

Mitsubishi Programmable Controller

SLMP Reference Manual

SAFETY PRECAUTIONS

(Read these precautions before using this product.)

Before using this product, please read this manual carefully and pay full attention to safety to handle the product correctly.

In this manual, the safety precautions are classified into two levels: "⚠WARNING" and "⚠CAUTION".

⚠ **WARNING**	Indicates that incorrect handling may cause hazardous conditions, resulting in death or severe injury.
⚠ **CAUTION**	Indicates that incorrect handling may cause hazardous conditions, resulting in minor or moderate injury or property damage.

Under some circumstances, failure to observe the precautions given under "⚠CAUTION" may lead to serious consequences.

Observe the precautions of both levels because they are important for personal and system safety.

Make sure that the end users read this manual and then keep the manual in a safe place for future reference.

[Design Precautions]

⚠WARNING

- When connecting an external device with an SLMP compatible device to modify data of a running programmable controller, configure an interlock circuit in the program to ensure that the entire system will always operate safely. For other forms of control (such as program modification, parameter change, forced output, or operating status change) of a running programmable controller, read the relevant manuals carefully and ensure that the operation is safe before proceeding. Improper operation may damage machines or cause accidents.
- Especially, when a remote programmable controller is controlled by an external device, immediate action cannot be taken if a problem occurs in the programmable controller due to a communication failure. To prevent this, configure an interlock circuit in the program, and determine corrective actions to be taken between the external device and programmable controller in case of a communication failure.
- Do not write any data to the "system area" and "write-protect area" of the buffer memory in the SLMP compatible device or intelligent function module. Also, do not use any "use prohibited" signals as an output signal from the CPU module to each module. Doing so may cause malfunction of the programmable controller system. For the "system area", "write-protect area", and the "use prohibited" signals, refer to the user's manual for the module used.

[Startup and Maintenance Precautions]

⚠CAUTION

- Before performing online operations (especially, program modification, parameter change, forced output, and operating status change) for the SLMP compatible device or CPU module on another station from the peripheral device connected, read relevant manuals carefully and ensure the safety. Improper operation may damage machines or cause accidents.

CONDITIONS OF USE FOR THE PRODUCT

(1) Mitsubishi programmable controller ("the PRODUCT") shall be used in conditions;
i) where any problem, fault or failure occurring in the PRODUCT, if any, shall not lead to any major or serious accident; and
ii) where the backup and fail-safe function are systematically or automatically provided outside of the PRODUCT for the case of any problem, fault or failure occurring in the PRODUCT.

(2) The PRODUCT has been designed and manufactured for the purpose of being used in general industries.
MITSUBISHI SHALL HAVE NO RESPONSIBILITY OR LIABILITY (INCLUDING, BUT NOT LIMITED TO ANY AND ALL RESPONSIBILITY OR LIABILITY BASED ON CONTRACT, WARRANTY, TORT, PRODUCT LIABILITY) FOR ANY INJURY OR DEATH TO PERSONS OR LOSS OR DAMAGE TO PROPERTY CAUSED BY the PRODUCT THAT ARE OPERATED OR USED IN APPLICATION NOT INTENDED OR EXCLUDED BY INSTRUCTIONS, PRECAUTIONS, OR WARNING CONTAINED IN MITSUBISHI'S USER, INSTRUCTION AND/OR SAFETY MANUALS, TECHNICAL BULLETINS AND GUIDELINES FOR the PRODUCT.
("Prohibited Application")
Prohibited Applications include, but not limited to, the use of the PRODUCT in;

- Nuclear Power Plants and any other power plants operated by Power companies, and/or any other cases in which the public could be affected if any problem or fault occurs in the PRODUCT.
- Railway companies or Public service purposes, and/or any other cases in which establishment of a special quality assurance system is required by the Purchaser or End User.
- Aircraft or Aerospace, Medical applications, Train equipment, transport equipment such as Elevator and Escalator, Incineration and Fuel devices, Vehicles, Manned transportation, Equipment for Recreation and Amusement, and Safety devices, handling of Nuclear or Hazardous Materials or Chemicals, Mining and Drilling, and/or other applications where there is a significant risk of injury to the public or property.

Notwithstanding the above, restrictions Mitsubishi may in its sole discretion, authorize use of the PRODUCT in one or more of the Prohibited Applications, provided that the usage of the PRODUCT is limited only for the specific applications agreed to by Mitsubishi and provided further that no special quality assurance or fail-safe, redundant or other safety features which exceed the general specifications of the PRODUCTs are required. For details, please contact the Mitsubishi representative in your region.

INTRODUCTION

Thank you for purchasing the Mitsubishi programmable controllers.
This manual describes the compatible devices, access range, communication procedure, and message format required for using the SLMP (SeamLess Message Protocol).
Before using this product, please read this manual carefully and develop familiarity with the SLMP to handle the product correctly.
When applying the program examples provided in this manual to an actual system, ensure the applicability and confirm that it will not cause system control problems.

CONTENTS

SAFETY PRECAUTIONS 1
CONDITIONS OF USE FOR THE PRODUCT 2
INTRODUCTION 3
TERMS 6

CHAPTER 1 OVERVIEW 7

CHAPTER 2 SPECIFICATIONS 9

2.1 SLMP Specifications 9
2.2 SLMP Compatible Device 9
2.3 Access Range and Accessible Modules with Other Stations 10

CHAPTER 3 COMMUNICATION PROCEDURE OF SLMP 12

3.1 When Using TCP/IP 12
3.2 When Using UDP/IP 13
3.3 Precautions 14

CHAPTER 4 MESSAGE FORMAT 16

4.1 Request Message 16
4.2 Response Message 21

CHAPTER 5 COMMANDS 24

5.1 Command List 25
5.2 Device (Device Access) 30
Data to be specified in command 30
Read (command: 0401) 40
Write (command: 1401) 44
Read Random (command: 0403) 47
Write Random (command: 1402) 51
Entry Monitor Device (command: 0801) 56
Execute Monitor (command: 0802) 60
Read Block (command: 0406) 63
Write Block (command: 1406) 67
5.3 Label (Label Access) 71
Data to be specified in command 71
Array Label Read (command: 041A) 80
Array Label Write (command: 141A) 89
Label Read Random (command: 041C) 99
Label Write Random (command: 141B) 106
5.4 Memory (Own Station Buffer Memory Access) 112
Data to be specified in command 112
Read (command: 0613) 114
Write (command: 1613) 116
5.5 Extend Unit (Accessing to Buffer Memory of Intelligent Function Module) 117
Data to be specified in command 118
Read (command: 0601) 120
Write (command: 1601) 122
5.6 Remote Control (Remote Operation) 124

Before the remote operation . . . 124
Remote Run (command: 1001) . . . 125
Remote Stop (command: 1002) . . . 127
Remote Pause (command: 1003) . . . 128
Remote Latch Clear (command: 1005) . . . 129
Remote Reset (command: 1006) . . . 130
Read Type Name (command: 0101) . . . 131
5.7 Remote Password (Remote Password) . . . 134
Data to be specified in command . . . 134
Lock (command: 1631) . . . 135
Unlock (command: 1630) . . . 137
5.8 File (File Control) . . . 139
Data to be specified in command . . . 140
Execution procedure . . . 144
Precautions . . . 148
Read Directory/File (command: 1810) . . . 149
Search Directory/File (command: 1811) . . . 159
New File (command: 1820) . . . 162
Delete File (command: 1822) . . . 165
Copy File (command: 1824) . . . 168
Change File State (command: 1825) . . . 172
Change File Date (command: 1826) . . . 175
Open File (command: 1827) . . . 178
Read File (command: 1828) . . . 181
Write File (command: 1829) . . . 184
Close File (command: 182A) . . . 187
5.9 Self Test (Loopback Test) (Command: 0619) . . . 189
5.10 Clear Error (Error Code Initialization, LED Off) (Command: 1617) . . . 191
5.11 Ondemand (Command: 2101) . . . 192

CHAPTER 6 TROUBLESHOOTING 193

APPENDICES 195

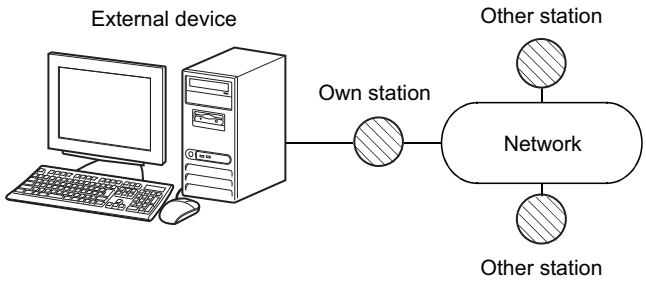
Appendix 1 Read or Write by Device Extension Specification . . . 195
Access to the link direct device . . . 195
Access to the module access device . . . 199
Access to the CPU buffer memory access device . . . 202
Access with indirect specification of the network No. and start I/O number by using the index register . . . 205
Access with indirect specification of the device No. by using the index register or long index register . . . 210
Access with indirect specification of the device No. by using the values stored in the word device . . . 216
Appendix 2 Correspondence Table of MC Protocol and SLMP . . . 220
Appendix 3 When Accessing Multiple CPU System . . . 222

INDEX 224

REVISIONS . . . 226
WARRANTY . . . 227
TRADEMARKS . . . 228

TERMS

Unless otherwise specified, this manual uses the following terms.

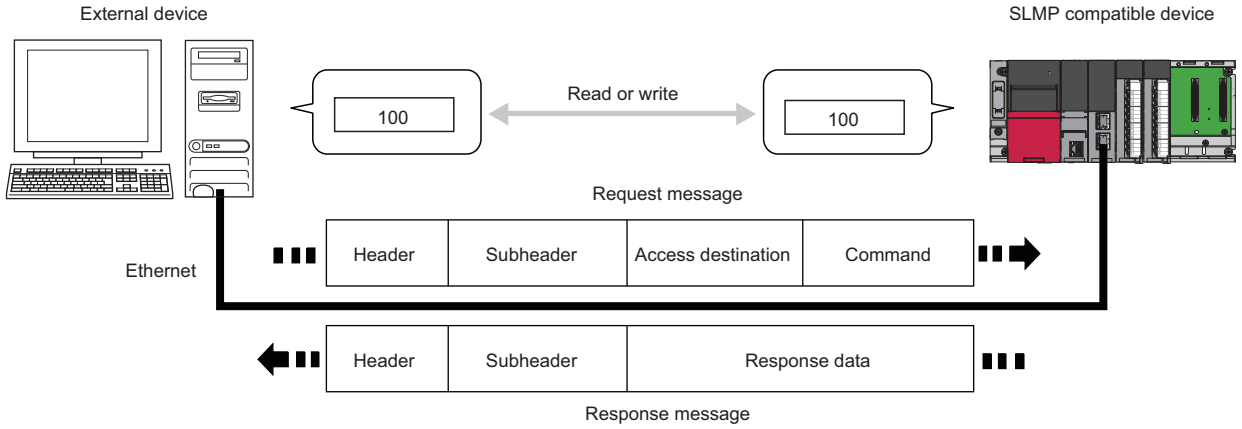
Term	Description
SLMP	The abbreviation for SeamLess Message Protocol. This protocol is used to access an SLMP compatible device or a programmable controller connected to an SLMP compatible device from an external device.
SLMP compatible device	A generic term for the devices of the Mitsubishi product that can transfer SLMP messages (Ethernet adapter module and Ethernet-equipped module)
External device	A generic term for devices that send SLMP request messages to a SLMP compatible device (personal computers, HMI (Human Machine Interface) and others)
MC protocol	The abbreviation for the MELSEC communication protocol. This protocol is used to access a MC protocol-compatible device or a programmable controller connected to a MC protocol-compatible device from an external device.
Own station	Own station indicates the station directly connected to external devices. Other station indicates a station connected to the own station on the network. 
Other station	
Request message	A processing request message sent from external devices to SLMP compatible devices
Response message	A processing result message sent from SLMP compatible devices in response to the request message
Engineering tool	Another term for the software package for the MELSEC programmable controllers
CC-Link IE Field Network	A high-speed and large-capacity open field network that is based on Ethernet (1000BASE-T)
CC-Link IE Field Network-equipped master/local module	A generic term for the CC-Link IE Field Network master/local modules: RJ71GF11-T2, QJ71GF11-T2, and LJ71GF11-T2, and the RJ71EN71 (when the CC-Link IE Field Network function is used)
CC-Link IE Field Network head module	The abbreviation for the LJ72GF15-T2 CC-Link IE Field Network head module
CC-Link IE Controller Network-equipped module	The abbreviation for the CC-Link IE Controller Network modules: RJ71GP21-SX, QJ71GP21-SX, and QJ71GP21S-SX
Ethernet-equipped module	A generic term for the RJ71EN71 (when the Ethernet function is used), the Ethernet interface modules: QJ71E71-100 and LJ71E71-100, MELSEC iQ-R series CPU module (when the Ethernet function is used), and built-in Ethernet port CPU
Ethernet adapter module	The abbreviation for the NZ2GF-ETB CC-Link IE Field Network Ethernet adapter module
Built-in Ethernet port CPU	A generic term for the Q03UDVCPU, Q04UDVCPU, Q04UDPVCPU, Q06UDVCPU, Q06UDPVCPU, Q13UDVCPU, Q13UDPVCPU, Q26UDVCPU, Q26UDPVCPU, L02CPU, L02CPU-P, L06CPU, L06CPU-P, L26CPU, L26CPU-P, L26CPU-BT, and L26CPU-PBT
Relay station	A station that includes two or more network modules. Data are passed through this station to stations on other networks
Device	A device (X, Y, W, or others) in a SLMP compatible device and the CPU module of the other stations
Link device	A device (RX, RY, RWr, or RWw) in a module on CC-Link IE Field Network
CPU module	A generic term for the MELSEC iQ-R series, MELSEC-Q series, and MELSEC-L series CPU module
RCPU	A generic term for the MELSEC iQ-R series CPU module
QCPU	A generic term for the MELSEC-Q series CPU module
Module access device	A generic term for the module access device of the MELSEC iQ-R series and intelligent function module device of the MELSEC-Q/L series
Intelligent function module	A generic term for the MELSEC iQ-R series and MELSEC-Q/L series module that has functions other than input and output, such as an A/D converter module and D/A converter module
Buffer memory	A memory in an intelligent function module and a SLMP compatible device, where data (such as setting values and monitoring values) are stored

1 OVERVIEW

SLMP is a protocol used for access from an external device to an SLMP compatible device through the Ethernet. SLMP communications are available among devices that can transfer messages by SLMP. (personal computers, human machine interface and others.)

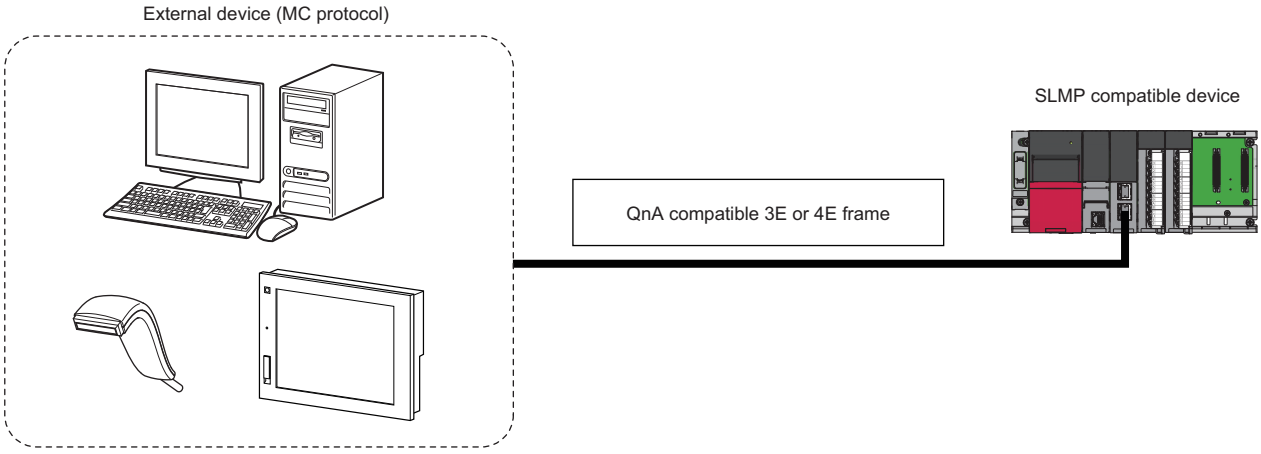
System monitoring from an external device

An external device connected through Ethernet can send a request message in the SLMP message format to read device data, allowing system monitoring. Using SLMP allows not only device data reading but also device data writing and resetting an SLMP compatible device. (☞ Page 24 COMMANDS)



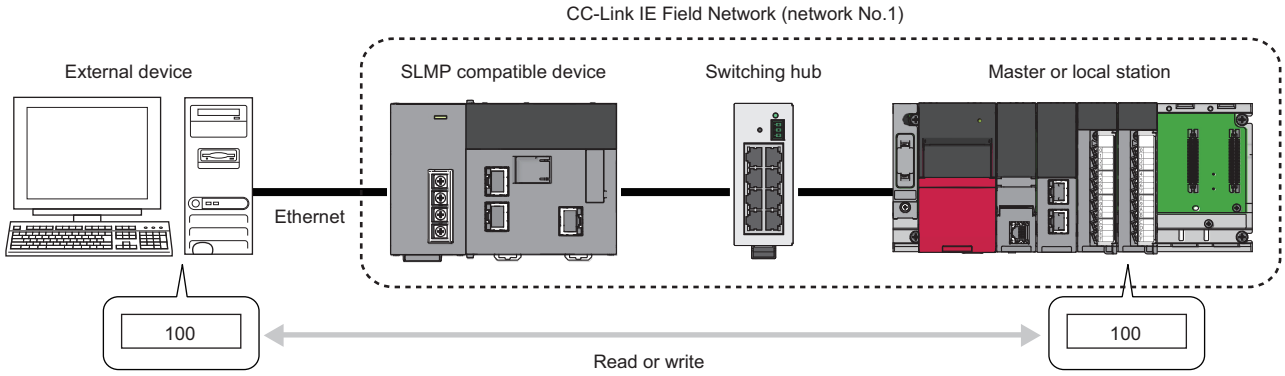
Connecting an external device used with MC protocol

The message format of SLMP is the same as that of the QnA compatible 3E and 4E frame of MC protocol. Therefore, external devices used with MC protocol can be connected to a SLMP compatible device directly. (☞ Page 220 Correspondence Table of MC Protocol and SLMP)



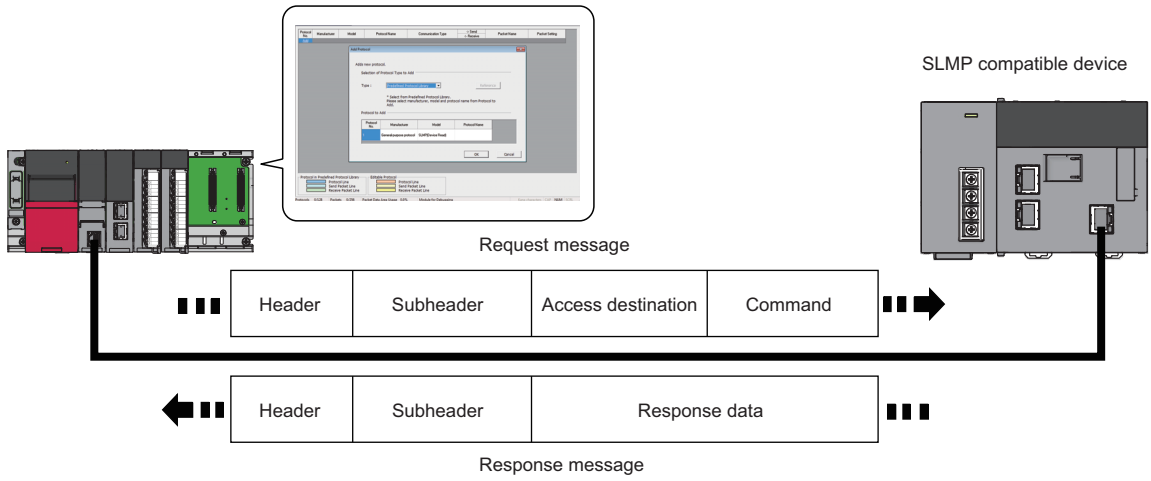
Access via network

The SLMP allows an external device to access the modules in the same network and other networks seamlessly via a SLMP compatible device. (☞ Page 10 Access Range and Accessible Modules with Other Stations)



Easy SLMP communication with the predefined protocol support function

The SLMP communication can be easily used with the predefined protocol support function of the engineering tool.
SLMP compatible devices can be controlled from an Ethernet-equipped module, as well as an external device communicates through the SLMP.



2 SPECIFICATIONS

This chapter describes communication specifications for SLMP compatible devices and the SLMP.

2.1 SLMP Specifications

This chapter describes the SLMP specifications for the message sent from an external device or by the predefined protocol support function.

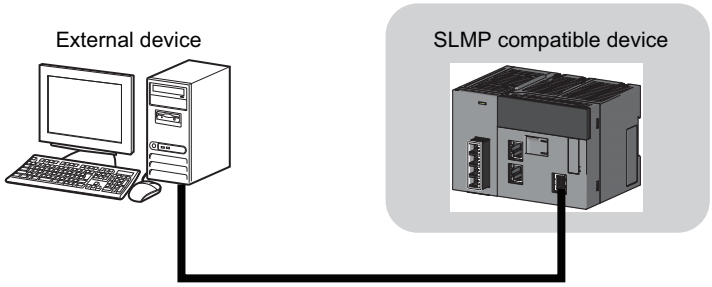
Item	Communication data code	Description	Reference
SLMP	• ASCII code • Binary code	The message format is the same as the QnA compatible 3E and 4E frames.	Page 16 Request Message

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 SLMP Compatible Device

For the SLMP compatibility, refer to the manual for the module used.



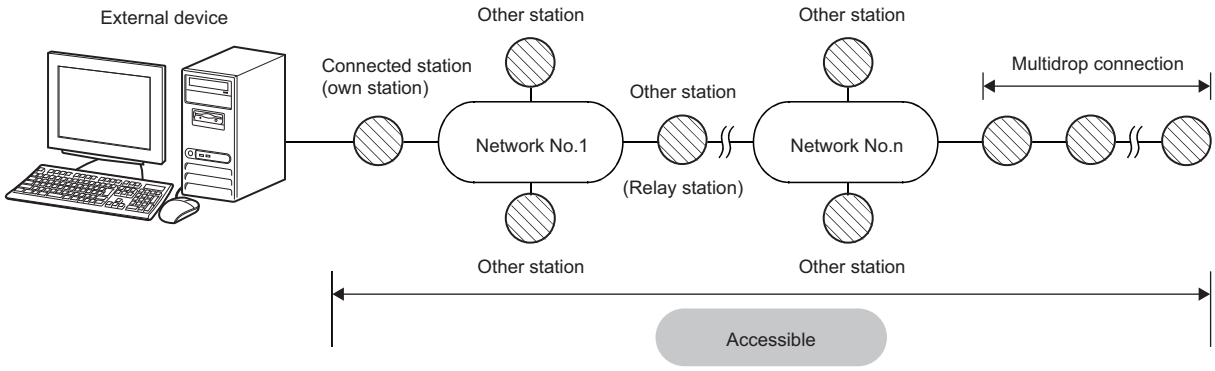
2.3 Access Range and Accessible Modules with Other Stations

Access range

The following devices are accessible.

- SLMP compatible devices directly connected to the external device (own station)
- Other stations on the same network with the SLMP compatible device
- Other stations on other networks connected to other stations on the same network with the SLMP compatible device*1

*1 The following targets are accessible: other stations in which the network No. and station No. are set and serial communication modules in multidrop connection.



Point

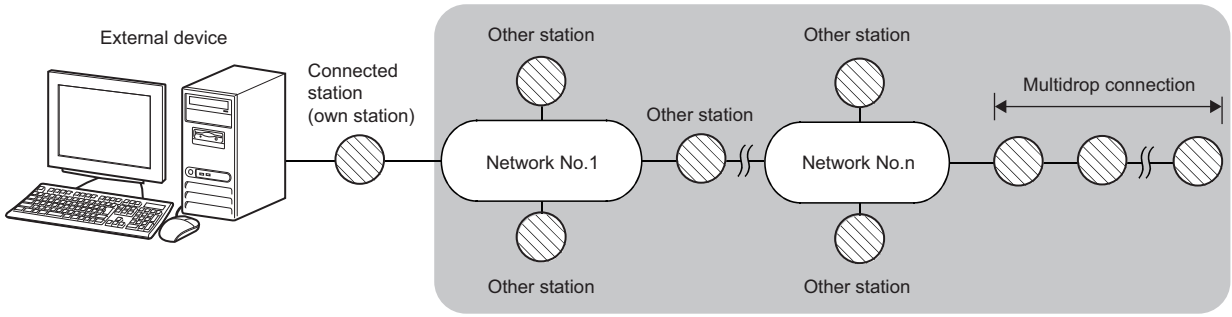
The following networks are accessible.

- Ethernet (The network No. and station No. must be set.)
- CC-Link IE Controller Network
- CC-Link IE Field Network
- MELSECNET/H

Eight networks (the number of relay stations: seven stations) are accessible at a maximum.

Modules of other stations that are accessible

When accessing other stations, the following modules are accessible.



■CPU module

CPU modules in the network corresponding to the SLMP compatible device are accessible. (📖 User's manual for each network module used)

■Other modules

The following modules are accessible.

- SLMP compatible device
- CC-Link IE Field Network-equipped master/local module
- CC-Link IE Field Network head module
- CC-Link IE Controller Network-equipped module
- Serial communication module in multidrop connection

3 COMMUNICATION PROCEDURE OF SLMP

An external device and a SLMP compatible device communicate in the following procedure.

3.1 When Using TCP/IP

The following is the communication procedure when executing SLMP communication with TCP/IP.
TCP/IP establishes a connection when communicating, and communicates checking that the data reached the receiver normally, so that the reliability is ensured. However, the load of line increases comparing to UDP/IP.



Connection request

Issues a connection request to the SLMP compatible device. (Active open)

Connection processing

Accepts the connection request from the external device. (Passive open)

Sending a request message

Upon establishment of a connection, sends a request message to the SLMP compatible device.

Processing

According to the message sent from the external device, performs the requested processing such as reading or writing.

Repeated as needed.

Receiving the response message

Receives the response message from the SLMP compatible device, and confirms the processing result.

Returning a response message

Upon completion of the processing, sends a response message to the external device.

Closing the connection

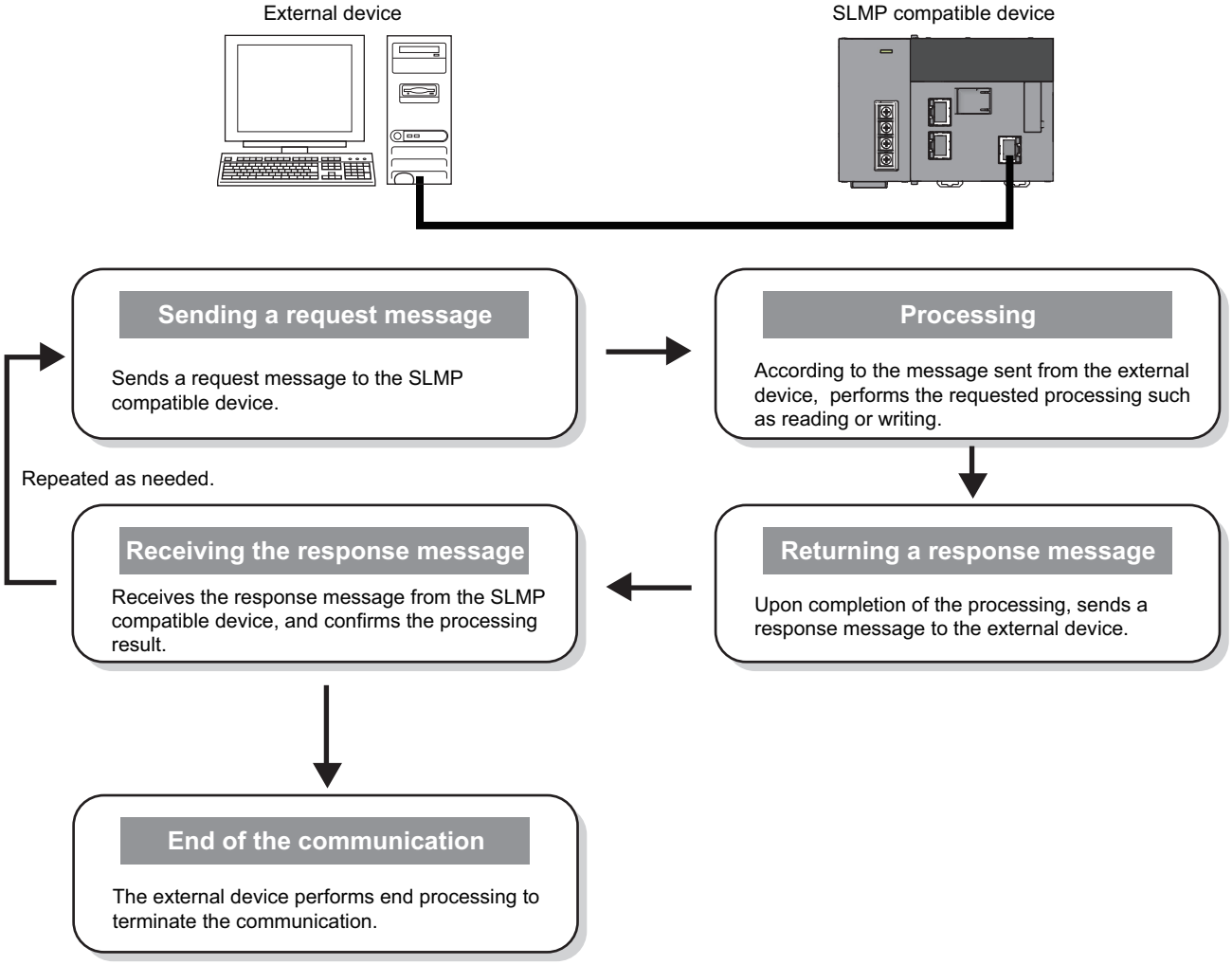
Issues a close request to the SLMP compatible device, and terminates the communication on the external device.

Closing the connection

Accepts the close request from the external device, and terminates the communication on the SLMP compatible device.

3.2 When Using UDP/IP

The following is the communication procedure when executing SLMP communication with UDP/IP.
UDP/IP neither establishes a connection when communicating nor communicates checking that the data reached the receiver normally, so that the load of line decreases. However, the reliability decreases comparing to TCP/IP.



3.3 Precautions

Request message transmission

Before sending a request message, the external device needs to confirm that the SLMP compatible device is ready to receive the message.

When sending several request messages

Add a "serial No." on the subheader of each request message at the external device side, then send them. Adding a "serial No.", the external device can identify the sender of the response message even if two or more request messages are sent.
(☞ Page 16 Request Message)

When sending the next request message continuously

When sending the next request message with "serial No." before receiving the response message continuously, the number of commands must not exceed the limit shown below.

SLMP compatible device		TCP/ UDP	Processable number of commands per one connection*1
Name	Model name		
Ethernet adapter module	NZ2GF-ETB	TCP	1 + (50 ÷ Number of connections to be used)
		UDP	
Ethernet-equipped module	RJ71EN71	TCP	(TCP window size (11680 bytes) ÷ Size of SLMP messages (byte))
		UDP	1 + (Number of messages that can be stored in receive buffer (190) ÷ Number of connections to be used)
	QJ71E71-100, LJ71E71-100	TCP	(6144 ÷ Size of SLMP messages (byte))
		UDP	1 + (57 ÷ Number of connections to be used)
	RCPU	TCP	(TCP window size (11680 bytes) ÷ Size of SLMP messages (byte))
		UDP	1 + Number of messages that can be stored in receive buffer (110)

*1 If the calculation result became decimal, round it down to the nearest whole number.

The following occurs when exceeded number of commands were sent. If the following occurred, decrease frequency of request message transmission.

- For TCP/IP, the overflow of the receive buffer of SLMP compatible device will occur. Since the window size becomes zero, transmission of request messages will be stopped temporarily until enough receive buffer space becomes available.
- For UDP/IP, an error may occur in the SLMP compatible device, or response messages may not return from the SLMP compatible device.

When the response message corresponding to the request message does not return

If the response message does not return from the SLMP compatible device, resend the request message from the external device after designated time set with "monitoring timer" of the request message.

Replacing SLMP compatible device

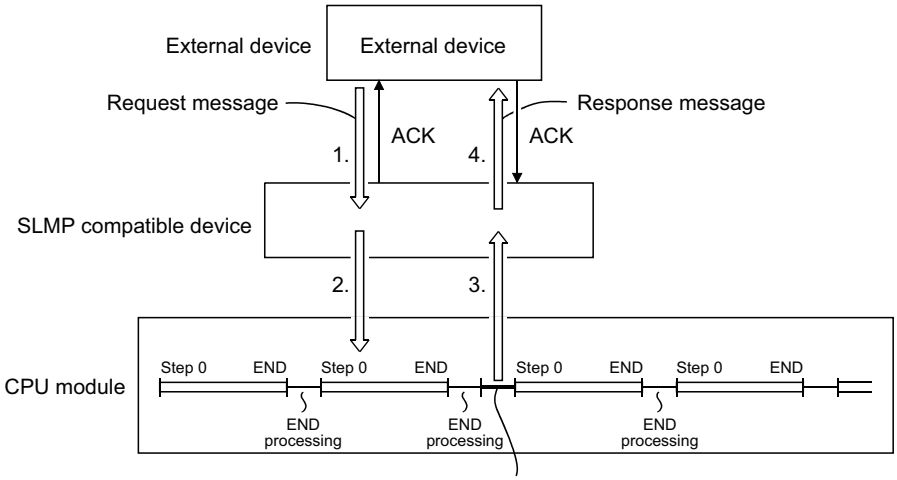
After replacing an external device or a SLMP compatible device due to failure and so on, the devices may not communicate by changing the MAC address. (When replaced with a device that has the same IP address)
When a device in the Ethernet network is replaced, restart all devices in the network.

When accessing the CPU module

Precautions for accessing the CPU module from the external device via a SLMP compatible device are shown below.

■Processing timing of the CPU module side

Processing for a request message is executed during an END processing of CPU module.



Processing for a request message from the external device

1. The external device sends a request message to the SLMP compatible device.
2. The SLMP compatible device receives the request message from the external device. Then, the SLMP compatible device sends a read request or a write request to the CPU module according to the message.
3. The CPU module reads or writes the data during END processing according to the request from the external device, and then sends back the processing result to the SLMP compatible device.
4. Once the SLMP compatible device receives a processing result from the CPU module, it sends a response message including the processing result to the external device.

■Read or write when the CPU module is running

- Scan time of the CPU module extends due to processing for the request from the external device. Access several times with less points when the control is affected by the extension of the scan time.
- Before writing, check that the CPU module is allowing the write processing during the run-time. (If system protection is unlocked, etc.)

■When the CPU module to be accessed is in system protection

An error occurs at the access destination, and an abnormal response is sent back to the external device. Unlock the system protection of the CPU module side, and resend the request message.

■When access requests are sent to one station from several external devices at the same time

Depending on the request timing, the processing requested from the external device may be on hold until several END processing take place. By using either of the following methods, multiple requests can be processed in one scan.

- Execute the COM instruction by program.
- Ensure 1 to 100ms of service processing time, using the "Service Processing Setting" of the engineering tool.

4 MESSAGE FORMAT

This chapter describes the message format of the SLMP.

4.1 Request Message

The following is the format of a request message sent from the external device to the SLMP compatible device.

Header	Subheader	Request destination network No.	Request destination station No.	Request destination module I/O No.	Request destination multidrop station No.	Request data length	Monitoring timer	Request data	Footer

Header

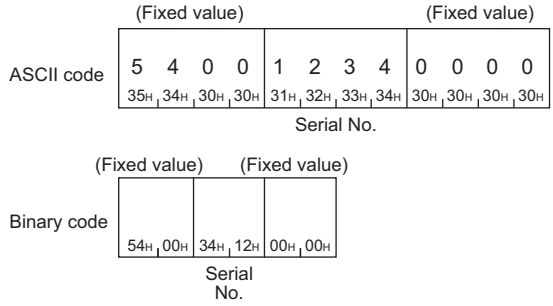
This is a header for TCP/IP and UDP/IP. Add the header on the external device side before sending the message. Normally it is added automatically by the external device.

Subheader

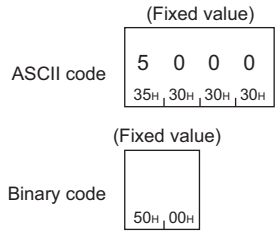
The subheader differs depending on whether or not a serial No. is added.

A serial No. is an optional number that is added on the external device side for message recognition. If a request message with serial No. is sent, the same serial No. will also be added on the response message. The serial No. is used when multiple request messages are sent from an external device to the same SLMP compatible device.

When adding a serial No. to the request message (When the serial No. is 1234H)



When not adding a serial No. to the request message



Point

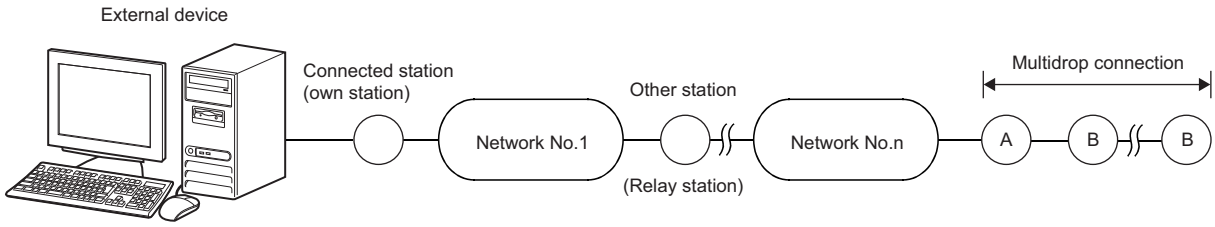
- Serial numbers must be managed at the external device side.
- When sending the message in ASCII code, the serial No. is stored from the upper byte to the lower byte.
- When sending the message in binary code, the serial No. is stored from the lower byte to the upper byte.

Request destination network No. and request destination station No.

Specify the network No. and station No. corresponding to the access destination. Specify the network No. and station No. in hexadecimal.

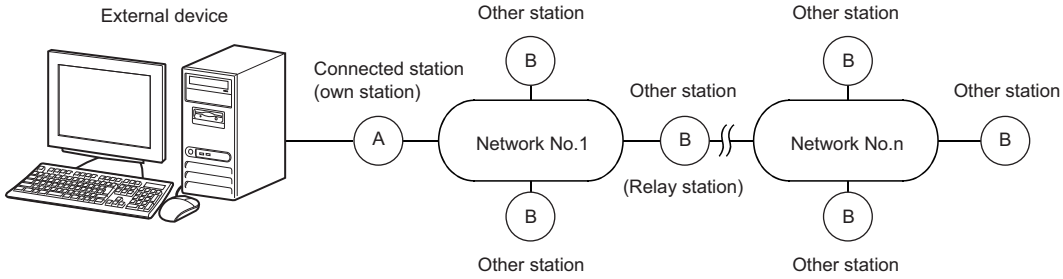
The request destination network No. and request destination station No. are sent in order from the upper byte to the lower byte.

■When the access destination is the multidrop connection station



Access destination	Request destination network No.	Request destination station No.
B (multidrop connection station)	Network No. of A (the station that relays the multidrop connection and network) and the network No. n connected	Station No. of the network module of A (the station that relays the multidrop connection and network)

■When the access destination is other than the multidrop connection station



Access destination	Request destination network No.	Request destination station No.
A (connected station)	00H	FFH
B (another station)	01H to EFH (1 to 239) The stations of network No.240 to 255 cannot be accessed.	01H to 78H (1 to 120): Station No. 7DH (125): Assigned control station/Master station*1 7EH (126): Present control station/Master station*2

*1 7DH (125): The assigned control station and master station access the station that is set as the control station or master station with a parameter.

*2 7EH (126): The present control station and master station access the station that actually operates as a control station or master station.

Ex.

When specifying 1AH (26) as the request destination network No.

ASCII code

1	A
31H	41H

Binary code

1AH

When specifying 1AH (26) as the request destination station No.

ASCII code

1	A
31H	41H

Binary code

1AH

Request destination module I/O No.

Specify the module of the access destination.

Access destination	Request destination module I/O No.
Own station	03FFH
Control CPU	03FFH
Multiple system CPU No.1	03E0H
Multiple system CPU No.2	03E1H
Multiple system CPU No.3	03E2H
Multiple system CPU No.4	03E3H
Multidrop connection station via a CPU module in multidrop connection	0000H to 01FFH

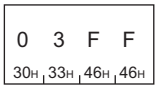
Point

When the CPU module in multidrop connection is relayed, specify the value in 4 digits (hexadecimal) obtained by dividing the I/O No. of the serial communication module of the multidrop connection source by 16.

Ex.

When specifying 03FFH as the request destination module I/O No.

ASCII code



Binary code



■When communicating data in ASCII code

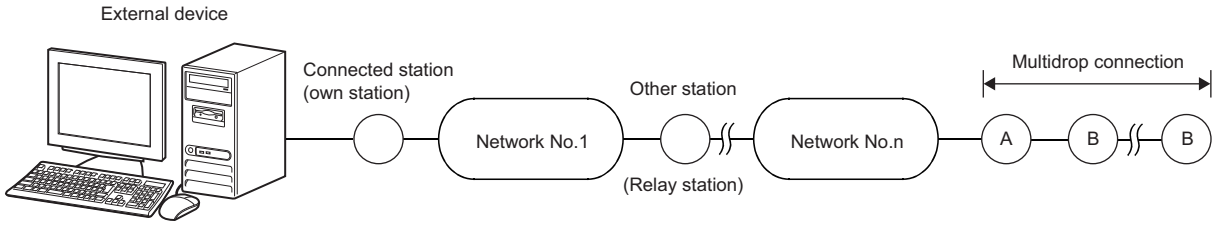
Send the data in order from the upper byte to the lower byte.

■When communicating data in binary code

Send the data in order from the lower byte to the upper byte.

Request destination multidrop station No.

Specify the request destination multidrop station No. when the access destination is a multidrop connection station.



Access destination of the external device	Request destination multidrop station No.
B (multidrop connection station)	00H to 1FH (0 to 31): Station No.
A (the station that relays the multidrop connection and network)	00H
Station other than the multidrop connection station	00H

Ex.

When 0 is specified as the request destination multidrop station No.

ASCII code

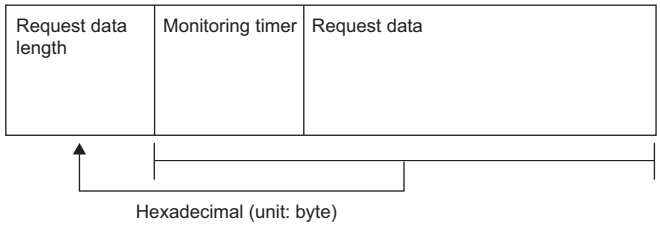
0	0
30H	30H

Binary code

00H

Request data length

Specify the data length from the monitoring timer to the request data in hexadecimal. (Unit: byte)



Ex.

When the request data length is 24 bytes

ASCII code

0	0	1	8
30H	30H	31H	38H

Binary code

18H 00H

■When communicating data in ASCII code

Send the data in order from the upper byte to the lower byte.

■When communicating data in binary code

Send the data in order from the lower byte to the upper byte.

Monitoring timer

This is a timer to set the waiting time until the access destination send back a response after the SLMP compatible device which received a request message from the external device requests a processing to the destination.

- 0000H (0): Unlimited wait (until the processing is completed)
- 0001H to FFFFH (1 to 65535): Waiting time (Unit: 250ms)

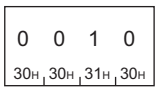
To execute normal data communication, it is recommended to use the timer with the following setting range depending on the access destination.

Access destination	Monitoring timer
Own station	01H to 28H (0.25s to 10s)
Other station	02H to F0H (0.5s to 60s)

Ex.

When specifying 10H for the monitoring timer

ASCII code



Binary code



■When communicating data in ASCII code

Send the data in order from the upper byte to the lower byte.

■When communicating data in binary code

Send the data in order from the lower byte to the upper byte.

Request data

Specify the command, the subcommand, and the data that indicate the request content. (☞ Page 24 COMMANDS)

Footer

This is a footer for TCP/IP and UDP/IP. Add the footer on the external device side before sending the message. Normally it is added automatically by the external device.

4.2 Response Message

The following is the format of a response message sent from the SLMP compatible device to the external device.

(When completed)

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

(When failed)

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

End code	Network No. (responding station)	Station No. (responding station)	Requested module I/O No.	Request destination multidrop station No.	Command	Subcommand	Footer

Error information

The same data as the request message is stored in the following items. (☞ Page 16 Request Message)

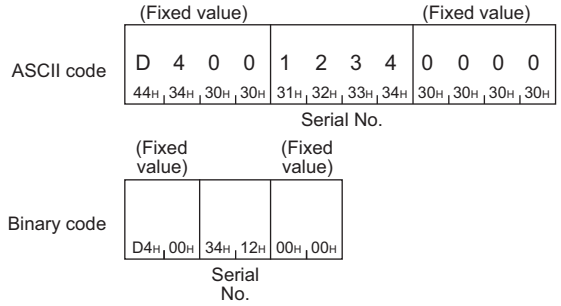
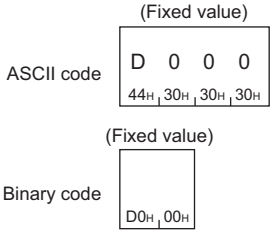
- Request destination network No.
- Request destination station No.
- Request destination module I/O No.
- Request destination multidrop station No.

Header

The header of Ethernet is stored.

Subheader

The subheader corresponding to the request message is stored.

When adding a serial No. to the request message (When the serial No. is 1234H)	When not adding a serial No. to the request message
	

■When communicating data in ASCII code

The serial No. is stored in order from the upper byte to the lower byte.

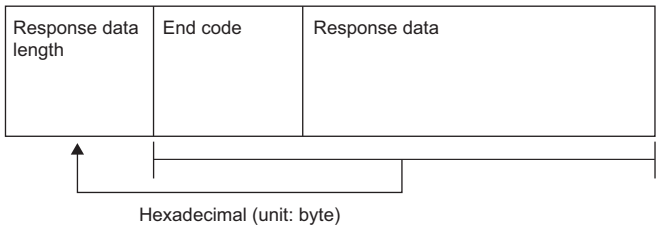
■When communicating data in binary code

The serial No. is stored in order from the lower byte to the upper byte.

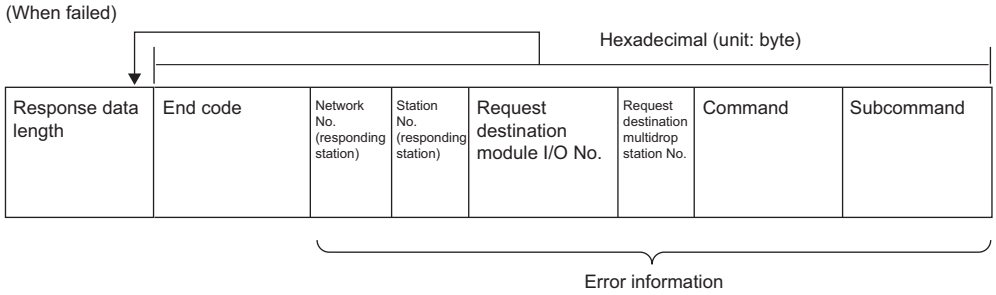
Response data length

The data length from the end code to the response data (when completed) or error information (when failed) is stored in hexadecimal. (Unit: byte)

(When completed)



(When failed)



Ex.

When the response data length is 22 bytes

ASCII code

0	0	1	6
30_H	30_H	31_H	36_H

Binary code

16_H	00_H

■When communicating data in ASCII code

The data is stored in order from the upper byte to the lower byte.

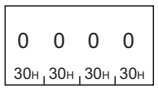
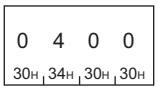
■When communicating data in binary code

The data is stored in order from the lower byte to the upper byte.

End code

The command processing result is stored.

When normally completed, 0 is stored. When failed, an error code of the access destination is stored. (📖 Manual for the SLMP compatible device)

When completed	When failed (for 0400H)
ASCII code 	ASCII code 
Binary code 	Binary code 

■When communicating data in ASCII code

The data is stored in order from the upper byte to the lower byte.

■When communicating data in binary code

The data is stored in order from the lower byte to the upper byte.

Response data

When the command is completed, the read data and others corresponding to the command are stored. For response data, refer to "Response data" of the command explanation part. (☞ From Page 40 Read (command: 0401))

Error information

The network No. (responding station), station No. (responding station), request destination module I/O No., and multidrop station No. of the stations which respond errors are stored. Doing so, numbers which does not correspond to the content of the request message may be stored. The command and the subcommand on which an error occurred are also stored.

5 COMMANDS

This chapter describes the SLMP commands.

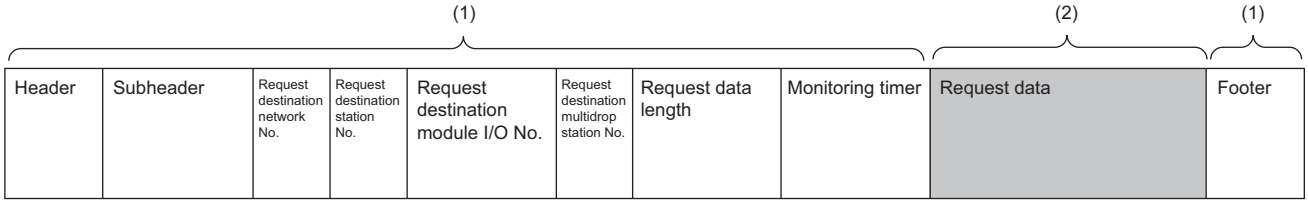
For message format except for the command part, refer to the following.

☞ Page 16 MESSAGE FORMAT

Request message

(1) ☞ Page 16 Request Message

(2) The request data includes commands and subcommands. For details, refer to the following page and after.
☞ Page 30 Device (Device Access)

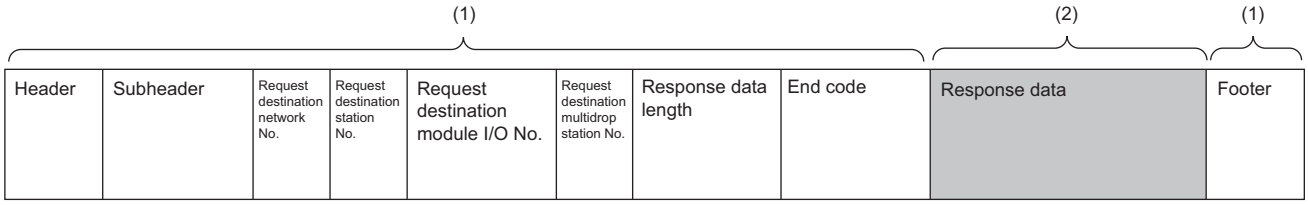


Response message

■When completed

(1) ☞ Page 21 Response Message

(2) Refer to the following page and after.
☞ Page 30 Device (Device Access)



■When failed

Refer to the following page and after.

☞ Page 16 MESSAGE FORMAT

5.1 Command List

Command list

The following table lists the commands. The ☐ part of "subcommand" differs depending on the type of a specified device. Refer to the following page and after.

☞ Page 30 Device (Device Access)

Item		Command	Subcommand	Description	Reference
Type	Operation				
Device	Read	0401	00☐1 00☐3	Reads value from the bit devices (consecutive device No.) in 1-point units.	Page 40 Read (command: 0401)
			00☐0 00☐2	• Reads value from the bit devices (consecutive device No.) in 16-point units. • Reads value from the word devices (consecutive device No.) in one-word units.	
	Write	1401	00☐1 00☐3	Writes value to the bit devices (consecutive device No.) in 1-point units.	Page 44 Write (command: 1401)
			00☐0 00☐2	• Writes value to the bit devices (consecutive device No.) in 16-point units. • Writes value to the word devices (consecutive device No.) in one-word units.	
	Read Random	0403	00☐0 00☐2	Specifies the device No. and reads the device value. This can be specified with inconsecutive device No. Reads value from the word devices in one-word units or two-word units.	Page 47 Read Random (command: 0403)
	Write Random	1402	00☐1 00☐3	Specifies the device No. to bit device in 1-point units and writes value. This can be specified with inconsecutive device No.	Page 51 Write Random (command: 1402)
			00☐0 00☐2	• Specifies the device No. to bit device in 16-point units and writes value. This can be specified with inconsecutive device No. • Specifies the device No. to word device in one-word units or two-word units and writes value. This can be specified with inconsecutive device No.	
	Entry Monitor Device	0801	00☐0 00☐2	Registers the device to be read by Execute Monitor (command: 0802).	Page 56 Entry Monitor Device (command: 0801)
	Execute Monitor	0802	0000	Reads the value of device registered by Entry Monitor Device (command: 0801).	Page 60 Execute Monitor (command: 0802)
	Read Block	0406	00☐0 00☐2	Reads data by treating n points of word devices or bit devices (one point is equivalent to 16 bits) as one block and specifying multiple blocks. This can be specified with inconsecutive device No.	Page 63 Read Block (command: 0406)
	Write Block	1406	00☐0 00☐2	Writes data by treating n points of word devices or bit devices (one point is equivalent to 16 bits) as one block and specifying multiple blocks. This can be specified with inconsecutive device No.	Page 67 Write Block (command: 1406)
Label	Array Label Read	041A	0000	Reads data from array type labels or labels whose structure members are the array.	Page 80 Array Label Read (command: 041A)
	Array Label Write	141A	0000	Writes data to array type labels or labels whose and structure members are the array.	Page 89 Array Label Write (command: 141A)
	Label Read Random	041C	0000	Specifies labels and reads the data.	Page 99 Label Read Random (command: 041C)
	Label Write Random	141B	0000	Specifies labels and writes data.	Page 106 Label Write Random (command: 141B)

Item		Command	Subcommand	Description	Reference
Type	Operation				
Memory	Read	0613	0000	Reads the buffer memory data of own station (SLMP compatible device).	Page 114 Read (command: 0613)
	Write	1613	0000	Writes the data in the buffer memory of own station (SLMP compatible device).	Page 116 Write (command: 1613)
Extend Unit	Read	0601	0000	Reads the data in the buffer memory of intelligent function module.	Page 120 Read (command: 0601)
	Write	1601	0000	Writes the data in the buffer memory of intelligent function module.	Page 122 Write (command: 1601)
Remote Control	Remote Run	1001	0000	Executes the remote RUN to the access destination module.	Page 125 Remote Run (command: 1001)
	Remote Stop	1002	0000	Executes the remote STOP to the access destination module.	Page 127 Remote Stop (command: 1002)
	Remote Pause	1003	0000	Executes the remote PAUSE to the access destination module.	Page 128 Remote Pause (command: 1003)
	Remote Latch Clear	1005	0000	Executes the remote latch clear to the access destination module.	Page 129 Remote Latch Clear (command: 1005)
	Remote Reset	1006	0000	Executes the remote RESET to the access destination module.	Page 130 Remote Reset (command: 1006)
	Read Type Name	0101	0000	This command reads the model name and model code of the access destination module.	Page 131 Read Type Name (command: 0101)
Remote Password	Lock	1631	0000	Specifies the remote password to disable the communication with other devices. (The locked state is activated from the unlocked state.)	Page 135 Lock (command: 1631)
	Unlock	1630	0000	Specifies the remote password to enable communication with other devices. (The unlocked state is activated from the locked state.)	Page 137 Unlock (command: 1630)
File	Read Directory/File	1810	0000 0040	Reads file list information.	Page 149 Read Directory/File (command: 1810)
	Search Directory/File	1811	0000 0040	Reads the presence of the specified file, file No., and file size.	Page 159 Search Directory/File (command: 1811)
	New File	1820	0000 0040	Reserves storage area for the specified file.	Page 162 New File (command: 1820)
	Delete File	1822	0000 0004 0040	Deletes a file.	Page 165 Delete File (command: 1822)
	Copy File	1824	0000 0004 0040	Copies the specified file.	Page 168 Copy File (command: 1824)
	Change File State	1825	0000 0004 0040	Changes file attributes.	Page 172 Change File State (command: 1825)
	Change File Date	1826	0000 0040	Changes the file creation date.	Page 175 Change File Date (command: 1826)
	Open File	1827	0000 0004 0040	Locks a file so that the content of the file is not changed by other devices.	Page 178 Open File (command: 1827)
	Read File	1828	0000	Reads the data of a file.	Page 181 Read File (command: 1828)
	Write File	1829	0000	Writes the data to a file.	Page 184 Write File (command: 1829)
	Close File	182A	0000	Cancels the file lock by open processing.	Page 187 Close File (command: 182A)
Self Test		0619	0000	Tests whether the communication with external devices is normally executed or not.	Page 189 Self Test (Loopback Test) (Command: 0619)

Item		Command	Subcommand	Description	Reference
Type	Operation				
Clear Error		1617	0000	Turns off the COM. ERR. LED of own station.	Page 191 Clear Error (Error Code Initialization, LED Off) (Command: 1617)
Ondemand		2101	0000	Outputs a send request to the SLMP compatible device from the CPU module and sends data to the external device.	Page 192 Ondemand (Command: 2101)

Accessible module for each command

The following table shows the access destination module that can be specified by a SLMP request message.

When specifying an Ethernet-equipped module to the access destination module, refer to the user's manual for the Ethernet-equipped module used.

○: Accessible, ×: Not accessible

Item		Command	Subcommand	Accessible module		
Type	Operation			CPU module		Intelligent device station on CC-Link IE Field Network
				MELSEC iQ-R series	MELSEC-Q/L series	
Device	Read	0401	00□1 00□0	○	○	○
			00□3 00□2		×	×
	Write	1401	00□1 00□0		○	○
			00□3 00□2		×	×
	Read Random	0403	00□0		○	○
			00□2		×	×
	Write Random	1402	00□1 00□0		○	○
			00□3 00□2		×	×
	Entry Monitor Device	0801	00□0		○	○
			00□2		×	×
	Execute Monitor	0802	0000		○	○
	Read Block	0406	00□0		○	○
			00□2		×	×
	Write Block	1406	00□0		○	○
			00□2		×	×
Label	Array Label Read	041A	0000		×	×
	Array Label Write	141A	0000			
	Label Read Random	041C	0000			
	Label Write Random	141B	0000			
Memory	Read	0613	0000		○	○*3
	Write	1613	0000			
Extend Unit	Read	0601	0000			○*4
	Write	1601	0000			
Remote Control	Remote Run	1001	0000			○
	Remote Stop	1002	0000			
	Remote Pause	1003	0000			×
	Remote Latch Clear	1005	0000			
	Remote Reset	1006	0000			○
	Read Type Name	0101	0000			
Remote Password*1	Lock	1631	0000		○	×
	Unlock	1630	0000			

Item		Command	Subcommand	Accessible module		
Type	Operation			CPU module		Intelligent device station on CC-Link IE Field Network
				MELSEC iQ-R series	MELSEC-Q/L series	
File	Read Directory/File	1810	0000	×	○	○*4
			0040	○	×	×
	Search Directory/File	1811	0000	×	○	○*4
			0040	○	×	×
	New File	1820	0000	×	○	○*4
			0040	○	×	×
	Delete File	1822	0000 0004	×	○*6*7	○*4*8
			0040	○	×	×
	Copy File	1824	0000 0004	×	○*6*7	○*4*8
			0040	○	×	×
	Change File State	1825	0000 0004	×	○*6*7	○*4*8
			0040	○	×	×
	Change File Date	1826	0000	×	○	○*4
			0040	○	×	×
	Open File	1827	0000 0004	×	○*6*7	○*4*8
			0040	○	×	×
	Read File	1828	0000	○	○	○*4
	Write File	1829	0000			
	Close File	182A	0000			
Self Test		0619	0000	—*2	×	×
Clear Error		1617	0000	×	×	○*1*3
Ondemand		2101	0000	—*5	—*5	—*5

*1 This can be used only for the connected stations connected to an external device.
*2 This can be used only for the Ethernet-equipped module connected to an external device.
*3 Only the CC-Link IE Field Network Ethernet adapter module is available.
*4 Only the CC-Link IE Field Network head module is available.
*5 This command is used to send data from the SLMP compatible device to the external device.
*6 The subcommand 0004 cannot access the QCPU.
*7 At the time of access to the LCPU, this can be used when a password is not set to the target file.
*8 This can be used when password is not set to the target file.

5.2 Device (Device Access)

This section describes commands which read/write data from/to a device.

Point

- Use the subcommand 00□1 or 00□0 when the access destination or connected station is a MELSEC-Q/L series module.
- Use the subcommand 00□3 or 00□2 when the access destination or connected station is a MELSEC iQ-R series module. Use the subcommands 00□1 and 00□0 to acquire compatibility with the MELSEC-Q/L series module.

Data to be specified in command

Device code

For request data, specify the access destination device using the following device code.
Specify the device code expressed in ( ) when the subcommand is 0001 or 0000.

Device	Type	Device code		Device No. range		Remarks
		ASCII code*1	Binary code			
Function input (FX)	Bit	—	—	—	Hexadecimal	Cannot be specified with SLMP.
Function output (FY)		—	—		Hexadecimal	
Function register (FD)	Word	—	—		Decimal	
Special relay (SM)	Bit	SM** (SM)	0091H (91H)	Specify within the device No. range of the access destination module.	Decimal	—
Special register (SD)	Word	SD** (SD)	00A9H (A9H)		Decimal	
Input (X)	Bit	X*** (X*)	009CH (9CH)		Hexadecimal	—
Output (Y)		Y*** (Y*)	009DH (9DH)		Hexadecimal	
Internal relay (M)		M*** (M*)	0090H (90H)		Decimal	Cannot access a local device.
Latch relay (L)		L*** (L*)	0092H (92H)		Decimal	—
Annunciator (F)		F*** (F*)	0093H (93H)		Decimal	
Edge relay (V)		V*** (V*)	0094H (94H)		Decimal	Cannot access a local device.
Link relay (B)		B*** (B*)	00A0H (A0H)		Hexadecimal	—
Data register (D)	Word	D*** (D*)	00A8H (A8H)		Decimal	Cannot access a local device.
Link register (W)		W*** (W*)	00B4H (B4H)		Hexadecimal	—

Device		Type	Device code		Device No. range		Remarks
			ASCII code*1	Binary code			
Timer (T)	Contact (TS)	Bit	TS** (TS)	00C1H (C1H)	Specify within the device No. range of the access destination module.	Decimal	Cannot access a local device.
	Coil (TC)		TC** (TC)	00C0H (C0H)			
	Current value (TN)	Word	TN** (TN)	00C2H (C2H)			
Long timer (LT)	Contact (LTS)	Bit	LTS* (—)	0051H (—)		Decimal	• Can be used with the subcommand 0003 or 0002 only. • Cannot access a local device.
	Coil (LTC)		LTC* (—)	0050H (—)			
	Current value (LTN)	Double word	LTN* (—)	0052H (—)			
Retentive timer (ST)	Contact (STS)	Bit	STS* (SS)	00C7H (C7H)		Decimal	Cannot access a local device.
	Coil (STC)		STC* (SC)	00C6H (C6H)			
	Current value (STN)	Word	STN* (SN)	00C8H (C8H)			
Long retentive timer (LST)	Contact (LSTS)	Bit	LSTS (—)	0059H (—)		Decimal	• Can be used with the subcommand 0003 or 0002 only. • Cannot access a local device.
	Coil (LSTC)		LSTC (—)	0058H (—)			
	Current value (LSTN)	Double word	LSTN (—)	005AH (—)			
Counter (C)	Contact (CS)	Bit	CS** (CS)	00C4H (C4H)		Decimal	Cannot access a local device.
	Coil (CC)		CC** (CC)	00C3H (C3H)			
	Current value (CN)	Word	CN** (CN)	00C5H (C5H)			
Long counter (LC)	Contact (LCS)	Bit	LCS* (—)	0055H (—)		Decimal	• Can be used with the subcommand 0003 or 0002 only. • Cannot access a local device.
	Coil (LCC)		LCC* (—)	0054H (—)			
	Current value (LCN)	Double word	LCN* (—)	0056H (—)			

Device	Type	Device code		Device No. range		Remarks
		ASCII code*1	Binary code			
Link special relay (SB	Bit	SB** (SB)	00A1H (A1H)	Specify within the device No. range of the access destination module.	Hexadecimal	—
Link special register (SW)	Word	SW** (SW)	00B5H (B5H)		Hexadecimal	
Step relay (S)	Bit	—	—	—	Decimal	Cannot be specified with SLMP.
Direct access input (DX)	Bit	DX** (DX)	00A2H (A2H)	Specify within the device No. range of the access destination module.	Hexadecimal	—
Direct access output (DY)		DY** (DY)	00A3H (A3H)		Hexadecimal	
Index register (Z)	Word	Z*** (Z*)	00CCH (CCH)		Decimal	Cannot access a local device.
Long index register (LZ)	Double word	LZ** (—)	0062H (—)			• Can be used with the subcommand 0003 or 0002 only. • Cannot access a local device.
File register (R, ZR)*2*3	Word	R*** (R*)	00AFH (AFH)		Decimal	Block switching method
		ZR** (ZR)	00B0H (B0H)		Hexadecimal	Serial number access method
Extended data register (D)*4	Word	— (D*)	— (A8H)	Binary code: Specify within the device No. range of the access destination module. ASCII code: 000000 to 999999	Decimal	—
Extended link register (W)*4	Word	— (W*)	— (B4H)	Specify within the device No. range of the access destination module.	Hexadecimal	
Module refresh register (RD)	Word	RD** (—)	002CH (—)		Decimal	Can be used with the subcommand 0003 or 0002 only.
Link direct device	☞ Page 195 Access to the link direct device					
Module access device	☞ Page 199 Access to the module access device					
CPU buffer memory access device	☞ Page 202 Access to the CPU buffer memory access device					

1 When communicating data in ASCII code, specify a device code in four digits for the subcommand 00□3 or 00□2. If the device code consists of three digits or less, add "" (ASCII code: 2AH) or a space (ASCII code: 20H) after the device code.
Specify a device code in two digits for the subcommand 00□1 or 00□0. If the device code consists of one digits, add "*" (ASCII code: 2AH) or a space (ASCII code: 20H) after the device code.

*2 The file register specified as "Use File Register of Each Program" in "CPU Parameters" or "PLC parameter" of the CPU module cannot be accessed from the external device.

*3 If the file register of the CPU module consists of multiple blocks, use the device code of the serial number access method ("ZR**, ZR" or "00B0H, B0H").
To specify the file register which consists of multiple blocks by the serial number access method, refer to the manual of the CPU module.

*4 If the access destination CPU module does not support the access to the extended data register D65536 or later, or the extended link register W10000 or later, transpose it to the file register (ZR) and specify again. For the transpose method, refer to the manual for the CPU module used.

■When communicating data in ASCII code

Use a 2- or 4-digit ASCII code converted from a device code, and send them from the upper byte to the lower byte. Use capitalized code for alphabetical letter.

The number of digits converted into an ASCII code differs depending on the subcommands.

Subcommand	Number of digits	Example
0003 0002	Converted into a four-digit ASCII code.	For input (X) (four digits)*1 
0001 0000	Converted into a two-digit ASCII code.	For input (X) (two digits)*1 

1 The device code of input replay is sent from "X". A space (code: 20H) can also be used instead of the second character and the following characters "".

■When communicating data in binary code

Send the data in order from the lower byte to the upper byte using two or one-byte numeral values.

The data size of the value differs depending on the subcommands.

Subcommand	Data size	Example
0003 0002	Two bytes	For input (X) (two bytes) 
0001 0000	One byte	For input (X) (one byte) 

Point

For devices that can be used with the access destination module, refer to the manual for the access destination module. (Manual for the module used)

Head device No. (Device No.)

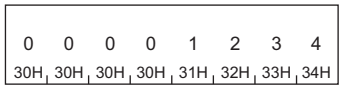
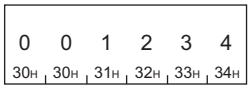
Specify the No. of the device which a file is to be read or written from/to. When specifying consecutive devices, specify the head device No. Specify the head device No. in decimal or in hexadecimal, depending on the device type.

(☞ Page 30 Device code)

■When communicating data in ASCII code

Use a 6- or 8-digit ASCII code converted from a device code, and send them from the upper byte to the lower byte.

The number of digits converted into an ASCII code differs depending on the subcommands.

Subcommand	Number of digits	Example
0003 0002	Converted into an eight-digit ASCII code.	For device No. 1234 (eight digits)*1 
0001 0000	Converted into a six-digit ASCII code.	For device No. 1234 (six digits)*1 

*1 Send the data in order from 0 in order. Spaces (code: 20H) can be also used for 0 at the upper digits.

■When communicating data in binary code

Send the data in order from the lower byte to the upper byte using four or three-byte numeral values. If the device No. is decimal, convert it to hexadecimal before sending.

The data size of the value differs depending on the subcommands.

Subcommand	Data size	Example
0003 0002	Four bytes	For internal relay M1234 and link relay B1234 (four bytes)*1 M1234: D2H, 04H, 00H, 00H B1234: 34H, 12H, 00H, 00H
0001 0000	Three bytes	For internal relay M1234 and link relay B1234 (three bytes)*2 M1234: D2H, 04H, 00H B1234: 34H, 12H, 00H

*1 Since the device No. of internal relay M1234 is decimal, convert it in hexadecimal. The internal relay M1234 becomes 000004D2H. Send them in order of D2H, 04H, 00H, 00H. The link relay B1234 becomes 00001234H. Send them in order of 34H, 12H, 00H, 00H.

*2 Since the device No. of internal relay M1234 is decimal, convert it in hexadecimal. The internal relay M1234 becomes 0004D2H. Send them in order of D2H, 04H, 00H. The link relay B1234 becomes 001234H. Send them in order of 34H, 12H, 00H.

Number of device points

Specify the number of points of the device to be read or written.

■When communicating data in ASCII code

Convert the points to a 4-digit ASCII code (hexadecimal), and send them in order from the upper byte to the lower byte. Use capitalized code for alphabetical letter.

Ex.

For 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)

■When 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.

For 5 points and 20 points

5 points: 05H, 00H

20 points: 14H, 00H

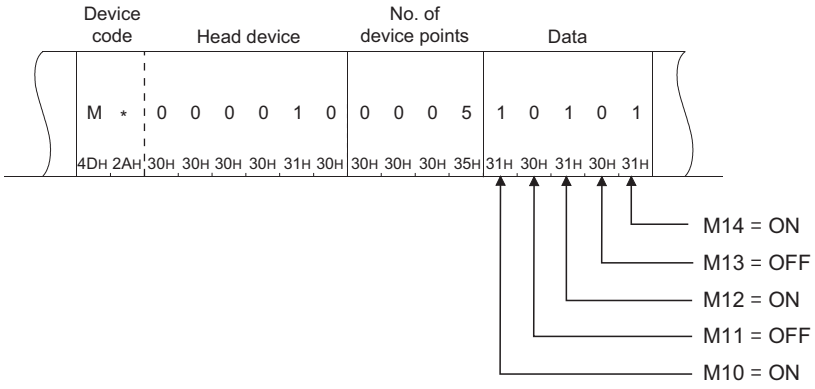
Read data, write data

In case of reading, the read data of the device is stored. In case of writing, the writing data is stored.
The data order differs depending on whether the data is read/written in bit units (subcommand: 00□1, 00□3) or word units (subcommand: 00□0, 00□2).

■For bit units (subcommand: 00□1, 00□3)

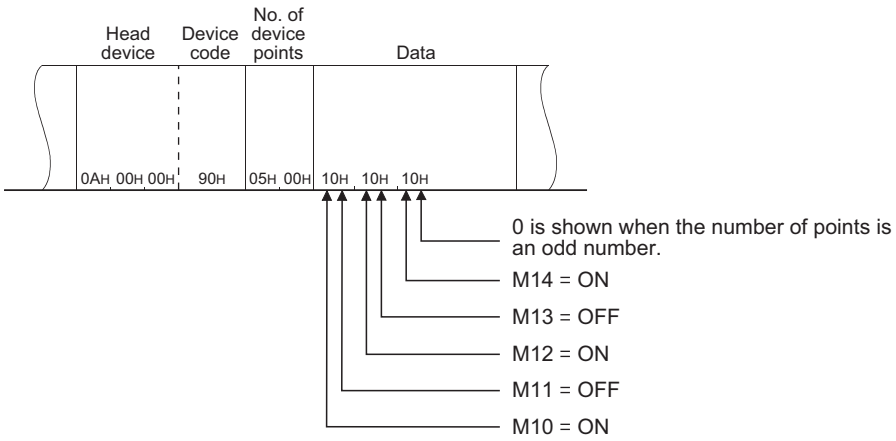
When communicating data in ASCII code, send the specified number of device points from the specified head device from the upper bit. ON is described as 31H (1) and OFF is described as 30H (0). Use capitalized code for alphabetical character.

Ex.
When indicating ON/OFF of five points from M10



When communicating data in binary code, specify one point as four bits, and send the specified number of device points from the specified head device from the upper bit. ON is described as "1" and OFF is described as "0".

Ex.
When indicating ON/OFF of five points from M10

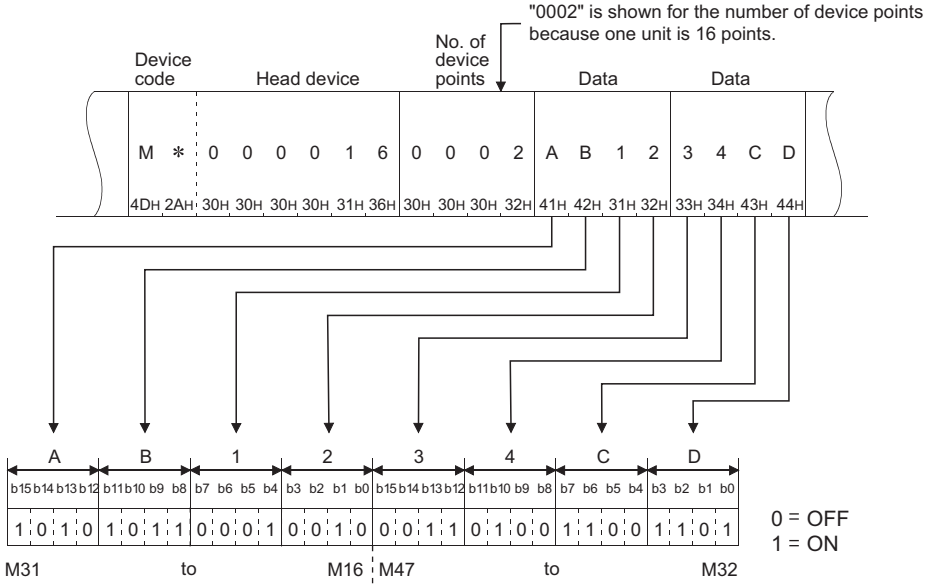


■For word units (subcommand: 00□0, 00□2)

When communicating data in ASCII code, send one word in four bit units from the upper bit to the lower bit. The data is described in hexadecimal. Use capitalized code for alphabetical letter.

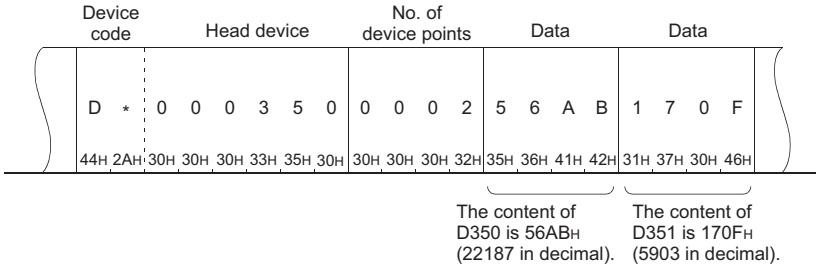
Ex.

When indicating ON/OFF of 32 points from M16



Ex.

When indicating the stored data of D350 and D351



Point

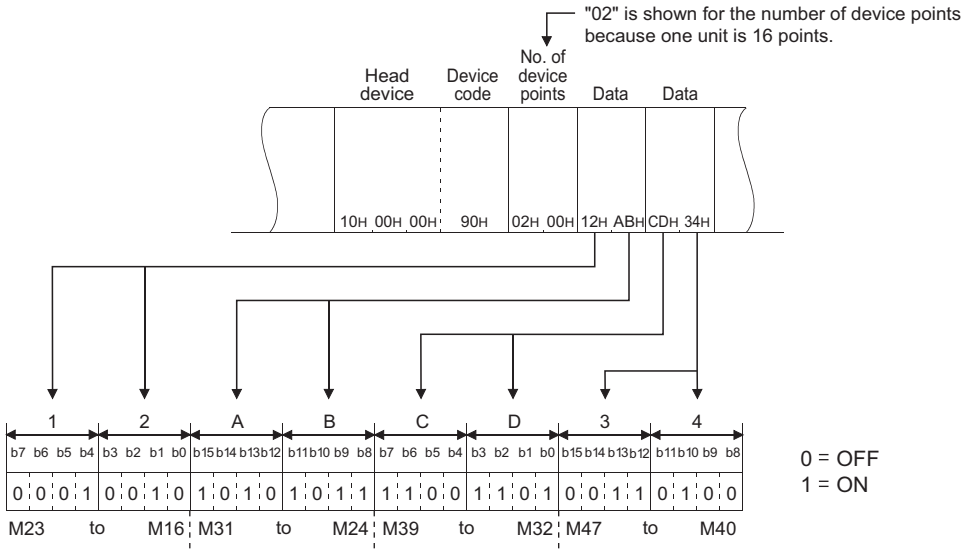
When real number or text is stored in the word devices to be read, the stored values are read as integral number.

- When real number (0.75) is stored in D0 and D1, D0 = 0000H and D1 = 3F40H.
- When text ("12AB") is stored in D2 and D3, D2 = 3231H and D3 = 4241H.

When communicating data in binary code and using bit devices in word units, specify one point as one bit as the following example. The storing order is from the lower byte (bit 0 to 7) to the upper byte (bit 8 to 15).

Ex.

When indicating ON/OFF of 32 points from M16



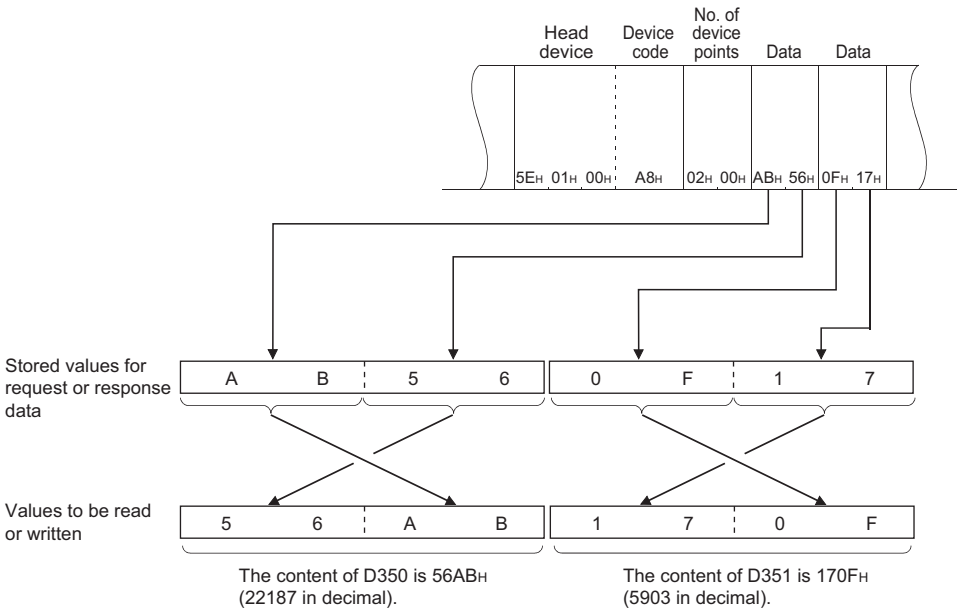
For the word device, one word is 16 bits as the following example. The storing order is from the lower byte (bit 0 to 7) to the upper byte (bit 8 to 15).

When reading, exchange the upper byte and the lower byte of the value stored in the response data on the user side.

When writing, exchange the upper byte and the lower byte of the value to be written on the user side before storing it into the request data.

Ex.

When indicating the stored data of D350 and D351



Point

When real number or text is stored in the word devices to be read, the stored values are read as integral number.

- When real number (0.75) is stored in D0 and D1, D0 = 0000H and D1 = 3F40H.
- When text ("12AB") is stored in D2 and D3, D2 = 3231H and D3 = 4241H.

■Precautions

When communicating in ASCII data, process the data as follows to pass the text from the external device to the CPU module. The following shows a procedure that the SLMP compatible device converts the data received from the external device to the binary code data and writes it to the specified device.

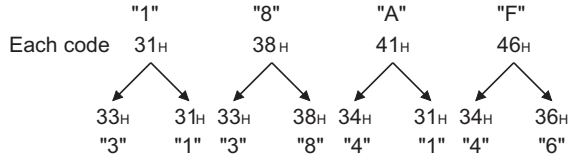
1. Expand the text to be sent from the external device to 2-byte code per one character.

2. Sort the expanded 2-byte text by every two characters and send them to the SLMP compatible device.

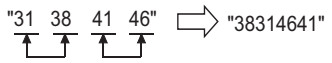
3. Write the data sent to the SLMP compatible device to the specified device.

The following shows an example that the text ("18AF") received from the external device is converted to the binary code data and written to D0 and D1.

1. Expand the text ("18AF") to be sent from the external device to 2-byte code per one character.



2. Sort the expanded 2-byte text by every two characters and send them to the SLMP compatible device.

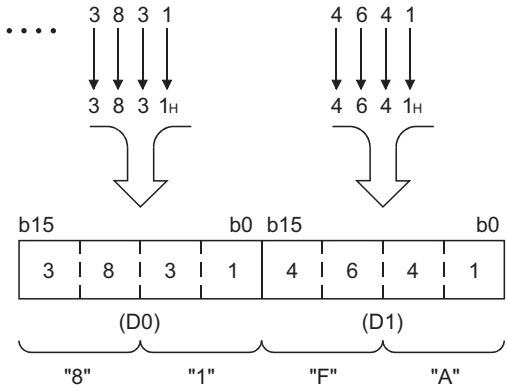


From the external device to the CPU module, "38314641" is sent.

3. Write the data "38314641" sent to the SLMP compatible device to D0 and D1.

The received data are converted to binary code data and written.

Each byte represents an alphanumeric value.



Number of bit access points

Specify the access points in bit units.

■When communicating data in ASCII code

Convert the points to 2-digit ASCII code (hexadecimal) and send them from the upper digit. Use capitalized code for alphabetical letter.

Ex.

For 5 points and 20 points

5 points



20 points



■When communicating data in binary code

Convert the points to hexadecimal and send.

Ex.

For 5 points and 20 points

5 points



20 points



Read (command: 0401)

This command reads value from a device.

Request data

ASCII

0 4 0 1 30H 34H 30H 31H	Subcommand	Device code	Head device No.	No. of device points

Binary

	Subcommand	Head device No.	Device code	No. of device points
01H 04H				

■Subcommand

Item	Subcommand*1	
	ASCII code	Binary code
When reading data in bit units	0 0 0 1 (30H 30H 30H 31H) or 0 0 8 1 (30H 30H 38H 31H)	01H 00H or 81H 00H
	0 0 0 3 (30H 30H 30H 33H) or 0 0 8 3 (30H 30H 38H 33H)	03H 00H or 83H 00H
When reading data in word units	0 0 0 0 (30H 30H 30H 30H) or 0 0 8 0 (30H 30H 38H 30H)	00H 00H or 80H 00H
	0 0 0 2 (30H 30H 30H 32H) or 0 0 8 2 (30H 30H 38H 32H)	02H 00H or 82H 00H

*1 The subcommand 008☐ is used to access the link direct device, module access device, or CPU buffer memory access device. When the subcommand is 008☐, the message format is different. (☞ Page 195 Read or Write by Device Extension Specification)

■Device code

Specify the type of the target device of reading. (☞ Page 30 Device code)

However, the contact, coil, or current value of the following devices cannot be specified.

- Long timer (LTS, LTC, LTN)
- Long retentive timer (LSTS, LSTC, LSTN)
- Long counter (LCS, LCC, LCN)

■Head device No.

Specify the head No. of the target device of reading. (☞ Page 33 Head device No. (Device No.))

■Number of device points

Specify the number of target device points of reading. (☞ Page 34 Number of device points)

Item	Number of points	
	ASCII code	Binary code
When reading data in bit units	1 to 3584 points	1 to 7168 points
When reading data in word units	1 to 960 points	

Response data

The value read from the device is stored in hexadecimal. The data order differs depending on the type of code, ASCII code or binary code. (☞ Page 35 Read data, write data)

Read data

Communication example (when reading data in bit units)

Read from M100 to M107.

■When communicating data in ASCII code

(Request data)

	Subcommand	Device code	Head device No.	No. of device points
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 ~ M107	

■When communicating data in binary code

(Request data)

Subcommand	Device code Head device No.	No. of device points
01H 04H	01H 00H 64H 00H 00H	90H 08H 00H

(Response data)

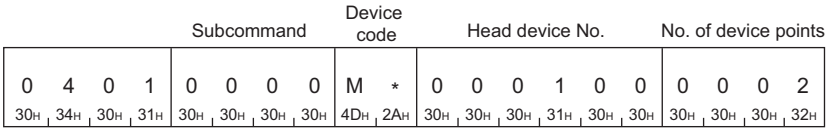
00H 01H 00H 11H	0 = OFF 1 = ON
M101 ~ M100	M107 M106

Communication example (when reading data in word units (bit device))

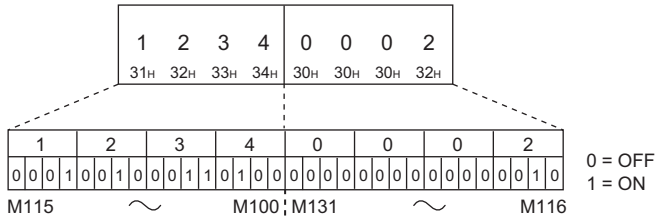
Read from M100 to M131 (two words).

■When communicating data in ASCII code

(Request data)

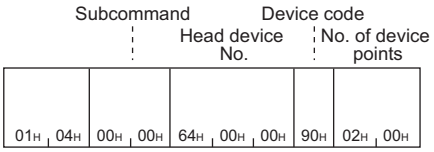


(Response data)

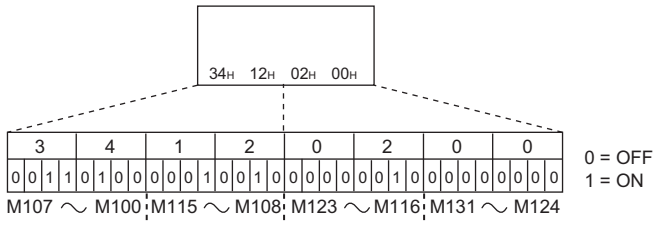


■When communicating data in binary code

(Request data)



(Response data)



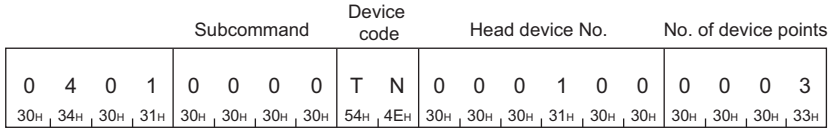
Communication example (when reading data in word units (word device))

Read from T100 to T102.

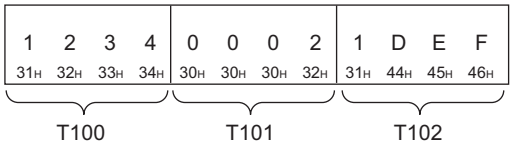
If T100 = 4660 (1234H), T101 = 2 (2H), T102 = 7663 (1DEFH) are stored, the response message will be as follows.

■When communicating data in ASCII code

(Request data)

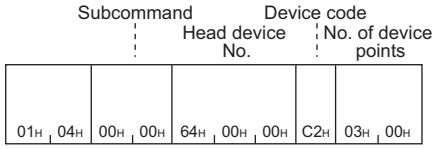


(Response data)

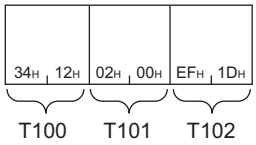


■When communicating data in binary code

(Request data)



(Response data)



Write (command: 1401)

This command writes the value in a device.

Request data

ASCII

1 4 0 1 31H 34H 30H 31H	Subcommand	Device code	Head device No.	No. of device points	Write data

Binary

01H 14H	Subcommand	Head device No.	Device code	No. of device points	Write data

■Subcommand

Item	Subcommand*1	
	ASCII code	Binary code
When writing data in bit units	0 0 0 1 (30H 30H 30H 31H) or 0 0 8 1 (30H 30H 38H 31H)	01H 00H or 81H 00H
	0 0 0 3 (30H 30H 30H 33H) or 0 0 8 3 (30H 30H 38H 33H)	03H 00H or 83H 00H
When writing data in word units	0 0 0 0 (30H 30H 30H 30H) or 0 0 8 0 (30H 30H 38H 30H)	00H 00H or 80H 00H
	0 0 0 2 (30H 30H 30H 32H) or 0 0 8 2 (30H 30H 38H 32H)	02H 00H or 82H 00H

*1 The subcommand 008☐ is used to access the link direct device, module access device, or CPU buffer memory access device. When the subcommand is 008☐, the message format is different. (☞ Page 195 Read or Write by Device Extension Specification)

■Device code

Specify the type of the target device of writing. (☞ Page 30 Device code)

However, the contact, coil, or current value of the following devices cannot be specified.

- Long timer (LTS, LTC, LTN)
- Long retentive timer (LSTS, LSTC, LSTN)
- Long counter (LCS, LCC, LCN)

■Head device No.

Specify the head No. of the target device of writing. (☞ Page 33 Head device No. (Device No.))

■Number of device points

Specify the target device points of writing. (☞ Page 34 Number of device points)

Item	Number of points	
	ASCII code	Binary code
When writing data in bit units	1 to 3584 points	1 to 7168 points
When writing data in word units	1 to 960 points	

■Write data

Specify the value to be written to the device of the number specified by "number of device points". (☞ Page 35 Read data, write data)

Response data

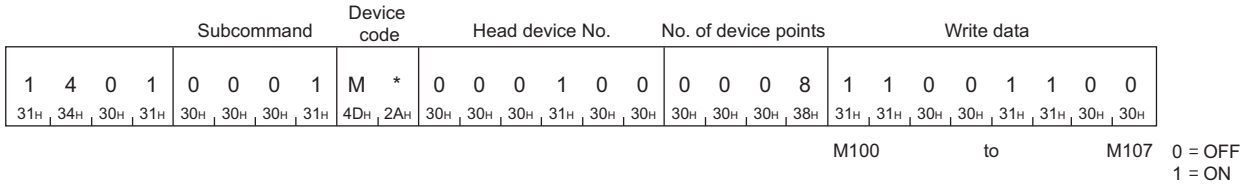
There is no response data for Write command.

Communication example (when writing data in bit units)

Write the value in from M100 to M107.

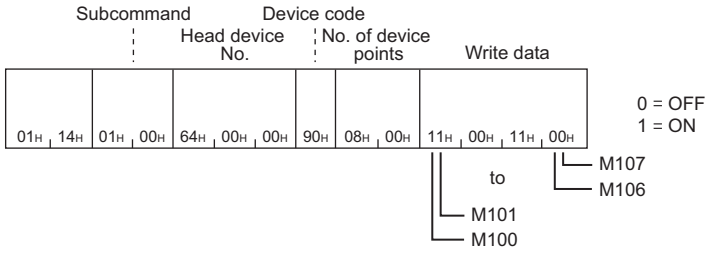
■When communicating data in ASCII code

(Request data)



■When communicating data in binary code

(Request data)

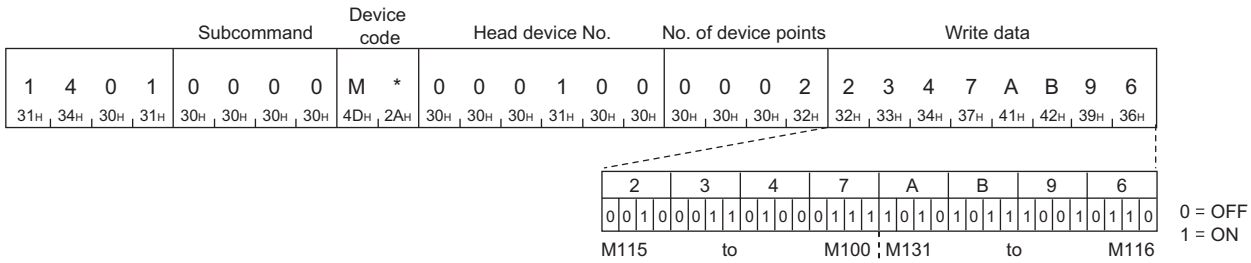


Communication example (when writing data in word units (bit device))

Write the value in from M100 to M131 (two words).

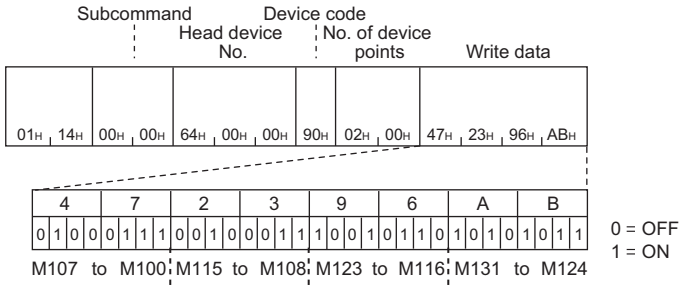
■When communicating data in ASCII code

(Request data)



■When communicating data in binary code

(Request data)

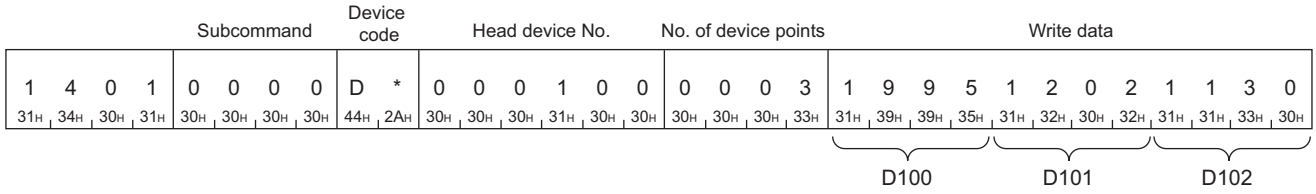


Communication example (when writing data in word units (word device))

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

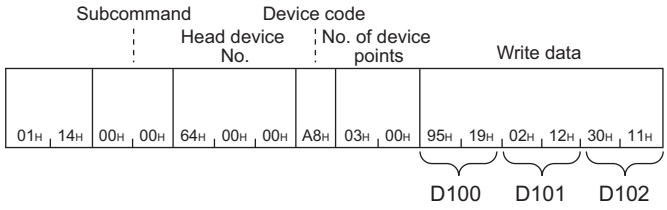
■When communicating data in ASCII code

(Request data)



■When communicating data in binary code

(Request data)



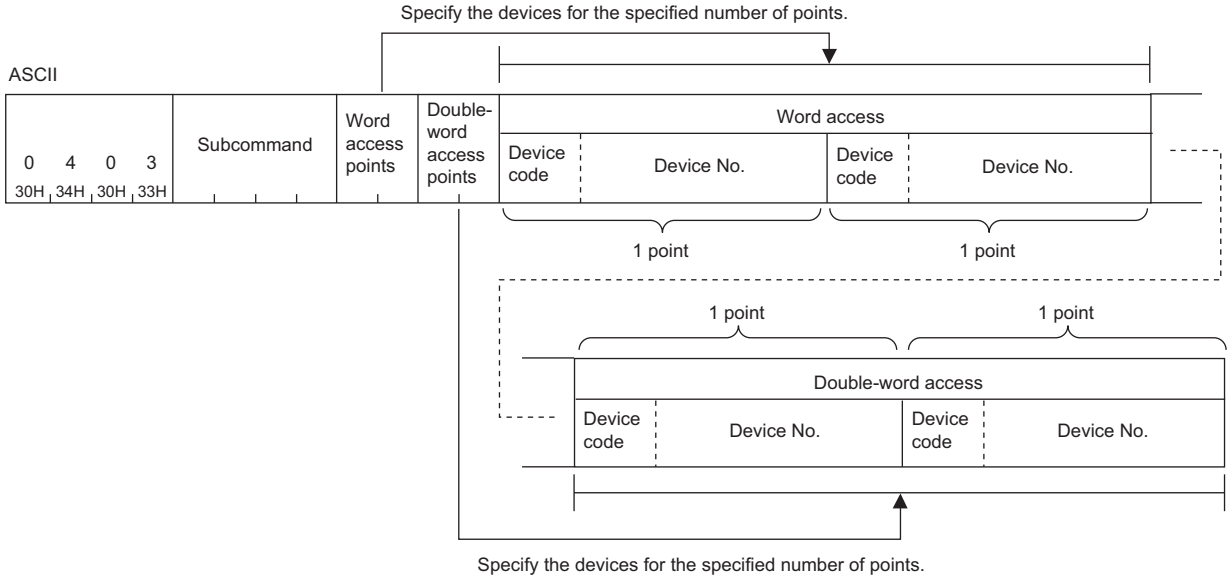
Read Random (command: 0403)

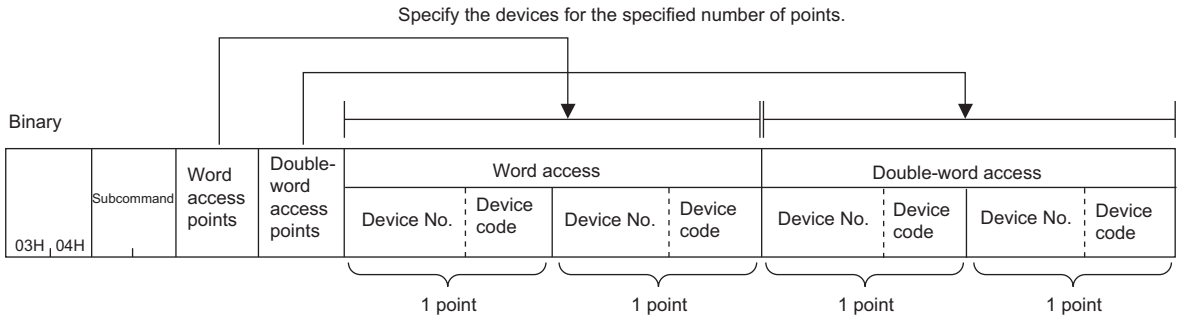
Specifies the device No. and reads the device value. This can be specified with inconsecutive device No.

Point

Do not execute the Read Random command to the CPU module during the conditional monitoring. The command of SLMP completes abnormally.
The command can be executed during unconditional monitoring.

Request data





■Subcommand

Subcommand*1	
ASCII code	**Binary code**
0 0 0 0 (30H 30H 30H 30H) or 0 0 8 0 (30H 30H 38H 30H)	00H 00H or 80H 00H
0 0 0 2 (30H 30H 30H 32H) or 0 0 8 2 (30H 30H 38H 32H)	02H 00H or 82H 00H

*1 The subcommand 008☐ is used to access the link direct device, module access device, or CPU buffer memory access device. When the subcommand is 008☐, the message format is different. (☞ Page 195 Read or Write by Device Extension Specification)

■Number of word access points, number of double-word access points

Specify the number of target device points of reading.

☞ Page 34 Number of device points

☞ Page 49 Communication example

Subcommand	Item	Description	Number of points
0002	Number of 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 ≤ number of word access points + number of double-word access points ≤ 96
	Number of 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.	
0000	Number of 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 ≤ number of word access points + number of double-word access points ≤ 192*1
	Number of double-word access points	Specify the number of points to be accessed in two-word units. The bit device is 32-point units, the word device is two-word units.	

*1 When the file register (ZR) of the High Performance model QCPU is specified, double the number of access points. In addition, when the subcommand 008□ is used, double the number of access points.

■Device code, device No.

Specify the device to be read in order from the word access to the double-word access.

☞ Page 30 Device code

☞ Page 33 Head device No. (Device No.)

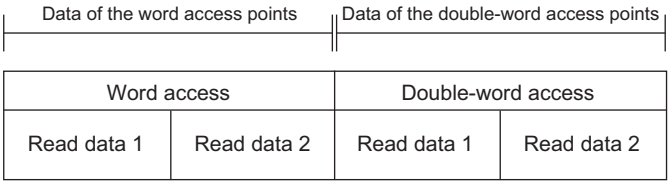
Item	Description
Word access	Specify the device of points designated by "number of word access points". The specification is not necessary when "number of word access points" is zero.
Double-word access	Specify the device of points designated by "number of double-word access points". The specification is not necessary when "number of double-word access points" is zero.

Response data

The value read from the device is stored in hexadecimal. The data order differs depending on the type of code, ASCII code or binary code.

☞ Page 35 Read data, write data

☞ Page 49 Communication example



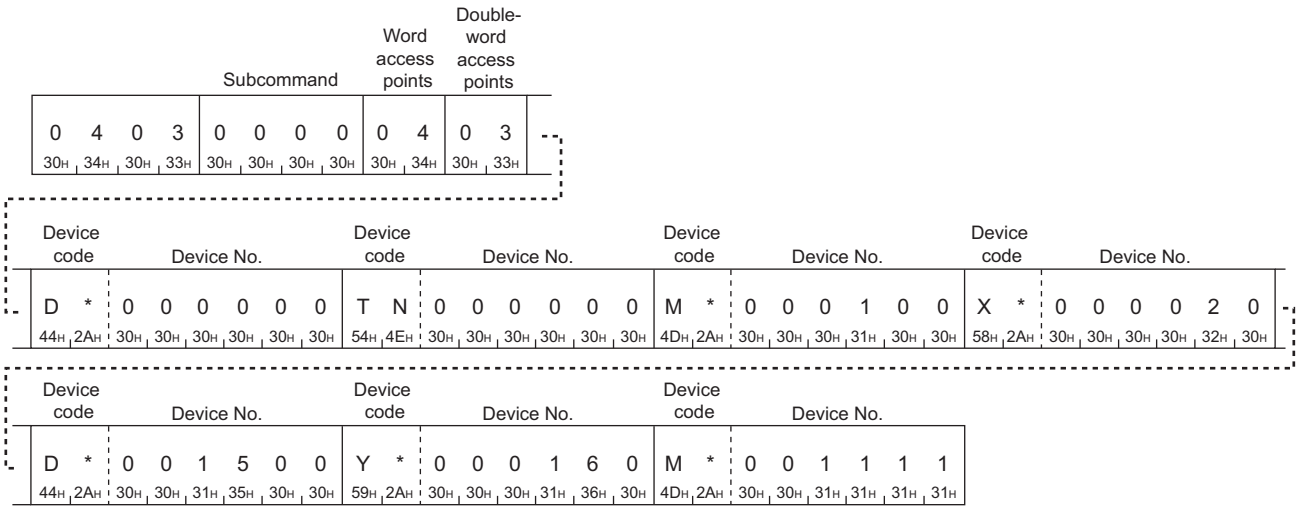
Communication example

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.

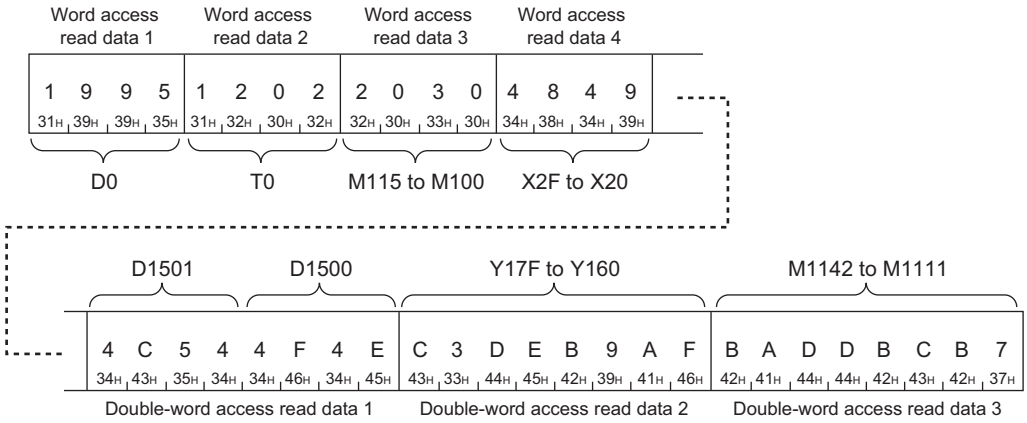
If D0 = 6549 (1995H), T0 = 4610 (1202H), D1500 = 20302 (4F4EH), D1501 = 19540 (4C54H) are stored, the response message will be as follows.

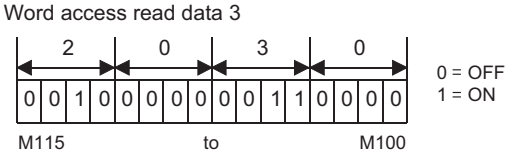
■When communicating data in ASCII code

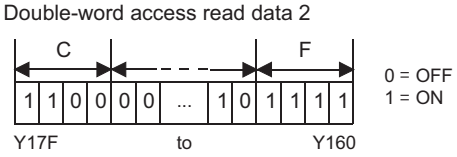
(Request data)

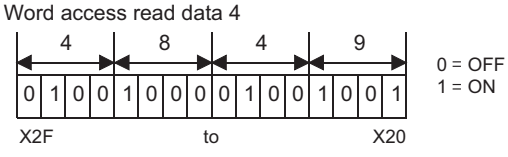


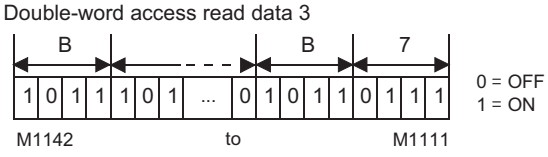
(Response data)





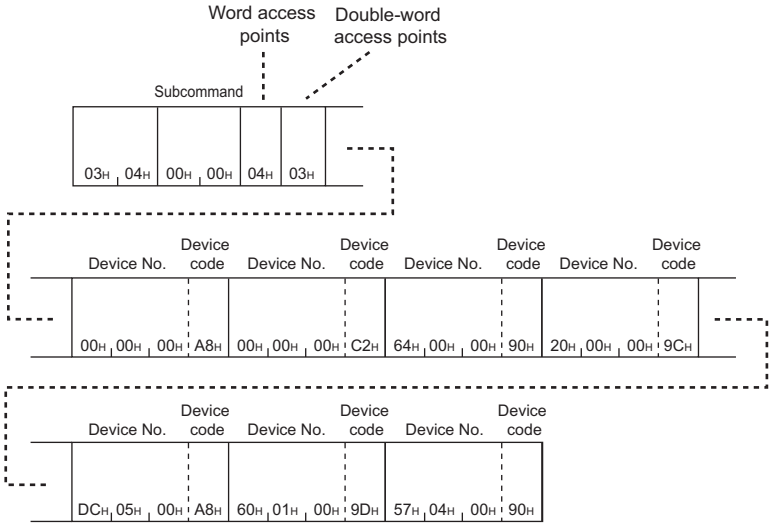




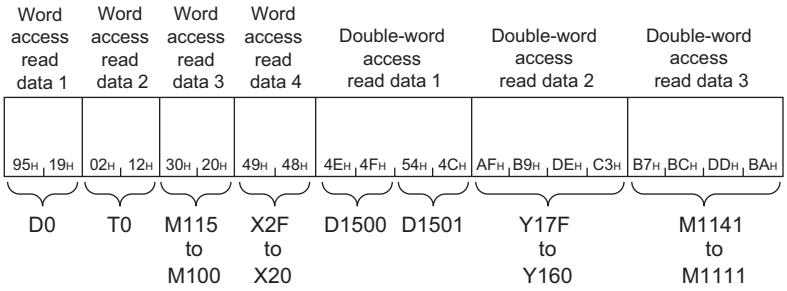


■When communicating data in binary code

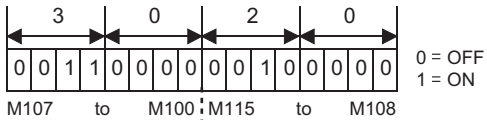
(Request data)



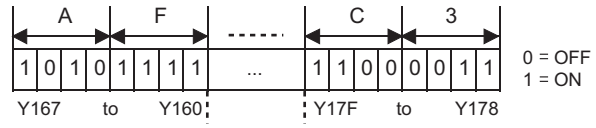
(Response data)



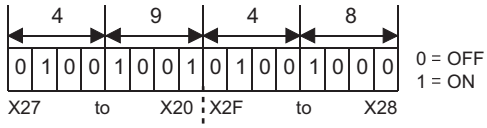
Word access read data 3



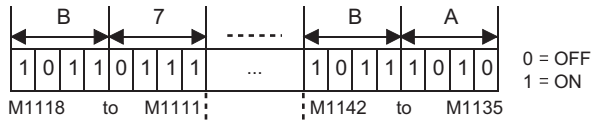
Double-word access read data 2



Word access read data 4



Double-word access read data 3

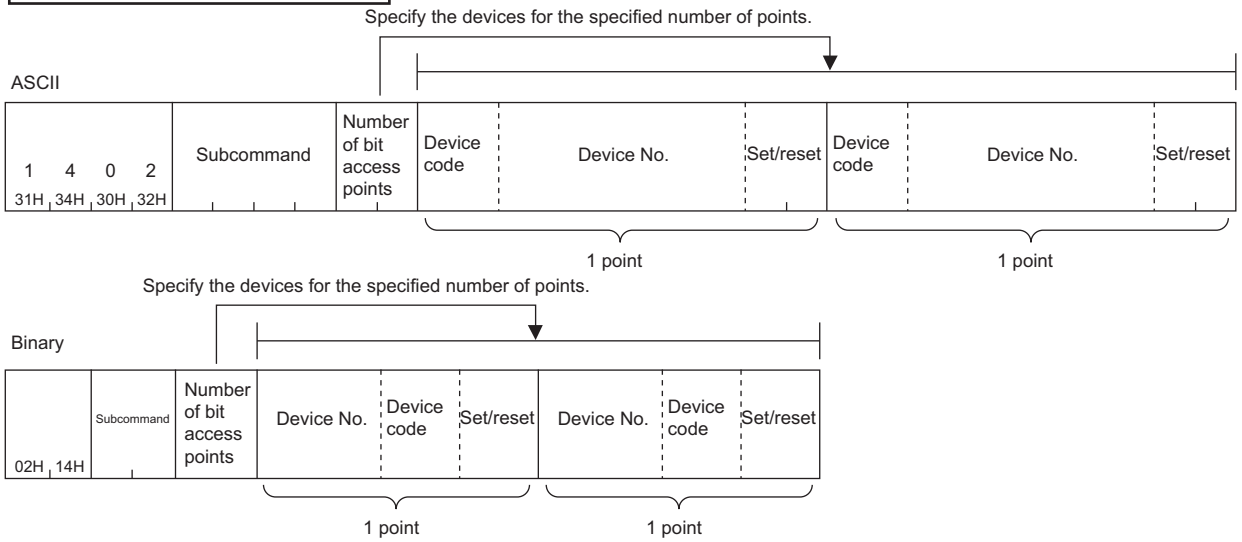


Write Random (command: 1402)

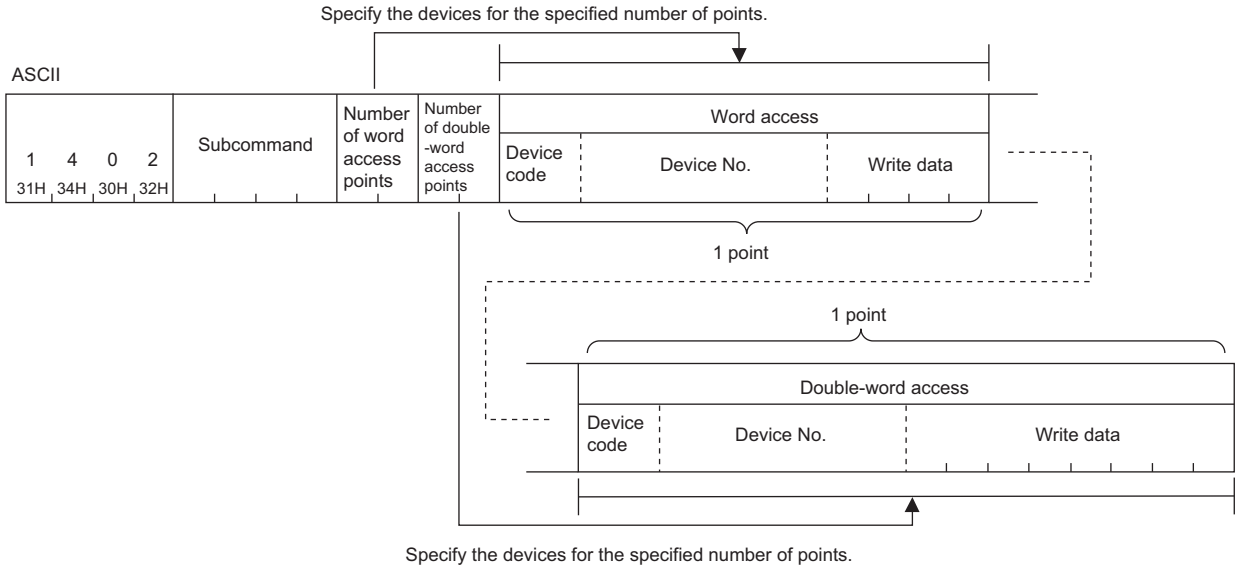
This command specifies the device No. and writes value to the device. This can be specified with inconsecutive device No.

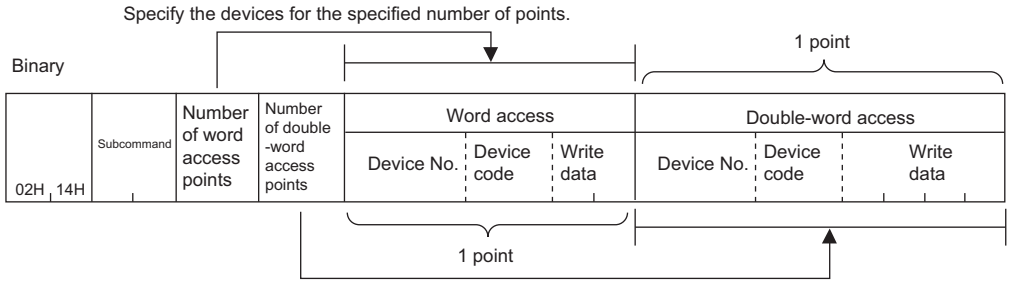
Request data

When writing data in bit units



When writing data in word units





■Subcommand

Item	Subcommand*1	
	ASCII code	Binary code
When writing data in bit units	0 0 0 1 (30H 30H 30H 31H) or 0 0 8 1 (30H 30H 38H 31H)	(01H 00H) or (81H 00H)
	0 0 0 3 (30H 30H 30H 33H) or 0 0 8 3 (30H 30H 38H 33H)	(03H 00H) or (83H 00H)
When writing data in word units	0 0 0 0 (30H 30H 30H 30H) or 0 0 8 0 (30H 30H 38H 30H)	(00H 00H) or (80H 00H)
	0 0 0 2 (30H 30H 30H 32H) or 0 0 8 2 (30H 30H 38H 32H)	(02H 00H) or (82H 00H)

*1 The subcommand 008☐ is used to access the link direct device, module access device, or CPU buffer memory access device. When the subcommand is 008☐, the message format is different. (☞ Page 195 Read or Write by Device Extension Specification)

■Number of bit access points, number of word access points, number of double-word access points

Specify the target device points of writing.

☞ Page 34 Number of device points

☞ Page 53 Communication example (when writing data in bit units)

Subcommand	Item	Description	Number of points
0003 0002	Number of bit access points	Specify the number of bit device points in one-point units.	1 to 94
	Number of 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 ≤ number of word access points × 12 + number of double-word access points × 14 ≤ 960
	Number of 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.	
0001 0000	Number of bit access points	Specify the number of bit device points in one-point units.	1 to 188
	Number of 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 ≤ number of word access points × 12 + number of double-word access points × 14 ≤ 1920*1
	Number of double-word access points	Specify the number of points to be accessed in two-word units. The bit device is 32-point units, the word device is two-word units.	

*1 When the access destination is the MELSEC iQ-R series module and the subcommand 008☐ is used, double the number of access points.

■Device code, device No., write data

Specify the target device of writing.

☞ Page 30 Device code

☞ Page 33 Head device No. (Device No.)

☞ Page 35 Read data, write data

The data is specified in hexadecimal number.

Item	Description
Word access	Specify the device of points designated by "number of word access points". The specification is not necessary when "number of word access points" is zero.
Double-word access	Specify the device of points designated by "number of double-word access points". The specification is not necessary when "number of double-word access points" is zero.

■Set/reset

Specify ON/OFF of the bit device.

Item	Subcommand	Data to write		Remark
		ON	OFF	
ASCII code	0003 0002	"0001"	"0000"	Four digits will be sent from 0 in order.
	0001 0000	"01"	"00"	Two digits will be sent from 0 in order.
Binary code	0003 0002	0100H	0000H	The 2-byte numerical value shown left will be sent.
	0001 0000	01H	00H	The one-byte numerical value shown left will be sent.

Response data

There is no response data for 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/reset	Device code	Device No.	Set/reset
	1 4 0 2	0 0 0 1	0 2	M *	0 0 0 0 5 0	0 0	Y *	0 0 0 0 2 F	0 1
	31H 34H 30H 32H	30H 30H 30H 31H	30H 32H	4DH 2AH	30H 30H 30H 30H 35H 30H	30H 30H	59H 2AH	30H 30H 30H 30H 32H 46H	30H 31H

■When communicating data in binary code

(Request data)

	Subcommand	Bit access points	Device No.	Device code	Set/reset	Device No.	Device code	Set/reset
	02H 14H	01H 00H	02H	32H 00H 00H	90H	00H	2FH 00H 00H	9DH	01H

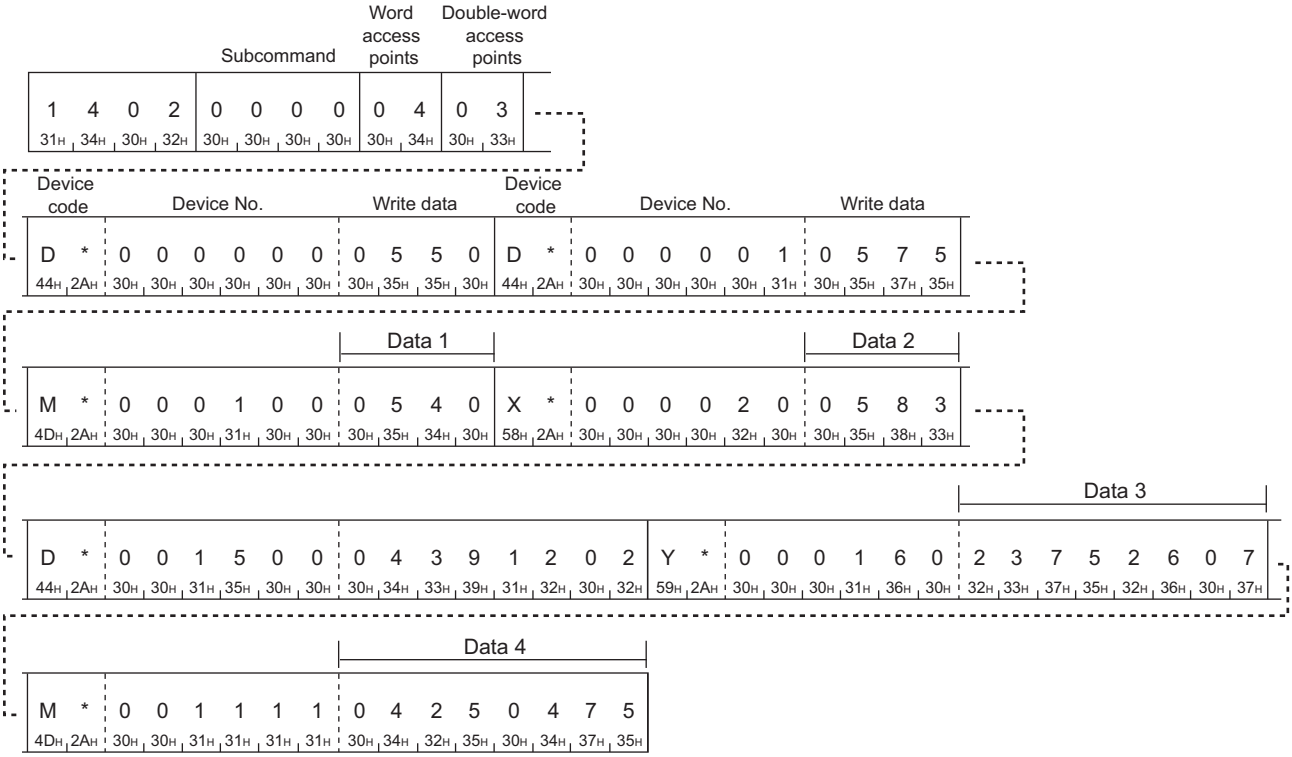
Communication example (when writing data in word units)

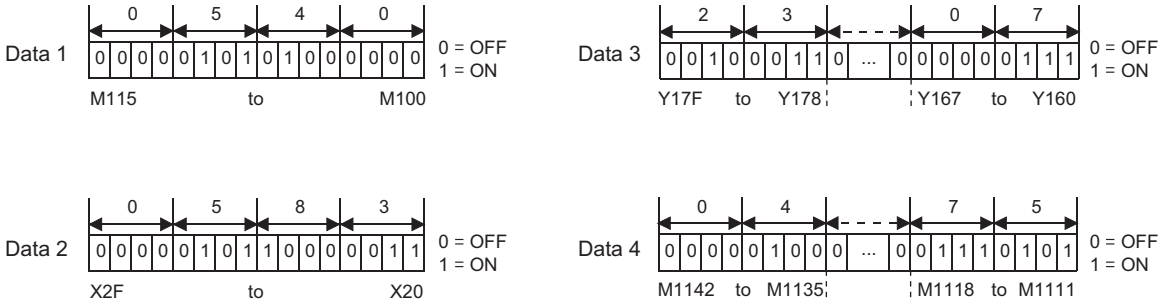
Write the value in a device as follows.

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

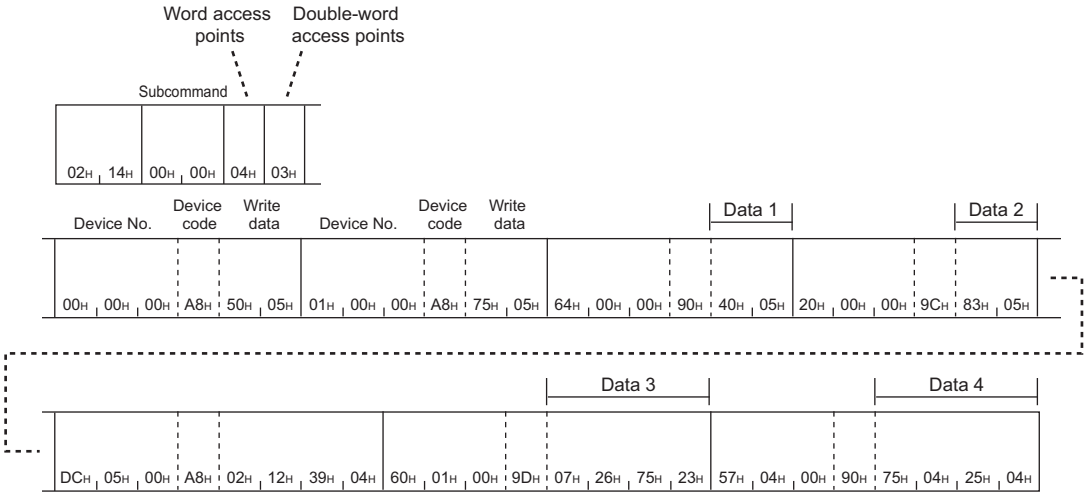
(Request data)

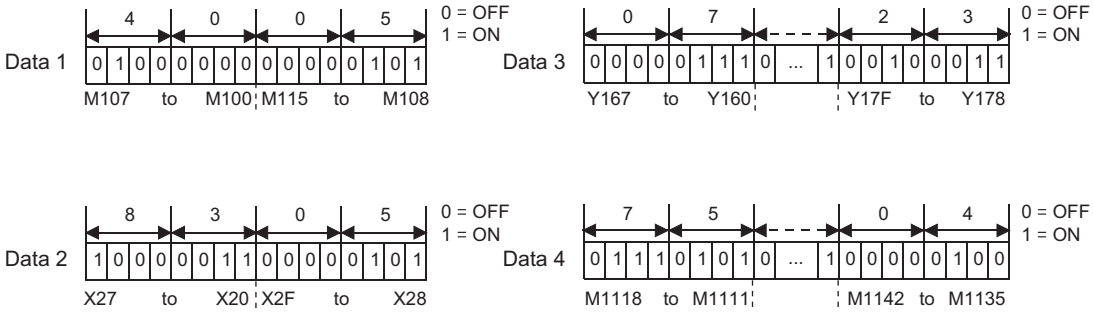




■When communicating data in binary code

(Request data)





Entry Monitor Device (command: 0801)

This command registers a device to be read by Execute Monitor (command: 0802). Registering the device in advance reduces the load of line because it shortens the request message when reading.
Entry Monitor Device (command: 0801) and Execute Monitor (command: 0802) are used as follows.

1. Monitoring device registration

By Entry Monitor Device (command: 0801), register a device to be read.

2. Monitoring execution

Execution of Execute Monitor (command: 0802) will read values from the device registered by Entry Monitor Device (command: 0801). (☞ Page 60 Execute Monitor (command: 0802))

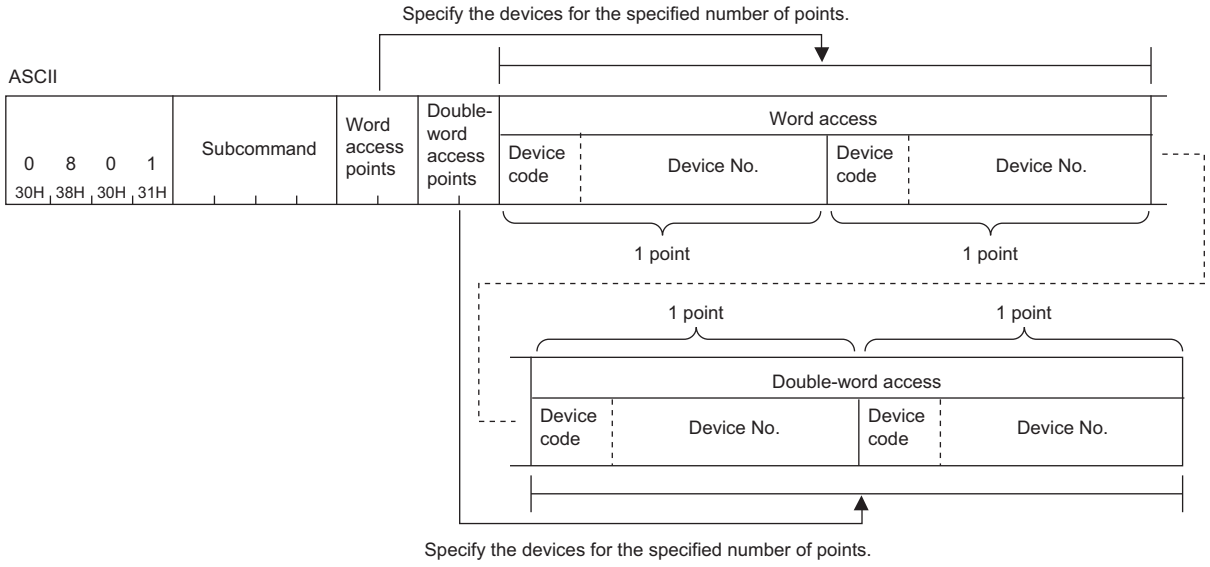
3. Monitoring device change

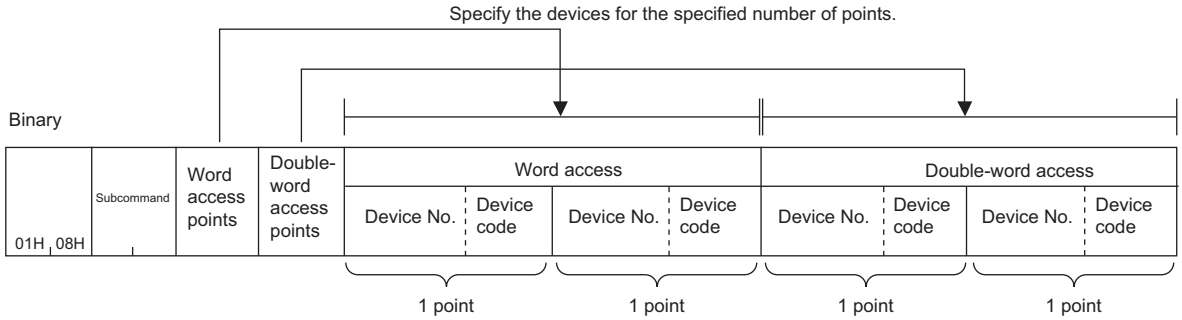
The device to be read can be changed by Entry Monitor Device (command: 0801).

Point

- Do not execute the Entry Monitor Device command to the CPU module during the conditional monitoring. The command of SLMP completes abnormally. The command can be executed during unconditional monitoring.
- If the access destination is restarted, the registered data will be deleted. Execute Entry Monitor Device again and register the device to be read.

Request data





■Subcommand*1

Subcommand*1	
ASCII code	**Binary code**
0 0 0 0 (30H 30H 30H 30H) or 0 0 8 0 (30H 30H 38H 30H)	00H 00H or 80H 00H
0 0 0 2 (30H 30H 30H 32H) or 0 0 8 2 (30H 30H 38H 32H)	02H 00H or 82H 00H

*1 The subcommand 008☐ is used to access the link direct device, module access device, or CPU buffer memory access device. When the subcommand is 008☐, the message format is different. (☞ Page 195 Read or Write by Device Extension Specification)

■Number of word access points, number of double-word access points

Specify the number of target device points of reading.

☞ Page 34 Number of device points

☞ Page 59 Communication example

Subcommand	Item	Description	Number of points
0002	Number of 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 ≤ number of word access points + number of double-word access points ≤ 96
	Number of 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.	
0000	Number of 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 ≤ number of word access points + number of double-word access points ≤ 192*1
	Number of double-word access points	Specify the number of points to be accessed in two-word units. The bit device is 32-point units, the word device is two-word units.	

*1 When the file register (ZR) of the High Performance model QCPU is specified, double the number of access points. In addition, when the subcommand 008□ is used, double the number of access points.

■Device code, device No.

Specify the device to be read in order from the word access to the double-word access.

☞ Page 30 Device code

☞ Page 33 Head device No. (Device No.)

Item	Description
Word access	Specify the device of points designated by "number of word access points". The specification is not necessary when "number of word access points" is zero.
Double-word access	Specify the device of points designated by "number of double-word access points". The specification is not necessary when "number of double-word access points" is zero.

Point

The contact and coil of timer, long timer, retentive timer, long retentive timer, counter, and long counter cannot be specified.

Response data

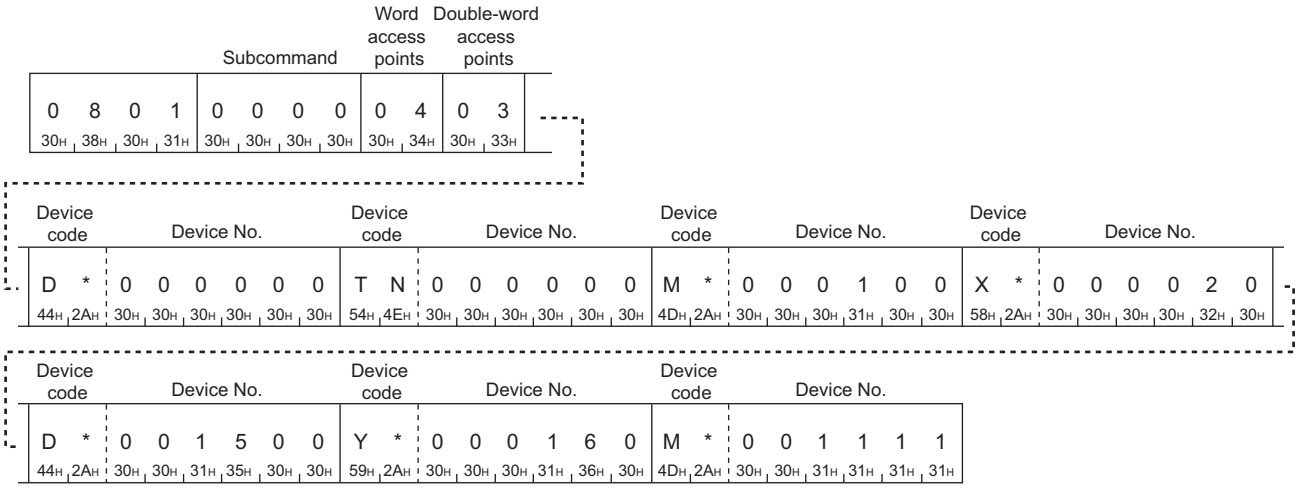
There is no response data for Entry Monitor Device.

Communication example

The following shows an example to register the devices for reading 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.

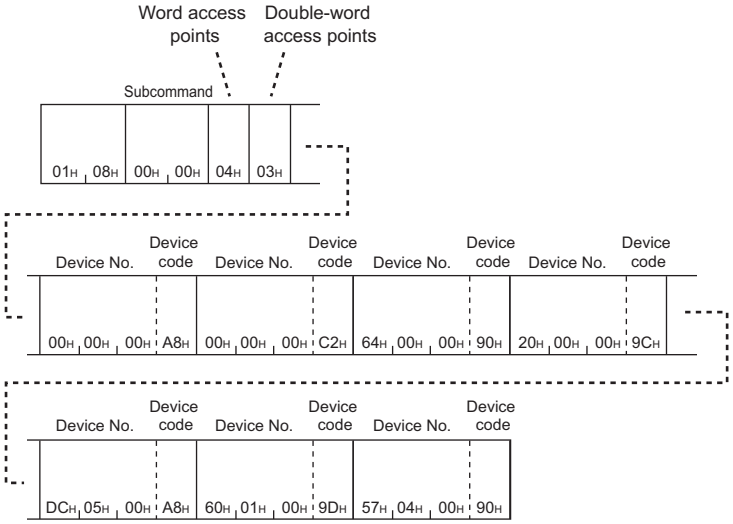
■When communicating data in ASCII code

(Request data)



■When communicating data in binary code

(Request data)



Execute Monitor (command: 0802)

This command reads the value of the device registered by Entry Monitor Device (command: 0801).

Point

- Before executing Execute Monitor (command: 0802), register the device to be read by Entry Monitor Device (command: 0801). Using this command without Entry Monitor Device (command: 0801) executed causes an error. (☞ Page 56 Entry Monitor Device (command: 0801))
- If the access destination is restarted, the registered data will be deleted. Execute the Entry Monitor Device (command: 0801) again and register the device to read.

Request data

ASCII

0	8	0	2	0	0	0	0
30H	38H	30H	32H	30H	30H	30H	30H

Binary

02H	08H	00H	00H

Response data

Data of the word access points		Data of the double-word access points	
Word access		Double-word access	
Read data 1	Read data 2	Read data 1	Read data 2

Communication example

Read the value from the device registered on Page 59 Communication example.

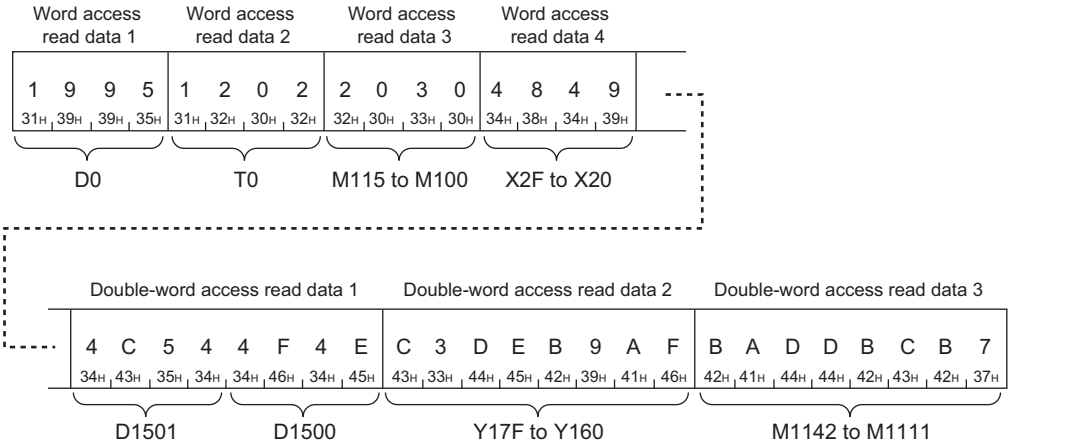
D0 = 6549 (1995H), T0 = 4610 (1202H), D1500 = 20302 (4F4EH), D1501 = 19540 (4C54H) are assumed to be stored.

■When communicating data in ASCII code

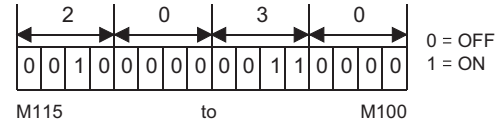
(Request data)

0	8	0	2	0	0	0	0
30H	38H	30H	32H	30H	30H	30H	30H

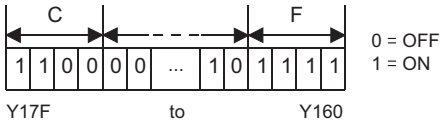
(Response data)



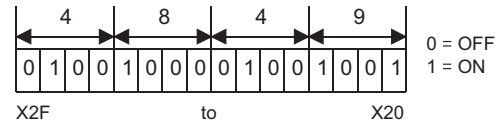
Word access read data 3



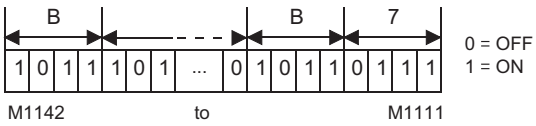
Double-word access read data 2



Word access read data 4

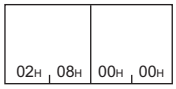


Double-word access read data 3

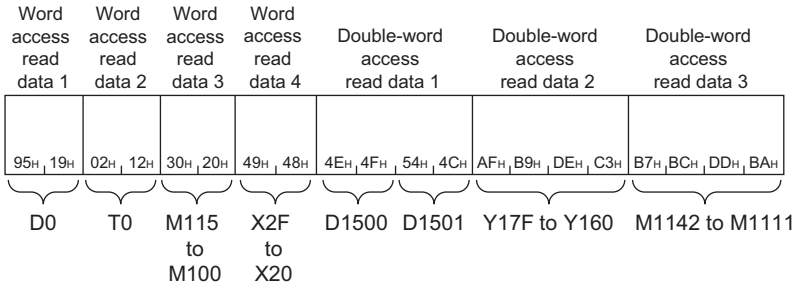


■When communicating data in binary code

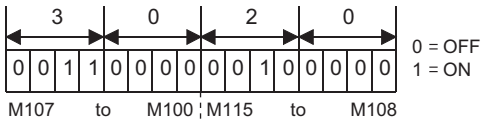
(Request data)



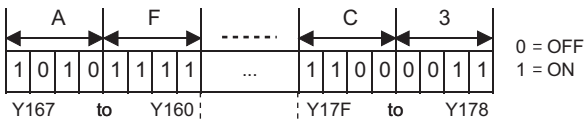
(Response data)



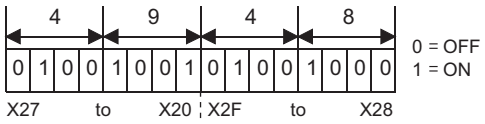
Word access read data 3



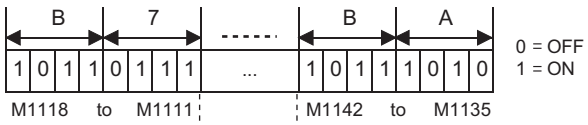
Double-word access read data 2



Word access read data 4



Double-word access read data 3



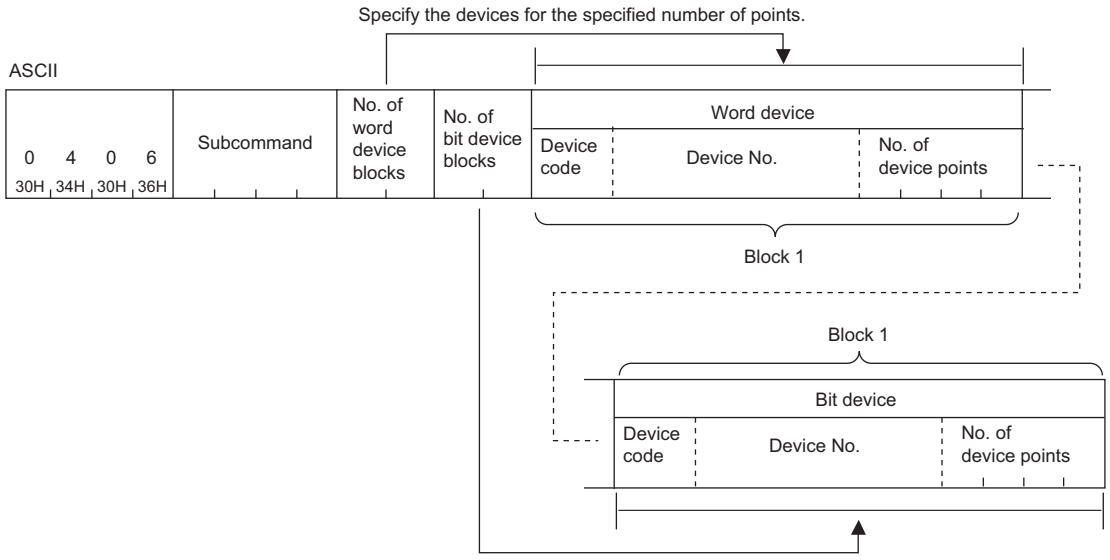
Read Block (command: 0406)

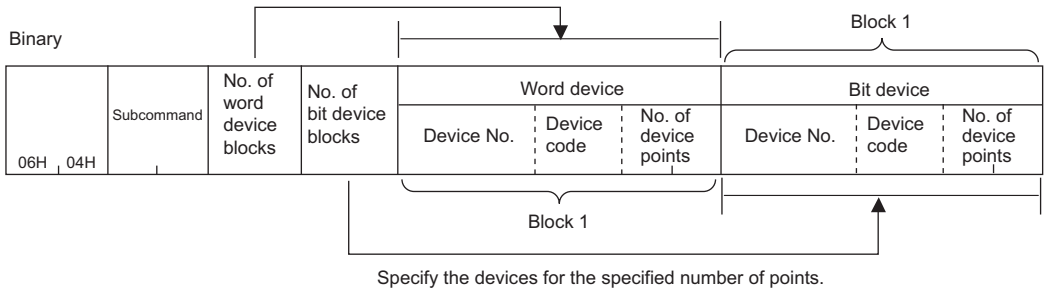
This command reads data by treating n points of word devices or bit devices (one point is equivalent to 16 bits) as one block and specifying multiple blocks. This can be specified with inconsecutive device No.

Point

When the SLMP compatible device communicates with the Universal model QCPU or LCPU, the data inconsistency may occur due to the settings other than "Specify service process execution counts" in "Service Processing Setting" of the CPU module. To prevent the data inconsistency, set "Specify service process execution counts".

Request data





■Subcommand

Subcommand*1	
ASCII code	**Binary code**
0 0 0 0 (30H 30H 30H 30H) or 0 0 8 0 (30H 30H 38H 30H)	00H 00H or 80H 00H
0 0 0 2 (30H 30H 30H 32H) or 0 0 8 2 (30H 30H 38H 32H)	02H 00H or 82H 00H

*1 The subcommand 008☐ is used to access the link direct device, module access device, or CPU buffer memory access device. When the subcommand is 008☐, the message format is different. (☞ Page 195 Read or Write by Device Extension Specification)

■Number of word device blocks, number of bit device blocks

Specify the number of blocks of the device to be read in hexadecimal number.

Subcommand	Item	Description	Number of points
0002	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 ≤ 60
	Number of bit device blocks	Specify the number of blocks of the bit device blocks to be read.	
0000	Number of word device blocks	Specify the number of blocks of the word device to be read.	Number of word device blocks + number of bit device blocks ≤ 120*1
	Number of bit device blocks	Specify the number of blocks of the bit device blocks to be read.	

*1 When the access destination is the MELSEC iQ-R series module and the subcommand 008□ is used, double the number of blocks.

■Device code, device No., number of device points

Specify the target device of reading.

☞ Page 30 Device code
☞ Page 33 Head device No. (Device No.)
☞ Page 34 Number of device points

However, the contact, coil, or current value of the following devices cannot be specified.

- Long timer (LTS, LTC, LTN)
- Long retentive timer (LSTS, LSTC, LSTN)
- Long counter (LCS, LCC, LCN)

Specify the number of device points to fulfill the following conditions.

- Total points of each block of the word device + total points of each block of the bit device ≤ 960

Specify the device number in order from the word device to the bit device.

Item	Description
Word device	Specify the device of points designated by "number of word device blocks". The specification is not necessary when "number of word device blocks" is zero.
Bit device	Specify the device of points designated by "number of bit device blocks". The specification is not necessary when "number of bit device blocks" is zero.

Point

When specifying the contact or coil of timer, retentive timer, or counter, use a bit device block.

Response data

The value read from the device is stored in hexadecimal. The data order differs depending on the type of code, ASCII code or binary code. (☞ Page 35 Read data, write data)



Word device		Bit device	
1st block data	2nd block data	1st block data	2nd block data

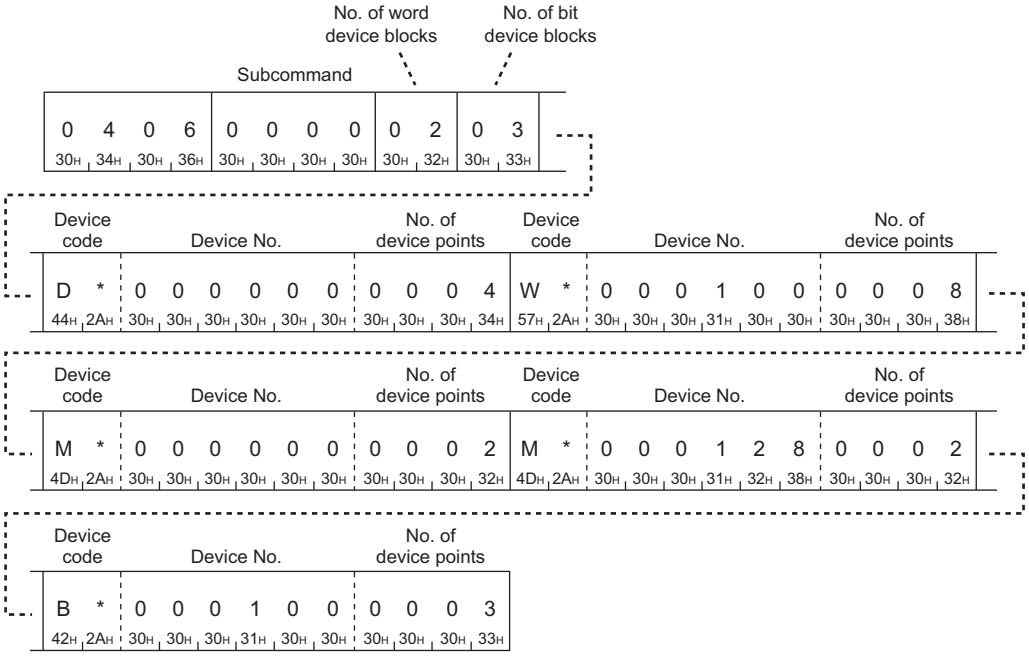
Communication example

Read the value from devices as follows.

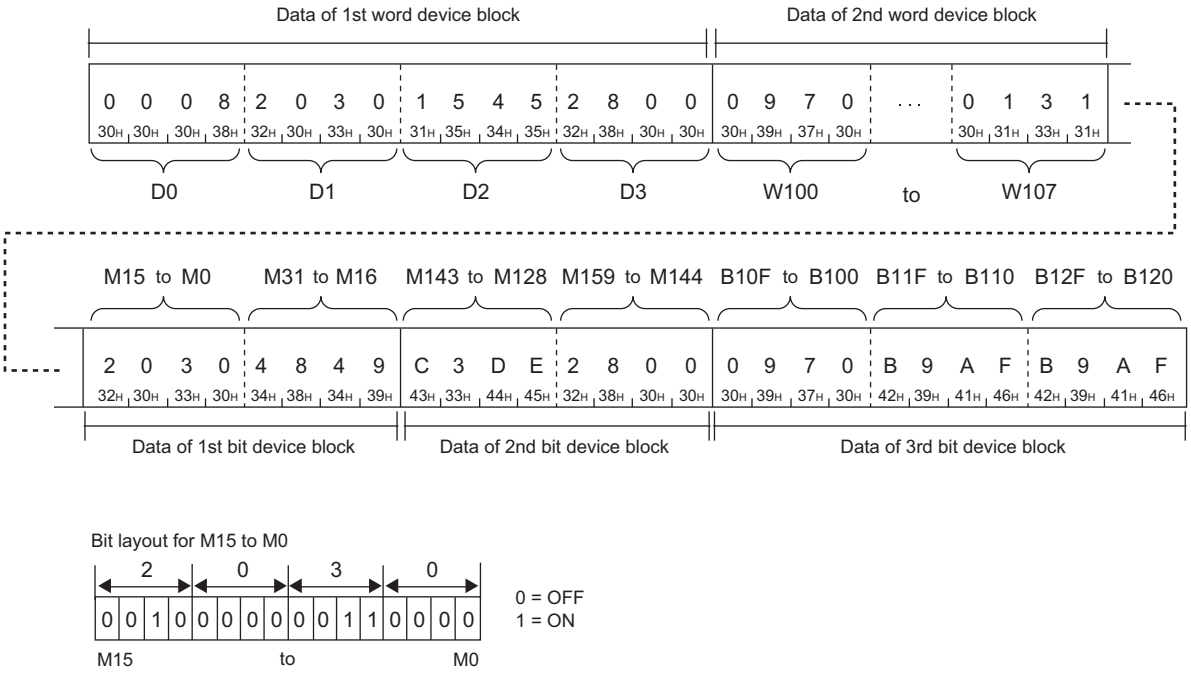
Item	Reading data
Word device	• Block 1: D0 to D3 (4 points) • Block 2: W100 to W107 (8 points)
Bit device	• Block 1: M0 to M31 (2 points) • Block 2: M128 to M159 (2 points) • Block 3: B100 to B12F (3 points)

■When communicating data in ASCII code

(Request data)



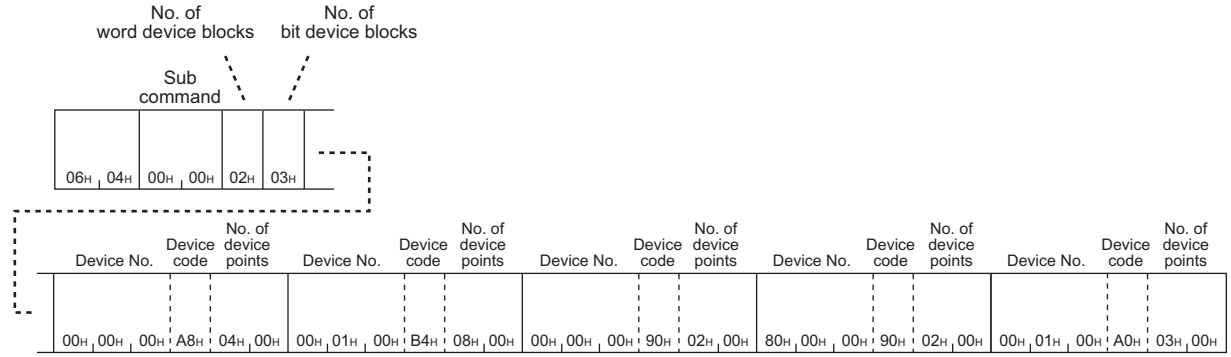
(Response data)



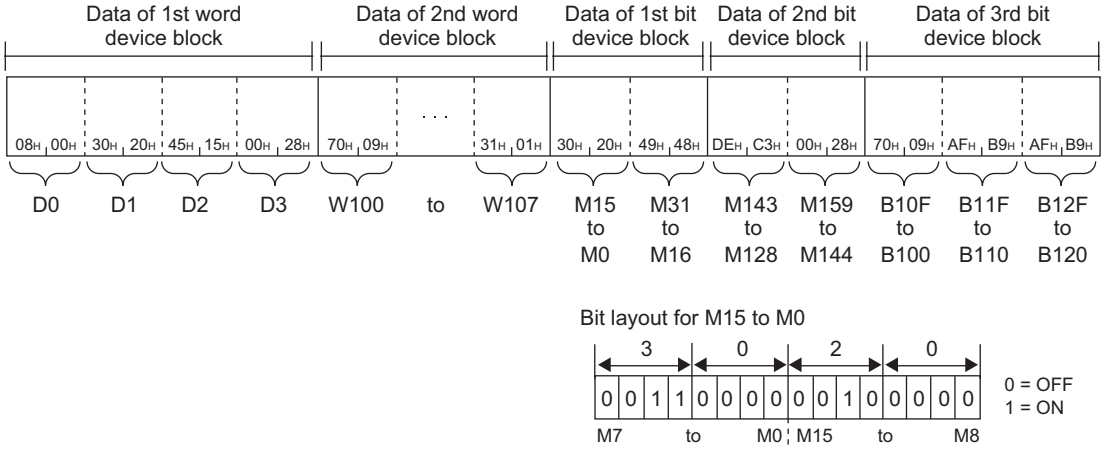
Bit layout for M15 to M0

■When communicating data in binary code

(Request data)



(Response data)



Bit layout for M15 to M0

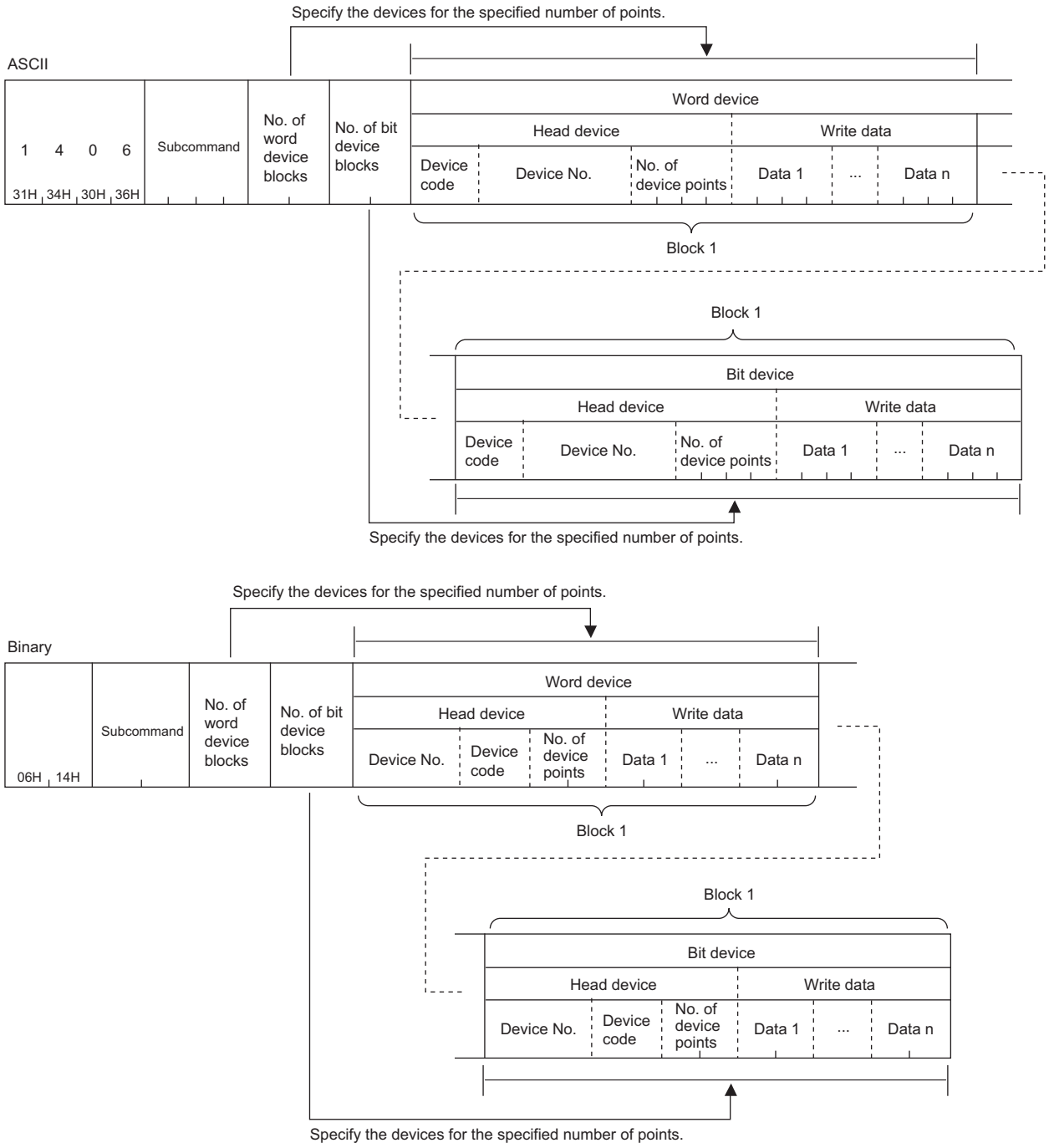
Write Block (command: 1406)

This command writes data by treating n points of word devices or bit devices (one point is equivalent to 16 bits) as one block and specifying multiple blocks. This can be specified with inconsecutive device No.

Point

When the SLMP compatible device communicates with the Universal model QCPU or LCPU, the data inconsistency may occur due to the settings other than "Specify service process execution counts" in "Service Processing Setting" of the CPU module. To prevent the data inconsistency, set "Specify service process execution counts".

Request data



■Subcommand

Subcommand[*1]	
ASCII code	**Binary code**
0 0 0 0 (30H 30H 30H 30H) or 0 0 8 0 (30H 30H 38H 30H)	(00H 00H) or (80H 00H)
0 0 0 2 (30H 30H 30H 32H) or 0 0 8 2 (30H 30H 38H 32H)	(02H 00H) or (82H 00H)

*1 The subcommand 008☐ is used to access the link direct device, module access device, or CPU buffer memory access device. When the subcommand is 008☐, the message format is different. (☞ Page 195 Read or Write by Device Extension Specification)

■Number of word device blocks, number of bit device blocks

Specify the number of blocks of the device to be written in. (☞ Page 34 Number of device points)

Subcommand	Item	Description	Number of points
0002	Number of word device blocks	Specifies the number of blocks of the word device to be written in.	Number of word device blocks + number of bit device blocks ≤ 60
	Number of bit device blocks	Specify the number of blocks of the bit device to be written in.	
0000	Number of word device blocks	Specifies the number of blocks of the word device to be written in.	Number of word device blocks + number of bit device blocks ≤ 120[*1]
	Number of bit device blocks	Specify the number of blocks of the bit device to be written in.	

*1 When the access destination is the MELSEC iQ-R series module and the subcommand 008☐ is used, double the number of blocks.

■Device code, device No., number of device points

Specify the target device of writing.

☞ Page 30 Device code

☞ Page 33 Head device No. (Device No.)

☞ Page 34 Number of device points

However, the contact, coil, or current value of the following devices cannot be specified.

- Long timer (LTS, LTC, LTN)
- Long retentive timer (LSTS, LSTC, LSTN)
- Long counter (LCS, LCC, LCN)

Specify the number of device points to fulfill the following conditions.

Subcommand	Condition
0002	(Number of word device blocks + number of bit device blocks) × 9 + total points of each block of a word device + total points of each block of a bit device ≤ 960
0000	(Number of word device blocks + number of bit device blocks) × 4 + total points of each block of a word device + total points of each block of a bit device ≤ 960

Specify the device number in order from the word device to the bit device.

Item	Description
Word device	Specify the device of points designated by "number of word device blocks". The specification is not necessary when "number of word device blocks" is zero.
Bit device	Specify the device of points designated by "number of bit device blocks". The specification is not necessary when "number of bit device blocks" is zero.

Point

When specifying the contact or coil of timer, retentive timer, or counter, use a bit device block.

Response data

There is no response data for Write Block command.

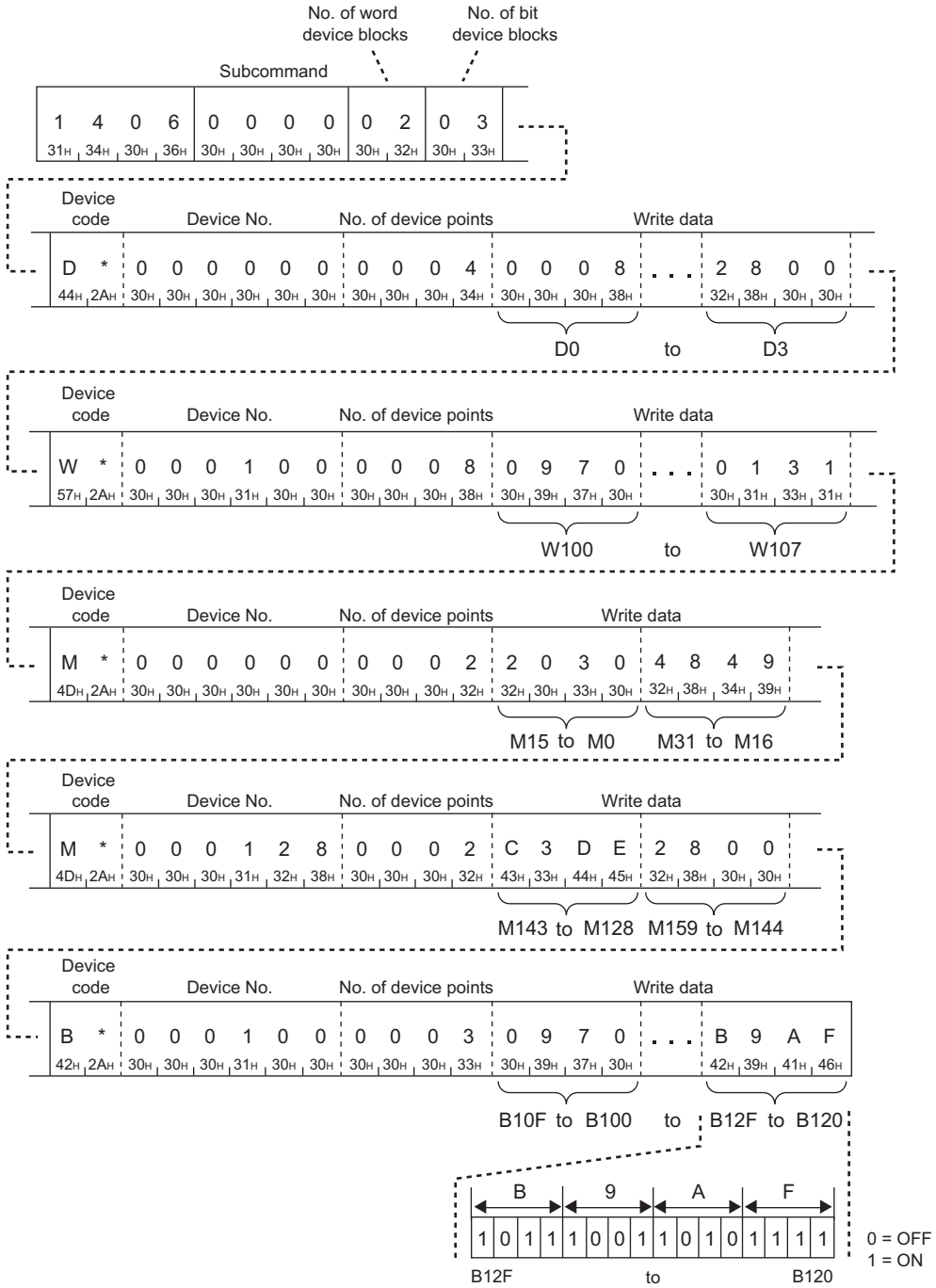
Communication example

Write the value in a device as follows.

Item	Writing data
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)

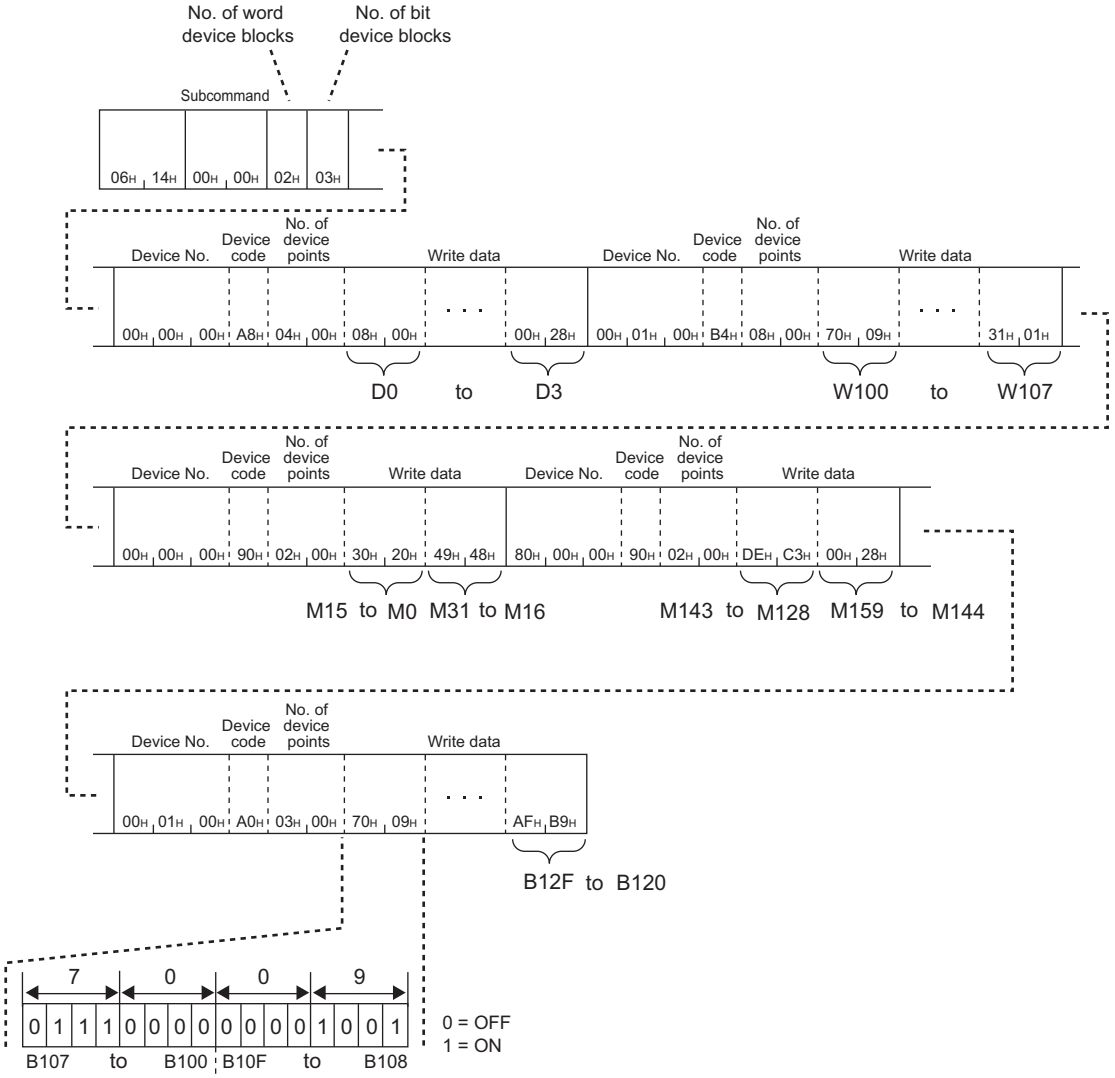
■When communicating data in ASCII code

(Request data)



■When communicating data in binary code

(Request data)



5.3 Label (Label Access)

This section describes commands which read or write data with a global label.

Restriction

- Local labels cannot be accessed.
- Global labels set in GX Works2 cannot be accessed.
- To access the global label, "Access from External Device" must be enabled with the global label editor in GX Works3. (The default setting is set to disabled.)
- When communicating data in ASCII code, the size of messages increases because the label name must be converted from UTF-16 to ASCII code.

Data to be specified in command

Number of array points

Specify the number of arrays to be read or written.

The maximum number that can be specified changes depending on the label name length because the maximum capacity of the send data is 1920 bytes.

■When communicating data in ASCII code

Convert the points to a 4-digit ASCII code, and send it in order from the upper byte to the lower byte.

Ex.

When the number of points is three

0	0	0	3
30H	30H	30H	33H

■When communicating data in binary code

Use numerical values in 2 bytes which indicate the number of points, and send it in order from the lower byte to the upper byte.

Ex.

When the number of points is three

03H	00H

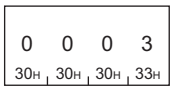
Number of read/write data points

Specify the number of labels to be read or written.
The maximum number that can be specified changes depending on the label name length because the maximum capacity of the send data is 1920 bytes.

■When communicating data in ASCII code

Convert the number of labels to a 4-digit ASCII code, and send it in order from the upper byte to the lower byte.

Ex.
When the number of labels is three



■When communicating data in binary code

Use numerical values in 2 bytes which indicate the number of labels, and send them in order from the lower byte to the upper byte.

Ex.
When the number of labels is three



Number of abbreviation points

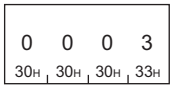
Specify the number of labels to which the abbreviation definition is applied. Specify 0 when the abbreviation definition is not used.

The abbreviation definition indicates that the label name is noted in an abbreviated form, such as "%1", "%2" ··· "%n" (n: specified number of points to be abbreviated). (☞ Page 76 Label name)

■When communicating data in ASCII code

When communicating data in ASCII code, convert the number of abbreviation points to four digits, and send it in order from the upper byte to the lower byte.

Ex.
When the number of abbreviation points is three



■When communicating data in binary code

When communicating data in binary code, use numerical values in 2 bytes which indicate the number of abbreviation points, and send them in order from the lower byte to the upper byte.

Ex.
When the number of abbreviation points is three



Label name length

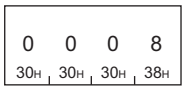
Specify the number of label name characters set in "Label name".

■When communicating data in ASCII code

Convert the number of characters to a 4-digit ASCII code, and send them in order from the upper byte to the lower byte.

Ex.

For eight characters



■When communicating data in binary code

Use 2-byte numerical values for the number of characters, and send them in order from the lower byte to the upper byte.

Ex.

For eight characters



Label name

Specify the label name.

- When communicating data in ASCII code, convert a UTF-16 value that indicates the global label name to an ASCII code, and send it in order from the upper byte to the lower byte.
- When communicating data in binary code, send a UTF-16 value that indicates the global label name in order from the lower byte to the upper byte.

■Label of primitive data type

Specify the global label name.

The following table lists the specification example of ASCII code and binary code when the global label name is "AAA".

Label name (UTF-16 (hexadecimal))	A (0041)	A (0041)	A (0041)
ASCII code (hexadecimal)	30303431	30303431	30303431
Binary code (hexadecimal)	4100	4100	4100

■Label of array specified type

Specify the label name and index (element number) of up to three-dimensional array elements.

The following table lists the specification example of ASCII code and binary code when the global label name is one-dimensional array "BBB[20]".

Label name (UTF-16 (hexadecimal))	B (0042)	B (0042)	B (0042)	[(005B)	2(0032)	0(0030)	] (005D)
ASCII code (hexadecimal)	30303432	30303432	30303432	30303542	30303332	30303330	30303544
Binary code (hexadecimal)	4200	4200	4200	5B00	3200	3000	5D00

The following table lists the specification example of ASCII code and binary code when the global label name is two-dimensional array "BBB[20,10]".

Label name (UTF-16 (hexadecimal))	B (0042)	B (0042)	B (0042)	[(005B)	2(0032)
ASCII code (hexadecimal)	30303432	30303432	30303432	30303542	30303332
Binary code (hexadecimal)	4200	4200	4200	5B00	3200

Label name (UTF-16 (hexadecimal))	0(0030)	,(002C)	1(0031)	0(0030)	] (005D)
ASCII code (hexadecimal)	30303330	30303243	30303331	30303330	30303544
Binary code (hexadecimal)	3000	2C00	3100	3000	5D00

The following table lists the specification example of ASCII code and binary code when the global label name is three-dimensional array "BBB[20,10,30]".

Label name (UTF-16 (hexadecimal))	B (0042)	B (0042)	B (0042)	[(005B)	2(0032)	0(0030)	,(002C)
ASCII code (hexadecimal)	30303432	30303432	30303432	30303542	30303332	30303330	30303243
Binary code (hexadecimal)	4200	4200	4200	5B00	3200	3000	2C00

Label name (UTF-16 (hexadecimal))	1(0031)	0(0030)	,(002C)	3(0033)	0(0030)	] (005D)	
ASCII code (hexadecimal)	30303331	30303330	30303243	30303333	30303330	30303544	
Binary code (hexadecimal)	3100	3000	2C00	3300	3000	5D00	

■Label of structured type

Connect the element names of the structure with one-byte periods, and specify the character string specified up to the last element.

The following table lists the specification example of ASCII code and binary code when the global label name is "XXX.YYY.ZZZ".

Label name (UTF-16 (hexadecimal))	X (0058)	X (0058)	X (0058)	.(002E)	Y (0059)	Y (0059)
ASCII code (hexadecimal)	30303538	30303538	30303538	30303245	30303539	30303539
Binary code (hexadecimal)	5800	5800	5800	2E00	5900	5900

Label name (UTF-16 (hexadecimal))	Y (0059)	.(002E)	Z (005A)	Z (005A)	Z (005A)	
ASCII code (hexadecimal)	30303539	30303245	30303541	30303541	30303541	
Binary code (hexadecimal)	5900	2E00	5A00	5A00	5A00	

■Label of structured type (when the member is an array)

Combine the specification methods of the label of structured type and label of array specified type.

The following table lists the specification example of ASCII code and binary code when the global label name is "XXX.YYY[20,10,30]".

Label name (UTF-16 (hexadecimal))	X (0058)	X (0058)	X (0058)	.(002E)	Y (0059)	Y (0059)
ASCII code (hexadecimal)	30303538	30303538	30303538	30303245	30303539	30303539
Binary code (hexadecimal)	5800	5800	5800	2E00	5900	5900

Label name (UTF-16 (hexadecimal))	Y (0059)	[(005B)	2(0032)	0(0030)	,(002C)	1(0031)
ASCII code (hexadecimal)	30303539	30303542	30303332	30303330	30303243	30303331
Binary code (hexadecimal)	5900	5B00	3200	3000	2C00	3100

Label name (UTF-16 (hexadecimal))	0(0030)	,(002C)	3(0033)	0(0030)	] (005D)	
ASCII code (hexadecimal)	30303330	30303243	30303333	30303330	30303544	
Binary code (hexadecimal)	3000	2C00	3300	3000	5D00	

■Data type with a label of structured type

When a data type is any of the following, the data is a label of structured type.

- Timer
- Counter
- Long timer
- Retentive timer
- Long retentive timer
- Long timer

The structure has the data type and member names including the contact, coil, and current value.

Member name	Data type	Description
S	Bit	Contact
C	Bit	Coil
N	Timer, counter, or retentive timer: Word [unsigned]/bit string [16 bits]	Current value
	Long timer, long counter, or long retentive timer: Double word [unsigned]/bit string [32 bits]	

■Label that cannot be specified

Label type	Description	Example
Bit specification of label	Specifying the label name and bit specification connected with one-byte periods as a character string is unavailable.	AAA.3
Digit specification of label	Specifying the label name and digit specification as a character string is unavailable.	K4AAA
Label of array specified type	Specifying the element number as a character string is unavailable.	BBB[XXX] BBB[XXX,YYY] BBB[XXX,YYY,ZZZ]
Label of structured type	Specifying the label name of structured type that is not the end member unavailable.	XXX
	Specifying the label to which a device is manually assigned and whose type is the structured type having a member of a label of structured type is unavailable.	Label1.Member1.Member2 (only for a label to which a device is manually assigned)

■Abbreviation definition of label name

For labels of structured type, the label name can be specified in the abbreviated form.

To use the abbreviation definition, specify the number of label names to be abbreviated, and specify and register the label name length and label name of the label to be abbreviated.

However, the label name must be specified in a unit separated by ".". The label name cannot be specified in a character unit.

For example, for the label of structured type of "LabelA.memberA3.memberB1", "LabelA" and "LabelA.memberA3" can be specified as the abbreviated label name. However, the label name abbreviated in a character unit, such as "Label" and "LabelA.member", is unacceptable.

The character string of the registered label can be specified in the abbreviated form consisting of "%" and the offset value (in serial order from 1), such as "%1", "%2" ··· "%n" (n: specified number of points to be abbreviated).

The following shows the procedure to register the labels of structured type shown below with "LabelA" and "memberA3" abbreviated such as "%1.memberA1", "%1.memberA2", "%1.%2.memberB1", and "%1.%2.memberB2".

- LabelA.memberA1
- LabelA.memberA2
- LabelA.memberA3.memberB1
- LabelA.memberA3.memberB2

1. Specify the number of label names to be abbreviated in the number of abbreviated points.

Two label names "LabelA" and "memberA3" are to be abbreviated, and thus specify "two" in the number of abbreviated points.

2. Specify the number of characters of the label names to be abbreviated in the label name length.

Label name	Number of characters	Label name length	
		When communicating data in ASCII code (hexadecimal)	When communicating data in binary code (hexadecimal)
LabelA	6	30303036	0600
memberA3	8	30303038	0800

3. Specify the label name to be noted in the abbreviated form.

Specify the label name to be abbreviated. Repeat the procedure 2 and 3 for the number of abbreviation points specified in the procedure 1.

Data	Value to specify	Description	
Number of abbreviation points	2	Specify the number of points of the label name to be abbreviated.	
Label name length	6	Specify the items for each label name to be abbreviated.	Specify the labels equivalent to the number of specified abbreviated points.
Label name	LabelA		
Label name length	8	Specify the items for each label name to be abbreviated.	
Label name	memberA3		

Restriction

When a label of array specified type is a member of a label of structured type, the abbreviated notation cannot be applied to the label name of array specified type.

Data type ID

The data type ID is stored in the response data.

When communicating data in ASCII code, the data type ID is indicated in a two-digit ASCII code.

When communicating data in binary code, the data type ID is indicated in a one-byte binary code.

The following table lists data type IDs stored in the response data.

Classification	Data type name	Data type ID
Label of primitive data type	Bit	1
	Word [unsigned]/bit string [16 bits]	2
	Double word [unsigned]/bit string [32 bits]	3
	Word [signed]	4
	Double word [signed]	5
	Single-precision real number	6
	Double-precision real number	7
	Hour	8
	Character string	9
	Character string [Unicode]	10
	Contact/coil of the following data types • Timer • Counter • Long timer • Retentive timer • Long retentive timer • Long timer	1
	Current value of the following data types • Timer • Counter • Retentive timer	2
	Current value of the following data types • Long timer • Long retentive timer • Long timer	3
Label of array specified type	Data type of array element (primitive data type)	
Label of structured type	Data type of end element (primitive data type)	

Read unit specification, write unit specification

Specify the unit of the read data length or write data length.

Value	Description
0	Specify this value when the data type of label is a bit. (Bit specification)
1	Specify this value when the data type of label is not a bit. (Byte specification)

The following table lists the read unit specification and write unit specification specified in each data type.

Classification	Data type name	Read unit specification, write unit specification
Label of primitive data type	Bit	0
	Word [unsigned]/bit string [16 bits]	1
	Double word [unsigned]/bit string [32 bits]	1
	Word [signed]	1
	Double word [signed]	1
	Single-precision real number	1
	Double-precision real number	1
	Hour	1
	Character string	1
	Character string [Unicode]	1
	Contact/coil of the following data types • Timer • Counter • Long timer • Retentive timer • Long retentive timer • Long timer	0
	Current value of the following data types • Timer • Counter • Retentive timer	1
	Current value of the following data types • Long timer • Long retentive timer • Long timer	1
Label of array specified type	Data type of array element (primitive data type)	
Label of structured type	Data type of end element (primitive data type)	

■When communicating data in ASCII code

Convert a value to a 2-digit ASCII code, and send it in order from the upper byte to the lower byte.

Ex.

When 0 is specified as the value



■When communicating data in binary code

Use a one byte numerical value that indicates the value to send.

Ex.

When 0 is specified as the value



Fixed value

Specify 0.

■When communicating data in ASCII code

Convert the value to a 2-digit ASCII code, and send it in order from the upper byte to the lower byte.

Ex.



■When communicating data in binary code

Use a one byte numerical value that indicates the value to send.

Ex.



Read data length, write data length

The sizes of the read data and write data of each label are shown in two-byte units.

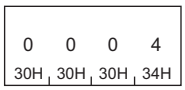
Specify "two" when the data type of the label is a bit. (☞ Page 77 Data type ID)

■When communicating data in ASCII code

Convert the size to a 4-digit ASCII code, and send it in order from the upper byte to the lower byte.

Ex.

When four is specified as the size



■When communicating data in binary code

Send the data in order from the lower byte to the upper byte using 2-byte numeral values that indicates the size.

Ex.

When four is specified as the size



Read array data length, write array data length

Specify the read or write data size of the array label.

Specify the size in the unit specified in the read unit specification or write unit specification (bit or byte). (☞ Page 78 Read unit specification, write unit specification)

For the bit unit, specify the size in units of 16 bits (2 bytes).

The order of sending data is the same as that of "Read data length, write data length". (☞ Page 79 Read data length, write data length)

Array Label Read (command: 041A)

Request data

This command reads data from a label of array specified type or a label of structured type when the members of the label are an array.

This command can read data even from other than a label of array specified type assuming the label having one element of the array.

Restriction

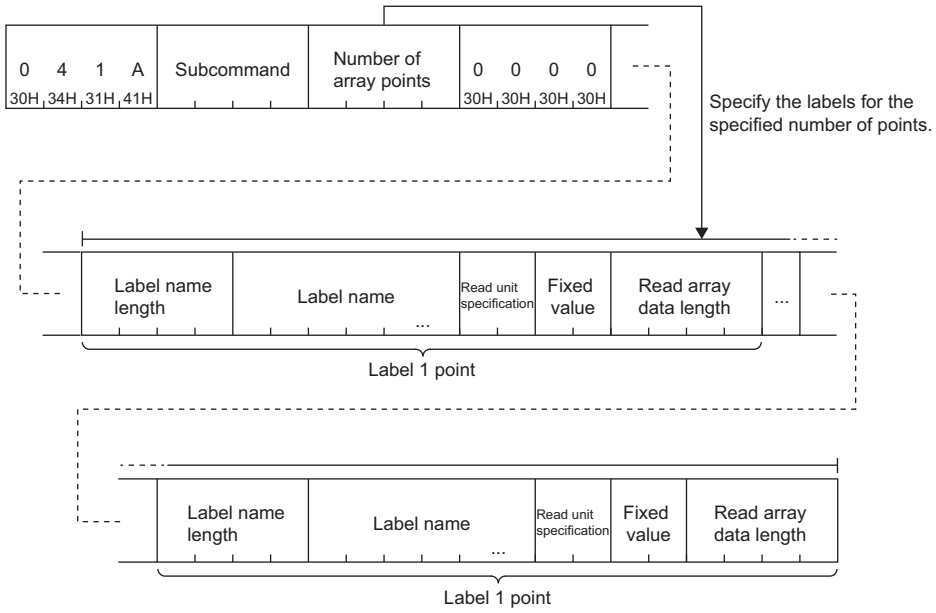
Labels of the following data types cannot be specified.

- Timer
- Counter
- Long timer
- Retentive timer
- Long retentive timer
- Long timer

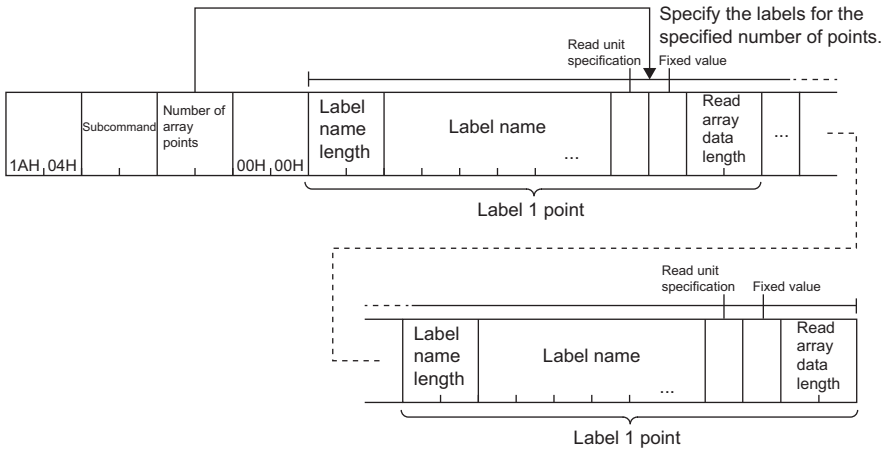
In addition, two-dimensional arrays or three-dimensional arrays whose data type is bit cannot be specified.

■Without the abbreviation definition

ASCII

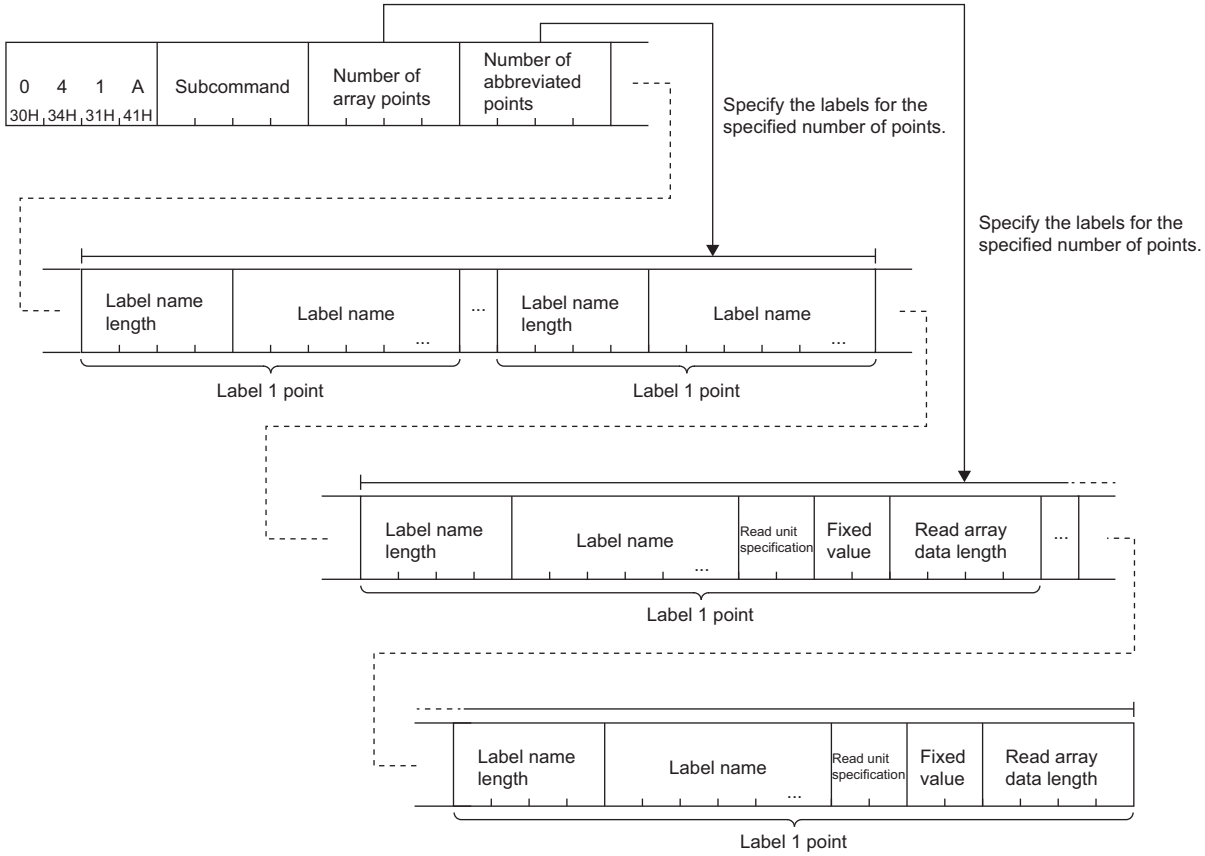


Binary

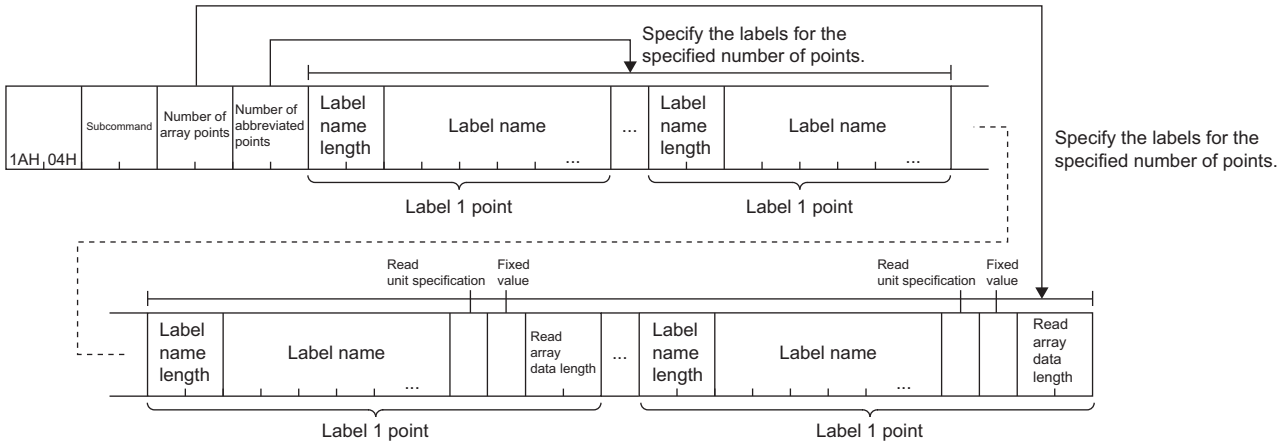


■With the abbreviation definition

ASCII



Binary



■Subcommand

Subcommand	
ASCII code	**Binary code**
0 0 0 0 30H 30H 30H 30H	00H 00H

■Number of array points

Specify the number of arrays to be read. (☞ Page 71 Number of array points)

■Number of abbreviation points

Specify the number of points of the label names to be abbreviated. (☞ Page 72 Number of abbreviation points)

■Label name length and label name equivalent to the number of abbreviation points

Specify the label name and label name length of the label to be abbreviated equivalent to the number of abbreviation points. (☞ Page 72 Number of abbreviation points)

■Label name length for number of array points, label name, read unit specification, fixed value, and read array data length

Specify the values equivalent to the number of labels specified in the number of array points.

☞ Page 73 Label name length

☞ Page 74 Label name

☞ Page 78 Read unit specification, write unit specification

☞ Page 79 Fixed value

☞ Page 79 Read array data length, write array data length

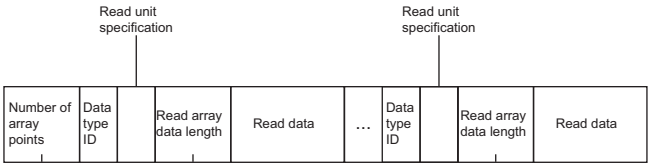
Response data

The value read from the label is stored in hexadecimal. The data order differs depending on the type of code, ASCII code or binary code.

ASCII



Binary



■Number of array points

The same data as the request data is stored.

■Data type ID, read unit specification, read array data length, and read data

The number of data points specified in the number of array points is read.

No.	Data name	Data configuration
(1)	Data type ID	
(2)	Read unit specification	
(3)	Read array data length	
(4)	Read data	

The read data differs depending on the read unit specification: bits or bytes.

When the data type is the character string or character string (Unicode), the size of the read data is the number of defined characters of the label + N. The characters to the NULL end are valid, and the later characters are undefined.

The following table lists values of N and the NULL end.

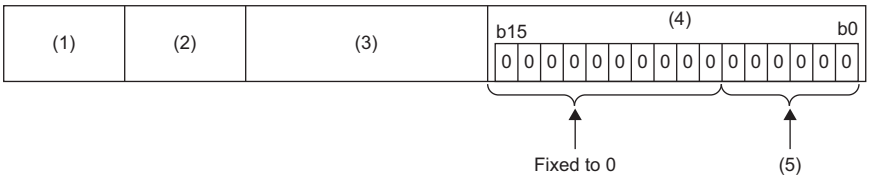
Data type	Value of N	Value of the NULL end
Character string	• The number of defined characters is odd: 1 • The number of defined characters is even: 2	00H
Character string (Unicode)	2	0000H

Point

The read data is stored in units of two bytes (words) regardless of the data type.

The following shows examples for communication data in ASCII code and binary code with the conditions below.

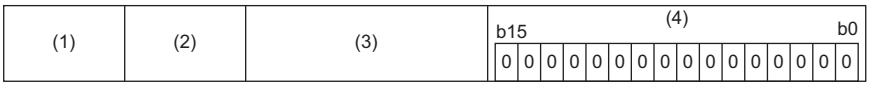
- Read unit specification: 0
- Read array data length: 6
- Read data: 0



No.	Data name	Data	
		When communicating data in ASCII code (hexadecimal)	When communicating data in binary code (hexadecimal)
(1)	Data type ID: Fixed to 1	3031	01
(2)	Read unit specification: 0	3030	00
(3)	Read array data length: 6	30303036	0600
(4)	The read data is stored in 16 bits (2 bytes).	30303030	0000
(5)	The read data of six bits is stored because the read array data length is "six".	—	

The following shows examples for communication data in ASCII code and binary code with the conditions below.

- Read unit specification: 1
- Read array data length: 2
- Read data: 0



No.	Data name	Data	
		When communicating data in ASCII code (hexadecimal)	When communicating data in binary code (hexadecimal)
(1)	Data type ID: Fixed to 2	3032	02
(2)	Read unit specification: 1	3031	01
(3)	Read array data length: 2	30303032	0200
(4)	The read data of two bytes is stored because the read array data length is "two".	30303030	0000

Communication example (label of array specified type (bit specification))

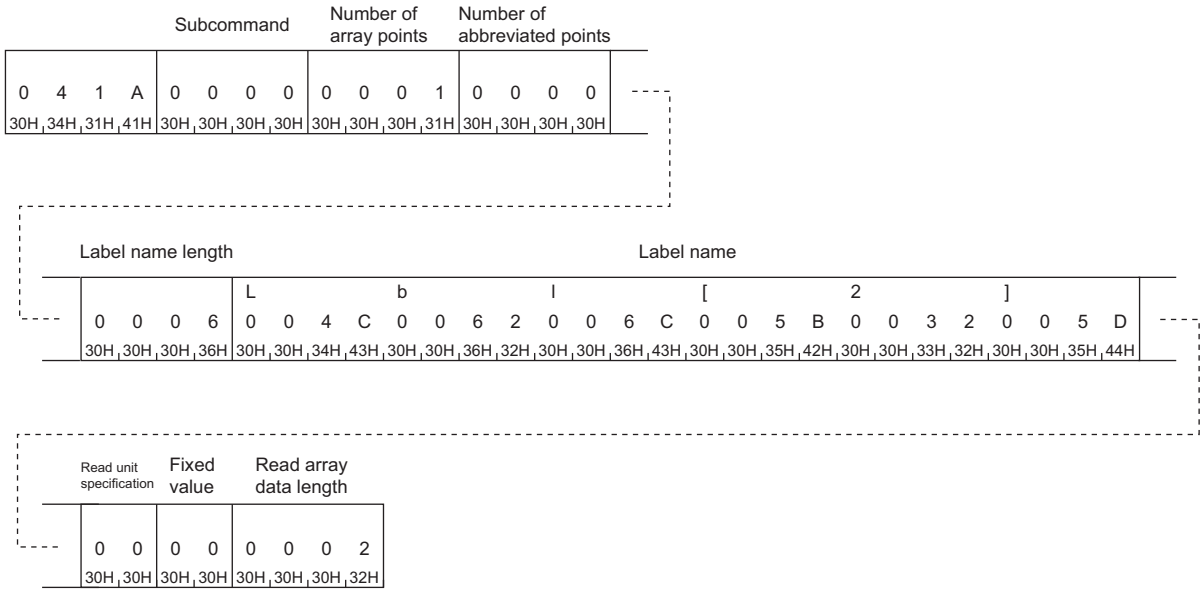
Data of two bits is read from the label of array specified type with the data type of bit, "Lbl[2]".

The following values are assumed to be stored in the label.

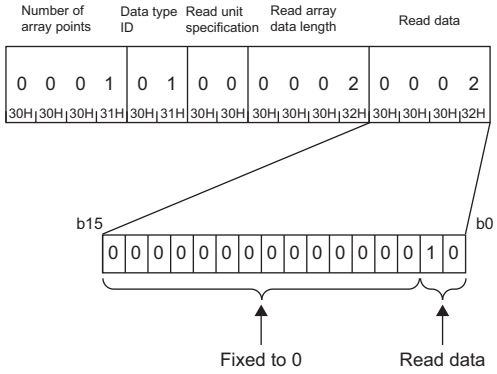
- Lbl[2]: 0(OFF)
- Lbl[3]: 1(ON)

■When communicating data in ASCII code

(Request data)

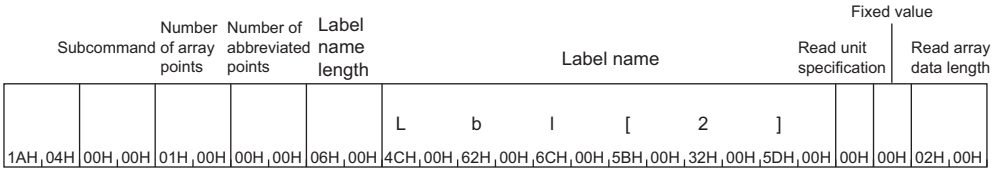


(Response data)

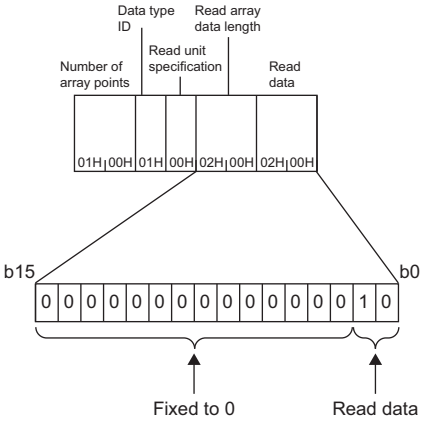


■When communicating data in binary code

(Request data)



(Response data)



Communication example (label of array specified type (byte specification))

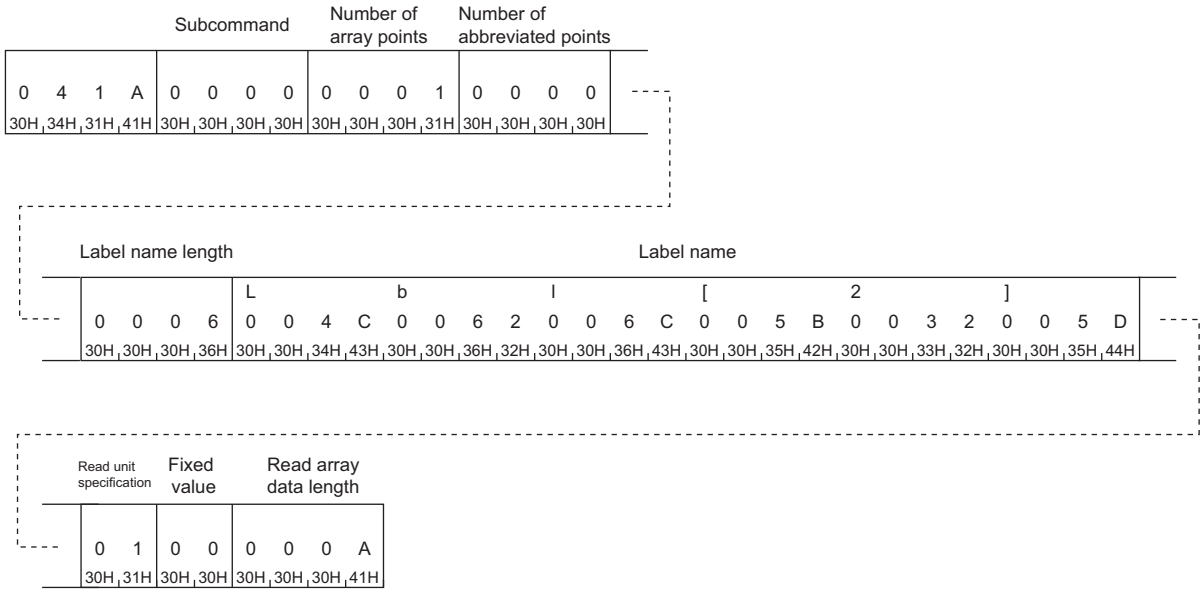
Data of five words is read from the label of array specified type with the data type of word, "Lbl[2]".

The following values are assumed to be stored in the label.

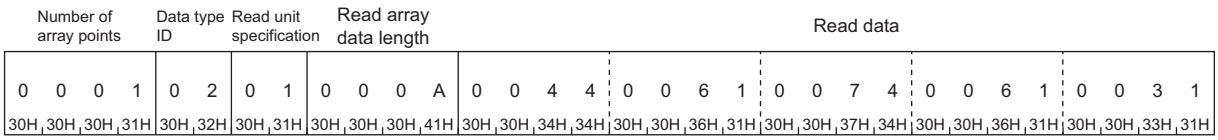
- Lbl[2]: 0044H
- Lbl[3]: 0061H
- Lbl[4]: 0074H
- Lbl[5]: 0061H
- Lbl[6]: 0031H

■When communicating data in ASCII code

(Request data)

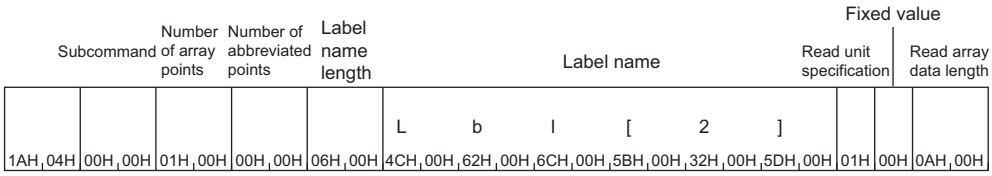


(Response data)

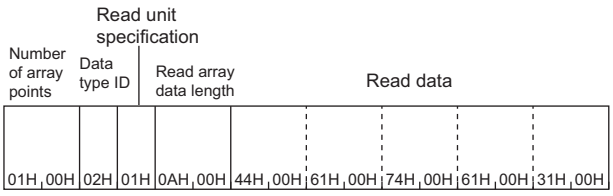


■When communicating data in binary code

(Request data)



(Response data)



Communication example (label of structured type)

Data of four words is read from the label of structured type with the data type of word, "Typ1.led[2]", and data of two words is read from the label of structured type with the data type of word, "Typ1.No[1]".

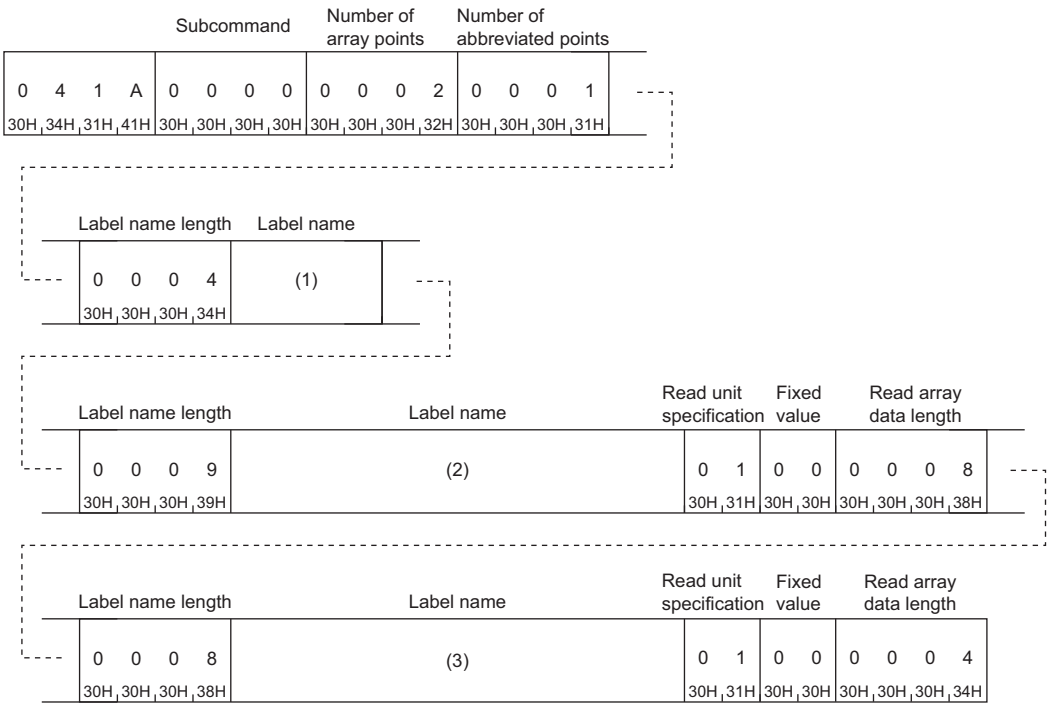
The following values are assumed to be stored in the label.

- Typ1.led[2]: 0031H
- Typ1.led[3]: 0032H
- Typ1.led[4]: 0033H
- Typ1.led[5]: 0034H
- Typ1.No[1]: 0030H
- Typ1.No[2]: 0031H

The abbreviation definition is used so that the label name "Type1" can be abbreviated as "%1".

■When communicating data in ASCII code

(Request data)

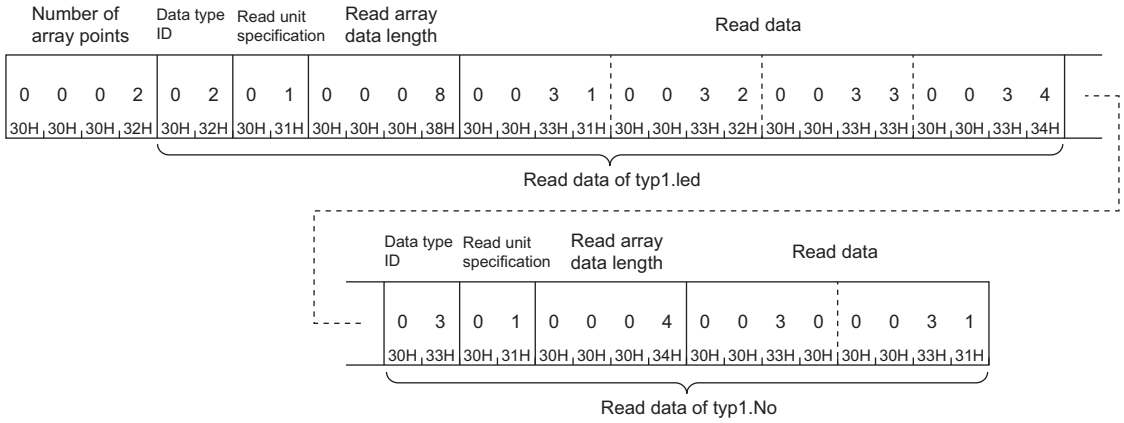


No.	Item	Value
—	Label name	Typ1
	UTF-16 (hexadecimal)	0054007900700031
(1)	ASCII code (hexadecimal)	3030353430303739303037303030333 1

No.	Item	Value
—	Label name	%1.led[2]
	UTF-16 (hexadecimal)	00250031002E006C00650064005B0032005D
(2)	ASCII code (hexadecimal)	3030323530303331303032453030364330303635303036343030354230303332303035 44

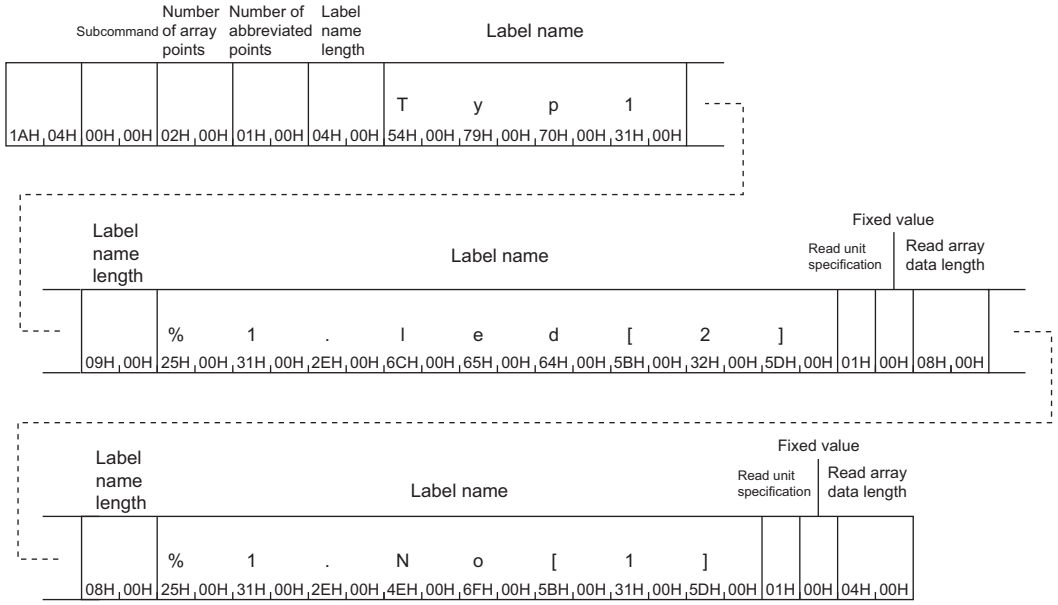
No.	Item	Value
—	Label name	%1.No[1]
	UTF-16 (hexadecimal)	00250031002E004E006F005B0031005D
(3)	ASCII code (hexadecimal)	3030323530303331303032453030344530303646303035423030333130303544

(Response data)

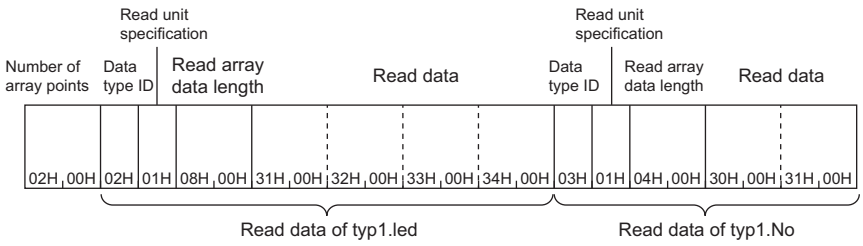


■When communicating data in binary code

(Request data)



(Response data)



Array Label Write (command: 141A)

This command writes data to a label of array specified type or label of structured type when the members of the label are an array.

This command can write data even to other than a label of array specified type assuming the label having one element of the array.

Restriction

Labels of the following data types cannot be specified.

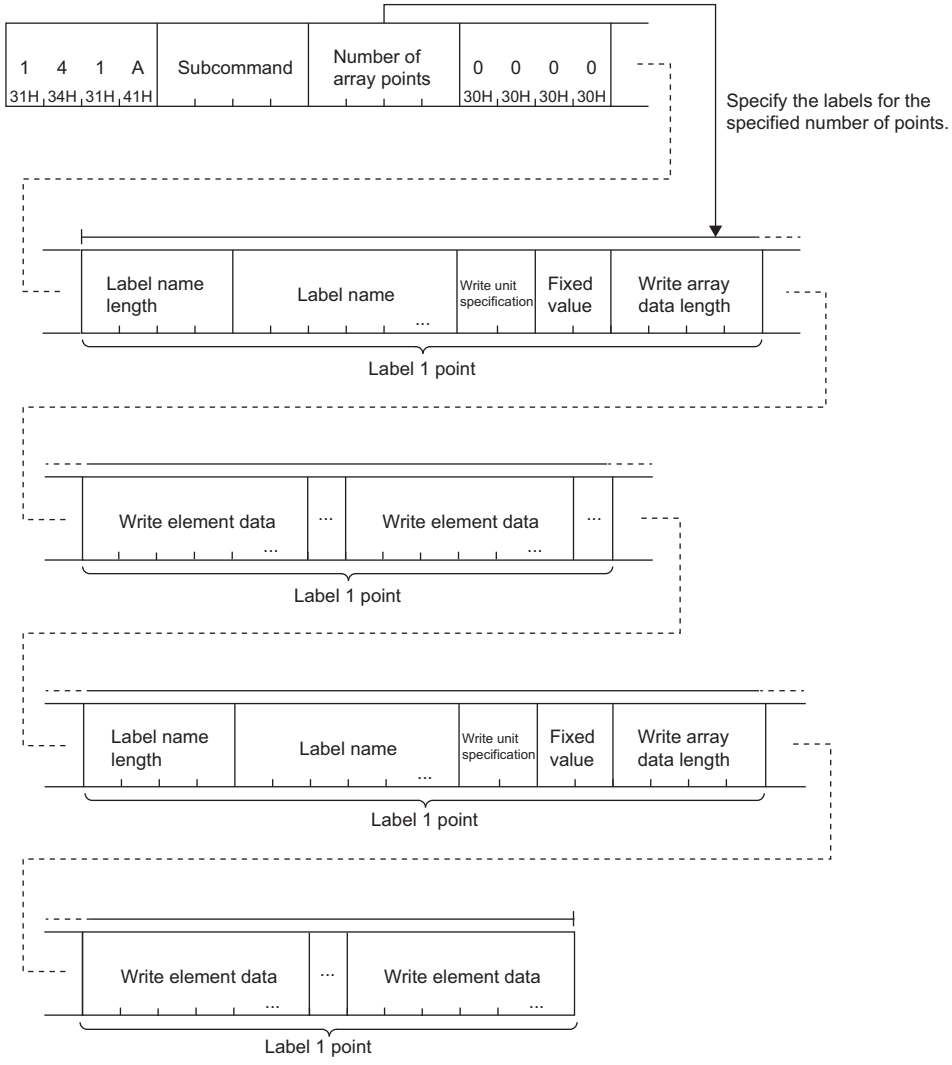
- Timer
- Counter
- Long timer
- Retentive timer
- Long retentive timer
- Long timer

In addition, two-dimensional arrays or three-dimensional arrays whose data type is bit cannot be specified.

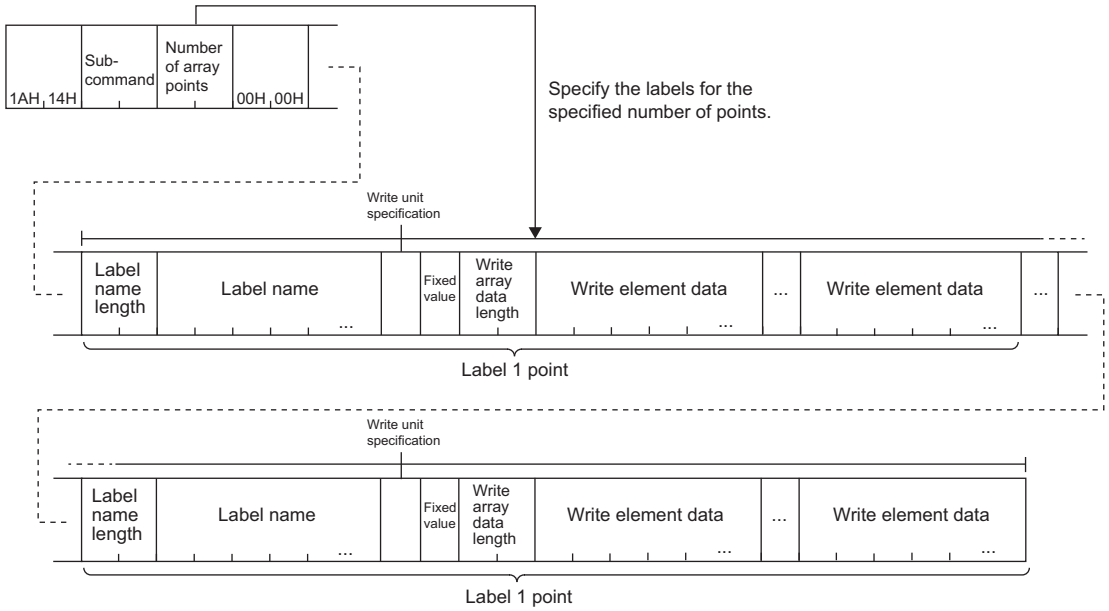
Request data

■Without the abbreviation definition

ASCII

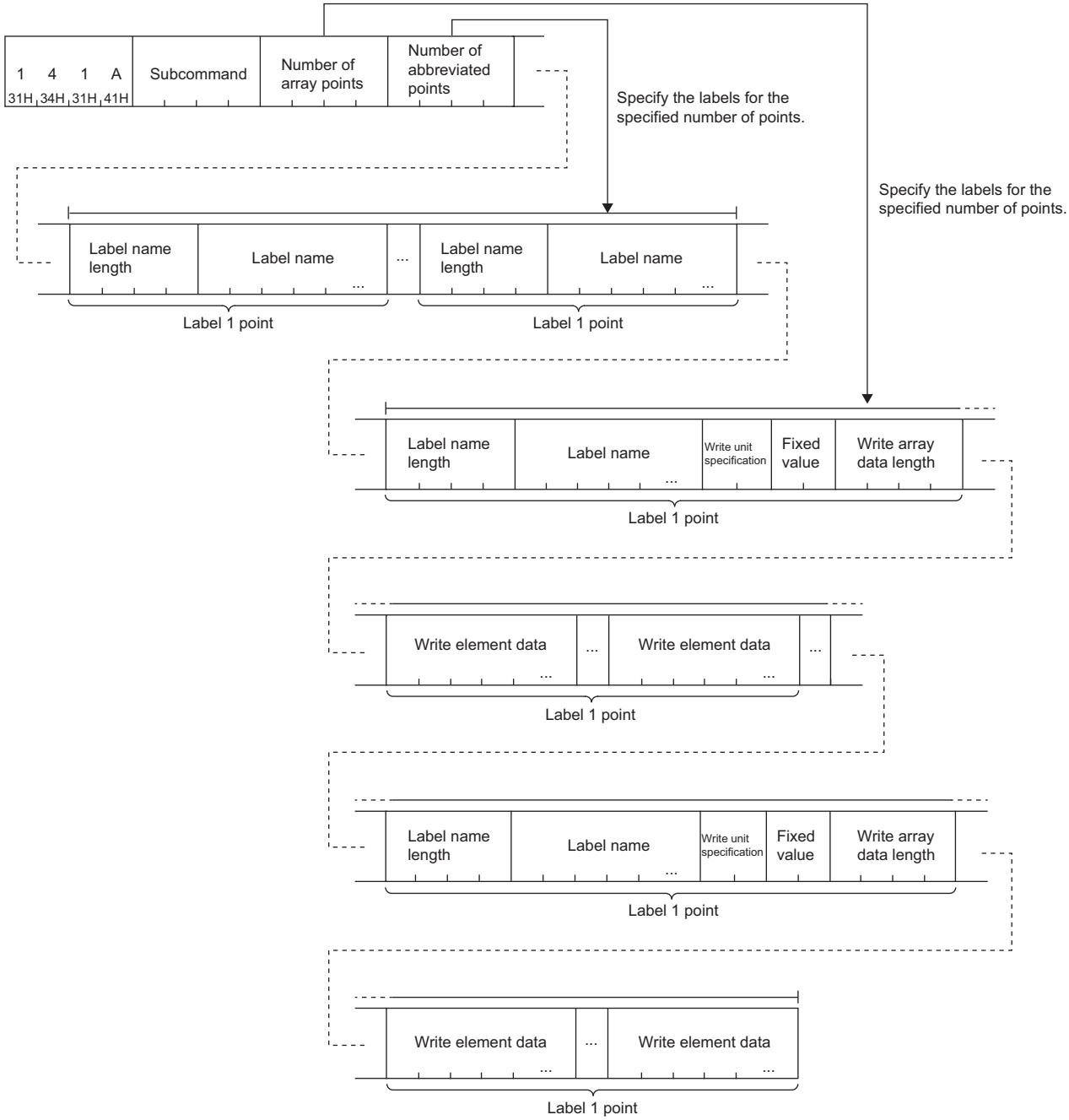


Binary

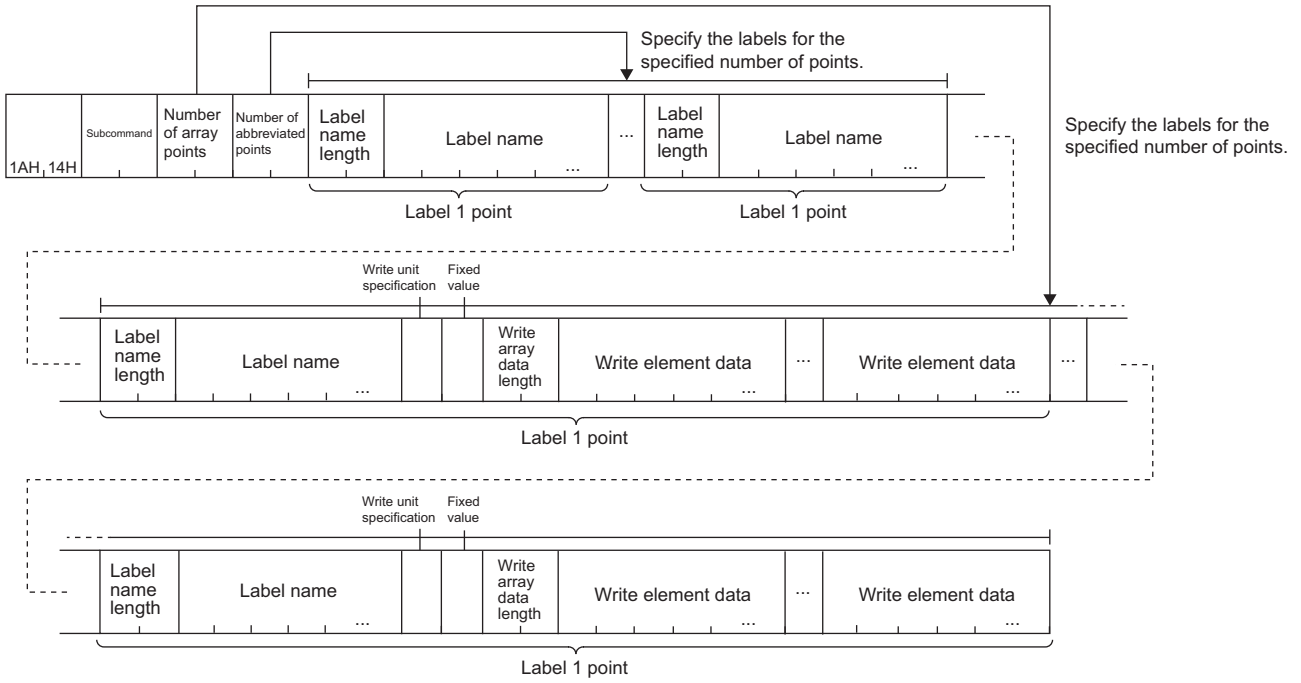

1AH
14H
Sub-command
Number of array points
00H
00H
Specify the labels for the specified number of points.
Write unit specification
Label name length
Label name
Fixed value
Write array data length
Write element data
Write element data
Label 1 point
Write unit specification
Label name length
Label name
Fixed value
Write array data length
Write element data
Write element data
Label 1 point

■With the abbreviation definition

ASCII



Binary



■Subcommand

Subcommand	
ASCII code	**Binary code**
0 0 0 0 30H 30H 30H 30H	00H 00H

■Number of array points

Specify the number of arrays to be written. (☞ Page 71 Number of array points)

■Number of abbreviation points

Specify the number of points of the label names to be abbreviated. (☞ Page 72 Number of abbreviation points)

■Label name length and label name equivalent to the number of abbreviation points

Specify the label name and label name length of the label to be abbreviated equivalent to the number of abbreviation points. (☞ Page 72 Number of abbreviation points)

■Label name length for number of array points, label name, write unit specification, fixed value, write layout data length, and write element data

Specify the values equivalent to the number of points specified in the number of array points.

☞ Page 73 Label name length
☞ Page 74 Label name
☞ Page 78 Read unit specification, write unit specification
☞ Page 79 Fixed value
☞ Page 79 Read array data length, write array data length

The following table lists the components of write data.

No.	Data name	Data configuration
(1)	Write unit specification	
(2)	Fixed value	
(3)	Write array data length	
(4)	Write element data	

The write element data differs depending on the write unit specification: bits or bytes.

For the write unit specification with bits, specify the write element data in the size rounded up in units of two bytes.

When the write unit specification does not correspond to the data type of the label, a communication error occurs and the error code is stored in the end code of the response message. For the error codes, refer to the manual for the CPU module. (MELSEC iQ-R CPU Module User's Manual (Application))

When the data type is the character string or an array of the character string (Unicode), specify the write element data for every one point of the array including the value of the NULL end, and specify all the elements in the size of the number of defined characters of the label + N.

The following table lists values of N and the NULL end.

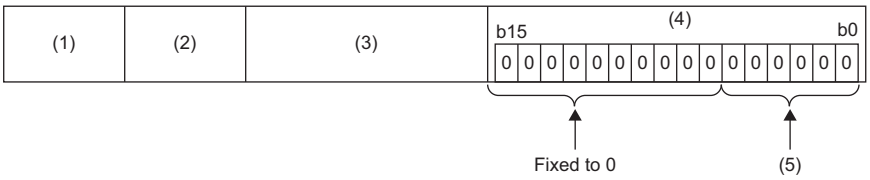
Data type	Value of N	Value of the NULL end
Character string	• The number of defined characters is odd: 1 • The number of defined characters is even: 2	00H
Character string (Unicode)	2	0000H

Point

Store the write element data in units of two bytes (words) regardless of the data type.

The following shows examples for communication data in ASCII code and binary code with the conditions below.

- Write unit specification: 0
- Write array data length: 6
- Write data: 0



No.	Data name	Data	
		When communicating data in ASCII code (hexadecimal)	When communicating data in binary code (hexadecimal)
(1)	Write unit specification: 0	3030	00
(2)	Fixed value	3030	00
(3)	Write array data length: 6	30303036	0600
(4)	The write data is stored in 16 bits (2 bytes).	30303030	0000
(5)	The write element data of six bits is stored because the write array data length is "six".	—	

The following shows examples for communication data in ASCII code and binary code with the conditions below.

- Write unit specification: 1
- Write array data length: 2
- Write data: 0



No.	Data name	Data	
		When communicating data in ASCII code (hexadecimal)	When communicating data in binary code (hexadecimal)
(1)	Write unit specification: 1	3031	01
(2)	Fixed value	3030	00
(3)	Write array data length: 2	30303032	02
(4)	The write element data of two bits is stored because the write array data length is "two".	30303030	0000

Response data

The Array Label Write command does not have response data.

Communication example (label of array specified type (bit specification))

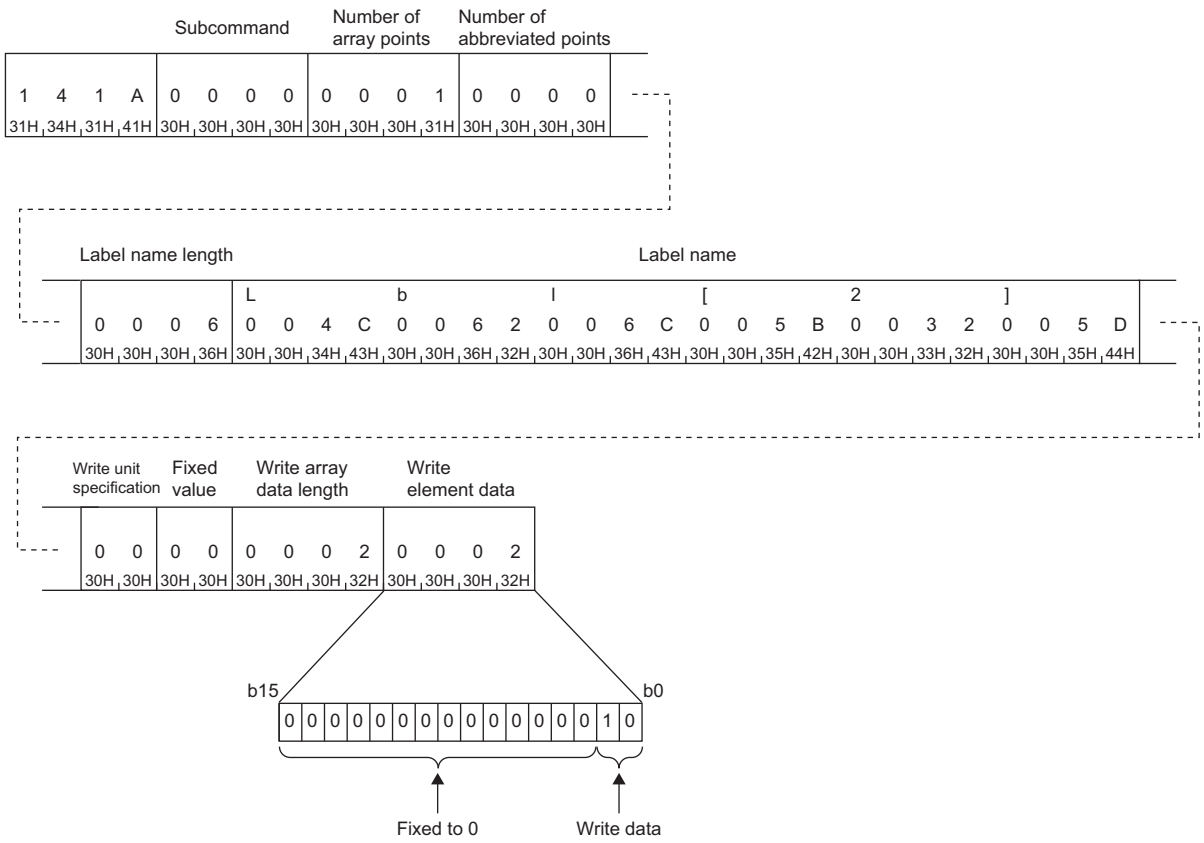
Data of two bits is written from the label of array specified type with the data type of bit, "Lbl[2]".

The following values are assumed to be written to the label.

- Lbl[2]: 0(OFF)
- Lbl[3]: 1(ON)

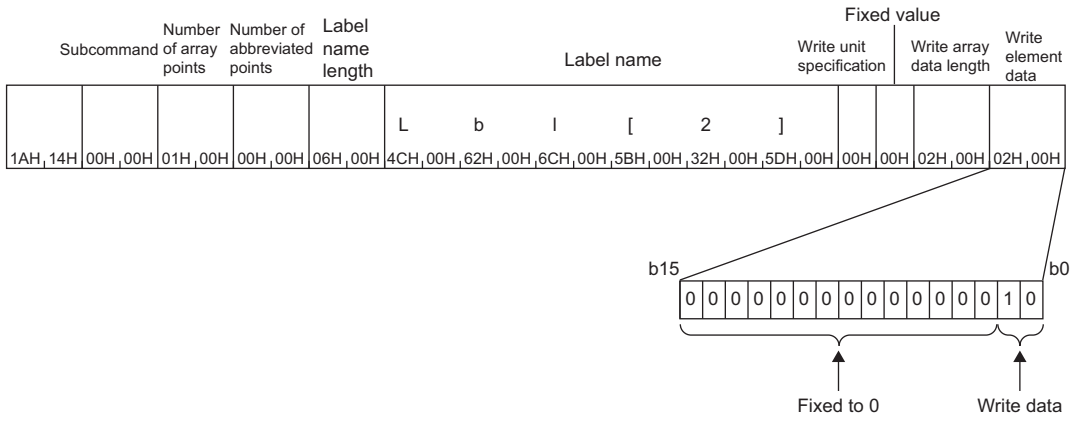
■When communicating data in ASCII code

(Request data)



■When communicating data in binary code

(Request data)



Communication example (label of array specified type (byte specification))

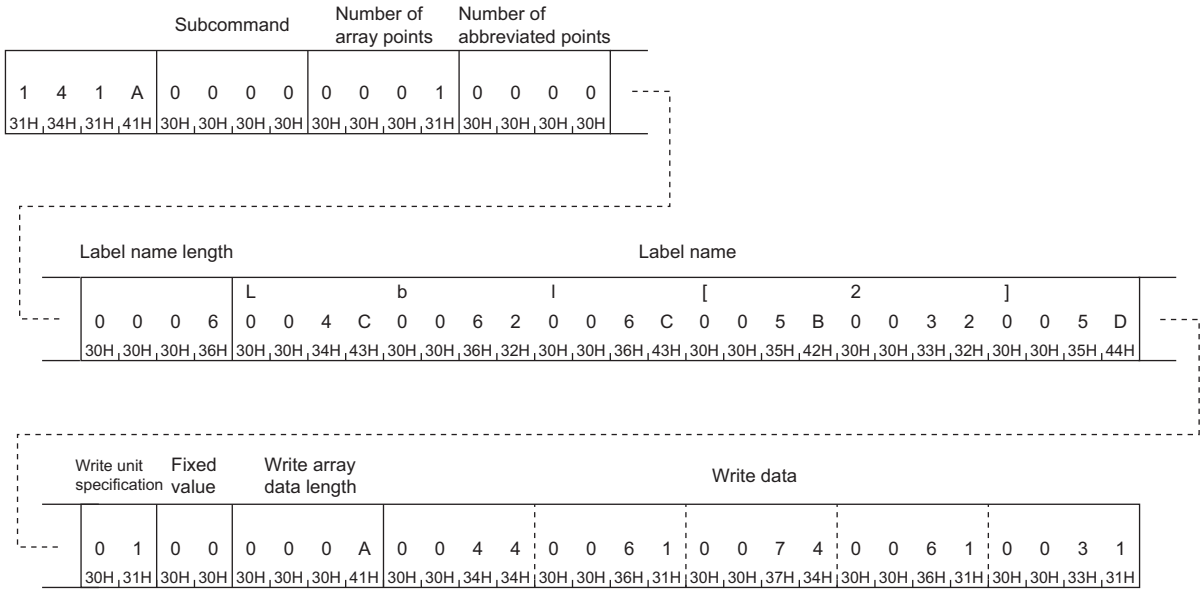
Data of five words is written from the label of array specified type with the data type of word, "Lbl[2]".

The following values are assumed to be written to the label.

- Lbl[2]: 4400H
- Lbl[3]: 6100H
- Lbl[4]: 7400H
- Lbl[5]: 6100H
- Lbl[6]: 3100H

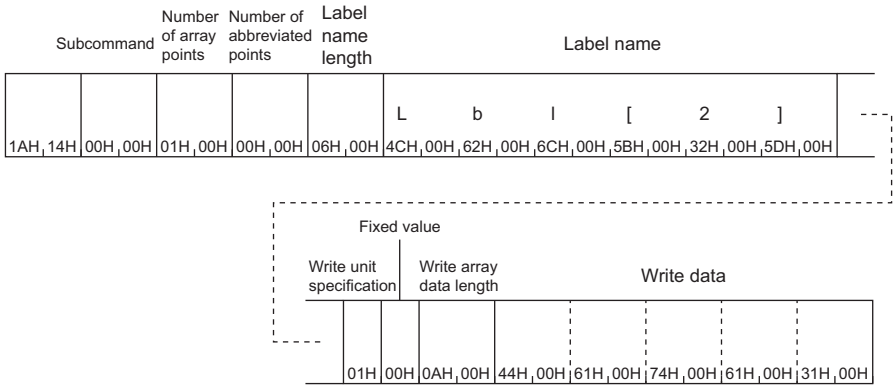
■When communicating data in ASCII code

(Request data)



■When communicating data in binary code

(Request data)



Communication example (label of structured type)

Data of four words is written from the label of structured type with the data type of word, "Typ1.led[5]", and data of two words is written from the label of structured type with the data type of word, "Typ1.No[7]".

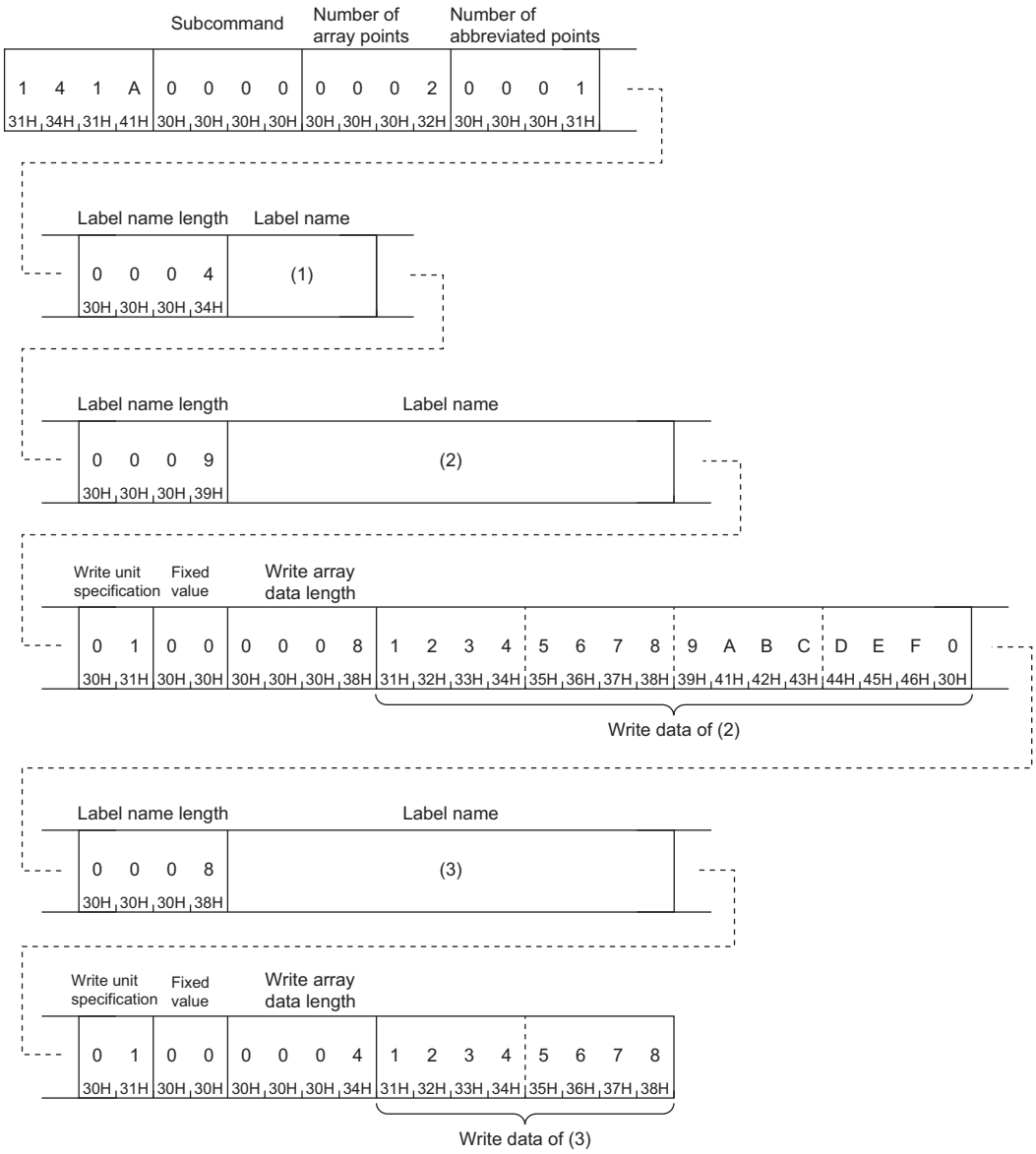
The following values are assumed to be written to the label.

- Typ1.led[5]: 1234H
- Typ1.led[6]: 5678H
- Typ1.led[7]: 9ABCH
- Typ1.led[8]: DEF0H
- Typ1.No[7]: 1234H
- Typ1.No[8]: 5678H

The abbreviation definition is used so that the label name "Type1" can be abbreviated as "%1".

■When communicating data in ASCII code

(Request data)



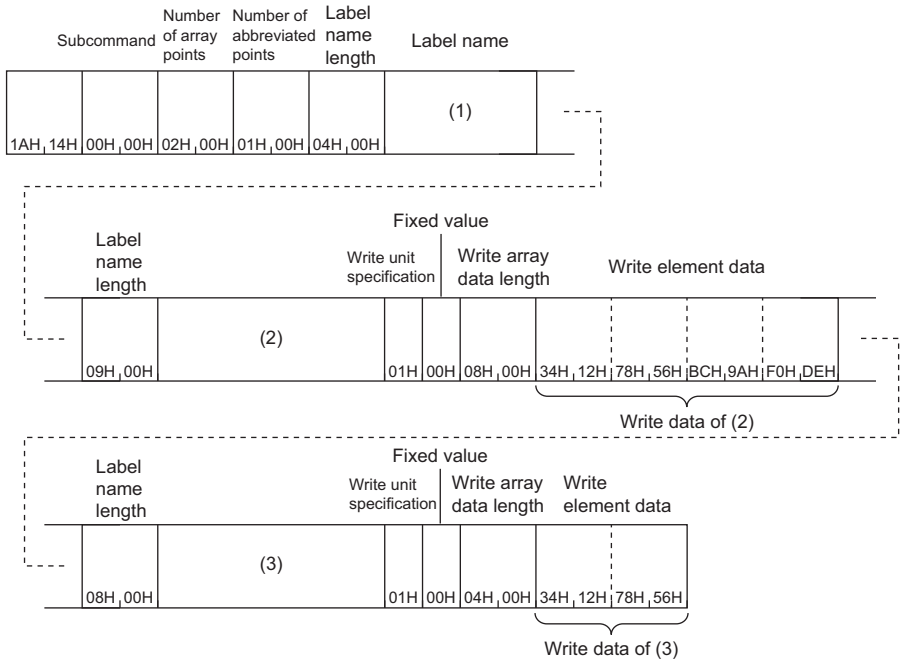
No.	Item	Value
—	Label name	Typ1
	UTF-16 (hexadecimal)	0054007900700031
(1)	ASCII code (hexadecimal)	3030353430303739303037303030333 1

No.	Item	Value
—	Label name	%1.led[5]
	UTF-16 (hexadecimal)	00250031002E006C00650064005B0035005D
(2)	ASCII code (hexadecimal)	3030323530303331303032453030364330303635303036343030354230303335303035 44

No.	Item	Value
—	Label name	%1.No[7]
	UTF-16 (hexadecimal)	00250031002E004E006F005B0037005D
(3)	ASCII code (hexadecimal)	3030323530303331303032453030344530303646303035423030333730303544

■When communicating data in binary code

(Request data)



No.	Item	Value
—	Label name	Typ1
	UTF-16 (hexadecimal)	0054007900700031
(1)	Binary code (hexadecimal)	5400790070003100

No.	Item	Value
—	Label name	%1.led[5]
	UTF-16 (hexadecimal)	00250031002E006C00650064005B0035005D
(2)	Binary code (hexadecimal)	250031002E006C00650064005B0035005D00

No.	Item	Value
—	Label name	%1.No[7]
	UTF-16 (hexadecimal)	00250031002E004E006F005B0037005D
(3)	Binary code (hexadecimal)	250031002E004E006F005B0037005D00

Label Read Random (command: 041C)

This command specifies labels and reads the data.

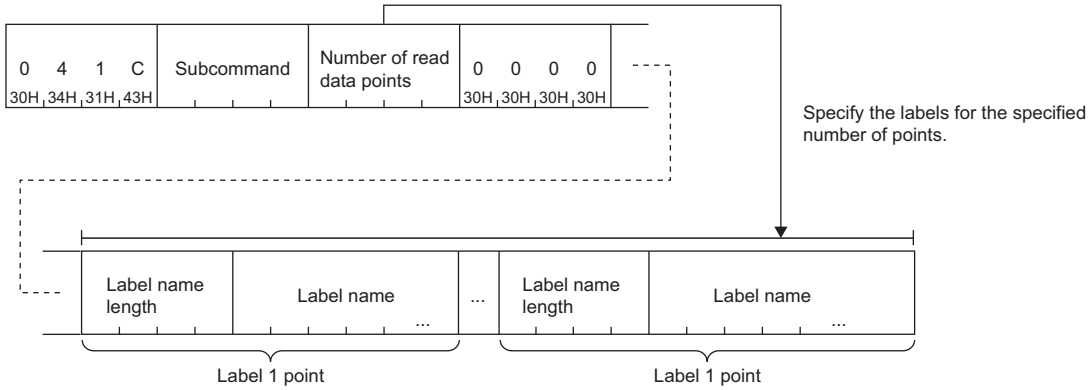
For an array, the data of each element can be specified and read.

The labels of the response data by the Label Read Random command are read in one-point units. To read array data continuously, use the Array Label Read command. (☞ Page 80 Array Label Read (command: 041A))

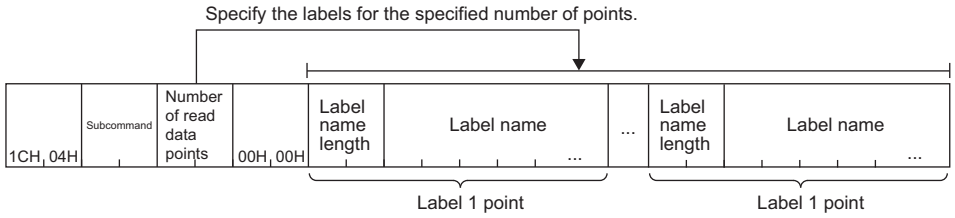
Request data

■Without the abbreviation definition

ASCII

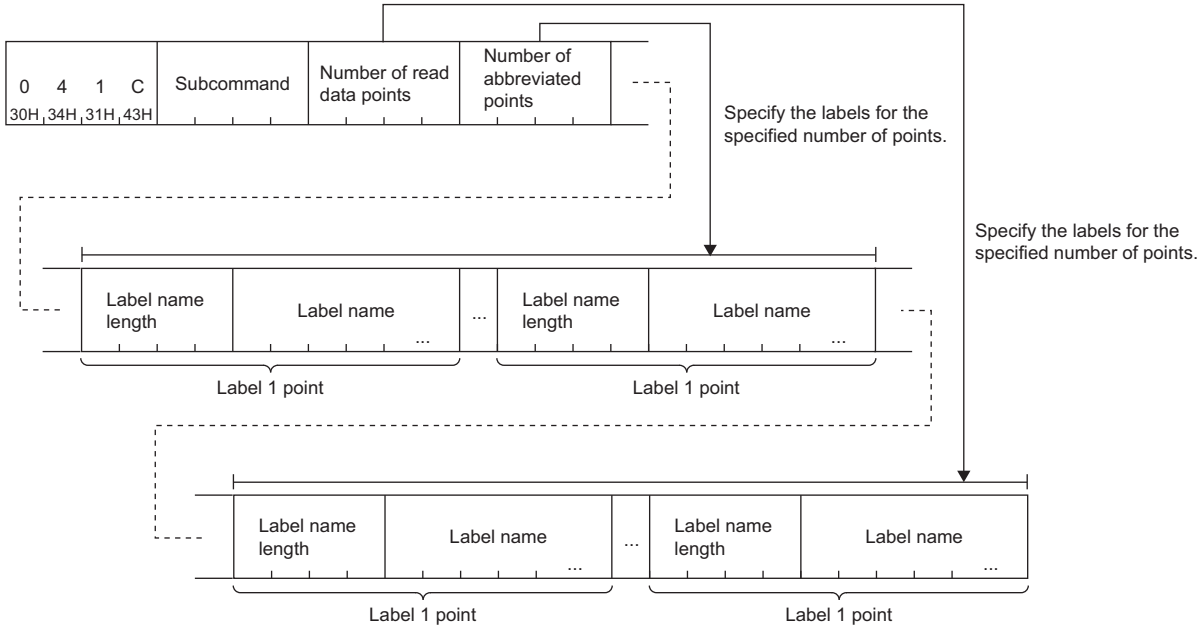


Binary

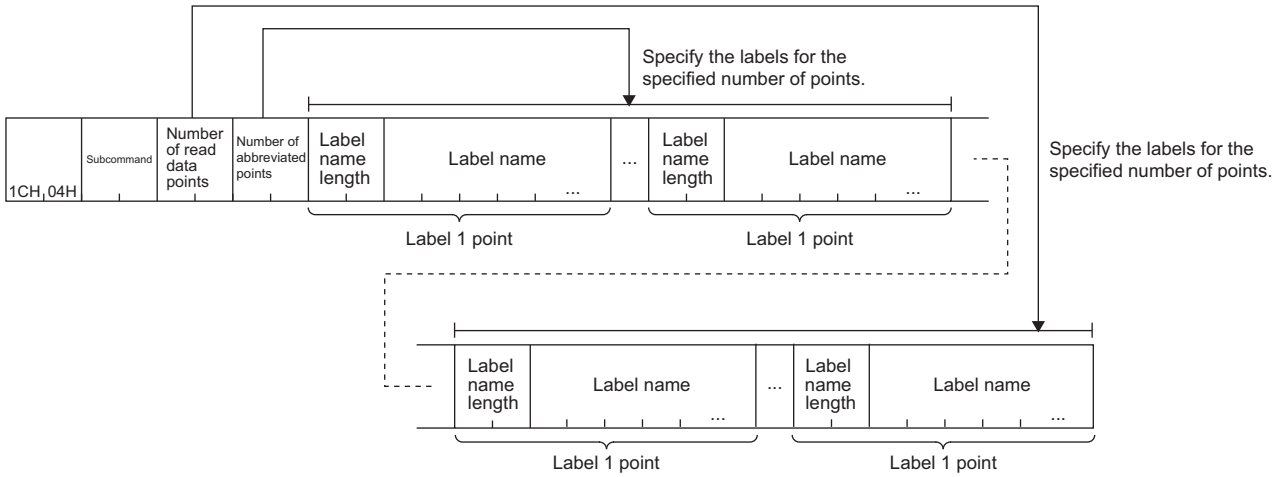


■With the abbreviation definition

ASCII



Binary



■Subcommand

Subcommand	
ASCII code	**Binary code**
0 0 0 0 30H 30H 30H 30H	00H 00H

■Number of read data points

Specify the number of labels to be read. (☞ Page 72 Number of read/write data points)

■Number of abbreviation points

Specify the number of points of the label names to be abbreviated. (☞ Page 72 Number of abbreviation points)

■Label name length and label name equivalent to the number of abbreviation points

Specify the label name and label name length of the label to be abbreviated equivalent to the number of abbreviation points. (☞ Page 72 Number of abbreviation points)

■Label name length and label name of the number of read data points

Specify the values equivalent to the number of labels specified in the number of read data points.

☞ Page 73 Label name length

☞ Page 74 Label name

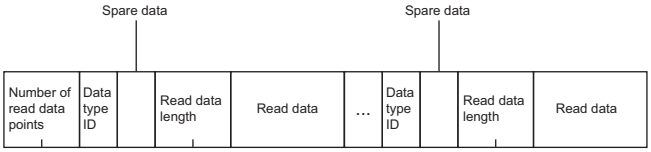
Response data

The value read from the label is stored in hexadecimal. The data order differs depending on the type of code, ASCII code or binary code.

ASCII



Binary



■Number of read data points

The same data as the request data is stored.

■Data type ID, read data length, spare data, and read data

Data equivalent to the number of data points specified in the number of read data points is read.

No.	Data name	Data configuration
(1)	Data type ID	
(2)	Spare data	
(3)	Read data length	
(4)	Read data	

The read data differs depending on the data type ID of the read label. (☞ Page 77 Data type ID)

When the data type is the character string or character string (Unicode), the size of the read data is the number of defined characters of the label + N. The characters to the NULL end are valid, and the later characters are undefined.

The following table lists values of N and the NULL end.

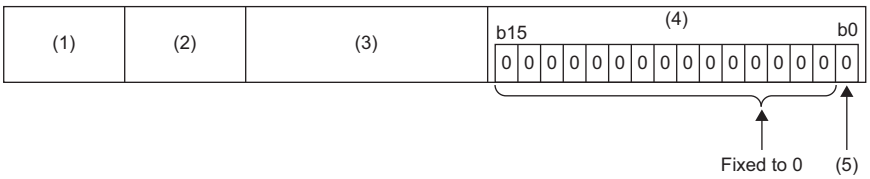
Data type	Value of N	Value of the NULL end
Character string	• The number of defined characters is odd: 1 • The number of defined characters is even: 2	00H
Character string (Unicode)	2	0000H

Point

- Specify the read data in units of two bytes (words) regardless of the data type.
- Do not use the spare data because an undefined value is stored.

The following shows examples for communication data in ASCII code and binary code with the conditions below.

- Data type ID: 1
- Read data length: 2
- Read data: 0



No.	Data name	Data	
		When communicating data in ASCII code (hexadecimal)	When communicating data in binary code (hexadecimal)
(1)	Data type ID: Fixed to 1	3031	01
(2)	Spare data	—	—
(3)	Read data length: Fixed to 2	30303032	0200
(4)	The read data is stored in 16 bits (2 bytes).	30303030	0000
(5)	The read data of one bit is stored because the data type ID is one.	—	

The following shows examples for communication data in ASCII code and binary code with the conditions below.

- Data type ID: 2
- Read data length: 2
- Read data: 2



No.	Data name	Data	
		When communicating data in ASCII code (hexadecimal)	When communicating data in binary code (hexadecimal)
(1)	Data type ID: 2	3032	02
(2)	Spare data	—	—
(3)	Read data length: 2	30303032	0200
(4)	The read data equivalent to the data size specified in the read data length is stored.	30303130	1000

The following shows examples for communication data in ASCII code and binary code with the conditions below.

- Data type ID: 10
- Read data length: 8
- Read data: AAAA



No.	Data name	Data	
		When communicating data in ASCII code (hexadecimal)	When communicating data in binary code (hexadecimal)
(1)	Data type ID: 10	3130	10
(2)	Spare data	—	—
(3)	Read data length: 8	30303038	0800
(4)	Read data: AAAA	30303431303034313030343130303431	4100410041004100

The following shows examples for communication data in ASCII code and binary code with the conditions below.

- Data type ID: 8
- Read data length: 4
- Read data: 20:31:23:647, 24th

(1)	(2)	(3)	(4)

No.	Data name	Data	
		When communicating data in ASCII code (hexadecimal)	When communicating data in binary code (hexadecimal)
(1)	Data type ID: 8	3038	08
(2)	Spare data	—	—
(3)	Read data length: 4	30303034	0400
(4)	Read data*1: 20:31:23:647, 24th	3746464646464646	7FFFFFFF

*1 Stored in increments of one millisecond in hexadecimal in the range of 80000000H (20:31:23:648, -24th) to 7FFFFFFFH (20:31:23:647, 24th).

Communication example

The data is read from the following three labels.

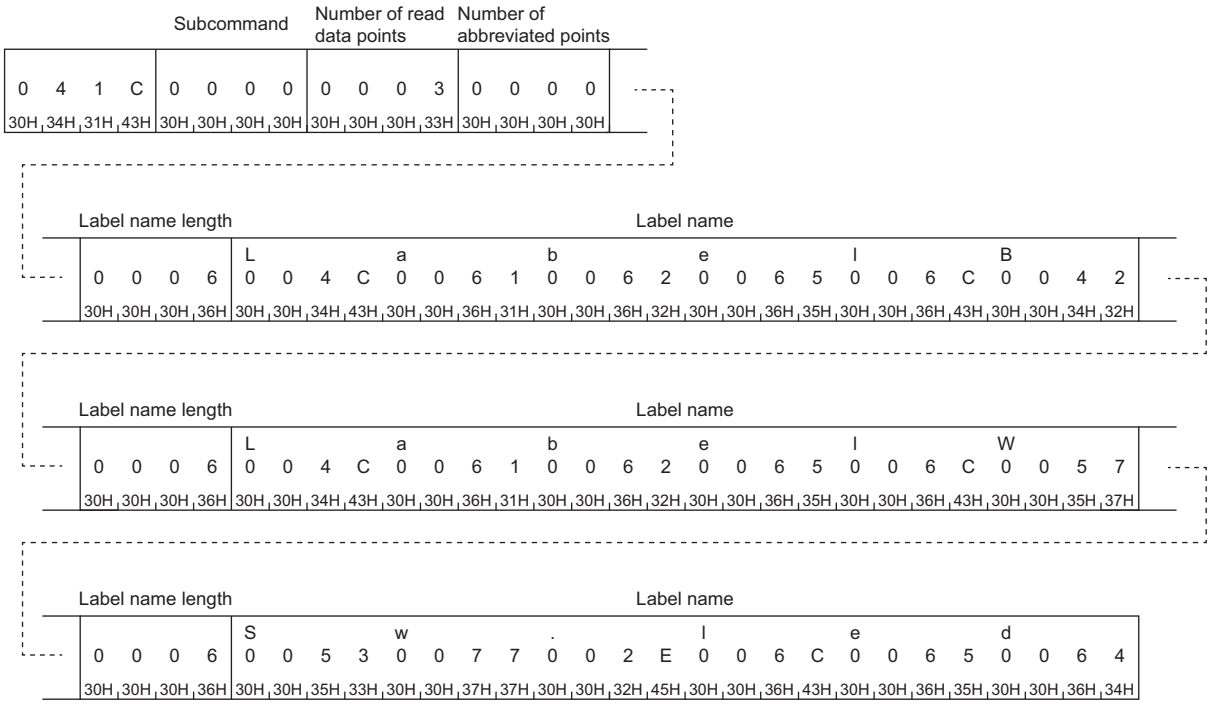
- Primitive data type label "LabelB" with the data type of bit
- Primitive data type label "LabelW" with the data type of word
- Structured type label "Sw.led" with the data type of word

The following values are assumed to be stored in the label.

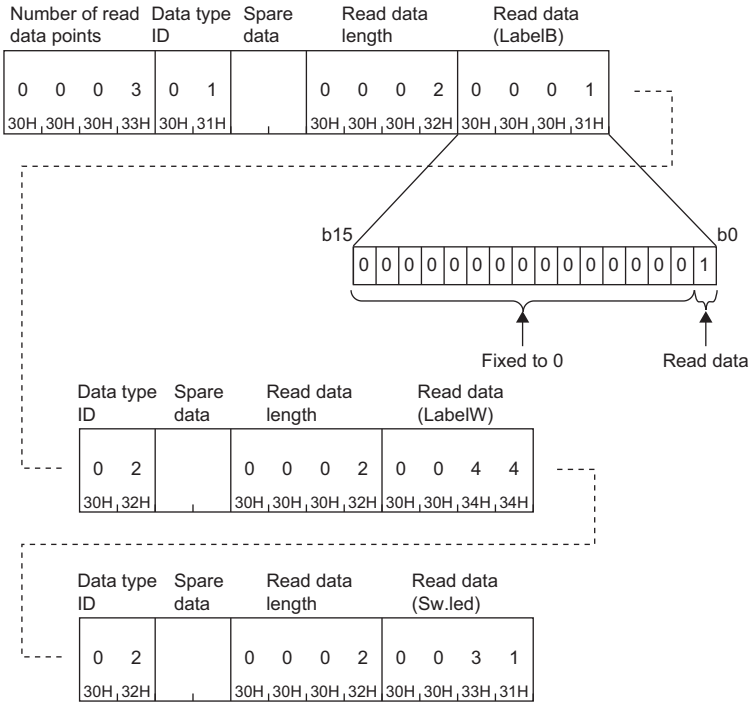
- LabelB: 1(ON)
- LabelW: 0044H
- Sw.led: 0031H

■When communicating data in ASCII code

(Request data)

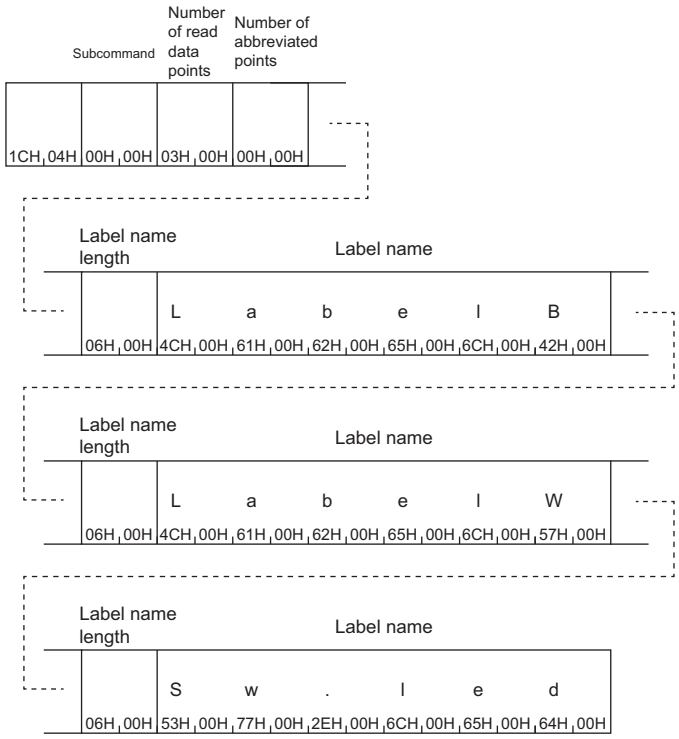


(Response data)

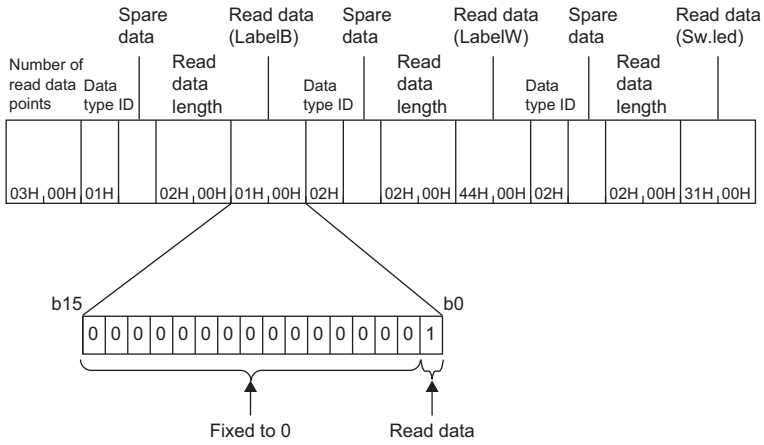


■When communicating data in binary code

(Request data)



(Response data)



Label Write Random (command: 141B)

The labels are written by the Label Write Random command in one-point units.

Specifies labels and writes data.

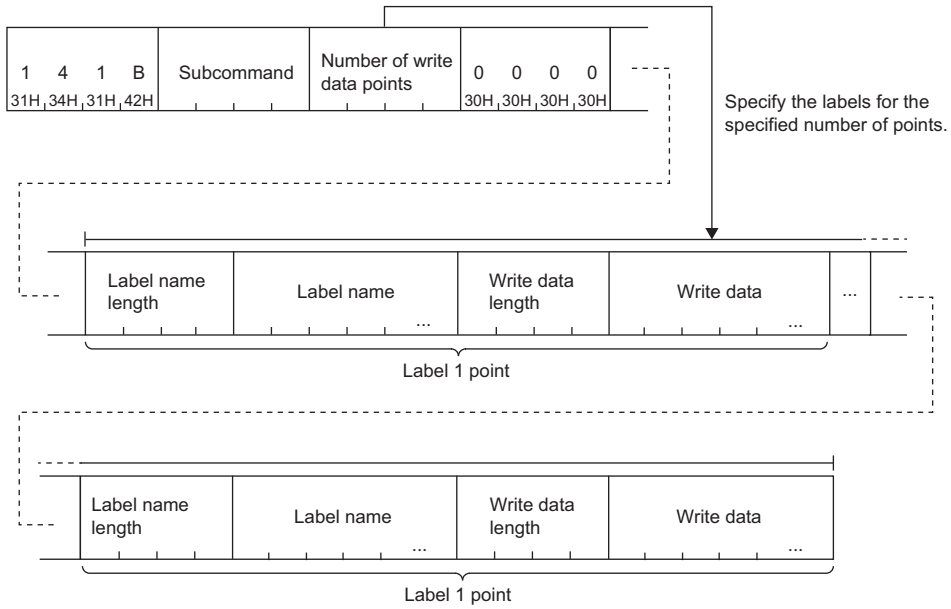
For an array, the data of each element can be specified and written.

The labels are written by the Label Write Random command in one-point units. To write array data continuously, use the ArrayLabel Write command. (☞ Page 89 Array Label Write (command: 141A))

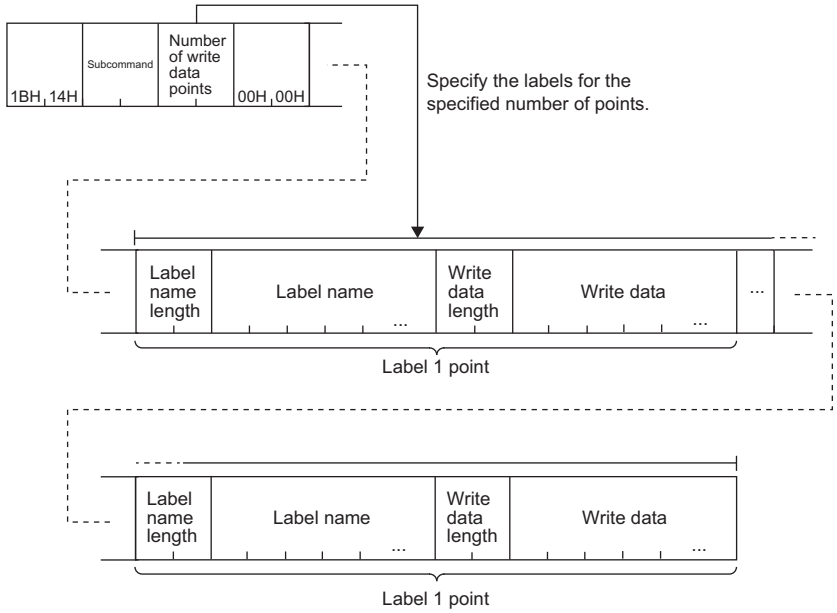
Request data

■Without the abbreviation definition

ASCII

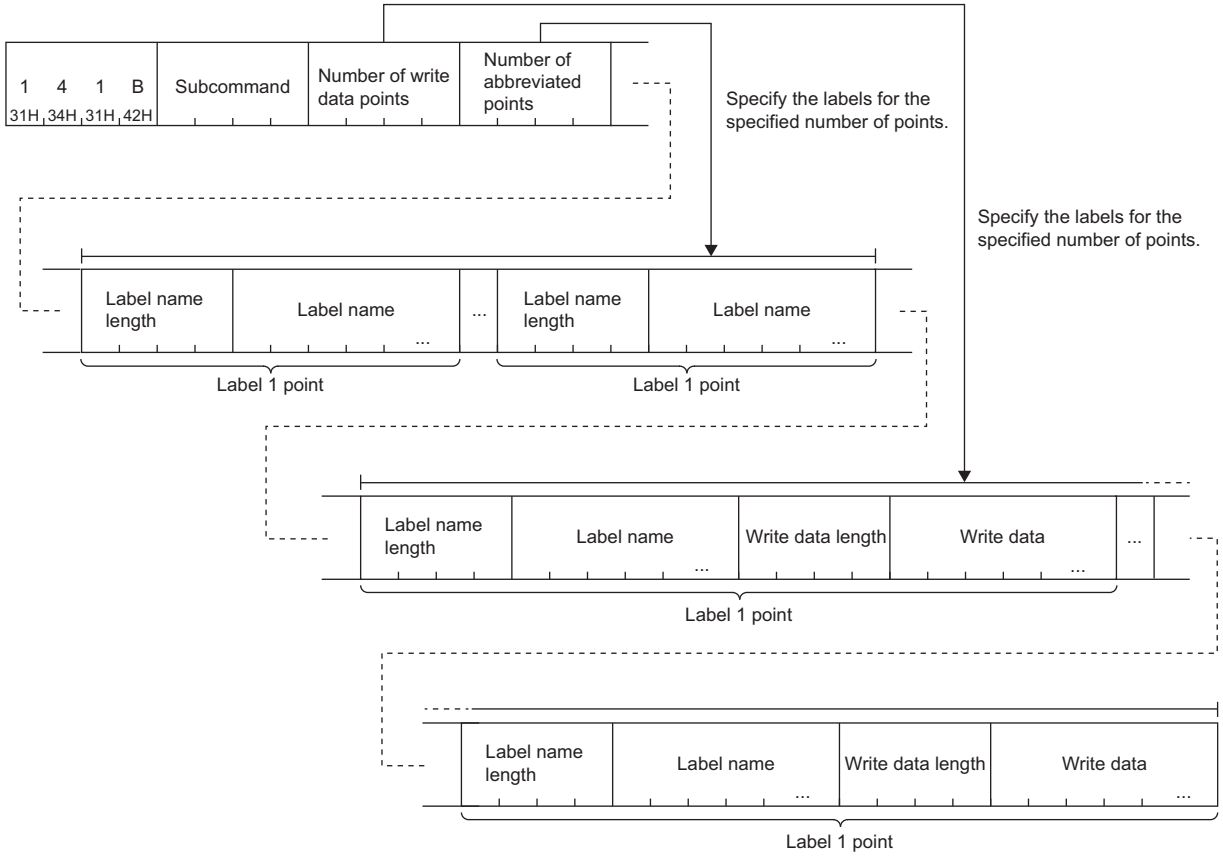


Binary

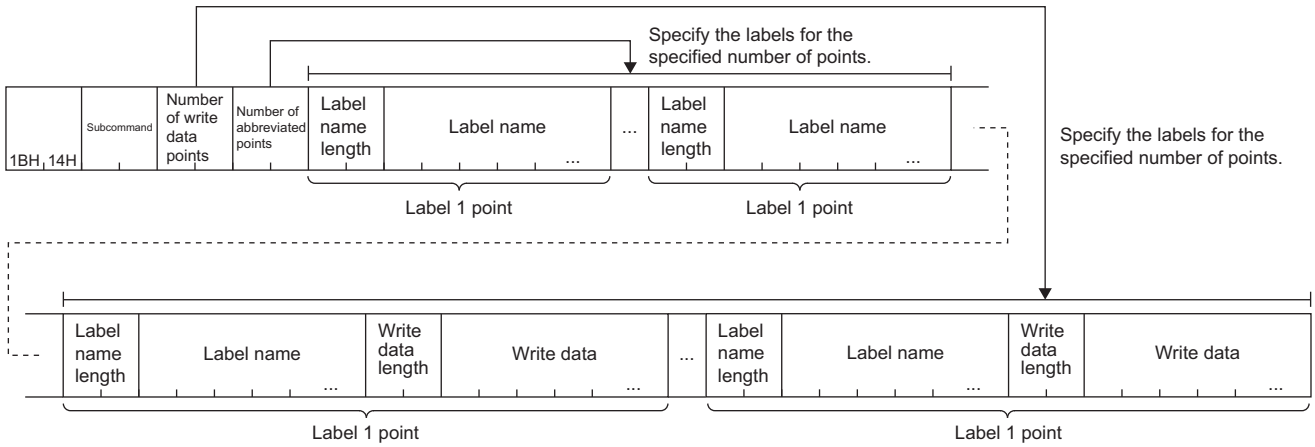


■With the abbreviation definition

ASCII



Binary



■Subcommand

Subcommand	
ASCII code	**Binary code**
0 0 0 0 30H 30H 30H 30H	00H 00H

■Number of write data points

Specify the number of labels to be written. (☞ Page 72 Number of read/write data points)

■Number of abbreviation points

Specify the number of points of the label names to be abbreviated. (☞ Page 72 Number of abbreviation points)

■Label name length and label name equivalent to the number of abbreviation points

Specify the label name and label name length of the label to be abbreviated equivalent to the number of abbreviation points. (☞ Page 72 Number of abbreviation points)

■Label name length, label name, write data length, and write data equivalent to the number of write data points

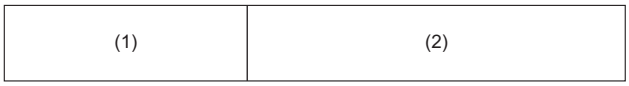
Specify the values equivalent to the number of points specified in the number of write data points.

☞ Page 73 Label name length

☞ Page 74 Label name

☞ Page 79 Read data length, write data length

The following table lists the components of write data.

No.	Data name	Data configuration
(1)	Write data length	
(2)	Write data	

The write data length of the Label Write Random command must correspond to the data type of the label.

The following table lists the write data lengths specified in each data type.

Classification	Data type name	Write data length
Label of primitive data type	Bit	2
	Word [unsigned]/bit string [16 bits]	2
	Double word [unsigned]/bit string [32 bits]	4
	Word [signed]	2
	Double word [signed]	4
	Single-precision real number	4
	Double-precision real number	8
	Hour	4
	Character string	Number of characters defined + N*1
	Character string [Unicode]	Doubled number of characters defined + N
	Contact/coil of the following data types • Timer • Counter • Long timer • Retentive timer • Long retentive timer • Long timer	2
	Current value of the following data types • Timer • Counter • Retentive timer	2
	Current value of the following data types • Long timer • Long retentive timer • Long timer	4
Label of array specified type	Data type of array element (primitive data type)	
Label of structured type	Data type of end element (primitive data type)	

*1 The value of N is 1 when the number of defined characters of the label is odd and 2 when the number of defined characters of the label is even.

When the write data length does not correspond to the data type of the label, a communication error occurs and the error code is stored in the end code of the response message. For the error codes, refer to the manual for the CPU module. (MELSEC iQ-R CPU Module User's Manual (Application))

When the data type is the character string or character string (Unicode), specify the write data length in the number of defined characters of the label + N. In addition, specify the write data including the value of the NULL end.

The following table lists values of N and the NULL end.

Data type	Value of N	Value of the NULL end
Character string	• The number of defined characters is odd: 1 • The number of defined characters is even: 2	00H
Character string (Unicode)	2	0000H

Point

• Specify the write data in units of two bytes (words) regardless of the data type.

The following shows examples for communication data in ASCII code and binary code with the conditions below.

• Write data length: 2

• Write data: 1

(1)	(2)

No.	Data name	Data	
		When communicating data in ASCII code (hexadecimal)	When communicating data in binary code (hexadecimal)
(1)	Write data length: 2	30303032	0200
(2)	Write data: 1	30303031	0100

Response data

There is no response data for Label Write Random command.

Communication example

The data is written to the following three labels.

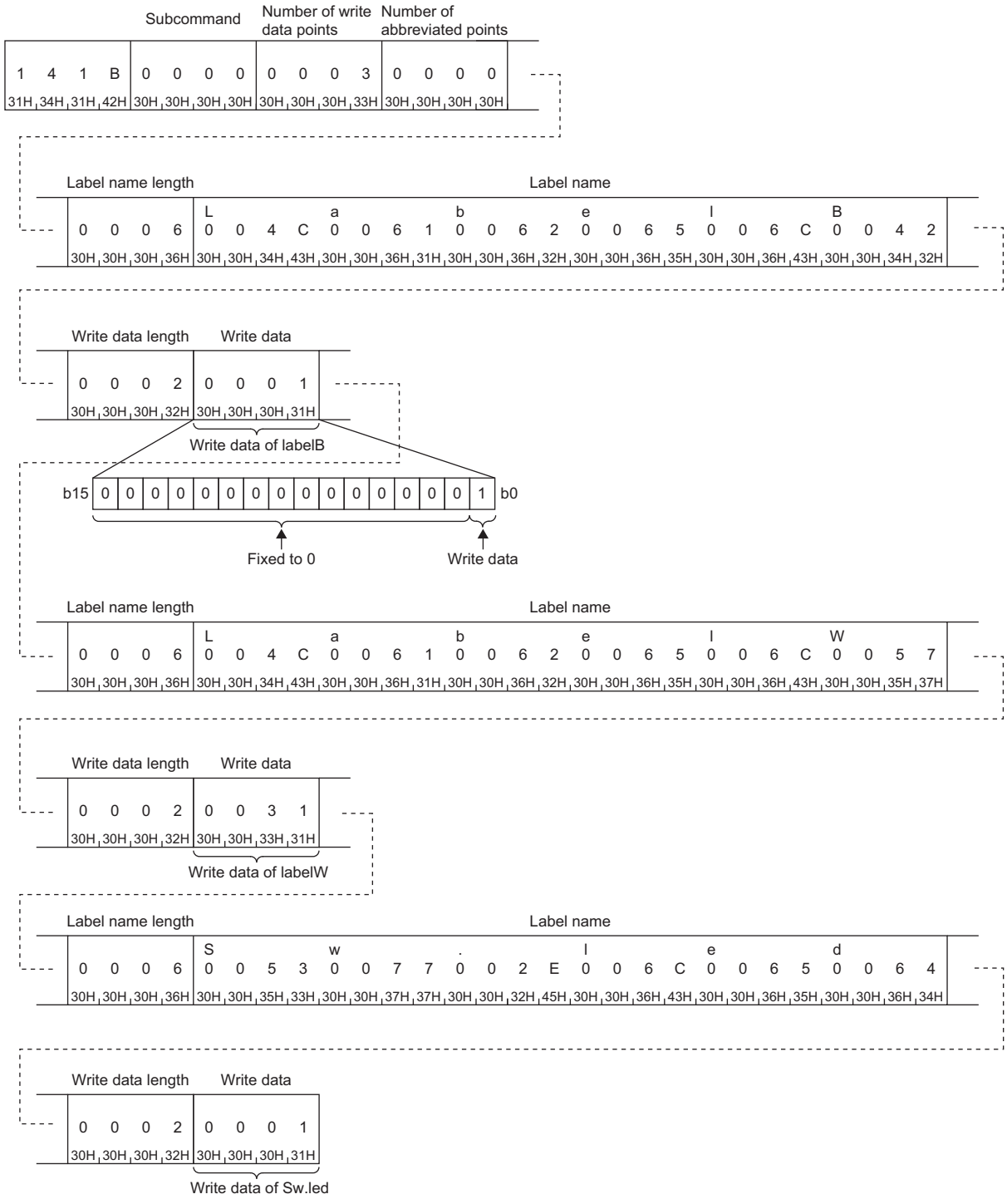
- Primitive data type label "LabelB" with the data type of bit
- Primitive data type label "LabelW" with the data type of word
- Structured type label "Sw.led" with the data type of word

The following values are assumed to be written to the label.

- LabelB: 1(ON)
- LabelW: 0031H
- Sw.led: 0001H

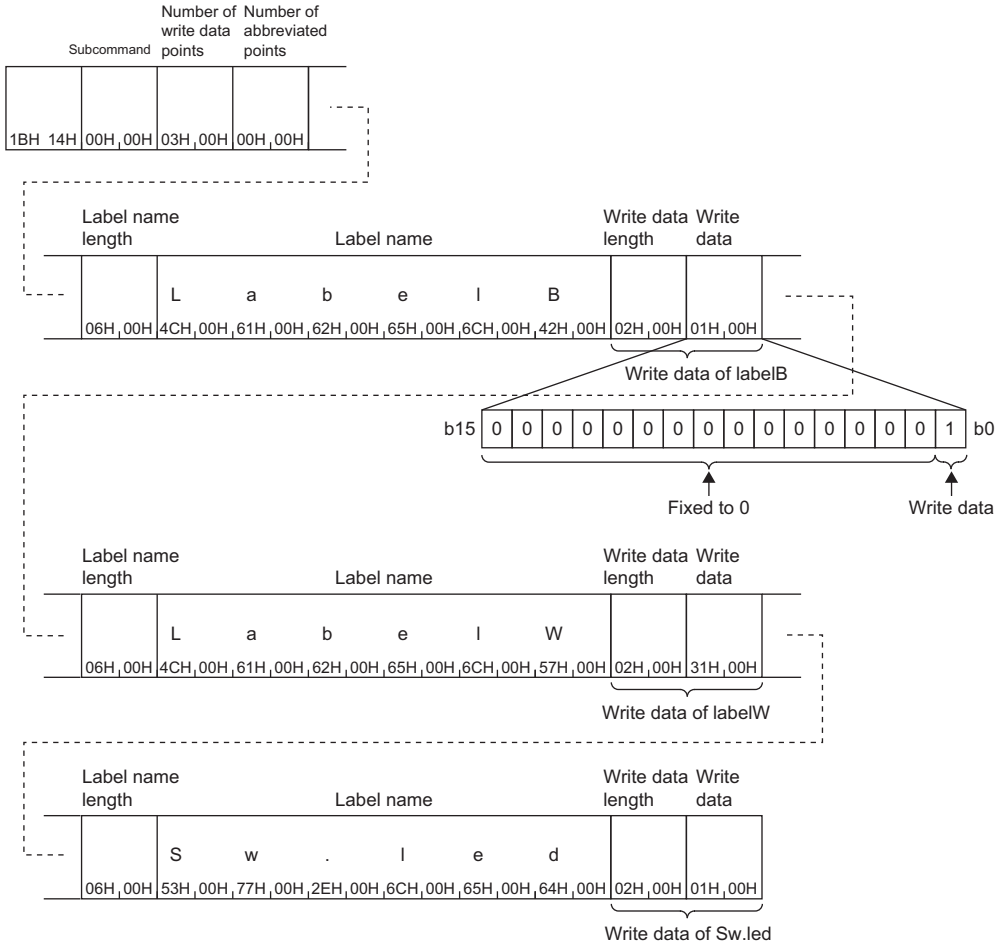
■When communicating data in ASCII code

(Request data)



■When communicating data in binary code

(Request data)



5.4 Memory (Own Station Buffer Memory Access)

This section describes the command which reads and writes the buffer memory of SLMP compatible device of own station.

Data to be specified in command

Request destination network No. and request destination station No.

Specify the station No. of the access destination. (The other stations cannot be specified.)

- Request destination network No.: 00H
- Request destination station No.: FFH

Head address

Specify the head address of buffer memory which is to be read from or to be written in.

■When communicating data in ASCII code

Convert the address to 8-digit ASCII code (hexadecimal), and send it in order the upper byte to the lower byte. Use capitalized code for alphabetical character.

Ex.

When the address is 1E1H

0	0	0	0	0	1	E	1
30H	30H	30H	30H	30H	31H	45H	31H

■When communicating data in binary code

Send the data in order from the lower byte to the upper byte using 4-byte numeral values.

Ex.

When the address is 1E1H

E1H	01H	00H	00H

Word length

Specify the word length of the buffer memory which is to be read from or to be written in.

■When communicating data in ASCII code

Convert the word length to 4-digit ASCII code (hexadecimal), and send it from the upper byte to the lower byte. Use capitalized code for alphabetical letter.

Ex.

For 5 words and 20 words

5 words

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

20 words

0	0	1	4
30H	30H	31H	34H

■When communicating data in binary code

Send the data in order from the lower byte to the upper byte using 2-byte numeral values.

Ex.

For 5 words and 20 words

5 words

05H	00H

20 words

14H	00H

Read data, write data

In case of reading, the read values of buffer memory are stored. In case of writing, the writing data is stored.

■When communicating data in ASCII code

The data is stored in 4-digit ASCII code (hexadecimal).

Ex.

For 09C1H

0	9	C	1
30H	39H	43H	31H

■When communicating data in binary code

Send the data in order from the data in order from the lower byte to the upper byte using 2-byte numeral values.

Ex.

For 09C1H

C1H	09H

Read (command: 0613)

This command reads the buffer memory data of own station (SLMP compatible device).

Point

This command cannot access to the following buffer memory areas.

- The intelligent function module which is mounted on own station (SLMP compatible device).
- The buffer memory on other station

When accessing the buffer memory areas described above, use the command of Device (device access) to access the buffer memory areas. (☞ Page 195 Read or Write by Device Extension Specification)

Request data

ASCII

0 6 1 3	0 0 0 0	Head address	Word length
30H 36H 31H 33H	30H 30H 30H 30H		

Binary

		Head address	Word length
13H 06H	00H 00H		

■Head address

Specify the head address of the buffer memory which is to be read from. (☞ Page 112 Head address)

■Word length

Specify the word length of the buffer memory which is to be read from. (☞ Page 113 Word length)

- Specification range: 1H to 1E0H (480)

Response data

The values read from the buffer memory areas are stored in order from the upper byte to the lower byte in hexadecimal. (☞ Page 113 Read data, write data)

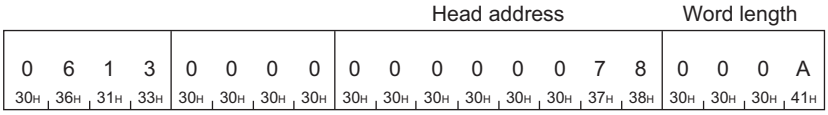
Read data 1	· · ·	Read data n

Communication example

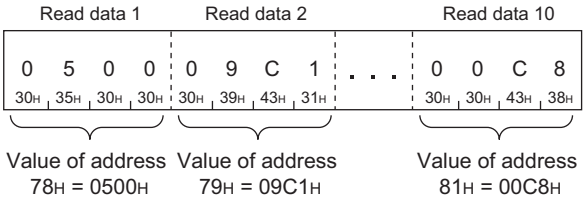
Read the data of buffer memory address 78H to 81H (120 to 129).

■When communicating data in ASCII code

(Request data)

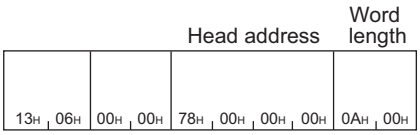


(Response data)

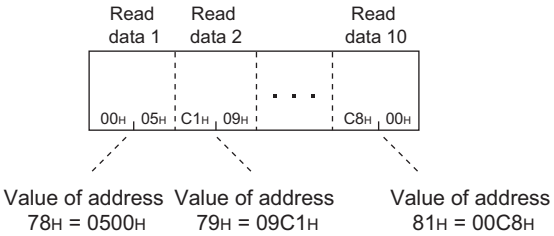


■When communicating data in binary code

(Request data)



(Response data)



5

Write (command: 1613)

This command writes the data in the buffer memory of own station (SLMP compatible device).

Point

This command cannot access to the following buffer memory areas.

- The intelligent function module which is mounted on own station (SLMP compatible device).
- The buffer memory on other station

When accessing the buffer memory areas described above, use the command of Device (device access) to access the buffer memory areas. (☞ Page 195 Read or Write by Device Extension Specification)

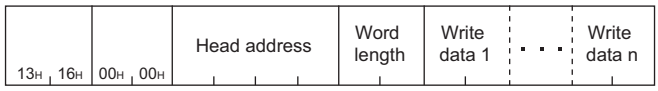
Do not write data in the "system area" of the buffer memory. Doing so may cause a programmable controller system malfunction.

Request data

ASCII



Binary



■Head address

Specify the head address of the buffer memory to be written in. (☞ Page 112 Head address)

■Word length

Specify the word length of the buffer memory to be written in. (☞ Page 113 Word length)

- Specification range: 1H to 1E0H (480)

Response data

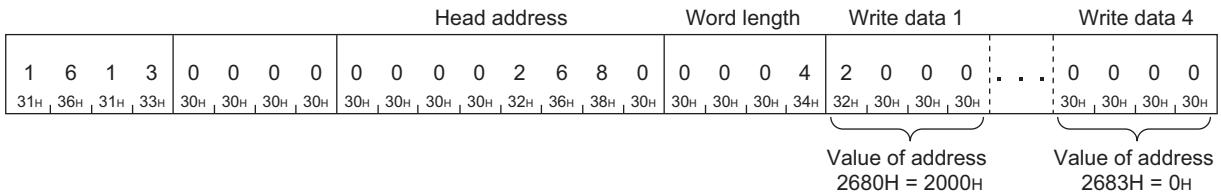
There is no response data for Write command.

Communication example

Write the data of buffer memory address 2680H to 2683H (9856 to 9859).

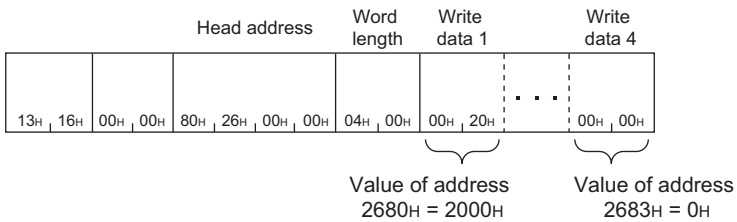
■When communicating data in ASCII code

(Request data)



■When communicating data in binary code

(Request data)



5.5 Extend Unit (Accessing to Buffer Memory of Intelligent Function Module)

The section describes the commands to read or write the buffer memory of intelligent function module.

The following intelligent function modules of MELSEC-Q series can be accessed by the command of Extend Unit. To access a buffer memory other than those of the following modules, specify the module access device from Read (command: 0401, subcommand: 008☐) or Write (command: 1401, subcommand: 008☐) and access the buffer memory.

☞ Page 199 Access to the module access device

Module model name	Head address*1	Module number when mounted in slot 0*1
QD35ID1/ID2 ID Interface module	4000H	0000H
Q62AD-DGH, Q64AD (-GH), Q66AD-DG, Q68AD-G, Q68ADV/ADI Analog-Digital Conversion module	1008H	
Q62DA (-FG), Q62DAN, Q64DA, Q64DAN, Q66DA-G, Q68DAV/Q68DAI, Q68DAVN/Q68DAIN Digital-Analogue Conversion module	1008H	
Q64AD2DA Analog Input/Output module	2000H	
Q62HLC Loop Control module	10000H	
Q64TCTT/Q64TCRT Temperature Control module	1000H	
Q61LD Load Cell Input module	2000H	
Q64TCTTBW/Q64TCRTBW Temperature Control module	1000H	0001H
Q64TD, Q64RD Thermocouple Input module (Function version B)	2000H	0000H
Q64TD, Q64TDV-GH, Q64RD (-G) Thermocouple Input module (Function version C)	8000H	
Q68TD-G-H01, Q68TD-G-H02 Channel Isolated Thermocouple Input module	1008H	
Q68RD3-G Channel Isolated RTD Input module	1008H	
QD51 (-R24) Intelligent Communication module	10000H	
QD60P8-G Channel Isolated Pulse Input module	2000H	
QD62, QD62E, QD62D High speed counter module	3CH	
QD63P6 Multichannel High-speed counter module	2000H	
QD63P6 4 Mpps compatible High-speed counter module	2000H	
QD70P4/P8 Positioning module	5000H	
QD70D4/D8 Positioning module	5000H	0001H
QD72P3C3 Positioning module with Built-in counter function	5000H	0000H
QD75P1/P2/P4, QD75D1/D2/D4, QD75M1/M2/M4, QD75MH1/MH2/MH4 Positioning module	10000H	
QD81DL96 High-Speed Data Logger module	10000H	
QJ61BT11 (N) CC-Link System Master/Local module	10000H	
QJ61CL12 CC-Link/LT Master module	01B4H	
QJ71C24N (-R2/R4), QJ71C24 (-R2) Serial Communication module	10000H	
QJ71AS92 AS-i Master module	10000H	
QJ71CMO (N) Modem Interface module	10000H	0000H
QJ71E71-100/-B5/-B2 Ethernet interface module	10000H	
QJ71FL71-T/-B5/-B2 -F01 FL-net (OPCN-2) Interface module	10000H	
QJ71MES96 MES Interface module	10000H	
QJ71WS96 Web server module	10000H	

*1 "Head address" and "Module number when mounted in slot 0" are used for the request data.

☞ Page 120 Read (command: 0601)
☞ Page 122 Write (command: 1601)

Data to be specified in command

Head address

Specify the head address of the buffer memory to be read or written. The order of sending data is the same as that of Memory (Accessing to Buffer Memory). (☞ Page 112 Head address)

Calculate the start address as follows.

Start address = {(Buffer memory address of the module × 2) in hexadecimal} + ("Start address" shown in the table of Extend Unit (Accessing to Buffer Memory of Intelligent Function Module))*1

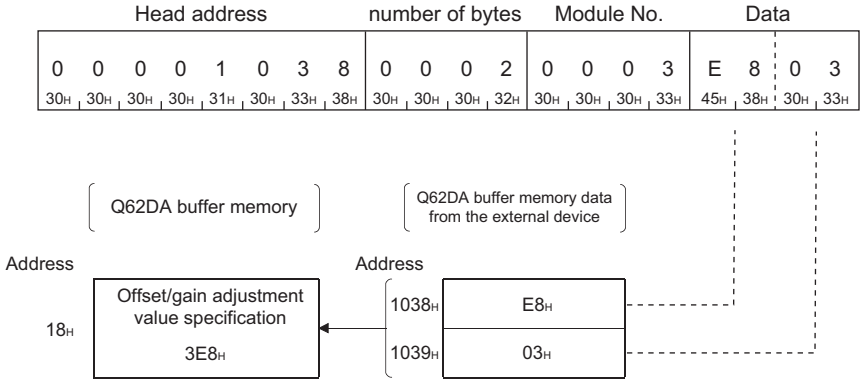
*1 Use the "start address" shown in the following table for the calculation formula.
☞ Page 117 Extend Unit (Accessing to Buffer Memory of Intelligent Function Module)

Ex.

When specifying the buffer memory address 18H of Q62DA whose input/output signal is from 30H to 4FH (Module No.: 03H)

(18H × 2) + 1008H = 30H + 1008H = 1038H

Q62DA buffer memory	Head address	Buffer memory address of module
D/A conversion enable/disable	1008H	0H
	1009H	
CH.1 digital value	100AH	1H
	100BH	
CH.2 digital value	100CH	2H
	100DH	
System area	—	3H to 10H
Offset/gain adjustment value specification	1038H	18H



Number of bytes

Specify the number of bytes of the buffer memory to be read or written. The order of sending data is the same as that of Memory (Accessing to Buffer Memory). (☞ Page 113 Word length)

Since one area consists of 2 bytes (one word) in the buffer memory of the intelligent function module, specify the number of bytes by doubling the number of addresses.

Module No.

Specify the intelligent function module which is to be read from or written to.

Calculate the module No. as follows.

Module No. = (First 3 digits when the start I/O number of the intelligent function module is expressed in 4 digits) + ("Module number when mounted in the slot 0" in the table of Extend Unit (Accessing to Buffer Memory of Intelligent Function Module)[*1]

*1 Use the "Module No. when mounted in the slot 0" shown in the following table as the module No. when the module is loaded in the slot 0 for the calculation formula.

☞ Page 117 Extend Unit (Accessing to Buffer Memory of Intelligent Function Module)

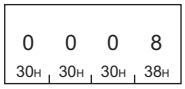
■When communicating data in ASCII code

Convert the module No. into a 4-digit ASCII code (hexadecimal), then send them in order from the upper byte to the lower byte.

Ex.

When the start I/O number is 0080H

The module No. becomes "0008". Send them in order from "0".



■When communicating data in binary code

Send the module No. in order from the lower byte to the upper byte.

Ex.

When the start I/O number is 0080H

The module No. becomes 0008H. Send 08H first, and then send 00H.



Read data, write data

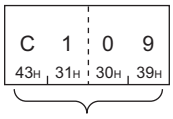
In case of reading, the read values of buffer memory are stored. In case of writing, the writing data is stored.

■When communicating data in ASCII code

The data is stored in 2-digit ASCII code (hexadecimal).

Ex.

For 09C1H



Data for one buffer memory address

■When communicating data in binary code

The data is stored in one byte unit in order from the lower byte to the upper byte.

Ex.

For 09C1H



Read (command: 0601)

This command reads the data in the buffer memory of intelligent function module.

Request data

ASCII

0 6 0 1	0 0 0 0	Head address	No. of bytes	Module No.
30H 36H 30H 31H	30H 30H 30H 30H			

Binary

		Head address	No. of bytes	Module No.
01H 06H	00H 00H			

■Head address

Specify the head address of the buffer memory which is to be read from. (☞ Page 118 Head address)

■Number of bytes

Specify the head address of the buffer memory which is to be read from. (☞ Page 118 Number of bytes)

- Specification range: 2H to 780H (1920)

■Module No.

Specify the intelligent function module which is to be read from. (☞ Page 119 Module No.)

Response data

The value read from buffer memory is stored in hexadecimal. (☞ Page 119 Read data, write data)

Read data 1	· · ·	Read data n

Communication example

The content of the buffer memory address 1H to 2H of Q62DA whose input/output signal is from 30H to 4FH (Module No.: 03H) is read.

■When communicating data in ASCII code

(Request data)

				Head address								No. of bytes				Module No.			
0	6	0	1	0	0	0	0	0	0	0	0	1	0	0	A	0	0	0	4	0	0	0	3
30H	36H	30H	31H	30H	30H	30H	30H	30H	30H	30H	30H	31H	30H	30H	41H	30H	30H	30H	34H	30H	30H	30H	33H

(Response data)

0	1	0	0	1	2	0	0
30H	31H	30H	30H	31H	32H	30H	30H
Value of address 1H = 0001H				Value of address 2H = 0012H			

■When communicating data in binary code

(Request data)

		Head address				No. of bytes		Module No.	
01H	06H	00H	00H	0AH	10H	00H	00H	04H	00H	03H	00H

(Response data)

01H	00H	12H	00H
Value of address 1H = 0001H		Value of address 2H = 0012H	

Write (command: 1601)

This command writes the data in the buffer memory of intelligent function module.

Request data

ASCII

1 6 0 1	0 0 0 0	Head address	No. of bytes	Module No.	Write data
31H 36H 30H 31H	30H 30H 30H 30H				

Binary

		Head address	No. of bytes	Module No.	Write data
01H 16H	00H 00H				

■Head address

Specify the head address of the buffer memory to be written in. (☞ Page 118 Head address)

■Number of bytes

Specify the number of bytes of the buffer memory to be written in. (☞ Page 118 Number of bytes)

- Specification range: 2H to 780H (1920)

■Module No.

Specify the intelligent function module which is to be written in. (☞ Page 119 Module No.)

■Write data

Specify the data to be written in the buffer memory. (☞ Page 119 Read data, write data)

Response data

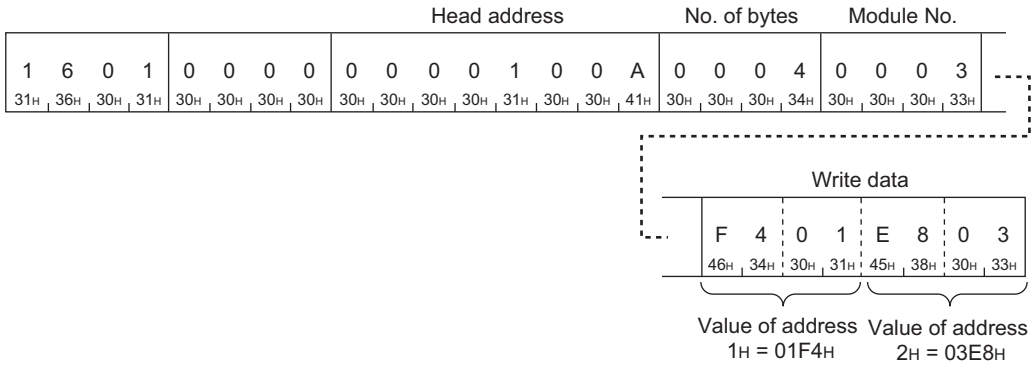
There is no response data for Write command.

Communication example

Write the data in the buffer memory address 1H to 2H of Q62DA whose input/output signal is from 30H to 4FH (Module No.: 03H).

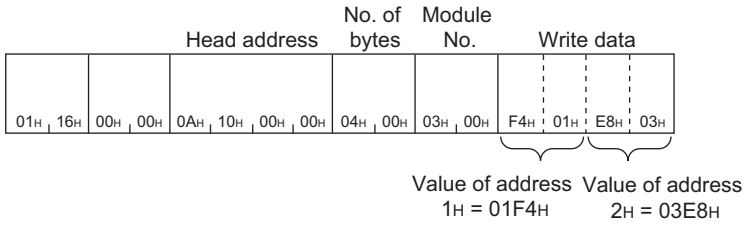
■When communicating data in ASCII code

(Request data)



■When communicating data in binary code

(Request data)



5

5.6 Remote Control (Remote Operation)

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

Before the remote operation

When the accessed device or module is turned on or 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 CPU module is in the RUN state, the switch will return to RUN state after resetting the module.

When the CPU module to be accessed is in system protection

Remote operation from the external device is not available. An error occurs at the access destination, and an abnormal response is sent back to the external device. Unlock the system protection of the CPU module side, and resend the request message.

When executing the remote operation to SLMP compatible device

It is recommended to use UDP protocol for remote operation. If TCP is used, the connection will be terminated when resetting. Therefore, reestablishing of connection is necessary.

Operable station in one command

Only one station can be operated remotely by one command.

Remote Run (command: 1001)

This command executes the 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 state. Even if the switch is in the STOP state, Remote Run (command: 1001) will be completed normally. However, the access destination does not become the RUN state.

Request data

ASCII

1 0 0 1	0 0 0 0	Mode	Clear mode	0 0
31H 30H 30H 31H	30H 30H 30H 30H			30H 30H

Binary

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

■Mode

This mode specifies whether the remote RUN can be executed forcibly by the device other than the external device which performed the remote STOP/remote PAUSE. If the forced execution is not allowed, remote RUN can be executed only by the external device which performed the remote STOP/remote PAUSE.

Forced execution is used when the external device which performed the remote operation cannot execute the remote RUN because of a trouble on the device.

Item	Mode	
	ASCII code	Binary code
Forced execution not allowed. (Remote RUN cannot be executed when other device is performing the remote STOP/remote PAUSE.)	0 0 0 1 30H 30H 30H 31H	01H 00H
Forced execution allowed. (Remote RUN can be executed even when other device is performing the remote STOP/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 or not when starting the calculation for the remote RUN. The device which received the remote RUN request turns to the RUN state after the clear (initialization) processing of device.

In case of CPU module, the clear (initialization) processing of device is executed according to the "Initial Device Value" of "PLC File" in PLC parameter.

Item	Mode	
	ASCII code	Binary code
Do not clear the device	0 0 30H 30H	00H
Clear all devices except that in the latch range	0 1 30H 31H	01H
Clear all devices including that in the latch range	0 2 30H 32H	02H

Response data

There is no response data for 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 the 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 (command: 1002)

This command executes the remote STOP to the access destination module.

Request data

ASCII

1	0	0	2	0	0	0	0	0	0	0	1
31H	30H	30H	32H	30H	30H	30H	30H	30H	30H	30H	31H

Binary

02H	10H	00H	00H	01H	00H

Response data

There is no response data for Remote Stop command.

Communication example

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

Remote Pause (command: 1003)

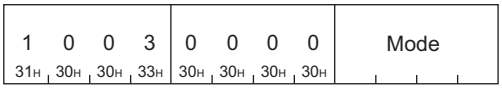
This command executes the remote PAUSE to the access destination module.

Point

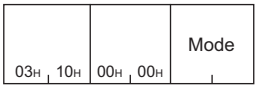
Remote PAUSE can be executed when the switch of the access destination module is in the RUN state. Even if the switch is in the STOP state, Remote Pause (command: 1003) will be completed normally. However, the access destination does not become the PAUSE state.

Request data

ASCII



Binary



■Mode

This mode specifies whether the remote PAUSE can be executed forcibly by the device other than the external device which performed the remote STOP/remote PAUSE. If the forced execution is not allowed, remote PAUSE can be executed only by the external device which performed the remote STOP/remote PAUSE.

Forced execution is used when the external device which performed the remote operation cannot execute the remote PAUSE because of a trouble on the device.

Item	Mode	
	ASCII code	Binary code
Forced execution not allowed. (Remote PAUSE cannot be executed when other device is performing the remote STOP/remote PAUSE.)	0 0 0 1 30H 30H 30H 31H	01H 00H
Forced execution allowed. (Remote PAUSE can be executed even when other device is performing the remote STOP/remote PAUSE.)	0 0 0 3 30H 30H 30H 33H	03H 00H

Response data

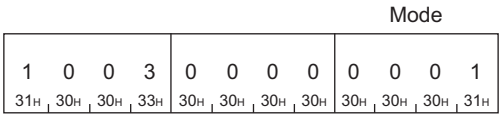
There is no response data for Remote Pause command.

Communication example

Set mode to "Forced execution not allowed" when executing the remote PAUSE.

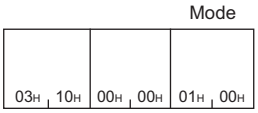
■When communicating data in ASCII code

(Request data)



■When communicating data in binary code

(Request data)



Remote Latch Clear (command: 1005)

This command executes the 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 the request from the other external device, the Remote Latch Clear cannot be executed. An abnormal completion of the command will occur. Cancel the remote STOP or remote PAUSE before executing the command.

Request data

ASCII

1	0	0	5	0	0	0	0	0	0	0	1
31H	30H	30H	35H	30H	30H	30H	30H	30H	30H	30H	31H

Binary

05H	10H	00H	00H	01H	00H

Response data

There is no response data for Remote Latch Clear command.

Communication example

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

Remote Reset (command: 1006)

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

Point

- If there is a setting of remote RESET enable/disable in the parameter of the access destination before the remote RESET is executed, enable the remote RESET. Before executing the 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.
- When remote RESET is executed, the response request may not be sent back to the external device since the access destination is reset.

Request data

ASCII

1	0	0	6	0	0	0	0	0	0	0	1
31H	30H	30H	36H	30H	30H	30H	30H	30H	30H	30H	31H

Binary

06H	10H	00H	00H	01H	00H

Response data

There is no response data for Remote Reset command.

Communication example

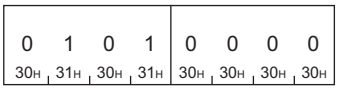
Send request messages from the external device by using the message format shown in "Request data" above.

Read Type Name (command: 0101)

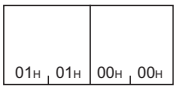
This command reads the model name and model code of the access destination module.

Request data

ASCII



Binary

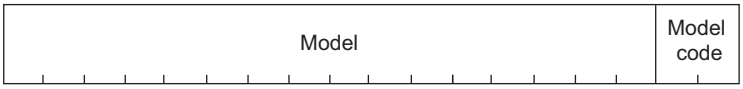


Response data

ASCII



Binary



■Model name

16 characters from the upper byte of the module model are stored.

If the model to be read is less than 16 characters, space (20H) is stored for the remaining character. When communicating data in binary code, the module model is stored in ASCII code.

■Model code

The following model codes will be stored.

When communicating in ASCII code, the data is stored in order from the upper byte to the lower byte.

When communicating in binary code, the data is stored in order from the lower byte to the upper byte. (☞ Page 133 Communication example)

Model name	Model code
Q00JCPU	250H
Q00CPU	251H
Q01CPU	252H
Q02CPU, Q02HCPU, Q02PHCPU	41H
Q06HCPU, Q06PHCPU	42H
Q12HCPU, Q12PHCPU	43H
Q25HCPU, Q25PHCPU	44H
Q12PRHCPU	4BH
Q25PRHCPU	4CH
Q00UJCPU	260H
Q00UCPU	261H
Q01UCPU	262H
Q02UCPU	263H
Q03UDCPU, Q03UDECPU	268H
Q03UDVCPU	366H
Q04UDHCPU, Q04UDEHCPU	269H
Q04UDVCPU	367H
Q06UDHCPU, Q06UDEHCPU	26AH
Q06UDVCPU	368H
Q10UDHCPU, Q10UDEHCPU	266H
Q13UDHCPU, Q13UDEHCPU	26BH
Q13UDVCPU	36AH
Q20UDHCPU, Q20UDEHCPU	267H
Q26UDHCPU, Q26UDEHCPU	26CH
Q26UDVCPU	36CH
Q50UDEHCPU	26DH
Q100UDEHCPU	26EH
L02SCPU, L02SCPU-P	543H
L02CPU, L02CPU-P	541H
L06CPU, L06CPU-P	544H
L26CPU, L26CPU-P	545H
L26CPU-BT, L26CPU-PBT	542H
QS001CPU	230H
LJ72GF15-T2	0641H
NZ2GF-ETB	0642H
R04CPU	4800H
R08CPU	4801H
R16CPU	4802H
R32CPU	4803H
R120CPU	4804H

Point

- Distinguish the model of CPU module by model code.
- When the command is executed for the RCPU with the connected station of other than the MELSEC iQ-R series, "RCPU" is stored in the model code and "0360H" in the model code.

Communication example

Execute the command to Q02UCPU, and read the model name and model code.

■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)

Model															Model code			
Q	0	2	U	C	P	U									0	2	6	3
51H	30H	32H	55H	43H	50H	55H	20H	20H	20H	20H	20H	20H	20H	20H	30H	32H	36H	33H

■When communicating data in binary code

(Request data)

01H	01H	00H	00H

(Response data)

Model															Model code	
Q	0	2	U	C	P	U										
51H	30H	32H	55H	43H	50H	55H	20H	20H	20H	20H	20H	20H	20H	20H	63H	02H

5.7 Remote Password (Remote Password)

This section describes the commands that execute the remote password unlock or lock.

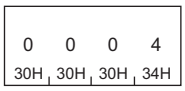
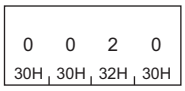
Precautions

- The number of password characters differs between the MELSEC iQ-R series module or MELSEC-Q/L series module. (☞ Page 134 Remote password length)

Data to be specified in command

Remote password length

- The number of password characters of the MELSEC-Q/L series module is fixed to four.
- The number of password characters of the MELSEC iQ-R series module can be specified in the range of 6 to 32.

Item	Remote password length	
	ASCII code	Binary code
MELSEC-Q/L series module (fixed to four characters)		
MELSEC iQ-R series module (when the number of remote password characters is 32)		

Remote password

The remote password is set for in the CPU module or MELSEC iQ-R series-compatible intelligent function module with an engineering tool.

When communicating data in binary code, specify the remote password in ASCII code.

■When communicating data in ASCII or binary code

The set remote password is sent from the first character.

Lock (command: 1631)

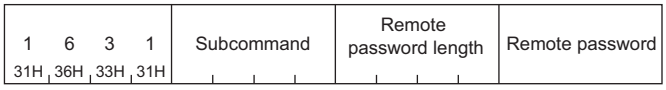
This command specifies the remote password and activates the locked state to unlocked state. (The communication with SLMP compatible devices is disabled.)

Point

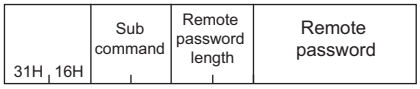
- When the Lock command is sent to an external device that is already in the locked state, the device remains in the state. (The password is not verified either.)
- This command can be executed only for the connected stations connected to an external device. This command cannot be executed for the modules of other stations via a network.

Request data

ASCII



Binary



■Subcommand

Subcommand	
ASCII code	**Binary code**
0 0 0 0 30H 30H 30H 30H	00H 00H

■Remote password length

Specify the number of remote password characters. (☞ Page 134 Remote password length)

■Remote password

Specify the set remote password. (☞ Page 134 Remote password)

Response data

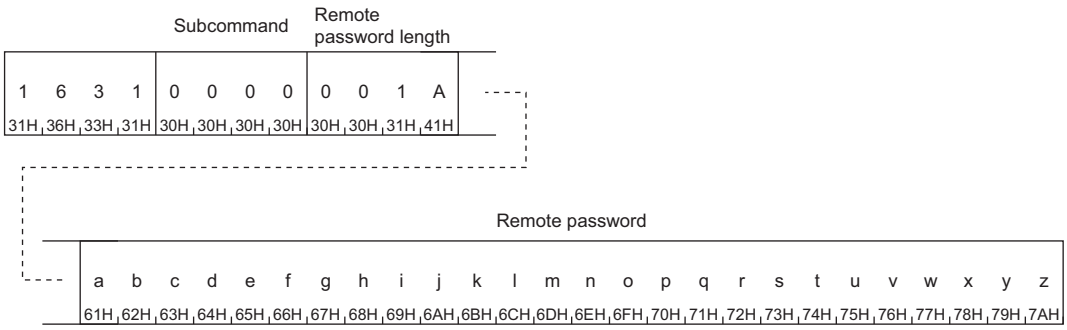
There is no response data for Lock command.

Communication example

The MELSEC iQ-R series module is set to the locked state with the remote password "abcdefghijklmnopqrstuvwxyz".

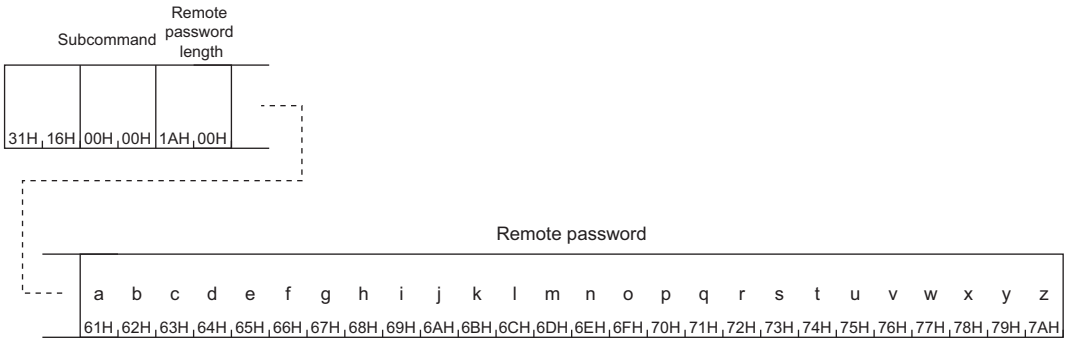
■When communicating data in ASCII code

(Request data)



■When communicating data in binary code

(Request data)



Unlock (command: 1630)

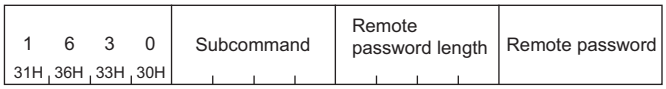
This command specifies the remote password and activates the unlocked state from the locked state. (The communication with SLMP compatible devices is enabled.)

Point

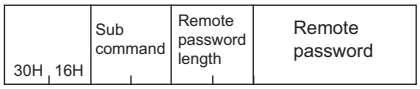
- If the password has been incorrectly entered continuously for the predetermined number of times, the lock cannot be disengaged for a certain period of time.
- When the Unlock command is sent to an external device that is already in the unlocked state, the device remains in the state. (The password is not verified either.)
- This command can be executed only for the connected stations connected to an external device. This command cannot be executed for the modules of other stations via a network.

Request data

ASCII



Binary



■Subcommand

Subcommand	
ASCII code	**Binary code**
0 0 0 0 30H 30H 30H 30H	00H 00H

■Remote password length

Specify the number of remote password characters. (☞ Page 134 Remote password length)

■Remote password

Specify the set remote password. (☞ Page 134 Remote password)

Response data

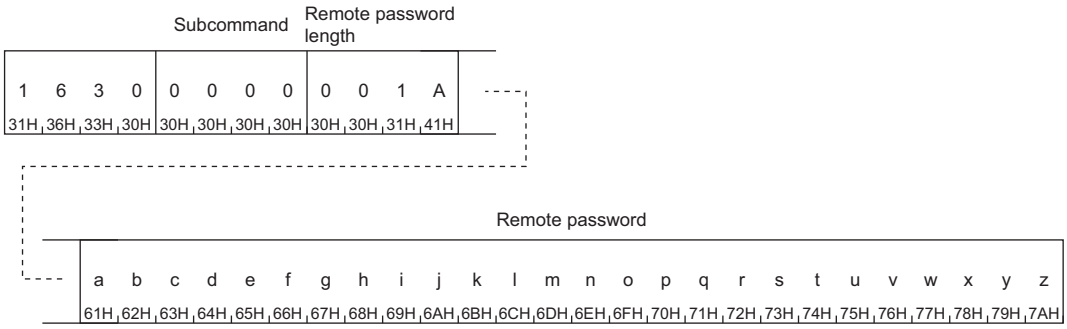
There is no response data for Unlock command.

Communication example

The MELSEC iQ-R series module is set to the unlocked state with the remote password "abcdefghijklmnopqrstuvwxyz".

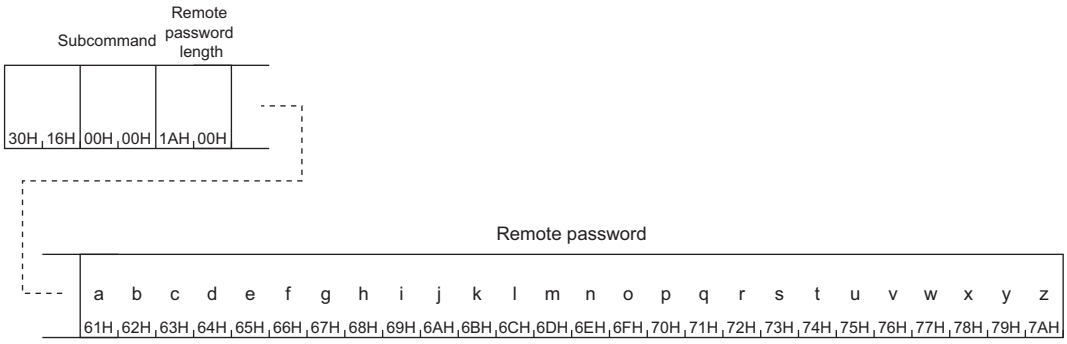
■When communicating data in ASCII code

(Request data)



■When communicating data in binary code

(Request data)



5.8 File (File Control)

This section describes the command to control files in the SLMP compatible device and the CPU module.
The File command is used for an external device to read parameters and programs from the CPU module and save them. The command is also used to write parameters and programs in an external device to the CPU module according to control contents.
For the file names, extensions, and storage locations of the files stored in the CPU modules, refer to the manual for the CPU module used.

Point

- Files not described in the user's manual for the CPU module used may be accessed using File (file control) commands. However, since the files are for system use, do not access them.

Data to be specified in command

Password

■When the subcommand is 0000 or 0004

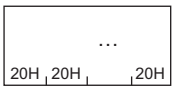
Specify the password for the access destination file. The length of a password is fixed to 4 characters when the subcommand is 0000 and 32 characters when the subcommand is 0004. When the password is shorter than the fixed length, spaces (20H) are entered to the blanks.

When communicating data in binary code, specify the password in ASCII code.

Point

- Access to program files, device comment files, and device initial files can be enabled or disabled. Each file as "read only" or "read/write disable" can be set.
- When not setting any password, add a space (code: 20H).

When communicating data in ASCII or binary code

When the password is set	When the password is not set
The following shows an example when the password is "ABCDEF". (Same regardless of ASCII code and binary code)  	Specify spaces (code: 20H) for the number of password characters. 

■When the subcommand is 0040

Specify the password and its number of characters for the access destination file.

Before specifying the password, specify the number of password characters in hexadecimal within 6 to 32 characters.

Specify the password within 6 to 32 characters.

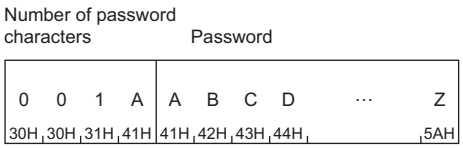
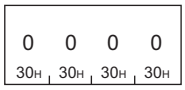
When communicating data in binary code, specify the password in ASCII code.

When the password is not set, "0" is specified as the number of password characters to specify no password and the data is aligned left.

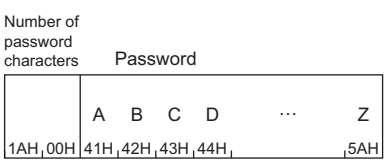
Point

If the password has been incorrectly entered continuously for the predetermined number of times, the lock cannot be disengaged for a certain period of time.

When communicating data in ASCII code

When the password is set	When the password is not set
The following shows an example when the password is "ABCDEFGHIJKLMNOPQRSTUVWXYZ" (the number of password characters is 26). Convert the number of password characters into a 4-digit ASCII code, and send them in order from the upper byte to the lower byte.  	Specify "30H" (0) as the number of password characters. No password is specified. 

When communicating data in binary code

When the password is set	When the password is not set
The following shows an example when the password is "ABCDEFGHIJKLMNOPQRSTUVWXYZ" (the number of password characters is 26). Send the data in order from the lower byte to the upper byte using 2-byte numeral values for the number of password characters.  	Specify "0" as the number of password characters. No password is specified. 

Drive No.

When the access destination is the MELSEC iQ-R series module, specify the drive to be file-controlled according to the following table.

Drive No.	Target drive	Drive No.	Target drive
0001H	Device/label memory*1	0004H	Data memory
0002H	SD memory card	—	
0003H	Device/label memory*2		

*1 This drive is the file storage area of the device/label memory. If 0001H is specified as the drive No., 0003H is accessed.

*2 The files related to the module control, such as the program files or parameter files, are stored in $MELPRJ$. When accessing to those files, refer to the precautions described in the following page.

☞ Page 148 Precautions

When the access destination is the MELSEC-Q/L series module, specify the drive for file control according to the following table.

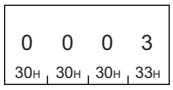
Drive No.	Target drive	Drive No.	Target drive
0000H	Program memory	0003H	Standard RAM
0001H	SRAM card	0004H	Standard ROM
0002H	Flash card, ATA card, SD memory card	—	—

■When communicating data in ASCII code

Send drive No. from the upper byte to the lower byte.

Ex.

When drive No. is 0003H



■When communicating data in binary code

Send drive No. in order from the lower byte to the upper byte.

Ex.

When drive No. is 0003H



Number of file name characters, file name

■When the subcommand is 0000 or 0004

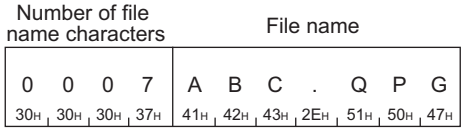
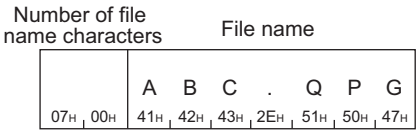
Specify the number of file name characters set in "File name" in hexadecimal as the number of file name characters. The number includes extensions.

Specify the file name within 12 characters (8 one-byte characters at a maximum[*1] + period + 3-character extension).

Both 1-byte characters (ASCII code) and 2-byte characters (Shift-JIS kanji code) can be used for file names.

*1 For 2-byte characters, the file name must be within 4 characters.

When communicating data in binary code, specify "File name" in ASCII code.

When communicating data in ASCII code	When communicating data in binary code
The following shows an example when "File name" is "ABC.QPG" (seven-character file name). Send the number of file name characters in order from the upper byte to the lower byte.  	The following shows an example when "File name" is "ABC.QPG" (seven-character file name). Send the number of file name characters in order from the lower byte to the upper byte. 

■When the subcommand is 0040

Specify the number of characters from the root directory to the file name specified in "File name" in hexadecimal as the number of file name characters. The number includes extensions.

Specify the file path from the root directory to the file name except for "Drive name:\" in UTF-16.

File names can also be specified with the path including "\" at the head.

Specify the file name within 64 characters (60 one- or two-byte characters at a maximum + period + 3-character extension).

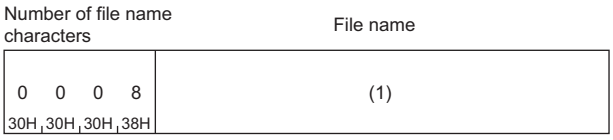
Specify the number of characters for the path from the file name and root directory within 252 characters.

Point

Some characters cannot be used in file names. For the characters that cannot be used in file names, refer to the manual for the module used. (Manual for the module used)

When communicating data in ASCII code

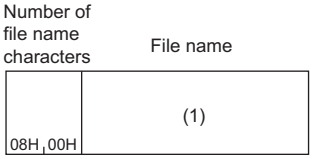
The following shows an example when "File name" is "LINE.CSV" (8-character file name).
As the file name, specify the ASCII code value that describes "File name" in UTF-16.



—	File name	LINE.CSV
	UTF-16 (hexadecimal)	004C0049004E0045002E004300530056
(1)	ASCII code (hexadecimal)	3030344330303439303034453030343530303245303034333030353330303536

When communicating data in binary code

The following shows an example when "File name" is "LINE.CSV" (8-character file name).
As the file name, specify the data in order from the lower byte to the upper byte with the value that describes "File name" in UTF-16.



—	File name	LINE.CSV
	UTF-16 (hexadecimal)	004C0049004E0045002E004300530056
(1)	Binary code (hexadecimal)	4C0049004E0045002E00430053005600

Attribute

Specify the file attributes.

There are two types for the file attributes: "Read only" and "Read, write enabled".

Point

- Existing file attributes can be checked by Read Directory/File (command: 1810). (☞ Page 149 Read Directory/File (command: 1810))
- Existing file attributes can be changed by Change File State (command:1825). (☞ Page 172 Change File State (command: 1825))

File pointer No.

Specify the number for the CPU module to manage files.

A file pointer No. is obtained at file open, and stored in the response data of Open File (command: 1827). When specifying File pointer No. in the request data, input the same value as stored in the response data of Open File (command: 1827).

■When communicating data in ASCII code

Send the data in order from the upper byte to the lower byte in 4-digits of ASCII code.

Ex.

When the file pointer No. is AH

0	0	0	A
30H	30H	30H	41H

■When communicating data in binary code

Send the data in order from the lower byte to the upper byte in 2-byte numerical values.

Ex.

When the file pointer No. is AH

0AH, 00H

Precautions

The MELSEC iQ-R series module cannot access some file types through SLMP.

For the file types that can be accessed through SLMP, refer to the manual for the module used.

Execution procedure

The following shows the procedure for file control.

Procedure for reading file contents

1. Checking for the presence of a file

By Read Directory/File (command: 1810) or Search Directory/File (command: 1811), check for the presence of a file.

☞Page 149 Read Directory/File (command: 1810)

☞Page 159 Search Directory/File (command: 1811)

2. Opening the file

By Open File (command: 1827), lock the file to prevent the file contents from being changed by another device. (☞ Page 178 Open File (command: 1827))

3. Reading data from the file

By Read File (command: 1828), read data from the file. (☞ Page 181 Read File (command: 1828))

4. Closing the file

By Close File (command: 182A), unlock the file. (☞ Page 187 Close File (command: 182A))

Point

Take a note of the following information about the file which is read to the external device. This information is required for commands such as writing data in a file. (Only when the access destination is the MELSEC-Q/L series module)

- File No. (read by Search Directory/File (command: 1811))
- File name, attribute, file size (read by Read Directory/File (command: 1810))

Procedure for creating a new file and writing data

Point

Before creating a new file, reserve an enough free area in the target memory. Use the engineering tool to check and reserve the free area of the target memory.

■When the file to be created is a headline sentence file (*.DAT), sequence program (*.PRG), program file (*.QPG), FB file (*.PFB), device comment file (*.QCD, *.DCM), or device initial file (*.QDI, *.DID)

1. Checking for the presence of a file

By Read Directory/File (command: 1810) or Search Directory/File (command: 1811), check for the presence of a file.

☞Page 149 Read Directory/File (command: 1810)
☞Page 159 Search Directory/File (command: 1811)

2. Registering a file name and reserving free space

By New File (command: 1820), create a new file. Use an extension other than DAT, PRG, QPG, PFB, QCD, DCM, QDI, or DID. (☞ Page 162 New File (command: 1820))

3. Opening the file

By Open File (command: 1827), lock the file to prevent the file contents from being changed by another device. (☞ Page 178 Open File (command: 1827))

4. Writing data to the file

By Write File (command: 1829), write data to the file. (☞ Page 184 Write File (command: 1829))

5. Closing the file

By Close File (command: 182A), unlock the file. (☞ Page 187 Close File (command: 182A))

6. Copying the file

By Copy File (command: 1824), copy the DAT, PRG, QPG, PFB, QCD, DCM, QDI, or DID file. After copying, delete the source file as needed. (☞ Page 168 Copy File (command: 1824))

■When the file to be created is not a headline sentence file (*.DAT), sequence program (*.PRG), program file (*.QPG), FB file (*.PFB), device comment file (*.QCD, *.DCM), or device initial file (*.QDI, *.DID)

1. Checking for the presence of a file

By Read Directory/File (command: 1810) or Search Directory/File (command: 1811), check for the presence of a file.

☞Page 149 Read Directory/File (command: 1810)
☞Page 159 Search Directory/File (command: 1811)

2. Registering a file name and reserving free space

By New File (command: 1820), create a new file. (☞ Page 162 New File (command: 1820))

3. Opening the file

By Open File (command: 1827), lock the file to prevent the file contents from being changed by another device. (☞ Page 178 Open File (command: 1827))

4. Writing data to the file

By Write File (command: 1829), write data to the file. (☞ Page 184 Write File (command: 1829))

5. Closing the file

By Close File (command: 182A), unlock the file. (☞ Page 187 Close File (command: 182A))

6. Confirming the file No.*1

By Search Directory/File (command: 1811), check the file No. and write it down. The file No. is required when Read Directory/File (command: 1810) is used. (☞ Page 159 Search Directory/File (command: 1811))

*1 This step is required only when the access destination is the MELSEC-Q/L series module.

Procedure for copying a file

Point

Before copying a file, reserve an enough free area in the target memory. Use the engineering tool to check and reserve the free area of the target memory.

1. Checking for the presence of a file

By Read Directory/File (command: 1810) or Search Directory/File (command: 1811), check for the presence of a file.

☞ Page 149 Read Directory/File (command: 1810)
☞ Page 159 Search Directory/File (command: 1811)

2. Copying the file

By Copy File (command: 1824), copy the file. (☞ Page 168 Copy File (command: 1824))

3. Confirming the file No.*1

When a new file is created by copying, by Search Directory/File (command 1811), check the file No. and write it down. The file No. is required when Read Directory/File (command: 1810) is used. (☞ Page 159 Search Directory/File (command: 1811))

*1 This step is required only when the access destination is the MELSEC-Q/L series module.

Procedure for overwriting data in the existing file

Point

- Before overwriting a file, reserve an enough free area in the target memory. Use the engineering tool to check and reserve the free area of the target memory.
- When the file sizes between the existing file and new file differ or when a PRG or PFB is used, delete a file to be overwritten by Delete File (command: 1822), and write the file data following the "Procedure for creating a new file and writing data" described in the following page.

☞ Page 145 Procedure for creating a new file and writing data

1. Checking for the presence of a file

By Read Directory/File (command: 1810) or Search Directory/File (command: 1811), check for the presence of a file.

☞ Page 149 Read Directory/File (command: 1810)
☞ Page 159 Search Directory/File (command: 1811)

2. Opening the file

By Open File (command: 1827), lock the file to prevent the file contents from being changed by another device. (☞ Page 178 Open File (command: 1827))

3. Writing data to the file

By Write File (command: 1829), write data to the file. (☞ Page 184 Write File (command: 1829))

4. Closing the file

By Close File (command: 182A), unlock the file. (☞ Page 187 Close File (command: 182A))

Procedure for changing file creation date

Execute Change File Date (command: 1826) to change the file creation date. It is not necessary to lock the file by Open File (command: 1827).

Procedure for deleting a file

1. Checking for the presence of a file

By Read Directory/File (command: 1810) or Search Directory/File (command: 1811), check for the presence of a file.

☞ Page 149 Read Directory/File (command: 1810)

☞ Page 159 Search Directory/File (command: 1811)

2. Deleting the file

By Delete File (command: 1822), delete the file. (☞ Page 165 Delete File (command: 1822))

Point

Deleting a file, while the programmable controller system is running, may stop the system. Determine the timing for deleting a file by considering a relationship with the whole programmable controller system.

Precautions

The followings are precautions for file control.

Read file

When the files related to the module control, such as the program files or parameter files, are read, they are used for backup in the external device. Do not edit the read file contents in the external device.

To back up or restore the data in the "$MELPRJ$" folder, execute the backup or restoration for all the files in the "$MELPRJ$" folder.

When not all of the read files are restored to the "$MELPRJ$" folder, the normal operation may not be obtained.

When the protection is executed

When executing the following commands, cancel the protection of the access destination (the system protection of the CPU module, lock of the protection switch of the SD memory card) in advance. If the command is executed while the file is protected, an abnormal completion of the command will occur.

Command	Reference
New File	Page 162 New File (command: 1820)
Delete File	Page 165 Delete File (command: 1822)
Copy File	Page 168 Copy File (command: 1824)
Change File State	Page 172 Change File State (command: 1825)
Change File Date	Page 175 Change File Date (command: 1826)
Write File	Page 184 Write File (command: 1829)

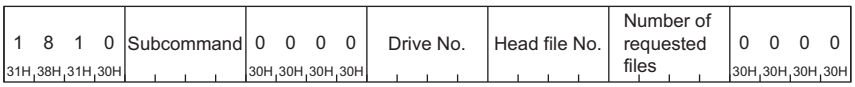
Read Directory/File (command: 1810)

Reads file list information.

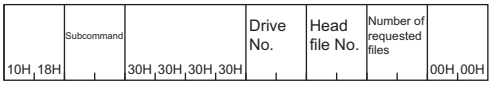
Request data

■When the subcommand is 0000

ASCII



Binary

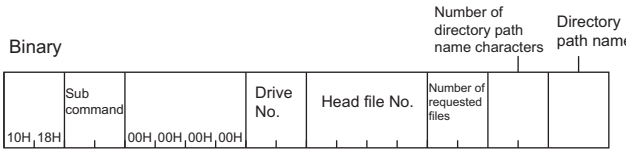


■When the subcommand is 0040

ASCII



Binary



■Subcommand

Subcommand	
ASCII code	**Binary code**
0 0 0 0 30H 30H 30H 30H	00H 00H
0 0 4 0 30H 30H 34H 30H	40H 00H

■Drive No.

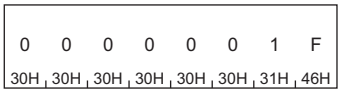
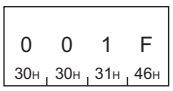
Specify the drive where the file list information is read out. (☞ Page 141 Drive No.)

■Head file No.

Specify the registered No. of the file written in the module. (Specification range: 1H or later)

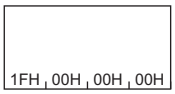
When communicating data in ASCII code, convert a file No. into an 8-digit or 4-digit ASCII code, and send them in order from the upper byte to the lower byte.

The number of digits converted into an ASCII code differs depending on the subcommand.

Subcommand	Number of digits	Example
0040	Converted into an eight-digit ASCII code.	In case of 1FH (8 digits) 
0000	Converted into a four-digit ASCII code.	In case of 1FH (4 digits) 

When communicating data in binary code, send the data in order from the lower byte to the upper byte using four or two-byte numeral values.

The data size of the value differs depending on the subcommand.

Subcommand	Data size	Example
0040	Four bytes	For input (X) (four bytes) 
0000	Two bytes	For input (X) (two bytes) 

Point

The file No. of the file stored in the module can be checked by Search Directory/File (command: 1811). (☞ Page 159 Search Directory/File (command: 1811))

■Number of requested files

Specify the number of files when the file information is read.

Subcommand	Specification range
0000	1 to 36
0040	1 to 36

The data sending order is the same as that for "Head file No.".

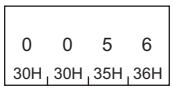
■Number of directory path name characters

Specify the number of directory path name characters in hexadecimal. When "0" is specified as the number of characters, it indicates the root directory.

When communicating data in ASCII code, convert the number of directory path name characters into a 4-digit ASCII code, and send them in order from the upper byte to the lower byte.

Ex.

When the number of directory path name characters is 86 (56H)



When communicating data in binary code, send the data in order from the lower byte to the upper byte using 2-byte numeral values.

Ex.

When the number of directory path name characters is 86 (56H)



■Directory path name

Specify the path name from the root directory in UTF-16.

- When communicating data in ASCII code, specify a numerical value that indicates the directory path name in UTF-16 using the ASCII code. Send the data in order from the upper byte to the lower byte.
- When communicating data in binary code, specify the directory path name with the numerical value indicated in UTF-16. Send the data in order from the lower byte to the upper byte.

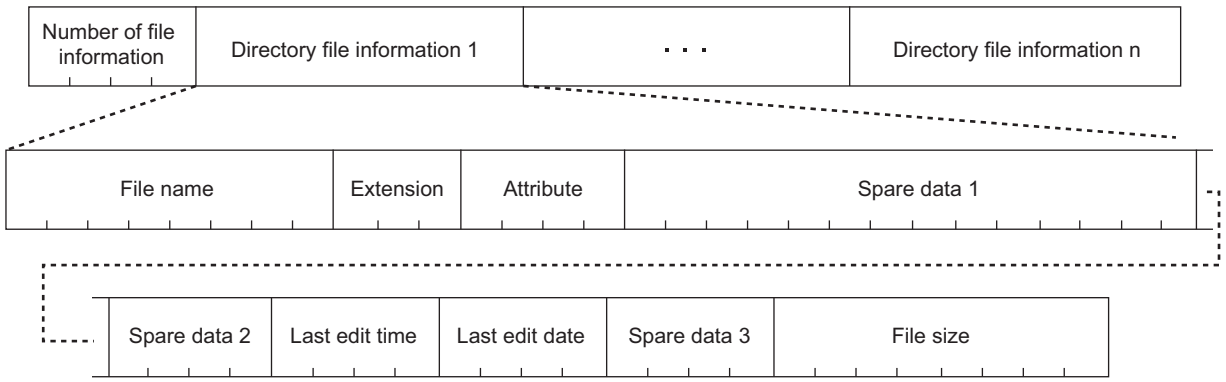
The following shows an example when the directory path name is "SUBDIR".

Path name (UTF-16 (hexadecimal))	S (0053)	U (0055)	B (0042)	D (0044)	I (0049)	R (0052)
ASCII code (hexadecimal)	30303533	30303535	30303432	30303434	30303439	30303532
Binary code (hexadecimal)	5300	5500	4200	4400	4900	5200

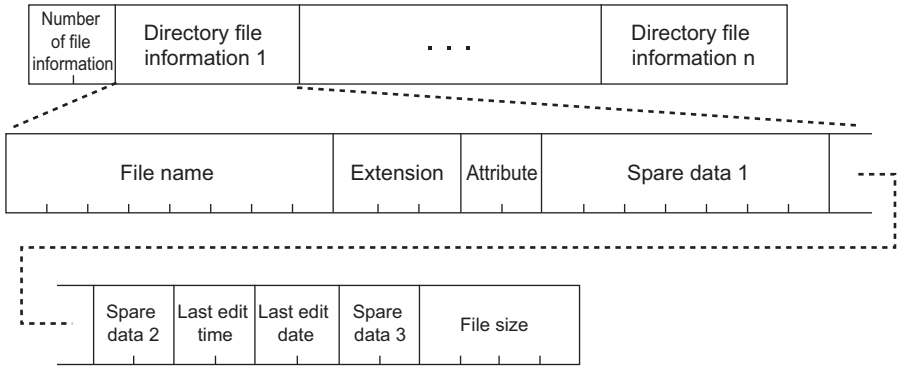
Response data

■When the subcommand is 0000

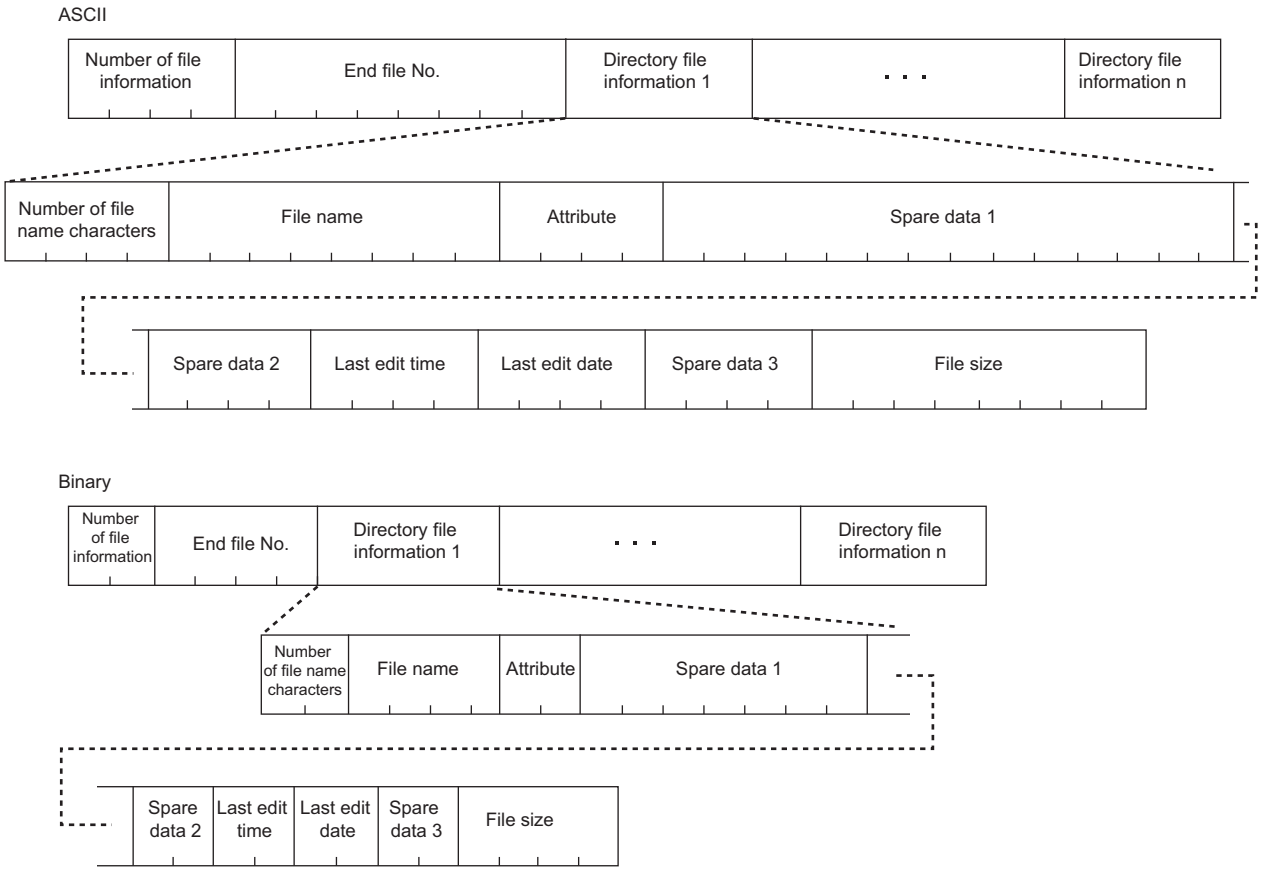
ASCII



Binary



■When the subcommand is 0040



■Number of file information

The number of the file information in the response data is stored. The data storing order is the same as that for "Head file No.

When there is no file after "Head file No." specified in the request data, 0 is stored.

Depending on the file name length, the number of file information may be less than the number of requested files.

■Last file No.

The file Nos. of the files whose file information have been read (including the deleted files that are not stored in the response data) are stored.

Use this area when the file information has not been read in one request. (☞ Page 155 Procedure to read directory file information with the subcommand 0040)

■File name, extension

Directories, file names, and extensions are stored. When communicating data in binary code, file names and extensions are stored in ASCII code.

When the file name is less than 8 characters, spaces (code: 20H) are stored for the remaining part.

When a directory name is stored, spaces are stored in the extension.

The deleted directory names or file names are not stored.

■Number of file name characters, file name

The file names and number of file name characters are stored.

However, the deleted directory names or file names are not stored.

■Attribute

Attributes of the file is stored.

- Read-only directory: 31H, 11H
- Read- and write-enabled directory: 30H, 10H
- Read-only file: 01H, 21H
- Read- and write-enabled file: 00H, 20H

The directory or file whose attribute stores the value other than the above is for the system. Do not access them.

■Spare data

Optional values are stored. Do not use this area.

Spare data 1: 14 digits in ASCII code. Seven bytes in binary code.

Spare data 2 and 3: Four characters in ASCII code. Two bytes in binary code.

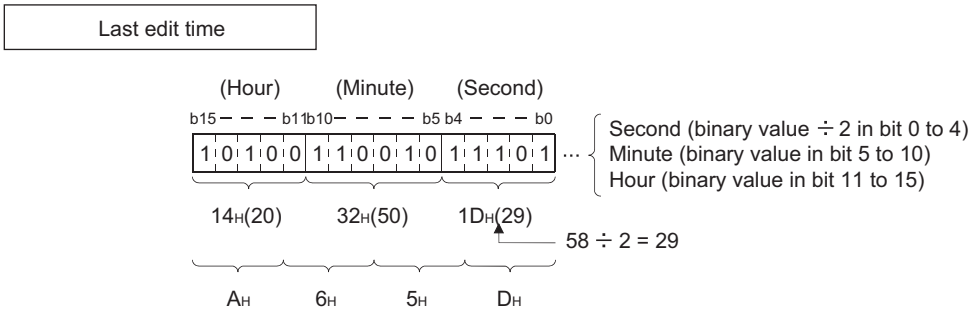
■Last edit time, last edit date

Last edit time and date of the file is stored.

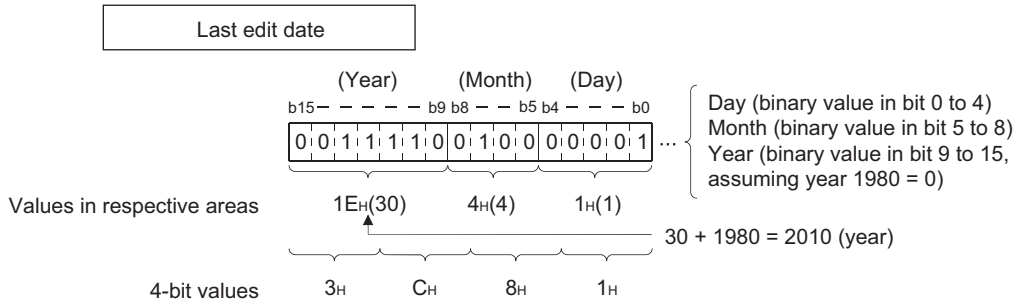
For the directory, the time and date when it was created are stored.

Ex.

When "Last edit time" is 20:50:58, and "Last edit date" is April 1, 2010



ASCII code communication : A65D (sending in order from "A")
Binary code communication : A65DH (sending 5DH and then A6H)



ASCII code communication : 3C81 (sending in order from "3")
Binary code communication : 3C81H (sending 81H and then 3CH)

■File size

File size is stored in bytes.

For the directory, the file size is 0.

Ex.

When the file size is 7168 bytes

ASCII code	Binary code
The file size is stored from the upper byte to the lower byte in 8-digits of ASCII code. (hexadecimal) 0 0 0 0 1 C 0 0 30H 30H 30H 30H 31H 43H 30H 30H	The file size is stored from the lower byte to the upper byte. (hexadecimal) 00H 1CH 00H 00H

■Procedure to read directory file information with the subcommand 0000

To obtain all the file information in the directory with the subcommand 0000, repeat the execution of this command for multiple times.

1. Specify one as the head file number, and execute this command.
2. Specify the value obtained by adding the number of requested files to the previously specified head file No. as the head file No., and execute this command.
3. Repeat step 2 until the number of file information becomes less than the number of requested files.

Point

If other file operation is executed while all the file information in the directory is being obtained, obtaining the information may fail. Do not execute other file operation other while the file information is being obtained.

■Procedure to read directory file information with the subcommand 0040

To obtain all the file information in the directory with the subcommand 0040, repeat the execution of this command for multiple times.

1. Specify one as the head file number, and execute this command.
2. Specify the value obtained by adding 1 to the last file No. of the response data as the head file No., and execute this command.
3. Repeat step 2 until the number of read information is -1 (0FFFFH).

Point

If other file operation is executed while all the file information in the directory is being obtained, obtaining the information may fail. Do not execute other file operation other while the file information is being obtained.

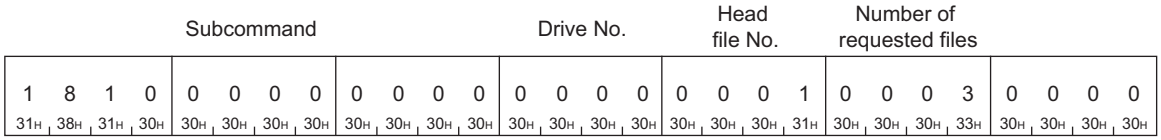
Communication example (when the subcommand is 0000)

Reads directory file information of the QCPU in the following conditions.

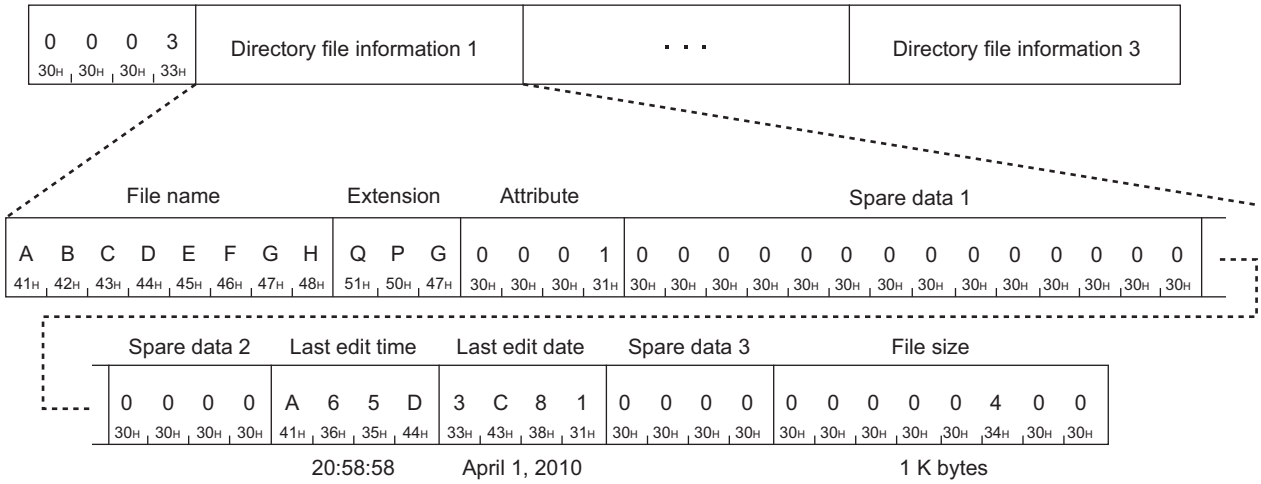
- Drive No.: 0
- Head file No.: 1
- Number of requested files: 3

■When communicating data in ASCII code

(Request data)

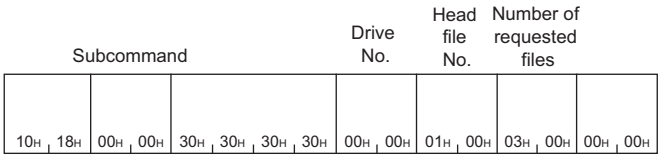


(Response data)

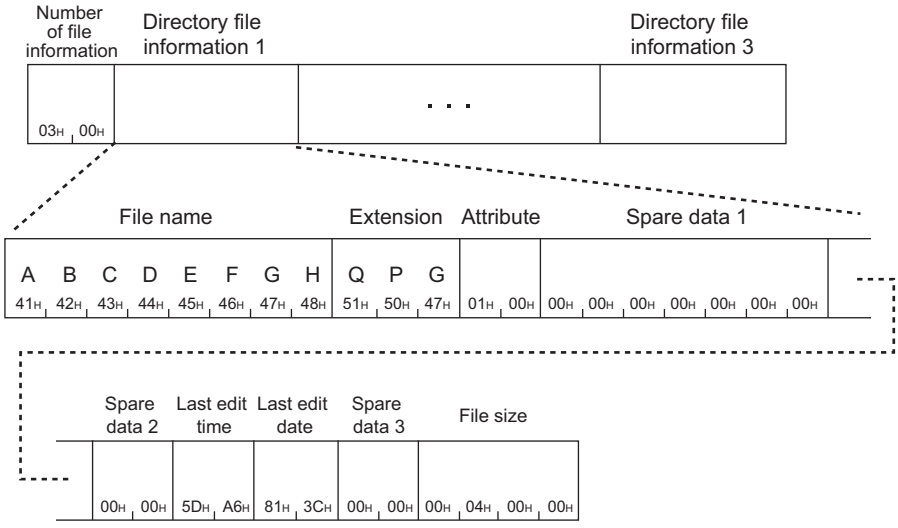


■When communicating data in binary code

(Request data)



(Response data)



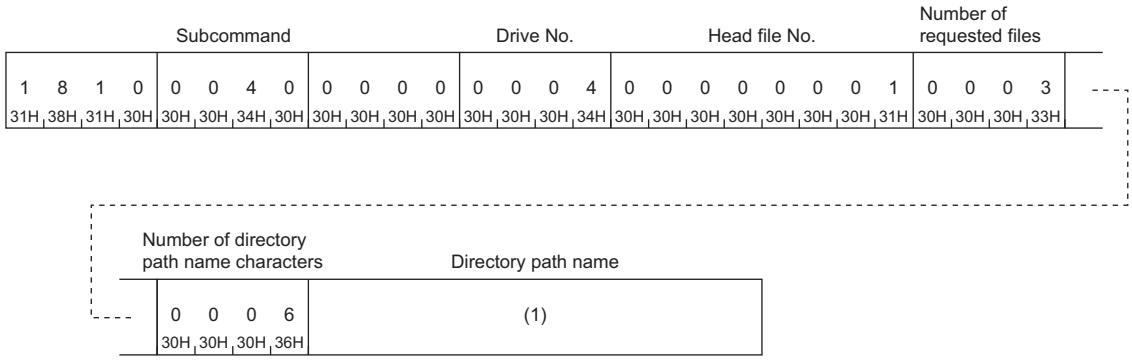
Communication example (when the subcommand is 0040)

Reads directory file information of the RCPU in the following conditions.

- Drive No.: 4
- Head file No.: 1
- Number of requested files: 3
- Directory path name: SUBDIR

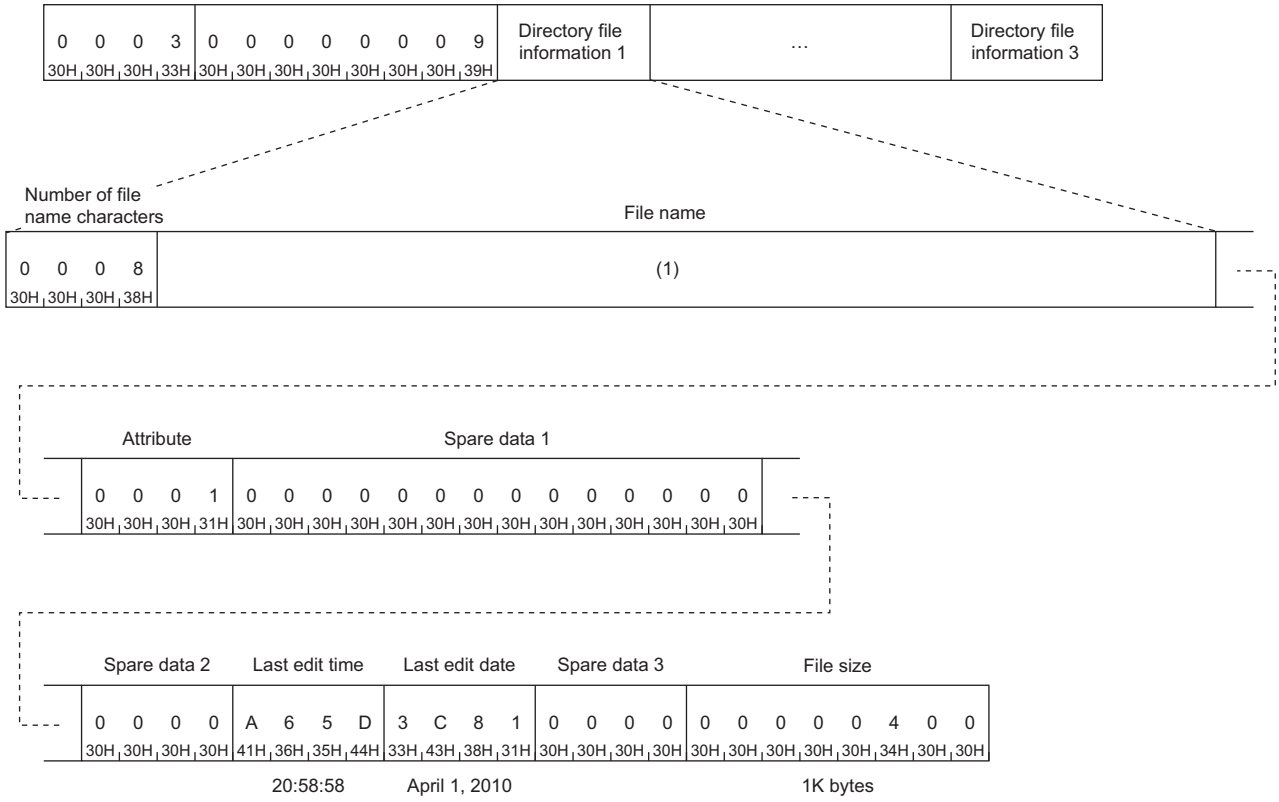
■When communicating data in ASCII code

(Request data)



No.	Item	Value
—	Directory path name	SUBDIR
	UTF-16 (hexadecimal)	00530055004200440049005200
(1)	ASCII code (hexadecimal)	3030353330303535303034323030343430303439303035323

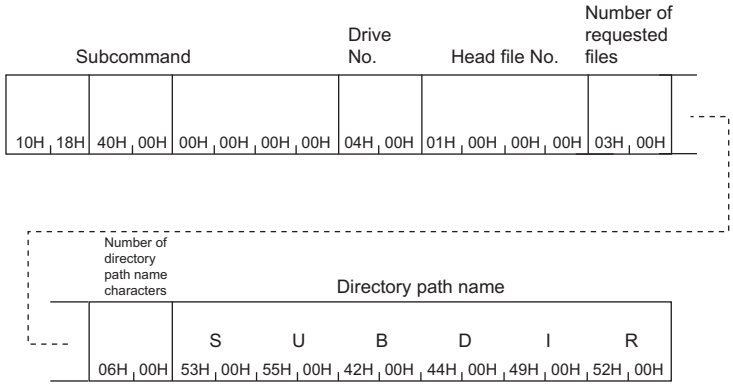
(Response data)



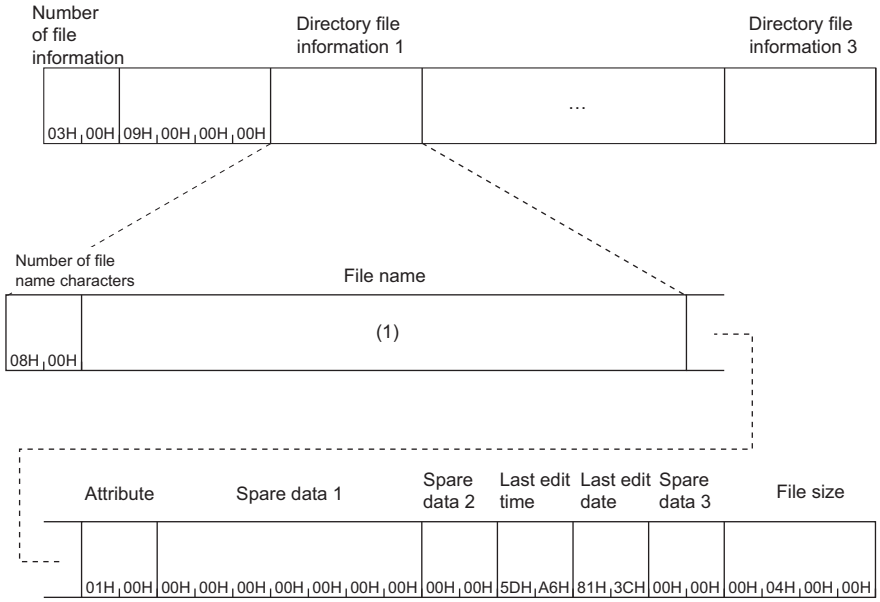
No.	Item	Value
—	File name	LINE.CSV
	UTF-16 (hexadecimal)	004C0049004E0045002E004300530056
(1)	ASCII code (hexadecimal)	3030344330303439303034453030343530303245303034333030353330303536

■When communicating data in binary code

(Request data)



(Response data)



No.	Item	Value
—	File name	LINE.CSV
	UTF-16 (hexadecimal)	004C0049004E0045002E004300530056
(1)	Binary code (hexadecimal)	4C0049004E0045002E00430053005600

Search Directory/File (command: 1811)

This command reads file No. of the specified file File No. is a registration number assigned when a file is written in the module.

Request data

ASCII

1 8 1 1 (31H 38H 31H 31H)	Subcommand	Password	Drive No.	0 0 0 0 (30H 30H 30H 30H)	Number of file name characters	File name

Binary

11H 18H	Sub command	Password	Drive No.	00H 00H	Number of file name characters	File name

■Subcommand

Subcommand	
ASCII code	**Binary code**
0 0 0 0 (30H 30H 30H 30H)	00H 00H
0 0 4 0 (30H 30H 34H 30H)	40H 00H

■Password

- Specify the file password when the access destination is the MELSEC-Q/L series module. (☞ Page 140 Password)
- The password is fixed to "0" when the access destination is the MELSEC iQ-R series module. Specify "30H" (0) when using ASCII code.

■Drive No.

Specify the drive where the file No. is read out. (☞ Page 141 Drive No.)

■Number of file name characters

Specify the number of file name characters set in "File name". (☞ Page 142 Number of file name characters, file name)

■File name

Specify the file name where the file No. is read out. Specify the file name with the extension. (☞ Page 142 Number of file name characters, file name)

Response data

File No. is stored.

When communicating data in ASCII code, the file No. is stored in order from the upper byte to the lower byte in an 8- or 4-digit ASCII code. (hexadecimal)

The number of digits of an ASCII code to be stored differs depending on the subcommand 0040 or 0000.

Subcommand	Number of digits	Example
0040	Eight-digit ASCII code	When the file No. is AH (eight digits) 0 0 0 0 0 0 0 A 30H 30H 30H 30H 30H 30H 30H 41H
0000	Four-digit ASCII code	When the file No. is AH (four digits) 0 0 0 A 30H 30H 30H 41H

When communicating data in binary code, the file No. is stored in numerical values (four or two bytes) in order from the lower byte to the upper byte.

The data size of the value to be stored differs depending on the subcommand 0040 or 0000.

Subcommand	Data size	Example
0040	Four bytes	When the file No. is AH (four bytes) 0AH 00H 00H 00H
0000	Two bytes	When the file No. is AH (two bytes) 0AH 00H

Communication example (when the subcommand is 0000)

The example is based on the following conditions with the QCPU.

- Password: 4 spaces (code: 20H)
- Drive No.: 0
- File name: ABC.QPG (file No. 6)

■When communicating data in ASCII code

(Request data)

	Subcommand	Password	Drive No.		Number of file name characters	File name
1 8 1 1	0 0 0 0		0 0 0 0	0 0 0 0	0 0 0 7	A B C . Q P G
31H 38H 31H 31H	30H 30H 30H 30H	20H 20H 20H 20H	30H 30H 30H 30H	30H 30H 30H 30H	30H 30H 30H 37H	41H 42H 43H 2EH 51H 50H 47H

(Response data)

File No.

0 0 0 6
30H 30H 30H 36H

■When communicating data in binary code

(Request data)

Subcommand	Password	Drive No.	Number of file name characters	File name
11H 18H	00H 00H	20H 20H 20H 20H	00H 00H	00H 00H	07H 00H	A B C . Q P G 41H 42H 43H 2EH 51H 50H 47H

(Response data)

File No.
06H 00H

Communication example (when the subcommand is 0040)

The example is based on the following conditions with the RCPU.

- Drive No.: 4
- File name: LINE.CSV (8 characters) (file No. 6)

■When communicating data in ASCII code

(Request data)

Subcommand	Drive No.	Number of file name characters	File name
1 8 1 1 31H 38H 31H 31H	0 0 4 0 30H 30H 34H 30H	0 0 0 0 30H 30H 30H 30H	0 0 0 4 30H 30H 30H 34H	0 0 0 0 30H 30H 30H 30H	0 0 0 8 30H 30H 30H 38H	(1)

No.	Item	Value
—	File name	LINE.CSV
	UTF-16 (hexadecimal)	004C0049004E0045002E004300530056
(1)	ASCII code (hexadecimal)	3030344330303439303034453030343530303245303034333030353330303536

(Response data)

File No.
0 0 0 0 0 0 0 6 30H 30H 30H 30H 30H 30H 30H 36H

■When communicating data in binary code

(Request data)

Subcommand	Drive No.	Number of file name characters	File name
11H 18H	40H 00H	00H 00H 00H 00H	04H 00H	00H 00H	08H 00H	(1)

No.	Item	Value
—	File name	LINE.CSV
	UTF-16 (hexadecimal)	004C0049004E0045002E004300530056
(1)	Binary code (hexadecimal)	4C0049004E0045002E00430053005600

(Response data)

File No.
06H 00H 00H 00H

New File (command: 1820)

This command specifies the file size, and creates a new file.

Request data

ASCII

1 8 2 0 31H 38H 32H 30H	Subcommand	Password	Drive No.	File size	Number of file name characters	File name

Binary

20H 18H	Sub command	Password	Drive No.	File size	Number of file name characters	File name

■Subcommand

Subcommand	
ASCII code	**Binary code**
0 0 0 0 30H 30H 30H 30H	00H 00H
0 0 4 0 30H 30H 34H 30H	40H 00H

■Password

- Specify the file password when the access destination is the MELSEC-Q/L series module. (☞ Page 140 Password)
- The password is fixed to "0" when the access destination is the MELSEC iQ-R series module. Specify "30H" (0) when using ASCII code.

■Drive No.

Specify the drive where a new file is created. (☞ Page 141 Drive No.)

■File size

Specify the file size in byte units.

Ex.

When the file size is 7168 bytes

ASCII code	Binary code
Specify the file size in a 8-digit ASCII code from the upper byte to the lower byte. (hexadecimal) 0 0 0 0 1 C 0 0 30H 30H 30H 30H 31H 43H 30H 30H	Specify the file size from the lower byte to the upper byte. (hexadecimal) 00H 1CH 00H 00H

■Number of file name characters

Specify the number of file name characters set in "File name". (☞ Page 142 Number of file name characters, file name)

■File name

Specify the name of a new file. (☞ Page 142 Number of file name characters, file name)

Precautions

For the new file creation, refer to the procedure described in the following page.

- ☞ Page 145 Procedure for creating a new file and writing data

When creating a new file by using this command, the last edit time is registered according to the time of the module.

The MELSEC iQ-R series module cannot access some file types through SLMP.

For the file types that can be accessed through SLMP, refer to the manual for the module used.

Response data

There is no response data for New File command.

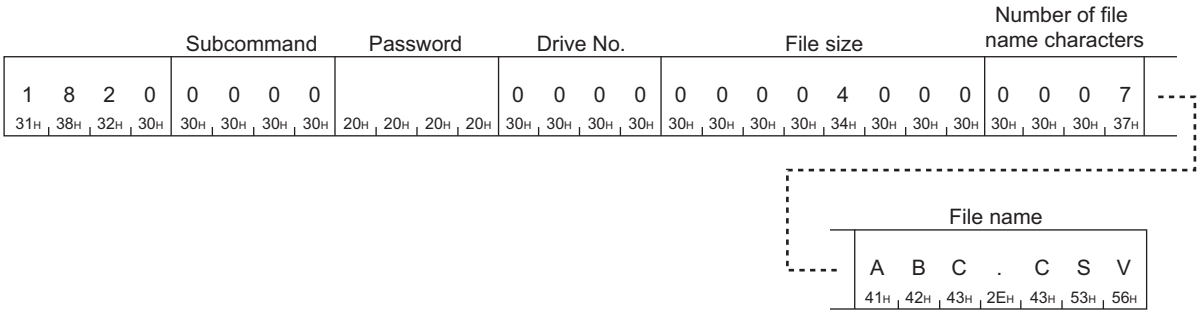
Communication example (when the subcommand is 0000)

Create a new file in the following conditions for the QCPU.

- Password: 4 spaces (code: 20H)
- Drive No.: 0
- File name: ABC.CSV
- File size: 1K bytes

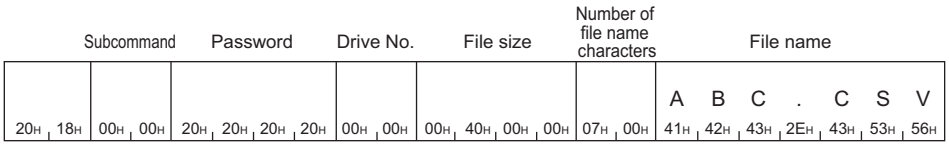
■When communicating data in ASCII code

(Request data)



■When communicating data in binary code

(Request data)



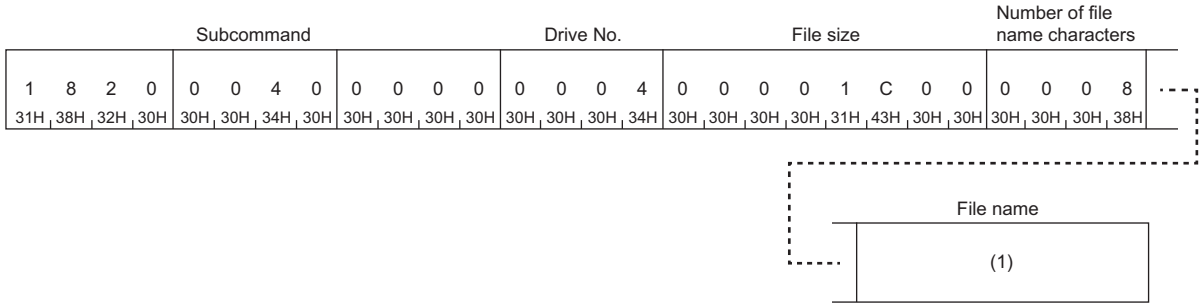
Communication example (when the subcommand is 0040)

Create a new file in the following conditions for the RCPU.

- Drive No.: 4
- File name: LINE.CSV (8 characters)
- File size: 7168 bytes

■When communicating data in ASCII code

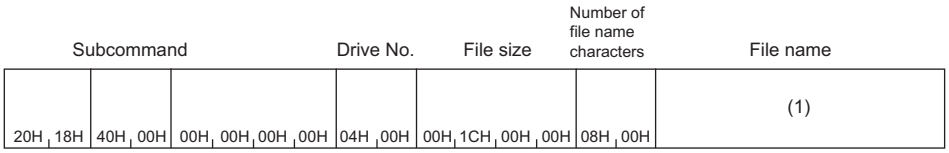
(Request data)



No.	Item	Value
—	File name	LINE.CSV
	UTF-16 (hexadecimal)	004C0049004E0045002E004300530056
(1)	ASCII code (hexadecimal)	3030344330303439303034453030343530303245303034333030353330303536

■When communicating data in binary code

(Request data)



No.	Item	Value
—	File name	LINE.CSV
	UTF-16 (hexadecimal)	004C0049004E0045002E004300530056
(1)	Binary code (hexadecimal)	4C0049004E0045002E00430053005600

Delete File (command: 1822)

This command deletes a file.

Request data

■When the subcommand is 0000 or 0004

ASCII

1 8 2 2 (31H 38H 32H 32H)	Subcommand	Password	Drive No.	Number of file name characters	File name

Binary

22H 18H	Sub command	Password	Drive No.	Number of file name characters	File name

■When the subcommand is 0040

ASCII

1 8 2 2 (31H 38H 32H 32H)	Subcommand	Number of password characters	Password	Drive No.	Number of file name characters	File name

Binary

22H 18H	Sub command	Number of password characters	Password	Drive No.	Number of file name characters	File name

■Subcommand

Subcommand	
ASCII code	**Binary code**
0 0 0 0 (30H 30H 30H 30H)	00H 00H
0 0 0 4 (30H 30H 30H 34H)	04H 00H
0 0 4 0 (30H 30H 34H 30H)	40H 00H

■Password

Specify the password for the access destination file. (☞ Page 140 Password)

■Drive No.

Specify the drive where the file is deleted. (☞ Page 141 Drive No.)

■Number of file name characters

Specify the number of file name characters set in "File name". (☞ Page 142 Number of file name characters, file name)

■File name

Specify the name of the file to be deleted. (☞ Page 142 Number of file name characters, file name)

Response data

There is no response data for Delete File command.

Precautions

- Deleting a file, while the programmable controller system is running, may stop the system. Determine the timing for deleting a file by considering a relationship with the whole programmable controller system.
- The file locked by Open File (command: 1827) cannot be deleted. Unlock the file by Close File (command: 182A), and then execute this command.
- When the MELSEC-Q/L series CPU module is in the RUN state, the program file, parameter file, and boot file cannot be deleted. Set the CPU module to the STOP state, and then delete the file.
- The MELSEC iQ-R series module cannot access some file types through SLMP. For the file types that can be accessed through SLMP, refer to the manual for the module used.

Communication example (when the subcommand is 0000)

Delete the file of the QCPU.

Information on the file to be deleted is as follows.

- Password: 1234
- Drive No.: 0
- File to delete: ABC.QPG

■When communicating data in ASCII code

(Request data)

	Subcommand	Password	Drive No.	Number of file name characters	File name
1 8 2 2	0 0 0 0	1 2 3 4	0 0 0 0	0 0 0 7	A B C . Q P G
31H 38H 32H 32H	30H 30H 30H 30H	31H 32H 33H 34H	30H 30H 30H 30H	30H 30H 30H 37H	41H 42H 43H 2EH 51H 50H 47H

■When communicating data in binary code

(Request data)

Subcommand		Password	Drive No.	Number of file name characters	File name
		1 2 3 4			A B C . Q P G
22H 18H	00H 00H	31H 32H 33H 34H	00H 00H	07H 00H	41H 42H 43H 2EH 51H 50H 47H

Communication example (when the subcommand is 0040)

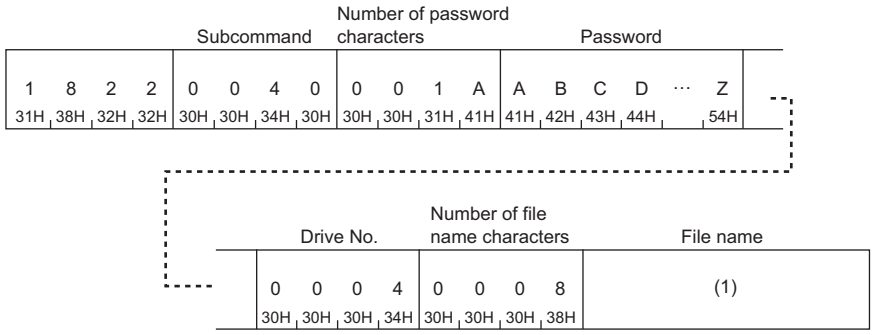
Delete the file of the RCPU.

Information on the file to be deleted is as follows.

- Password: A to Z (26 characters)
- Drive No.: 4
- File to delete: LINE.CSV (8 characters)

■When communicating data in ASCII code

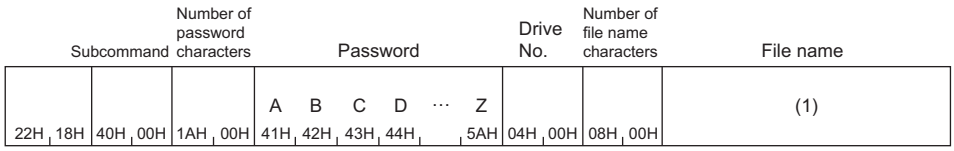
(Request data)



No.	Item	Value
—	File name	LINE.CSV
	UTF-16 (hexadecimal)	004C0049004E0045002E004300530056
(1)	ASCII code (hexadecimal)	3030344330303439303034453030343530303245303034333030353330303536

■When communicating data in binary code

(Request data)



No.	Item	Value
—	File name	LINE.CSV
	UTF-16 (hexadecimal)	004C0049004E0045002E004300530056
(1)	Binary code (hexadecimal)	4C0049004E0045002E00430053005600

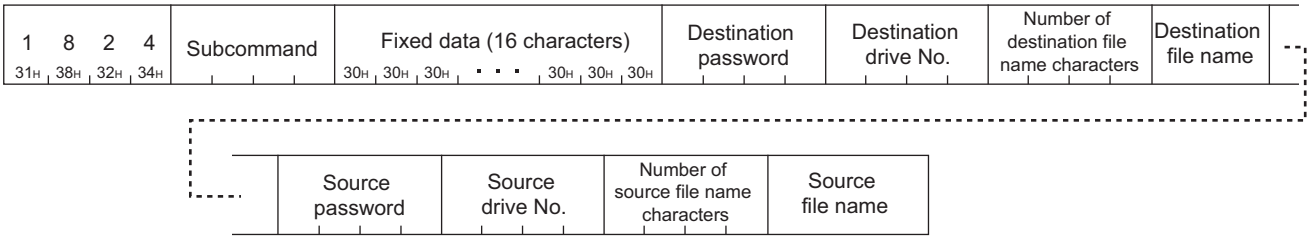
Copy File (command: 1824)

This command copies the specified file.

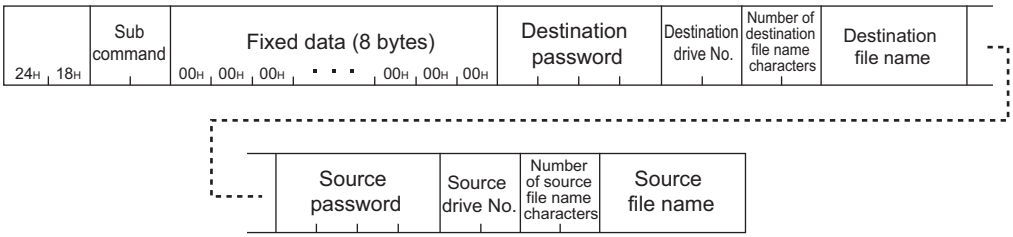
Request data

■When the subcommand is 0000 or 0004

ASCII

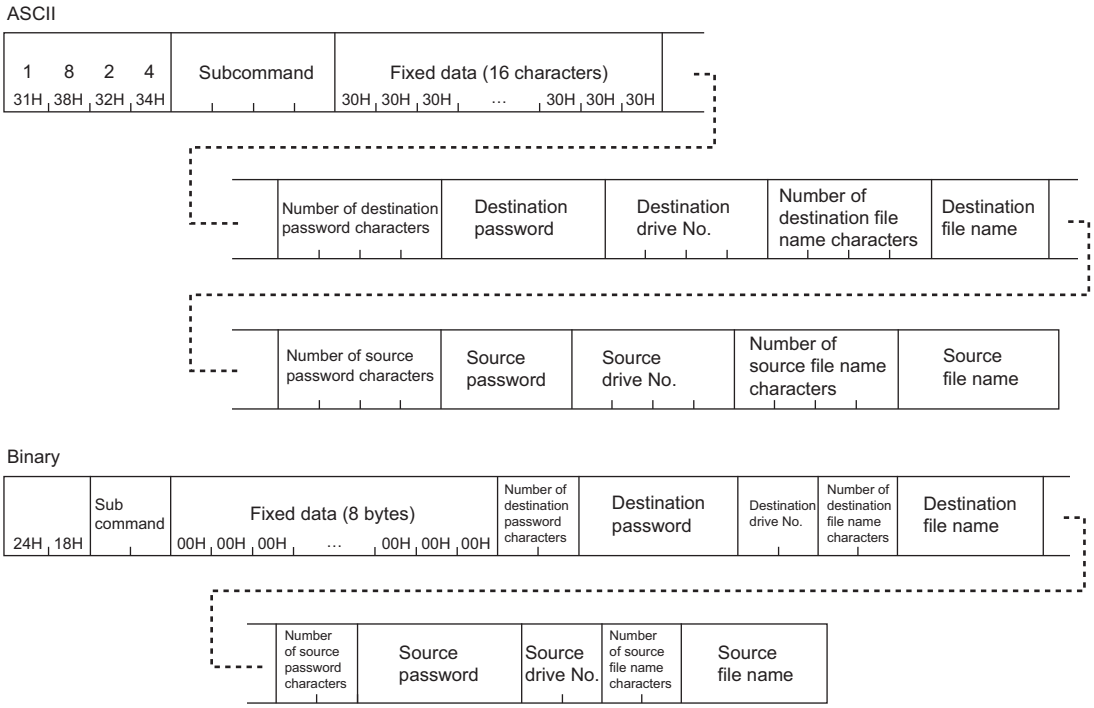


Binary



■When the subcommand is 0040

ASCII



Binary

■Subcommand

Subcommand	
ASCII code	Binary code
0 0 0 0 30H 30H 30H 30H	00H 00H
0 0 0 4 30H 30H 30H 34H	04H 00H
0 0 4 0 30H 30H 34H 30H	40H 00H

■Fixed data (16 characters)

Specify "0". Specify "30H" (0) when using ASCII code.

■Destination password, Source password

Specify the password for the access destination file. (☞ Page 140 Password)

■Destination drive No., source drive No.

Specify the copy destination drive and copy source drive. (☞ Page 141 Drive No.)

Point

When the RCPU is the copy source or copy destination, the drive No. 0 (program memory) cannot be specified.

■Number of destination file name characters, number of source file name characters

Specify the number of file name characters set in "File name". (☞ Page 142 Number of file name characters, file name)

■Destination file name, source file name

Specify the file name of the file to be copied. (☞ Page 142 Number of file name characters, file name)

Response data

There is no response data for Copy File command.

Precautions

Set the MELSEC-Q/L series CPU module to the STOP state to copy the following files. An error occurs when copying the files during RUN state.

- Parameter file
- Currently running files of program memory (drive No.: 0000H)

The MELSEC iQ-R series module cannot access some file types through SLMP. For the file types that can be accessed through SLMP, refer to the manual for the module used.

Communication example (when the subcommand is 0000)

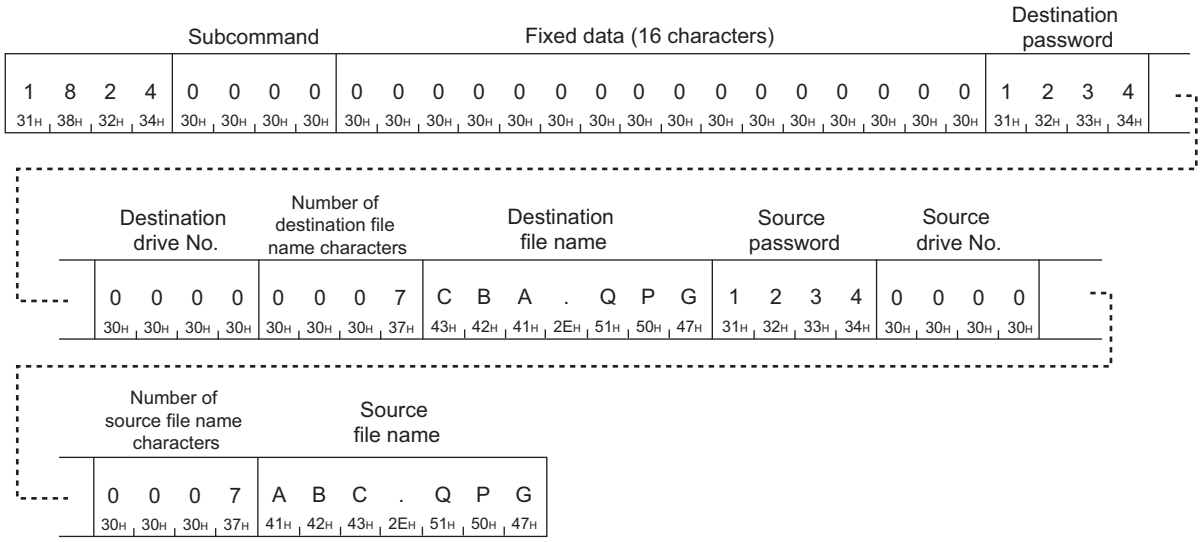
Copy the file of the QCPU.

This example is based on the following conditions.

- Destination password, source password: 1234
- Source drive No.: 0
- Destination drive No.: 1
- Source file name: ABC.QPG
- Destination file name: CBA.QPG

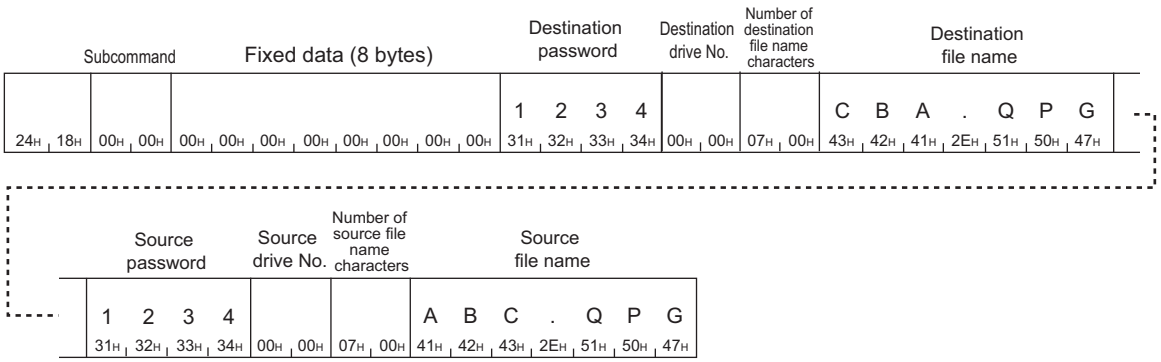
■When communicating data in ASCII code

(Request data)



■When communicating data in binary code

(Request data)



Communication example (when the subcommand is 0040)

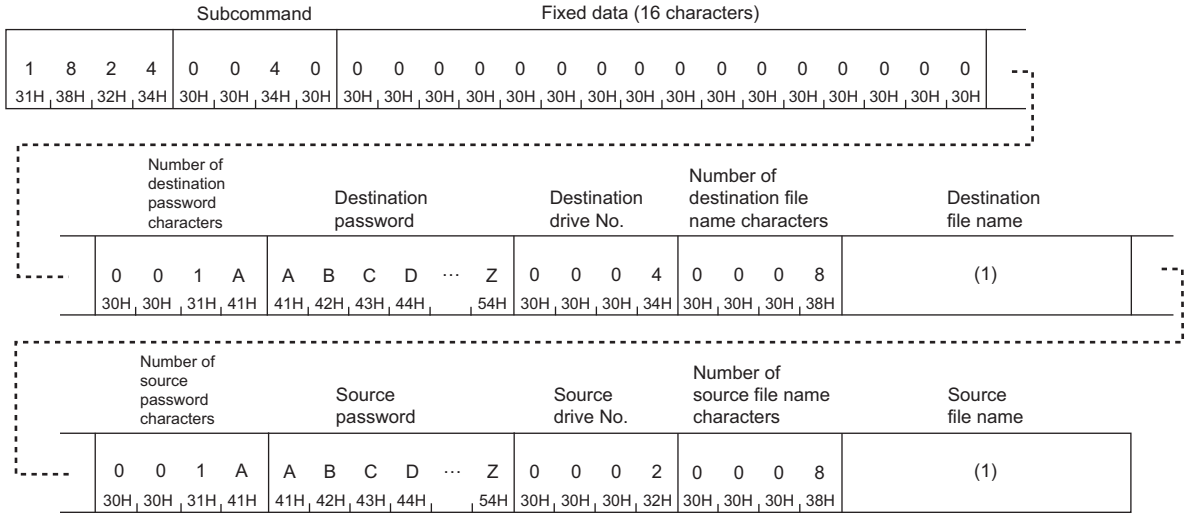
Copy the file of the RCPU.

This example is based on the following conditions.

- Destination password, source password: A to Z (26 characters)
- Source drive No.: 2
- Destination drive No.: 4
- Source file name: LINE.CSV (8 characters)
- Destination file name: LINE.CSV (8 characters)

■When communicating data in ASCII code

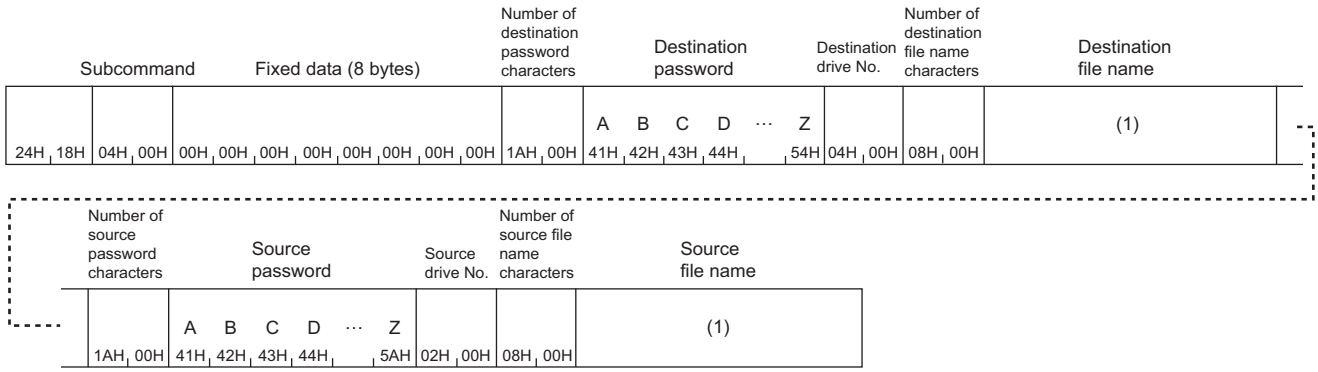
(Request data)



No.	Item	Value
—	File name	LINE.CSV
	UTF-16 (hexadecimal)	004C0049004E0045002E004300530056
(1)	ASCII code (hexadecimal)	3030344330303439303034453030343530303245303034333030353330303536

■When communicating data in binary code

(Request data)



No.	Item	Value
—	File name	LINE.CSV
	UTF-16 (hexadecimal)	004C0049004E0045002E004300530056
(1)	Binary code (hexadecimal)	4C0049004E0045002E00430053005600

Change File State (command: 1825)

This command changes file attributes.

Request data

■When the subcommand is 0000 or 0004

ASCII

1 8 2 5 (31H 38H 32H 35H)	Subcommand	Password	Drive No.	Attribute to change	Number of file name characters	File name

Binary

25H 18H	Sub command	Password	Drive No.	Attribute to change	Number of file name characters	File name

■When the subcommand is 0040

ASCII

1 8 2 5 (31H 38H 32H 35H)	Subcommand	Number of password characters	Password	Drive No.	Attribute to change	Number of file name characters	File name

Binary

25H 18H	Sub command	Number of password characters	Password	Drive No.	Attribute to change	Number of file name characters	File name

■Subcommand

Subcommand	
ASCII code	**Binary code**
0 0 0 0 (30H 30H 30H 30H)	00H 00H
0 0 0 4 (30H 30H 30H 34H)	04H 00H
0 0 4 0 (30H 30H 34H 30H)	40H 00H

■Password

Specify the password for the access destination file. (☞ Page 140 Password)

■Drive No.

Specify the drive of the file whose attributes are to be changed. (☞ Page 141 Drive No.)

■Attribute to change

Specify the file attributes.

- Read only: 01H
- Read, write enable: 20H

Point

No error may occur even with a value other than the above. However, do not use such values because they are for the system.

■Number of file name characters

Specify the number of file name characters set in "File name". (☞ Page 142 Number of file name characters, file name)

■File name

Specify the file name of the file whose attributes are to be changed. (☞ Page 142 Number of file name characters, file name)

Response data

There is no response data for Change File State command.

Precautions

Set the MELSEC-Q/L series CPU module to the STOP state to change the attribute of the following files. An error occurs when changing attributes during RUN state.

- Parameter file
- Currently running files of program memory (drive No.: 0000H)

The MELSEC iQ-R series module cannot access some file types through SLMP. For the file types that can be accessed through SLMP, refer to the manual for the module used.

Communication example (when the subcommand is 0000)

Change attributes of files stored in the QCPU.

This example is based on the following conditions.

- Password: 1234
- Drive No.: 0
- Target file of attribute change: ABC.QPG
- Attribute to change: Read only

■When communicating data in ASCII code

(Request data)

	Subcommand	Password	Drive No.	Attribute to change	Number of file name characters	File name
	1 8 2 5	0 0 0 0	1 2 3 4	0 0 0 0	0 0 0 1	0 0 0 7	A B C . Q P G
	31H 38H 32H 35H	30H 30H 30H 30H	31H 32H 33H 34H	30H 30H 30H 30H	30H 30H 30H 31H	30H 30H 30H 37H	41H 42H 43H 2EH 51H 50H 47H

■When communicating data in binary code

(Request data)

	Subcommand	Password	Drive No.	Attribute to change	Number of file name characters	File name
		1 2 3 4				A B C . Q P G
	25H 18H	00H 00H	31H 32H 33H 34H	00H 00H	01H 00H	07H 00H	41H 42H 43H 2EH 51H 50H 47H

Communication example (when the subcommand is 0040)

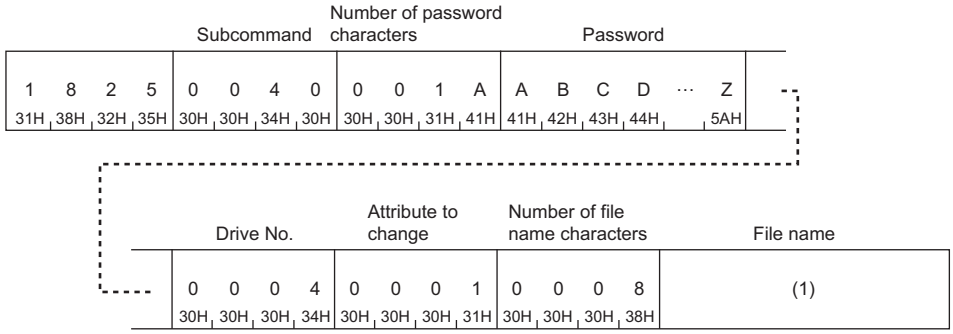
Change attributes of files stored in the RCPU.

This example is based on the following conditions.

- Password: A to Z (26 characters)
- Drive No.: 4
- Target file of attribute change: LINE.CSV (8 characters)
- Attribute to change: Read only

■When communicating data in ASCII code

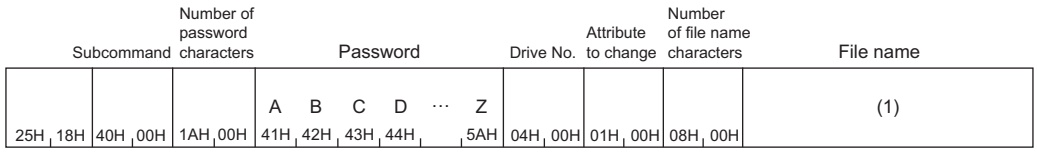
(Request data)



No.	Item	Value
—	File name	LINE.CSV
	UTF-16 (hexadecimal)	004C0049004E0045002E004300530056
(1)	ASCII code (hexadecimal)	3030344330303439303034453030343530303245303034333030353330303536

■When communicating data in binary code

(Request data)



No.	Item	Value
—	File name	LINE.CSV
	UTF-16 (hexadecimal)	004C0049004E0045002E004300530056
(1)	Binary code (hexadecimal)	4C0049004E0045002E00430053005600

Change File Date (command: 1826)

This command changes file creation date.

Request data

ASCII

1 8 2 6 (31H 38H 32H 36H)	Subcommand	0 0 0 0 (30H 30H 30H 30H)	Drive No.	Date to change	Time to change	Number of file name characters	File name

Binary

26H 18H	Sub command	00H 00H 00H 00H	Drive No.	Date to change	Time to change	Number of file name characters	File name

■Subcommand

Subcommand	
ASCII code	**Binary code**
0 0 0 0 (30H 30H 30H 30H)	00H 00H
0 0 4 0 (30H 30H 34H 30H)	40H 00H

■Drive No.

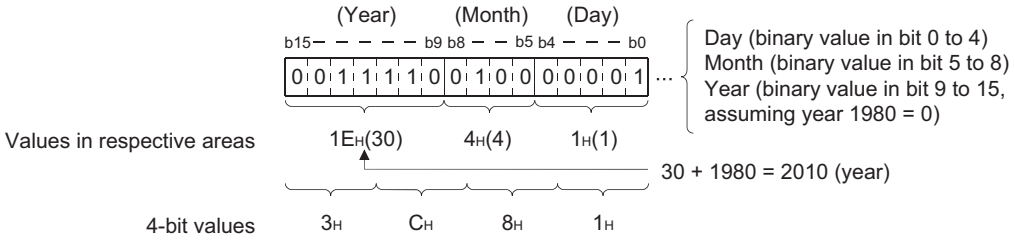
Specify the drive of the file whose creation date is to be changed. (☞ Page 141 Drive No.)

■Date to change

Specify a new date.

Ex.

When "date to change" is April 1, 2010

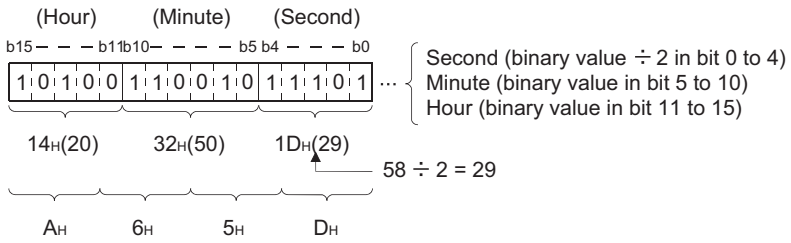


ASCII code communication : 3C81 (sending in order from "3")
Binary code communication : 3C81H (sending 81H and then 3CH)

■Time to change

Ex.

When "time to change" is 20:50:58



ASCII code communication : A65D (sending in order from "A")
Binary code communication : A65DH (sending 5DH and then A6H)

■Number of file name characters

Specify the number of file name characters set in "File name". (☞ Page 142 Number of file name characters, file name)

■File name

Specify the file name of the file whose date is to be changed. (☞ Page 142 Number of file name characters, file name)

Response data

There is no response data for Change File Date command.

Precautions

Set the MELSEC-Q/L series CPU module to the STOP state to change the date of the following files. An error occurs when changing the date during RUN state.

- Parameter file
- Currently running files of program memory (drive No.: 0000H)

The MELSEC iQ-R series module cannot access some file types through SLMP. For the file types that can be accessed through SLMP, refer to the manual for the module used.

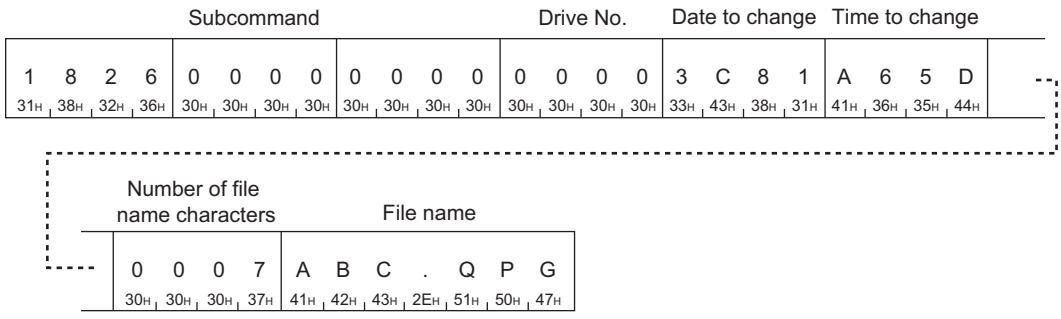
Communication example (when the subcommand is 0000)

Change the file creation date of the QCPU as follows.

- Drive No.: 0
- Date to change: April 1, 2010
- Time to change: 20:50:58
- File name: ABC.QPG

■When communicating data in ASCII code

(Request data)



■When communicating data in binary code

(Request data)

Subcommand			Drive No.	Date to change	Time to change	Number of file name characters	File name							
26H 18H	00H 00H	00H 00H 00H 00H	00H 00H	81H 3CH	5DH A6H	07H 00H	A 41H	B 42H	C 43H	. 2EH	Q 51H	P 50H	G 47H

Communication example (when the subcommand is 0040)

Change the file creation date of the RCPU as follows.

- Drive No.: 4
- Date to change: April 1, 2010
- Time to change: 20:50:58
- File name: LINE.CSV (8 characters)

■When communicating data in ASCII code

(Request data)

Subcommand				Drive No.	Date to change		Time to change			
1 31H	8 38H	2 32H	6 36H	0 30H	0 30H	4 34H	0 30H	0 30H	0 30H	0 30H	0 30H	0 30H	0 30H	4 34H	3 33H	C 43H	8 38H	1 31H	A 41H	6 36H	5 35H	D 44H

Number of file name characters				File name
0 30H	0 30H	0 30H	8 38H	(1)

No.	Item	Value
—	File name	LINE.CSV
	UTF-16 (hexadecimal)	004C0049004E0045002E004300530056
(1)	ASCII code (hexadecimal)	3030344330303439303034453030343530303245303034333030353330303536

■When communicating data in binary code

(Request data)

Subcommand			Drive No.	Date to change	Time to change	Number of file name characters	File name
26H 18H	40H 00H	00H 00H 00H 00H	04H 00H	81H 3CH	5DH A6H	08H 00H	(1)

No.	Item	Value
—	File name	LINE.CSV
	UTF-16 (hexadecimal)	004C0049004E0045002E004300530056
(1)	Binary code (hexadecimal)	4C0049004E0045002E00430053005600

Open File (command: 1827)

This command locks a file so that the content of file is not changed by other devices.

Point

The file can be unlocked by either of the followings.

- Execution of Close File (command: 182A) (☞ Page 187 Close File (command: 182A))
- Restart of the module (e.g. reset of CPU module)

Request data

■When the subcommand is 0000 or 0004

ASCII

1 8 2 7 (31H 38H 32H 37H)	Subcommand	Password	Open mode	Drive No.	Number of file name characters	File name

Binary

27H 18H	Sub command	Password	Open mode	Drive No.	Number of file name characters	File name

■When the subcommand is 0040

ASCII

1 8 2 7 (31H 38H 32H 37H)	Subcommand	Number of password characters	Password	Open mode	Drive No.	Number of file name characters	File name

Binary

27H 18H	Sub command	Number of password characters	Password	Open mode	Drive No.	Number of file name characters	File name

■Subcommand

Subcommand	
ASCII code	**Binary code**
0 0 0 0 (30H 30H 30H 30H)	00H 00H
0 0 0 4 (30H 30H 30H 34H)	04H 00H
0 0 4 0 (30H 30H 34H 30H)	40H 00H

■Password

Specify the password for the access destination file. (☞ Page 140 Password)

■Open mode

Specify whether to lock the specified file for reading or writing.

Item	Open mode	
	ASCII code	Binary code
Lock the file for data reading	0 0 0 0 30H 30H 30H 30H	00H 00H
Lock the file for data writing	0 1 0 0 30H 31H 30H 30H	00H 01H

■Drive No.

Specify the drive in which the file is to be locked. (☞ Page 141 Drive No.)

■Number of file name characters

Specify the number of file name characters set in "File name". (☞ Page 142 Number of file name characters, file name)

■File name

Specify the name of the file to be locked. (☞ Page 142 Number of file name characters, file name)

Response data

The file pointer No. is stored. (☞ Page 143 File pointer No.)

Communication example (when the subcommand is 0000)

Lock the file of the QCPU.

This example is based on the following conditions.

- Password: 1234
- Drive No.: 0
- File name: ABC.QPG
- Open mode: Write open

■When communicating data in ASCII code

(Request data)

	Subcommand	Password	Open mode	Drive No.	Number of file name characters	File name
1 8 2 7	0 0 0 0	1 2 3 4	0 1 0 0	0 0 0 0	0 0 0 7	A B C . Q P G
31H 38H 32H 37H	30H 30H 30H 30H	31H 32H 33H 34H	30H 31H 30H 30H	30H 30H 30H 30H	30H 30H 30H 37H	41H 42H 43H 2EH 51H 50H 47H

(Response data)

File pointer No.
0 0 0 0
30H 30H 30H 30H

■When communicating data in binary code

(Request data)

Subcommand		Password	Open mode	Drive No.	Number of file name characters	File name
		1 2 3 4				A B C . Q P G
27H 18H	00H 00H	31H 32H 33H 34H	00H 01H	00H 00H	07H 00H	41H 42H 43H 2EH 51H 50H 47H

(Response data)

File pointer No.
00H 00H

Communication example (when the subcommand is 0040)

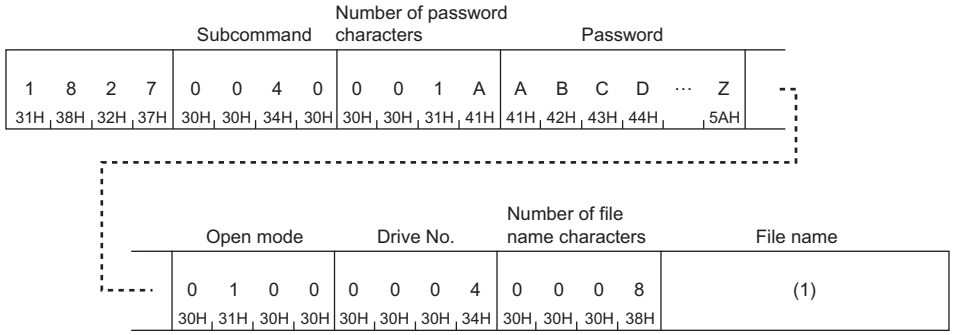
Lock the file of the RCPU.

This example is based on the following conditions.

- Password: A to Z (26 characters)
- Drive No.: 4
- File name: LINE.CSV (8 characters)
- Open mode: Write open

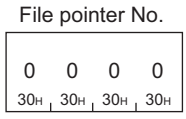
■When communicating data in ASCII code

(Request data)



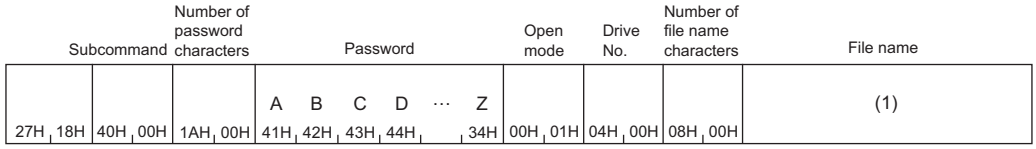
No.	Item	Value
—	File name	LINE.CSV
	UTF-16 (hexadecimal)	004C0049004E0045002E004300530056
(1)	ASCII code (hexadecimal)	3030344330303439303034453030343530303245303034333030353330303536

(Response data)



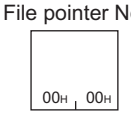
■When communicating data in binary code

(Request data)



No.	Item	Value
—	File name	LINE.CSV
	UTF-16 (hexadecimal)	004C0049004E0045002E004300530056
(1)	Binary code (hexadecimal)	4C0049004E0045002E00430053005600

(Response data)

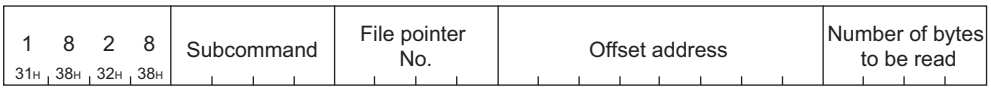


Read File (command: 1828)

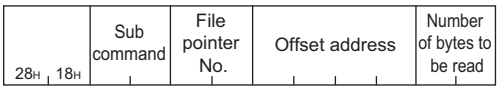
This command reads the contents of a file.

Request data

ASCII



Binary



■Subcommand

Subcommand	
ASCII code	**Binary code**
0 0 0 0 30H 30H 30H 30H	00H 00H

■File pointer No.

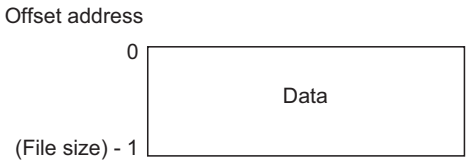
Specify the file pointer No. (☞ Page 143 File pointer No.)

■Offset address

Specify the start position for file read. The offset address is used when a file is separately read out.

When reading a file at once, specify "0" and set the file size in the number of bytes to be read.

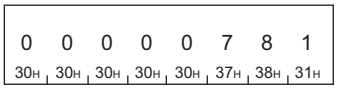
For the offset address, specify an even number that indicates the offset (1 address/1 byte) from the head of the file (offset address: 0H).



When communicating data in ASCII code, specify the offset address in an 8-digit ASCII code in order from the upper byte to the lower byte. (hexadecimal)

Ex.

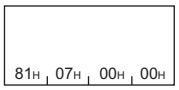
When the offset address is 781H (1921)



When communicating in binary code, specify the offset address in order from the lower byte to the upper byte. (hexadecimal)

Ex.

When the offset address is 781H (1921)



Point

When the file size is 1921 bytes or more, use the offset address and read the file in multiple times. The file size can be checked in the following commands.

- Read Directory/File (command: 1810) (☞ Page 149 Read Directory/File (command: 1810))
- Search Directory/File (command: 1811) (☞ Page 159 Search Directory/File (command: 1811))

Leave the read data in the external device as it is stored. The read data cannot be edited from the external device side.

■Number of bytes to be read

Specify the size (number of bytes) of the file to be read. The size is specified as 1 address/1 byte. (Specification range: 0 to 1920)

Ex.

When the number of bytes to be read is 780H (1920)

ASCII code	Binary code
Specify the file size in a 4-digit ASCII code from the upper byte to the lower byte. (hexadecimal) 0 7 8 0 30H 37H 38H 30H	Specify the file size from the lower byte to the upper byte. (hexadecimal) 80H 07H

Response data

The number of bytes to be read and the read data are stored.

ASCII



Binary



■Number of bytes to be read

The number of bytes of the read file is stored in the same format as that of "the number of bytes to be read" of the request data.

■Read data

The contents of the read file are stored.

Precautions

The MELSEC iQ-R series module cannot access some file types through SLMP. For the file types that can be accessed through SLMP, refer to the manual for the module used.

Communication example

This example explains how to read the following file.

- File pointer No.: 0
- Number of bytes to be read: 1K bytes

■When communicating data in ASCII code

(Request data)

	Subcommand	File pointer No.	Offset address	Number of bytes to be read
1 8 2 8	0 0 0 0	0 0 0 0	0 0 0 0 0 0 0 0	0 4 0 0
31H 38H 32H 38H	30H 30H 30H 30H	30H 30H 30H 30H	30H 30H 30H 30H 30H 30H 30H 30H	30H 34H 30H 30H

(Response data)

0 4 0 0	Read data
30H 34H 30H 30H	

■When communicating data in binary code

(Request data)

Subcommand	File pointer No.	Offset address	Number of bytes to be read
28H 18H	00H 00H	00H 00H	00H 00H 00H 00H	00H 04H

(Response data)

00H 04H	Read data

Write File (command: 1829)

This command writes the contents in a file.

Request data

ASCII

1 8 2 9 (31H 38H 32H 39H)	Subcommand	File pointer No.	Offset address	Number of bytes to be written	Write data

Binary

29H 18H	Sub command	File pointer No.	Offset address	Number of bytes to be written	Write data

■Subcommand

Subcommand	
ASCII code	**Binary code**
0 0 0 0 (30H 30H 30H 30H)	00H 00H

■File pointer No.

Specify the file pointer No. (☞ Page 143 File pointer No.)

■Offset address

Specify the start position for file write. The offset address is used when a file is separately written. Specify "0" when writing a file at once.

For the offset address, specify an even number or multiple of four that indicates the offset (1 address/1 byte) from the head of the file (offset address: 0H).

- When writing to drive No. 0000 (program memory, parameter memory): Specify a multiple of four.
- When writing to drive numbers other than 0000: Specify an even number.



When communicating data in ASCII code, specify the offset address in an 8-digit ASCII code in order from the upper byte to the lower byte. (hexadecimal)

Ex.

When the offset address is 781H (1921)

0 0 0 0 0 7 8 1 (30H 30H 30H 30H 30H 37H 38H 31H)

When communicating in binary code, specify the offset address in order from the lower byte to the upper byte. (hexadecimal)

Ex.

When the offset address is 781H (1921)

81H 07H 00H 00H

Point

When the file size is 1921 bytes or more, use the offset address and write to the file in multiple times. The file size can be checked in the following commands.

- Read Directory/File (command: 1810) (☞ Page 149 Read Directory/File (command: 1810))
- Search Directory/File (command: 1811) (☞ Page 159 Search Directory/File (command: 1811))

Set the CPU module to the STOP state to write to the following files. An error occurs when writing to the files during RUN state.

- Parameter file
- Currently running files of program memory (drive No.: 0000H)

■Number of bytes to be written

Specify the size (number of bytes) of the file to be written in. The size is specified as 1 address/1 byte.
(specification range: 0 to 1920 or 0 to the file size specified in New File (command: 1820))

Ex.

When the number of bytes to be written is 780H (1920)

ASCII code	Binary code
Specify the file size in a 4-digit ASCII code from the upper byte to the lower byte. (hexadecimal) 0 7 8 0 30H 37H 38H 30H	Specify the file size from the lower byte to the upper byte. (hexadecimal) 80H 07H

■Write data

Specify the data read by Read File (command: 1828).

Response data

The number of bytes of the written file is stored in the same format as that of "the number of bytes to be written" of the request data.

Precautions

The MELSEC iQ-R series module cannot access some file types through SLMP. For the file types that can be accessed through SLMP, refer to the manual for the module used.

Communication example

This example explains how to write to the following files.

- File pointer No.: 0
- Offset address: 0
- Number of bytes to be written: 1K bytes

■When communicating data in ASCII code

(Request data)

	Subcommand	File pointer No.	Offset address	Number of bytes to be written	Write data
1 8 2 9	0 0 0 0	0 0 0 0	0 0 0 0 0 0 0 0	0 4 0 0	
31H 38H 32H 39H	30H 30H 30H 30H	30H 30H 30H 30H	30H 30H 30H 30H 30H 30H 30H 30H	30H 34H 30H 30H	

(Response data)

Number of bytes to be written

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

■When communicating data in binary code

(Request data)

Subcommand	File pointer No.	Offset address	Number of bytes to be written	Write data
29H 18H	00H 00H	00H 00H	00H 00H 00H 00H	00H 04H	

(Response data)

Number of bytes to be written

00H 04H

Close File (command: 182A)

This command cancels the file lock by open processing.

Request data

ASCII

1	8	2	A	Subcommand	File pointer No.	Close type
31H	38H	32H	41H			

Binary

	Sub command	File pointer No.	Close type
2AH 18H			

■Subcommand

Subcommand	
ASCII code	**Binary code**
0 0 0 0 30H 30H 30H 30H	00H 00H

■File pointer No.

Specify the file pointer No. (☞ Page 143 File pointer No.)

■Close type

Select whether to unlock only the target file or unlock all the locked files.

Unlocking target	Close type	
	ASCII code	**Binary code**
Only the files locked by the external device that executes the command*1	0 0 0 0 30H 30H 30H 30H	00H 00H
All the files locked by the external device that executes the command*2	0 0 0 1 30H 30H 30H 31H	01H 00H
	0 0 0 2 30H 30H 30H 32H	02H 00H

*1 If the command is executed to a file locked by other external devices, the command gets rejected and ends as an error.

*2 Use when the external device that locked a file cannot unlock it due to an external device error and others.

Point

Restart of the module (reset of CPU module, etc.) also unlocks the files.

Response data

There is no response data for Close File command.

Communication example

The example is based on the following conditions.

- File pointer No.: 0
- Close type: 2 (All locked files)

■When communicating data in ASCII code

(Request data)

	Subcommand	File pointer No.	Close type
1 8 2 A	0 0 0 0	0 0 0 0	0 0 0 2
31H 38H 32H 41H	30H 30H 30H 30H	30H 30H 30H 30H	30H 30H 30H 32H

■When communicating data in binary code

(Request data)

	Subcommand	File pointer No.	Close type
2AH 18H	00H 00H	00H 00H	02H 00H

5.9 Self Test (Loopback Test) (Command: 0619)

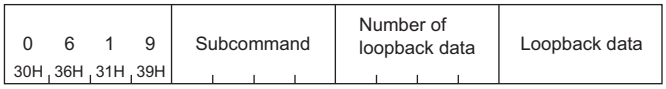
This command tests whether the communication between the external device and Ethernet-equipped module is normally executed or not. By conducting the loopback test, the connection and data communication with an external device are checked.

Point

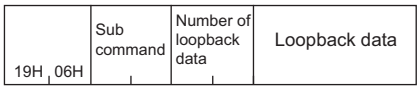
The loopback test can be conducted only for the Ethernet-equipped module connected to an external device. The loopback test cannot be conducted for the modules of other stations via a network.

Request data

ASCII



Binary



■Subcommand

Subcommand	
ASCII code	**Binary code**
0 0 0 0 30H 30H 30H 30H	00H 00H

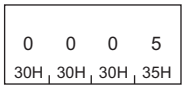
■Number of loopback data

Specify the number of data of "Loopback data" in the number of bytes. The specification range is 1 to 960.

Ex.

When the number of loopback data is five bytes

When using the ASCII code, convert the number of bytes to a 4-digit ASCII code (hexadecimal), and send it in order from the upper byte to the lower byte.



When using the binary code, specify the numerical values in 2 bytes that describe the number of bytes in order from the lower byte to the upper byte.



■Loopback data

Specify the data to be sent/received in the loopback test.

When communicating data in ASCII code, specify a 1-byte character string, "0" to "9" and "A" to "F", as the loopback data, and send it from its head character. The maximum number of characters is 960.

When communicating data in binary code, convert the code to a 1-byte numerical value, "0" to "9" or "A" to "F", and send it from the head character code. The maximum capacity is 960 bytes.

Response data

The same data as those specified in "Number of loopback data" and "Loopback data" in the request message is stored.

ASCII

Number of loopback data	Loopback data

Binary

Number of loopback data	Loopback data

Communication example

Conduct the loopback test with the loopback data "ABCDE".

■When communicating data in ASCII code

(Request data)

	Subcommand	Number of loopback data	Loopback data
0 6 1 9	0 0 0 0	0 0 0 5	A B C D E
30H 36H 31H 39H	30H 30H 30H 30H	30H 30H 30H 35H	41H 42H 43H 44H 45H

(Response data)

Number of loopback data	Loopback data
0 0 0 5	A B C D E
30H 30H 30H 35H	41H 42H 43H 44H 45H

■When communicating data in binary code

(Request data)

Subcommand		Number of loopback data	Loopback data
			A B C D E
19H 06H	00H 00H	05H 00H	41H 42H 43H 44H 45H

(Response data)

Number of loopback data	Loopback data
	A B C D E
05H 00H	41H 42H 43H 44H 45H

5.10 Clear Error (Error Code Initialization, LED Off) (Command: 1617)

This command turns off COM. ERR. LED of the own station.

Request data

ASCII

1	6	1	7	0	0	0	0
31H	36H	31H	37H	30H	30H	30H	30H

Binary

17H	16H	00H	00H

Response data

There is no response data for Clear Error command.

Communication example

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

5.11 Ondemand (Command: 2101)

This command outputs Send request to the SLMP compatible device from the CPU module and sends data to the external device.

Data from the SLMP compatible device

ASCII

2	1	0	1	0	0	0	0	Send data
32H	31H	30H	31H	30H	30H	30H	30H	

Binary code

		Send data
01H 21H	00H 00H	

■Send data

Data sent from the SLMP compatible device is stored (up to 1920 bytes (up to 960 words)).

Communication example

Data is received from the SLMP compatible device using the message format shown in "Data from the SLMP compatible device" above.

For how to send data from the SLMP compatible device, refer to the manual for the SLMP compatible device used.

6 TROUBLESHOOTING

When an external device cannot communicate with a SLMP compatible device, read this chapter to specify the cause on the external device side and to take corrective actions.
For the troubleshooting on the SLMP compatible device side, refer to the SLMP compatible device manual.

Check item	Corrective action
Is a connection with the SLMP compatible device established when using TCP/IP?	Issue a connection request from the external device to the SLMP compatible device. (Active open)
Is a request message sent from the external device?	• Send a request message from the external device to the SLMP compatible device. • Check if the destination of the request message is a SLMP compatible device. (☞ Page 9 SLMP Compatible Device)
Is the network load high?	• Decrease the frequency of request message transmission from the external device. • Reduce the network load.
Is the IP address correct?	• Match the network part of the external device IP address with that of the SLMP compatible device IP address. • Do not set the same IP address as other Ethernet devices. • Set the destination IP address of the request message to the IP address of the SLMP compatible device.
Is a correct protocol (TCP/IP or UDP/IP) used?	Match the external device protocol with the protocol set for the SLMP compatible device.
Is the port No. correct?	Match the destination port No. of the request message with the own station port No. set on the SLMP compatible device.
Is the communication data code (ASCII or binary code) the same for both the external device and the SLMP compatible device?	Match the communication data code of the request message (ASCII code or binary code) with the communication data code set on the SLMP compatible.
Is the request message format correct?	Send the request message in the message format described in this manual. (☞ Page 16 Request Message)
Is the storing order and the value range of the specified data within the request message correct?	Set the specified data within the request message in the storing order and the value range described in this manual. ☞ Page 16 MESSAGE FORMAT ☞ Page 24 COMMANDS
Is the "end code" of the response message 0?	When the "end code" is not 0, there is an error on the SLMP compatible device. Check the meaning of the end code in the manual of the SLMP compatible device used, and take a corrective action.
When using TCP/IP, is the length of the response message that the external device actually received the same as the one expected?	• If the response message is shorter than expected, take action to receive the remaining data. • If the response message is longer than expected, check the corresponding request message. • Reset the SLMP compatible device.
Is the firewall set?	Check the firewall settings.
When the label access is used, is "Access from External Device" enabled with the global label setting editor in GX Works3?	Enable "Access from External Device" with the global label setting editor in GX Works3.

MEMO

APPENDICES

Appendix 1 Read or Write by Device Extension Specification

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

- Access to the link direct device
- Access to the module access device
- Access to the CPU buffer memory access device
- Access with indirect specification of the network No. and start I/O number by using the index register
- Access with indirect specification of the device No. by using the index register or long index register
- Access with indirect specification of the device No. by using the values stored in the word device

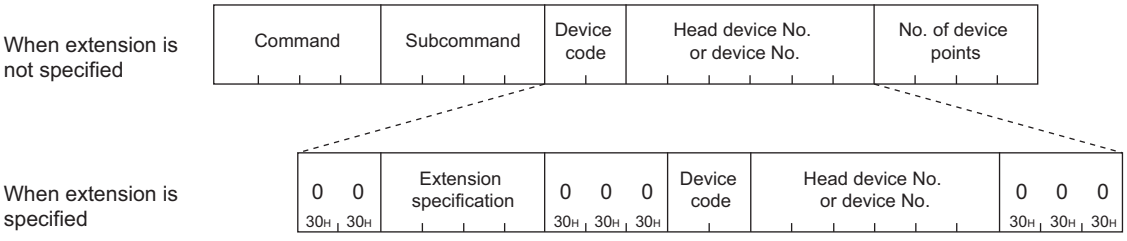
Access to the link direct device

Link devices of the network module, such as remote input (RX), remote output (RY) and link special relay (SB) can be accessed.

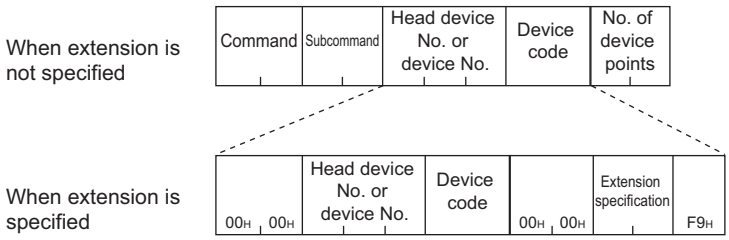
Request data

■When the subcommand is 0081 or 0080

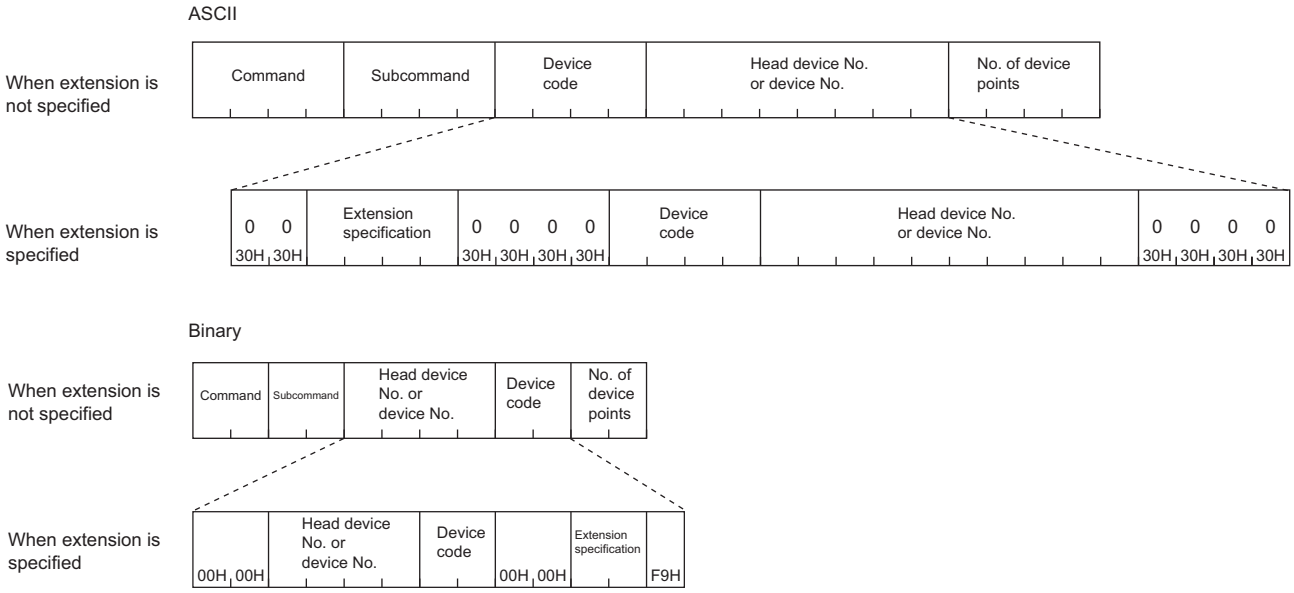
ASCII



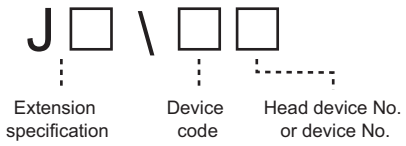
Binary



■When the subcommand is 0083 or 0082



The following shows the approach for link direct device and request data.



Point

Devices described in the following page can be accessed by specifying 0 in "extension specification" of commands which can specify multiple devices.

- ☞ Page 30 Device code

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
	Entry Monitor Device	0801
	Read Block	0406
	Write Block	1406

■Subcommand

Item	Subcommand	
	ASCII code	Binary code
When accessing bit units	0 0 8 1 30H 30H 38H 31H	81H 00H
	0 0 8 3 30H 30H 38H 33H	83H 00H
When accessing 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 network No. corresponding to the access.

ASCII code	Binary code
Specify the network No. in hexadecimal (3-digit ASCII code). J □ □ □ 4AH Example Network No.8 J 0 0 8 4AH 30H 30H 38H	Specify the network No. in hexadecimal (2 bytes). □□H □□H Example Network No.8 08H 00H

Point

Indirect specification of the access target network No. can also be performed by using the CPU module index register. (☞ Page 205 Access with indirect specification of the network No. and start I/O number by using the index register)

■Device code

Specify the following device codes.

Device	Type	Device code				Device No. range	
		ASCII code		Binary code			
		MELSEC iQ-R series*1	MELSEC-Q/L series*2	MELSEC iQ-R series	MELSEC-Q/L series		
Link input (X)	Bit	X***	X*	009CH	9CH	Specify within the device No. range of the access destination module.	Hexadecimal
Link output (Y)		Y***	Y*	009DH	9DH		Hexadecimal
Link relay (B)		B***	B*	00A0H	A0H		Hexadecimal
Link special relay (SB		SB**	SB	00A1H	A1H		Hexadecimal
Link register (W)	Word	W***	W*	00B4H	B4H		Hexadecimal
Link special register (SW)		SW**	SW	00B5H	B5H		Hexadecimal

1 For ASCII codes, the device code is specified with 4 digits. If the device code has three digits or less, add "" (ASCII code: 2AH) or a space (ASCII code: 20H) after the device code.

2 For ASCII codes, the device code is specified with 2 digits. If the device code has one digit, add "" (ASCII code: 2AH) or a space (ASCII code: 20H) after the device code.

■Head device or device No.

Specify the head device or device No. in hexadecimal. (☞ Page 33 Head device No. (Device No.))

Point

Indirect specification of the access target device No. can be performed by using the CPU module index register or long index register. (☞ Page 210 Access with indirect specification of the device No. by using the index register or long index register)

Response data

The same as when extension is not specified.

Communication example

Access to W100(J1\W100) of network No.1.

■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	J 0 0 1	0 0 0	W *	0 0 0 1 0 0	0 0 0
30H 30H 38H 30H	30H 30H	4AH 30H 30H 31H	30H 30H 30H	57H 2AH	30H 30H 30H 31H 30H 30H	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	00H 01H 00H	B4H	00H 00H	01H 00H	F9H

Access to the module access device

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

Request data

■When the subcommand is 0080

ASCII

When extension is not specified

Command	Subcommand	Device code	Head device No. or device No.	No. of device points

When extension is specified

0 0 (30H 30H)	Extension specification	0 0 0 (30H 30H 30H)	Device code	Head device No. or device No.	0 0 0 (30H 30H 30H)

Binary code

When extension is not specified

Command	Subcommand	Head device No. or device No.	Device code	No. of device points

When extension is specified

00H 00H	Head device No. or device No.	Device code	00H 00H	Extension specification	F8H

■When the subcommand is 0082

ASCII

When extension is not specified

Command	Subcommand	Device code	Head device No. or device No.	No. of device points

When extension is specified

0 0 (30H 30H)	Extension specification	0 0 0 0 (30H 30H 30H 30H)	Device code	Head device No. or device No.	0 0 0 0 (30H 30H 30H 30H)

Binary code

When extension is not specified

Command	Subcommand	Head device No. or device No.	Device code	No. of device points

When extension is specified

00H 00H	Head device No. or device No.	Device code	00H 00H	Extension specification	F8H

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

U□ \ G□

- U□: Extension specification
- G□: Device code
- Head device No. or device No.

Point

Devices described in the following page can be accessed by specifying 0 in "extension specification" of commands which can specify multiple devices.

• ☞ Page 30 Device code

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
	Entry Monitor Device	0801
	Read Block	0406
	Write Block	1406

■Subcommand

ASCII code	Binary code
0 0 8 0 30H 30H 38H 30H	80H 00H
0 0 8 2 30H 30H 38H 32H	82H 00H

■Extension specification

Specify the start I/O number of intelligent function modules.

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

Point

- Specify 0 when accessing to a buffer memory of other than intelligent function modules, such as CC-Link IE Field Network Ethernet adapter module.
- Indirect specification of the start I/O number can also be performed by using the CPU module index register. (☞ Page 205 Access with indirect specification of the network No. and start I/O number by using the index register)

■Device code

Specify the following device codes.

Type	Device code				Device No. range	
	ASCII code		Binary code			
	MELSEC iQ-R series*1	MELSEC-Q/L series*2	MELSEC iQ-R series	MELSEC-Q/L series		
Word	G***	G*	00ABH	ABH	Specify within the device No. range of the access destination module.	Decimal

1 For ASCII codes, the device code is specified with 4 digits. If the device code has three digits or less, add "" (ASCII code: 2AH) or a space (ASCII code: 20H) after the device code.

2 For ASCII codes, the device code is specified with 2 digits. If the device code has one digit, add "" (ASCII code: 2AH) or a space (ASCII code: 20H) after the device code.

■Head device or device No.

Specify the head device or device No. in decimal. (☞ Page 33 Head device No. (Device No.))

Point

Indirect specification of the access target device No. can be performed by using the CPU module index register or long index register. (☞ Page 210 Access with indirect specification of the device No. by using the index register or long index register)

Response data

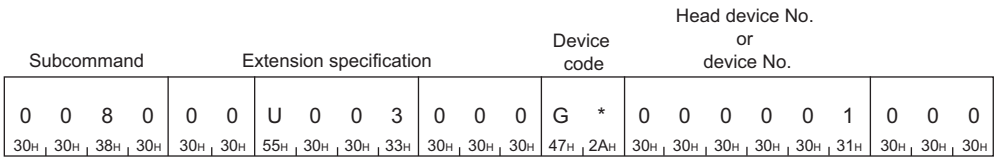
The same as when extension is not specified.

Communication example

Access to the buffer memory (Address: 1) of the intelligent function module whose start I/O number is 0030H.

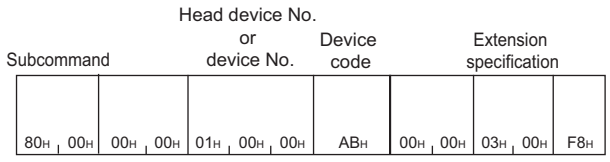
■When communicating data in ASCII code

(Request data)



■When communicating data in binary code

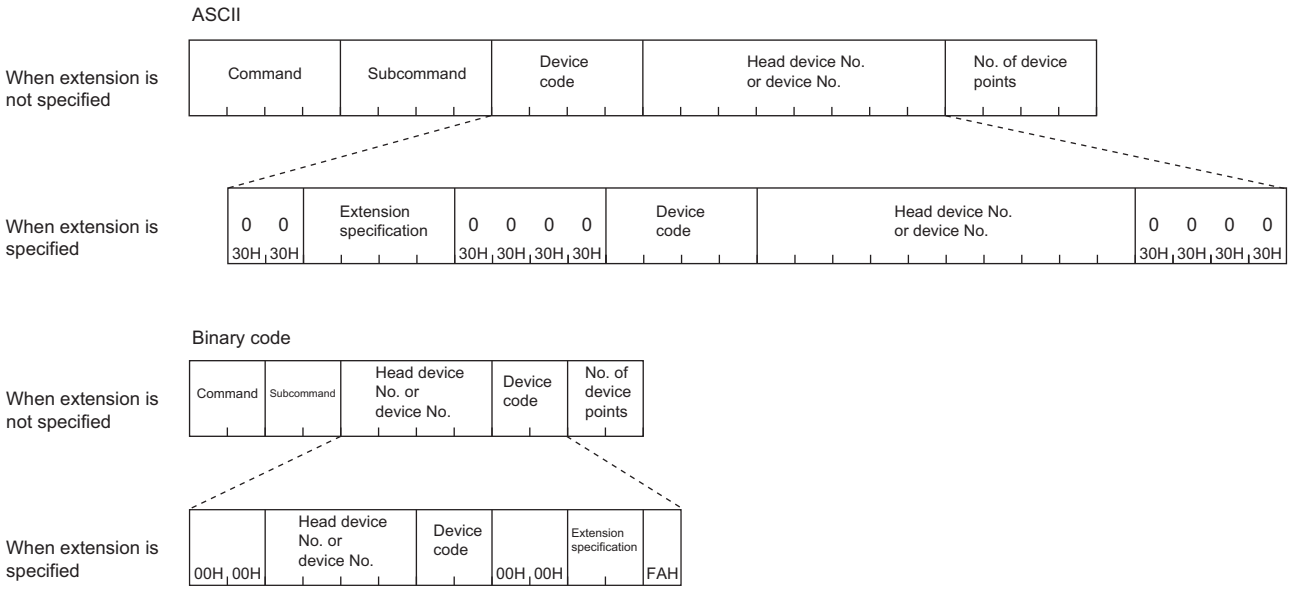
(Request data)



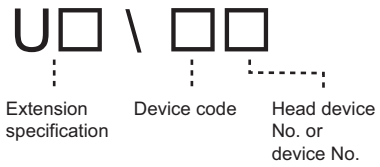
Access to the CPU buffer memory access device

Access the buffer memory of the RCPU.

Request data



The following shows the approach for the CPU module access device and request data.



■Command

The following commands can be used for accessing.

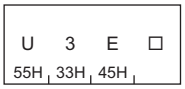
Item		Command
Type	Operation	
Device	Read	0401
	Write	1401
	Read Random	0403
	Write Random	1402
	Entry Monitor Device	0801
	Read Block	0406
	Write Block	1406

■Subcommand

ASCII code	Binary code
0 0 8 2 30H 30H 38H 32H	82H 00H

■Extension specification

Specify the start I/O number of CPU modules.

ASCII code	Binary code
Specify the start I/O number in hexadecimal (3-digit ASCII code). When described with 4-digits, specify the start I/O number with the upper 3-digits. 	Specify the start I/O number in hexadecimal (2 bytes). When described with 4-digits, specify the start I/O number with the upper 3-digits. 

The following shows the start I/O numbers of the CPU modules to be specified.

CPU module number	Start I/O number
CPU No.1	03E0H
CPU No.2	03E1H
CPU No.3	03E2H
CPU No.4	03E3H

Point

Indirect specification of the start I/O number of the CPU module can also be performed by using the CPU module index register. (☞ Page 210 Access with indirect specification of the device No. by using the index register or long index register)

■Device code

Specify the following device codes.

Device	Type	Device code		Device No. range	
		ASCII code	Binary code		
		MELSEC iQ-R series*1	MELSEC iQ-R series		
CPU buffer memory	Word	G***	00ABH	Specify within the device No. range of the access destination module.	Decimal
Fixed-cycle area of the CPU buffer memory		HG**	002EH		

1 For ASCII codes, the device code is specified with 4 digits. If the device code has three digits or less, add "" (ASCII code: 2AH) or a space (ASCII code: 20H) after the device code.

■Head device or device No.

Specify the head device or device No. in decimal. (☞ Page 33 Head device No. (Device No.))

Point

Indirect specification of the access target device No. can be performed by using the CPU module index register or long index register. (☞ Page 210 Access with indirect specification of the device No. by using the index register or long index register)

Response data

The same as when extension is not specified.

Communication example

Access the buffer memory (Address: 1) of the CPU module whose start I/O number is 03E0H.

The following shows request data when communicating data in ASCII code.

■When communicating data in ASCII code

(Request data)

Subcommand	Extension specification	Device code	Head device No. or device No.	
0 0 8 2	0 0	U 3 E 0	0 0 0 0	G * * *	0 0 0 0 0 0 0 0 0 1	0 0 0 0
30H 30H 38H 32H	30H 30H	55H 33H 45E 30H	30H 30H 30H 30H	47H 2AH 2AH 2AH	30H 30H 30H 30H 30H 30H 30H 30H 30H 31H	30H 30H 30H 30H

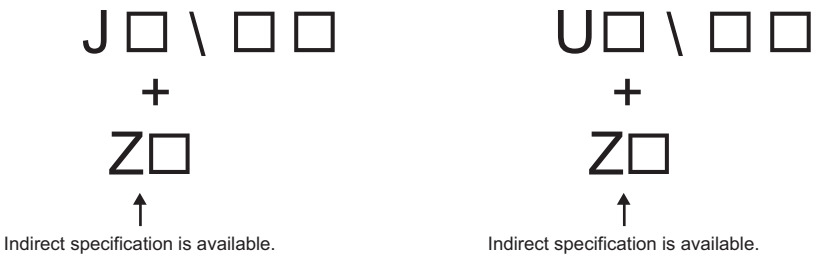
■When communicating data in binary code

(Request data)

Subcommand		Head device No. or device No.	Device code		Extension specification	
82H 00H	00H 00H	01H 00H 00H 00H	ABH 00H	00H 00H	E0H 03H	FAH

Access with indirect specification of the network No. and start I/O number by using the index register

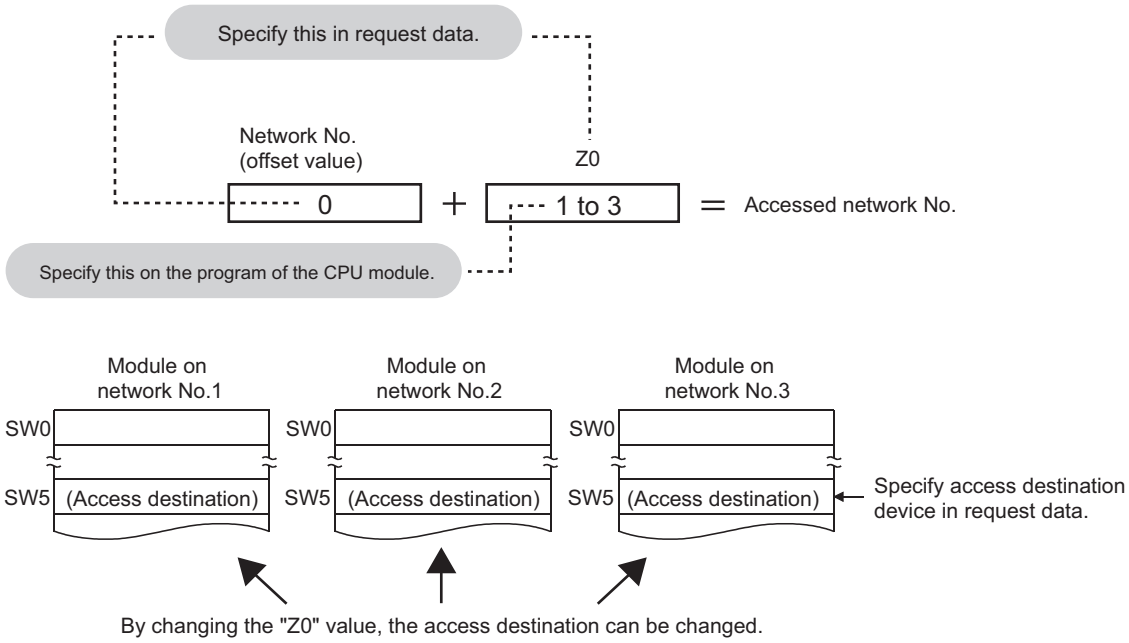
Indirect specification of the access target network No. can be performed with index register, when accessing to the link direct device. Also, indirect specification of the access target start I/O number can be performed when accessing the module access device or CPU buffer memory access device.



The access destination can be switched with one message, by changing the value of the index register in CPU module programs.

Ex.

The access destination can be switched by changing the value of "Z0", when multiple network modules are mounted onto the access destination.

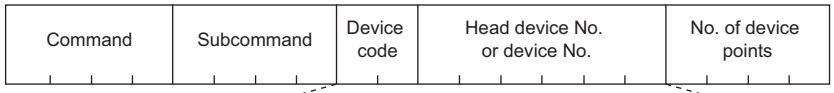


Request data

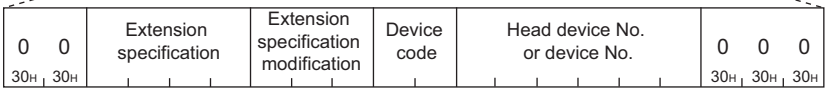
■When the subcommand is 0081 or 0080

ASCII

When extension is not specified

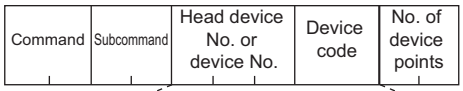


When extension is specified

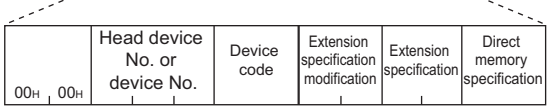


Binary

When extension is not specified



When extension is specified



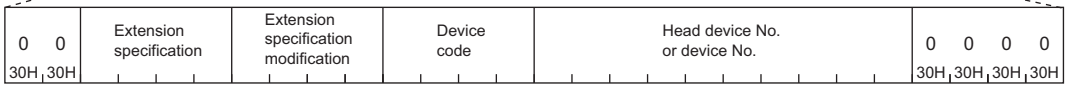
■When the subcommand is 0083 or 0082

ASCII

When extension is not specified

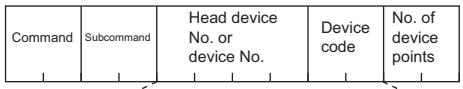


When extension is specified

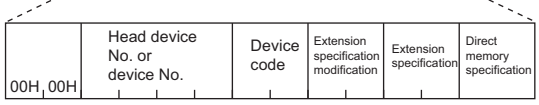


Binary

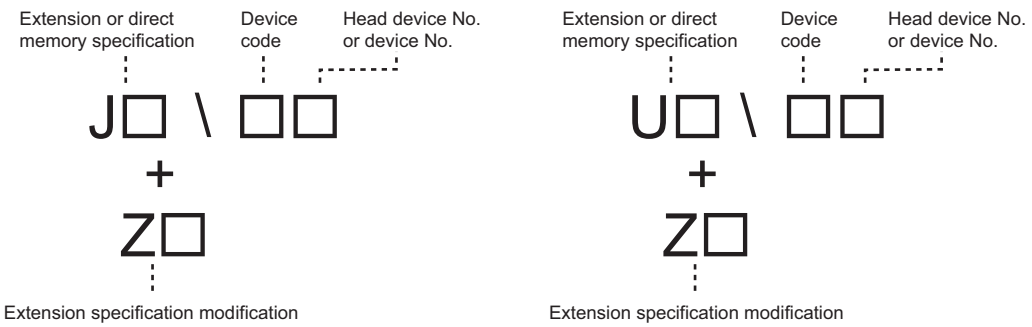
When extension is not specified



When extension is specified



The following shows the approach for the link direct device, module access device, access to the CPU buffer memory access device, index, and request data.



Point

Devices described in the following page can be accessed by specifying 0 to "extension specification", "extension specification modification" and "direct memory specification".

- ☞ Page 30 Device code

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 Random	0403
	Write Random	1402
	Entry Monitor Device	0801

■Subcommand

Item	Subcommand	
	ASCII code	Binary code
When accessing bit units	0 0 8 1 30H 30H 38H 31H	81H 00H
	0 0 8 3 30H 30H 38H 33H	83H 00H
When accessing 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 access target network No. and the offset value of start I/O number.

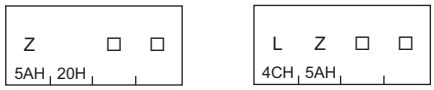
For the extension specification of each access device, refer to the following.

Item	Reference
Link direct device	☞ Page 197 Extension specification
Module access device	☞ Page 200 Extension specification
CPU buffer memory access device	☞ Page 203 Extension specification

■Extension specification modification

Treat the value specified in "extension specification" as the offset value. Specify the index register number when performing indirect specification of the network No. and start I/O number with index register.

Specify the following values when the access destination is the MELSEC iQ-R series module.

Subcommand	ASCII code	Binary code
0083 0082	Specify the number of the index register (Z) 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) 
0081 0080	Specify the number of the index register (Z) 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) 

Specify the following values when the access destination is the MELSEC-Q/L series module.

ASCII code	Binary code
Specify the number of the index register in decimal (2-digit ASCII code). (Specification range: 0 to 15) 	Specify the number of the index register in hexadecimal. (Specification range: 00H to 0FH) 

Point

- When performing indirect specification to the I/O number of the module access device with the values of the index register, store "value of the upper 3-digits when describing the start I/O number with 4 characters" in the index register.
- When performing indirect specification to the I/O number of the CPU buffer memory access device with the values of the index register, store "3E0H to 3E3H" in the index register.

■Device code

Specify the device code. For the device codes of each access device, refer to the following.

Item	Reference
Link direct device	☞ Page 198 Device code
Module access device	☞ Page 201 Device code
CPU buffer memory access device	☞ Page 203 Device code

■Head device or device No.

Specify the head device or device No. in decimal or hexadecimal.

☞ Page 33 Head device No. (Device No.)

■Direct memory specification (only when communicating in binary code)

Specify the type of the access device.

Item	Binary code
Link direct device	Specify F9H.
Module access device	Specify F8H.
CPU buffer memory access device	Specify FAH.

Response data

The same as when extension is not specified.

Communication example

Access to W100 (J1 + Z0\W100) of network No.1 + Z0.

■When communicating data in ASCII code

(Request data)

Subcommand	Extension specification	Extension specification modification	Device code	Head device No. or device No.
0 0 8 0	0 0	J 0 0 1	Z 0 0	W *	0 0 0 1 0 0	0 0 0
30H 30H 38H 30H	30H 30H	4AH 30H 30H 31H	5AH 30H 30H	57H 2AH	30H 30H 30H 31H 30H 30H	30H 30H 30H

■When communicating data in binary code

(Request data)

Subcommand	Head device No. or device No.	Device code	Extension specification modification	Extension specification	Direct memory specification
80H 00H	00H 00H	00H 01H 00H	B4H	00H 40H	01H 00H	F9H

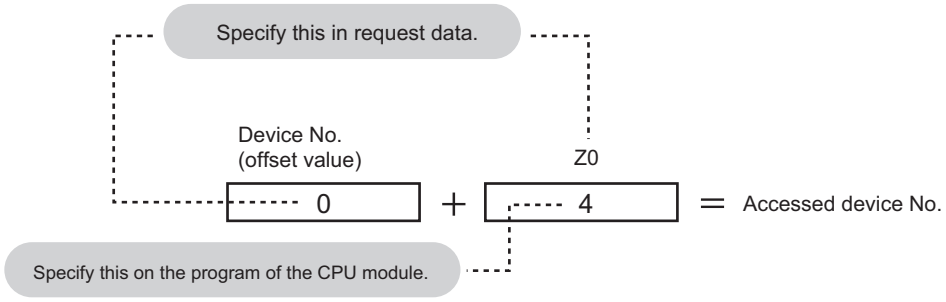
Access with indirect specification of the device No. by using the index register or long index register

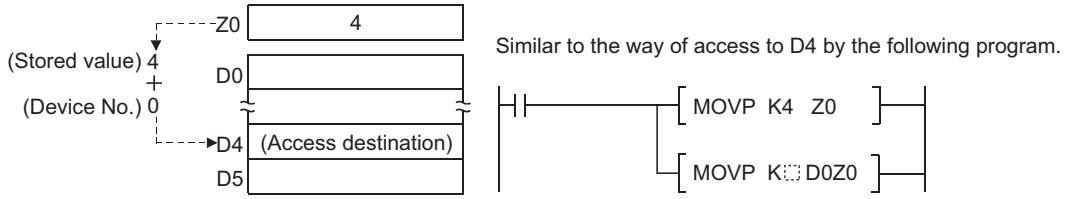
Indirect specification to the device No. can be performed by using the index register or long index register when accessing the device.

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.

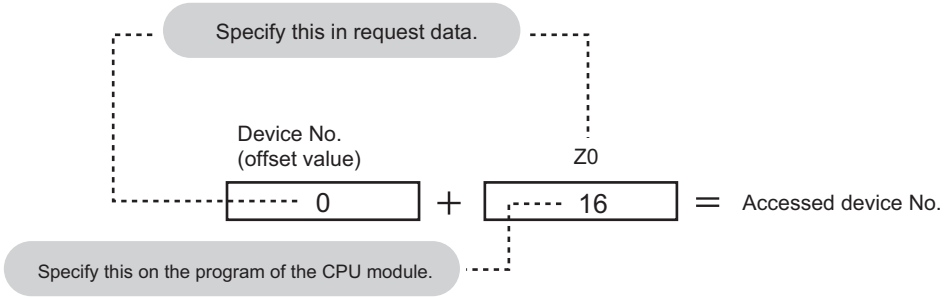
When accessing D4 with D0 and Z0 specifications

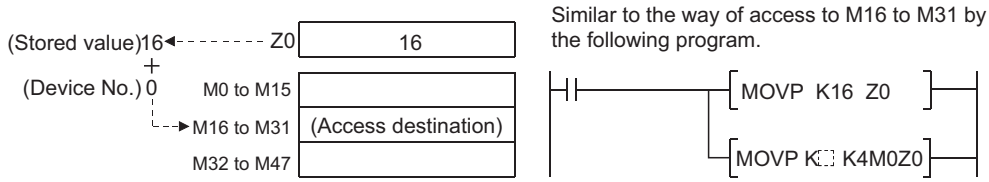




Ex.

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



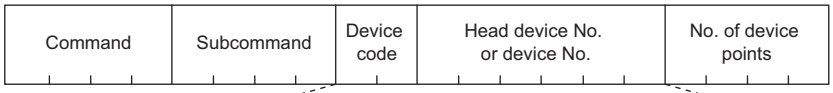


Request data

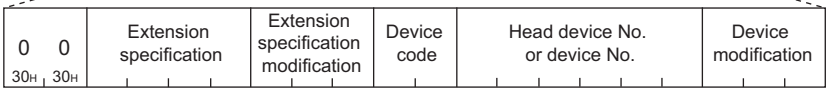
■When the subcommand is 0081 or 0080

ASCII

When extension is not specified

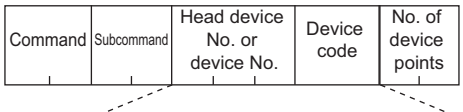


When extension is specified

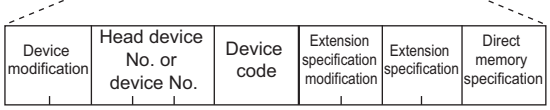


Binary

When extension is not specified



When extension is specified



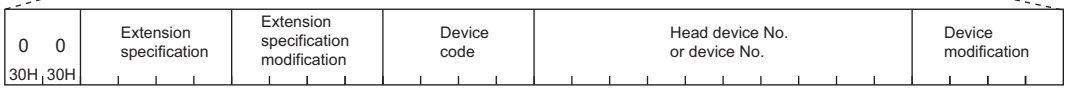
■When the subcommand is 0083 or 0082

ASCII

When extension is not specified

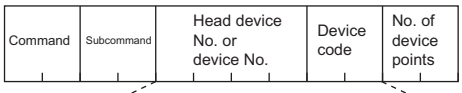


When extension is specified

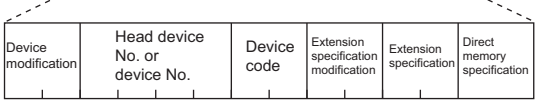


Binary

When extension is not specified

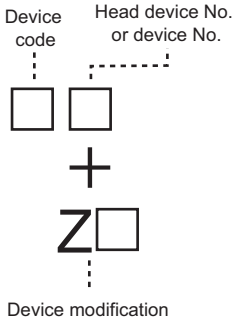


When extension is specified

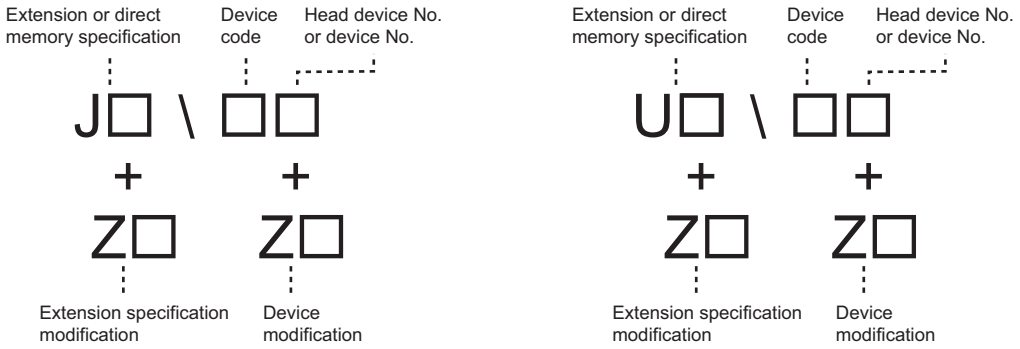


The following shows the approach for the device, index register, long index register, and request data.

- Other than the link direct device, module access device, or CPU buffer memory access device



• Link direct device, module access device, or CPU buffer memory 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
	Entry Monitor Device	0801

■Subcommand

Item	Subcommand	
	ASCII code	**Binary code**
When accessing bit units	0 0 8 1 30H 30H 38H 31H	81H 00H
	0 0 8 3 30H 30H 38H 33H	83H 00H
When accessing 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 access target network No. and the start I/O number.

The values specified in this item turn to the offset value when performing indirect specification of the network No. and start I/O number in "extension specification modification".

For the extension specification of each access device, refer to the following.

Item	Reference
Link direct device	☞ Page 198 Device code
Module access device	☞ Page 201 Device code
CPU buffer memory access device	☞ Page 203 Device code

Specify "0" when accessing a device other than the link direct device, module access device, or CPU buffer memory access device.

ASCII code	Binary code
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 number when performing indirect specification of the network No. and start I/O number with index register. (☞ Page 208 Extension specification modification)

Point

When performing indirect specification to the start I/O number with the values of the index register, store "value of the upper 3-digits when describing the start I/O number with 4 characters" in the index register.

■Device code

Specify the code of the device to be accessed. (☞ Page 30 Device code)

Refer to the following device codes.

Item	Reference
Link direct device	☞ Page 198 Device code
Module access device	☞ Page 201 Device code
CPU buffer memory access device	☞ Page 203 Device code

■Head device or device No.

Specify the head device or device No. in decimal or hexadecimal. (☞ Page 33 Head device No. (Device No.))

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 number or long index register number when performing indirect specification of the device No. with the index register or long index register.

Specify the following values when the access destination is the MELSEC iQ-R series module.

Subcommand	ASCII code	Binary code
0083 0082	Specify the number of the index register (Z) 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 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). (📖 MELSEC iQ-R CPU Module User's Manual (Application))

Specify the following values when the access destination is the MELSEC-Q/L series module.

ASCII code	Binary code
Specify the number of the index register in decimal (2-digit ASCII code). (Specification range: 0 to 15) 	Specify the number of the index register in hexadecimal. (Specification range: 00H to 0FH) 

■Direct memory specification (only when communicating in binary code)

Specify the device type when accessing the link direct device, module access device, or CPU buffer memory access device. (☞ Page 208 Direct memory specification (only when communicating in binary code))

Specify 0 when accessing a device other than the link direct device, module access device, or CPU buffer memory access device.

Binary code
Specify 0. 

Response data

The same as when extension is not specified.

Communication example

Access to 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	D	*	0	0	0	1	0	0	Z	0	4
30H	30H	38H	30H	30H	30H	30H	30H	30H	30H	30H	44H	2AH	30H	30H	30H	31H	30H	30H	5AH	30H	34H

■When communicating data in binary code

(Request data)

Subcommand	Device modification	Head device No. or device No.	Device code	Extension specification modification	Extension specification	Direct memory specification
80H 00H	04H 40H	64H 00H 00H	A8H	00H 00H	00H 00H	00H

Access with indirect specification of the device No. by using the values stored in the word device

Access to the device corresponding to the address stored in the word device (for 2 points).

Ex.

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

Using the ADRSET instruction on the CPU module side, store the address in D100 into D0.



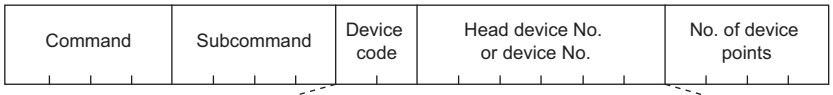
By specifying "@D0" in request data, D100 can be indirectly accessed.

Request data

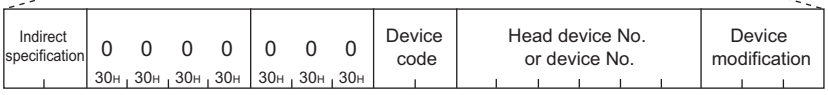
■When the subcommand is 0080

ASCII

When extension is not specified

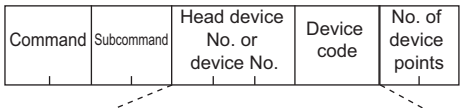


When extension is specified

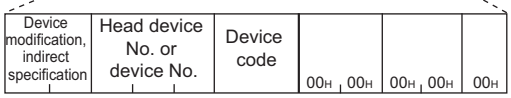


Binary

When extension is not specified



When extension is specified



■When the subcommand is 0082

ASCII

When extension is not specified

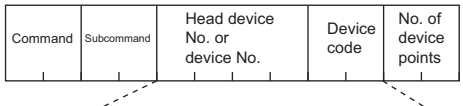


When extension is specified

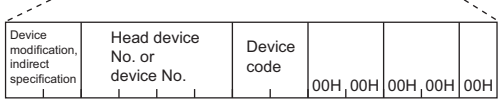


Binary

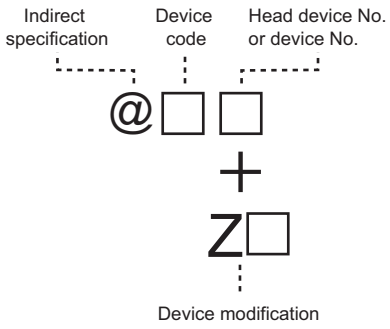
When extension is not specified



When extension is specified



The following shows the approach for indirect specification devices, index registers, long index registers, 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.

■Command

The following commands can be used for accessing.

Item		Command
Type	Operation	
Device	Read Random	0403
	Write Random	1402
	Entry Monitor Device	0801

■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

■Indirect specification, device modification

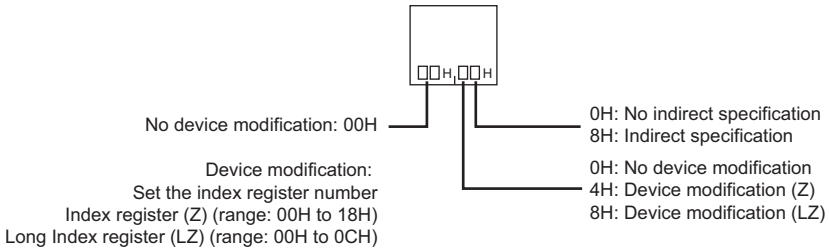
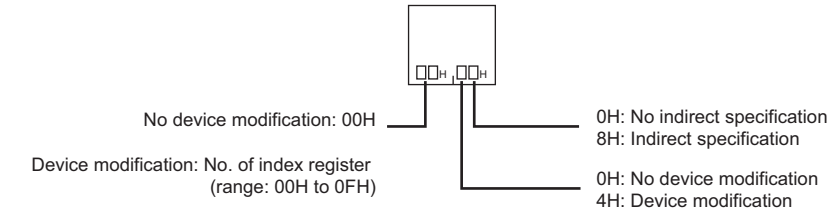
Specify the following.

- For the indirect specification: Specify the "@" part of the indirect specification device. Indirect specification can be specified only for word devices.
- For the device modification: Specify the index register number when performing indirect specification with index register or long index register to an indirectly specified device.

When communicating in ASCII code

Item	Subcommand	Description	
Indirect specification	0082 0080	0 @ 30H 40H	
Device modification	0082	For the device modification with the index register Z □ □ 5AH 20H	For no device modification with the index register 0 0 0 0 30H 30H 30H 30H
		For the device modification with the long index register L Z □ □ 4CH 5AH	For no device modification with the long index register 0 0 0 0 30H 30H 30H 30H
	0080	For the device modification with the index register Z □ □ 5AH	For no device modification with the index register 0 0 0 30H 30H 30H

When communicating in binary code

Subcommand	Description
0082	
0080	

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

Specify the code of the device to be accessed. (☞ Page 30 Device code)

Specify the following device codes when accessing the link direct device.

Device	Type	Device code				Device No. range	
		ASCII code		Binary code			
		MELSEC iQ-R series*1	MELSEC-Q/L series*2	MELSEC iQ-R series	MELSEC-Q/L series		
Link register	Word	W***	W*	00B4H	B4H	Specify within the device No. range of the access destination module.	Hexadecimal
Link special register		SW**	SW	00B5H	B5H		Hexadecimal

1 For ASCII codes, the device code is specified with 4 digits. If the device code has three digits or less, add "" (ASCII code: 2AH) or a space (ASCII code: 20H) after the device code.

2 For ASCII codes, the device code is specified with 2 digits. If the device code has one digit, add "" (ASCII code: 2AH) or a space (ASCII code: 20H) after the device code.

When accessing the module access device, refer to the device codes described in the following.

☞ Page 201 Device code

When accessing the CPU buffer memory access device, refer to the device codes described in the following.

☞ Page 203 Device code

■Head device or device No.

Specify the head device or device No. in decimal or hexadecimal. (☞ Page 33 Head device No. (Device No.))

Response data

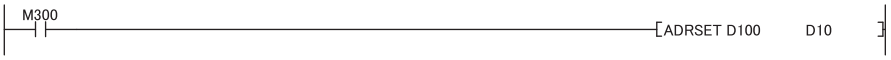
The same as when extension is not specified.

Communication example

Access to @D0 + Z4.

At command execution, store the address of the device (D100) to be accessed in D10 with the following programs.

In addition, K10 is assumed to be stored in Z4.



■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	Z	0	4
30H	30H	38H	30H	30H	40H	30H	30H	30H	30H	30H	30H	30H	44H	2AH	30H	30H	30H	30H	30H	30H	5AH	30H	34H

■When communicating data in binary code

(Request data)

Subcommand	Device modification, indirect specification	Head device No. or device No.			Device code				
80H	00H	04H	48H	00H	00H	00H	A8H	00H	00H	00H	00H	00H

Appendix 2 Correspondence Table of MC Protocol and SLMP

The message format of SLMP is the same as the QnA compatible 3E and 4E frames. 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.

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		
Monitor data registration	0801	00□0		Entry Monitor Device
Monitor of registered device memory	0802	0000		Execute Monitor
Multiple block batch read	0406	00□0		Read Block
Multiple block batch write	1406	00□0		Write Block
Buffer memory read	0613	0000	Memory	Read
Buffer memory write	1613	0000		Write
Intelligent function module buffer memory read	0601	0000	Extend Unit	Read
Intelligent function module buffer memory write	1601	0000		Write
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
Drive memory usage status read	0205	0000	If these commands are used in the external device, delete them from the programs of external device.	
Drive memory defragmentation	1207	0000		
File information table read	0201	0000		
	0202			
	0204			
New file creation (File name registration)	1202	0000	Replace this command with New File (command: 1820).	
File information modification	1204	0000	Replace this command with Change File Date (command: 1826).	
		0001	If these commands are used in the external device, delete them from the programs of external device.	
		0002		
File presence read (File search)	0203	0000	If these commands are used in the external device, delete them from the programs of external device.	
File contents read	0206	0000	Replace these commands with Read File (command: 1828) or Write File (command: 1829).	
File write	1203	0000		
		0001		
File lock register/cancel	0808	000□	Replace this command with Open File (command: 1827) and Close File (command: 182A).	
File copy	1206	0000	Replace this command with Copy File (command: 1824).	
File delete	1205	0000	Replace this command with Delete File (command: 1822).	

MC protocol			SLMP	
Item	**Command**	**Subcommand**	**Type**	**Operation**
Directory file information read	1810	0000	File	Read Directory/File
Directory file information search	1811	0000		Search Directory/File
New file creation	1820	0000		New File
File delete	1822	0000		Delete File
File copy	1824	0000		Copy File
File attribute modification	1825	0000		Change File State
File creation date modification	1826	0000		Change File Date
File open	1827	0000		Open File
File read	1828	0000		Read File
File write	1829	0000		Write File
File close	182A	0000		Close File
Loopback test	0619	0000	Self Test	
COM.ERR.LED off	1617	000□	Clear Error	
Remote password unlock	1630	0000	Remote Password	Unlock
Remote password lock	1631	0000		Lock

Appendix 3 When Accessing Multiple CPU System

This section describes the SLMP communication for accessing the multiple CPU system.

Point

Read this section when accessing the multiple CPU system. For the multiple CPU system, refer to the manual for the CPU module used. (User's Manual (Multiple CPU System) for the CPU module used)

Access range

The control CPU and non-control CPU are accessible.

The following table lists the accessible commands.

Item		Reference
Type	Operation	
Device	Read	Page 40 Read (command: 0401)
	Write	Page 44 Write (command: 1401)
	Read Random	Page 47 Read Random (command: 0403)
	Write Random	Page 51 Write Random (command: 1402)
	Entry Monitor Device*1	Page 56 Entry Monitor Device (command: 0801)
	Execute Monitor*1	Page 60 Execute Monitor (command: 0802)
	Read Block	Page 63 Read Block (command: 0406)
	Write Block	Page 67 Write Block (command: 1406)
Label	Array Label Read	Page 80 Array Label Read (command: 041A)
	Array Label Write	Page 89 Array Label Write (command: 141A)
	Label Read Random	Page 99 Label Read Random (command: 041C)
	Label Write Random	Page 106 Label Write Random (command: 141B)
Extend Unit	Read	Page 120 Read (command: 0601)
	Write	Page 122 Write (command: 1601)
Remote Control	Remote Run	Page 125 Remote Run (command: 1001)
	Remote Stop	Page 127 Remote Stop (command: 1002)
	Remote Pause	Page 128 Remote Pause (command: 1003)
	Remote Latch Clear	Page 129 Remote Latch Clear (command: 1005)
	Remote Reset	Page 130 Remote Reset (command: 1006)
	Read Type Name	Page 131 Read Type Name (command: 0101)
Remote Password	Lock	Page 135 Lock (command: 1631)
	Unlock	Page 137 Unlock (command: 1630)
File	Read Directory/File	Page 149 Read Directory/File (command: 1810)
	Search Directory/File	Page 159 Search Directory/File (command: 1811)
	New File	Page 162 New File (command: 1820)
	Delete File	Page 165 Delete File (command: 1822)
	Copy File	Page 168 Copy File (command: 1824)
	Change File State	Page 172 Change File State (command: 1825)
	Change File Date	Page 175 Change File Date (command: 1826)
	Open File	Page 178 Open File (command: 1827)
	Read File	Page 181 Read File (command: 1828)
	Write File	Page 184 Write File (command: 1829)
	Close File	Page 187 Close File (command: 182A)

*1 Cannot access a non-control CPU.

Specification of the CPU of multiple CPU system to be accessed

Specify the CPU with the request destination module I/O No. in the request message. (☞ Page 18 Request destination module I/O No.)

INDEX

0 to 9

4E frame 220

A

Access range 10
Access to the CPU buffer memory access device 202
Access to the link direct device 195
Access to the module access device 199
Access with indirect specification of the device No. by using the index register or long index register 210
Access with indirect specification of the device No. by using the values stored in the word device. . . 216
Access with indirect specification of the network No. and start I/O number by using the index register . 205
Accessible module for each command 28
Accessing to buffer memory of intelligent function module 117
Array Label Read (command: 041A) 80
Array Label Write (command: 141A) 89
Attribute 143

B

Buffer memory 6

C

CC-Link IE Field Network 6
Change File Date (command: 1826) 175
Change File State (command: 1825) 172
Clear Error (error code initialization, LED off) (command: 1617) 191
Close File (command: 182A) 187
Command list 25
Commands 24
Communication procedure of SLMP 12
Copy File (command: 1824) 168
Correspondence table of MC protocol and SLMP 220
CPU module 6,11

D

Data type ID 77
Delete File (command: 1822) 165
Device 6
Device access 30
Device code 30
Drive No. 141

E

End code 23
Entry Monitor Device (command: 0801) 56
Error information 23
Execute Monitor (command: 0802) 60
External device 6

F

File control 139
File name 142
File pointer No. 143

H

Head device No. (device No.) 33
Header 16

I

Intelligent function module 6

L

Label access 71
Label name 74
Label name length 73
Label Read Random (command: 041C) 99
Label Write Random (command: 141B) 106
Link device 6
Lock (command: 1631) 135

M

MC protocol 6
Message format 16
Modules of other stations that are accessible 11
Monitoring timer 20
Multiple CPU system 222

N

New File (command: 1820) 162
Number of abbreviation points 72
Number of array points 71
Number of bit access points 39
Number of device points 34
Number of file name characters 142
Number of read/write data points 72

O

Ondemand (command: 2101) 192
Open File (command: 1827) 178
Other station 6
Overview 7
Own station 6
Own station buffer memory access 112

P

Password 140
Procedure for changing file creation date 147
Procedure for copying a file 146
Procedure for creating a new file and writing data 145
Procedure for deleting a file 147
Procedure for overwriting data in the existing file 146

Procedure for reading file contents 144

Q

QnA compatible 3E frame 220

R

Read (command: 0401) 40
Read (command: 0601) 120
Read (command: 0613) 114
Read array data length, write array data length . . 79
Read Block (command: 0406) 63
Read data length, write data length 79
Read data, write data 35
Read Directory/File (command: 1810) 149
Read File (command: 1828) 181
Read Random (command: 0403) 47
Read Type Name (command: 0101) 131
Read unit specification, write unit specification . . . 78
Relay station 6
Remote Latch Clear (command: 1005) 129
Remote operation 124
Remote Pause (command: 1003) 128
Remote Reset (command: 1006) 130
Remote Run (command: 1001) 125
Remote Stop (command: 1002) 127
Request data length 19
Request destination module I/O No. 18
Request destination multidrop station No. 19
Request destination network No. and request destination station No. 17
Request message 6,16
Response data length 22
Response message 6,21

S

Safety precautions 1
Search Directory/File (command: 1811) 159
Self Test (loopback test) (command: 0619) 189
Serial No. 16
SLMP 6
SLMP compatible device 6,9
SLMP specifications 9
Subheader 16

T

TCP/IP 12
Term 6

U

UDP/IP 13
Unlock (command: 1630) 137

W

When accessing the CPU module 15
Write (command: 1401) 44
Write (command: 1601) 122
Write (command: 1613) 116
Write Block (command: 1406) 67
Write File (command: 1829) 184
Write Random (command: 1402) 51

REVISIONS

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

Print date	*Manual number	Description
October 2010	SH(NA)-080956ENG-A	First edition
September 2011	SH(NA)-080956ENG-B	[Addition] Section 5.8 [Partial correction] Section 2.3, 5.1, 5.6, 5.6.1
June 2013	SH(NA)-080956ENG-C	[Partial correction] Chapter 1, Section 2.2, 2.3, 3.3, 5.2.8, 5.2.9, Appendix 3
June 2014	SH(NA)-080956ENG-D	Descriptions regarding MELSEC iQ-R series are added.

Japanese manual number: SH-080931-E

This manual confers no industrial property rights of any other kind, nor does it confer any patent licenses. Mitsubishi Electric Corporation cannot be held responsible for any problems involving industrial property rights which may occur as a result of using the contents noted in this manual.

© 2010 MITSUBISHI ELECTRIC CORPORATION

WARRANTY

Please confirm the following product warranty details before using this product.

1. Gratis Warranty Term and Gratis Warranty Range

If any faults or defects (hereinafter "Failure") found to be the responsibility of Mitsubishi occurs during use of the product within the gratis warranty term, the product shall be repaired at no cost via the sales representative or Mitsubishi Service Company.

However, if repairs are required onsite at domestic or overseas location, expenses to send an engineer will be solely at the customer's discretion. Mitsubishi shall not be held responsible for any re-commissioning, maintenance, or testing on-site that involves replacement of the failed module.

[Gratis Warranty Term]

The gratis warranty term of the product shall be for one year after the date of purchase or delivery to a designated place. Note that after manufacture and shipment from Mitsubishi, the maximum distribution period shall be six (6) months, and the longest gratis warranty term after manufacturing shall be eighteen (18) months. The gratis warranty term of repair parts shall not exceed the gratis warranty term before repairs.

[Gratis Warranty Range]

(1) The range shall be limited to normal use within the usage state, usage methods and usage environment, etc., which follow the conditions and precautions, etc., given in the instruction manual, user's manual and caution labels on the product.

(2) Even within the gratis warranty term, repairs shall be charged for in the following cases.

1. Failure occurring from inappropriate storage or handling, carelessness or negligence by the user. Failure caused by the user's hardware or software design.
2. Failure caused by unapproved modifications, etc., to the product by the user.
3. When the Mitsubishi product is assembled into a user's device, Failure that could have been avoided if functions or structures, judged as necessary in the legal safety measures the user's device is subject to or as necessary by industry standards, had been provided.
4. Failure that could have been avoided if consumable parts (battery, backlight, fuse, etc.) designated in the instruction manual had been correctly serviced or replaced.
5. Failure caused by external irresistible forces such as fires or abnormal voltages, and Failure caused by force majeure such as earthquakes, lightning, wind and water damage.
6. Failure caused by reasons unpredictable by scientific technology standards at time of shipment from Mitsubishi.
7. Any other failure found not to be the responsibility of Mitsubishi or that admitted not to be so by the user.

2. Onerous repair term after discontinuation of production

(1) Mitsubishi shall accept onerous product repairs for seven (7) years after production of the product is discontinued. Discontinuation of production shall be notified with Mitsubishi Technical Bulletins, etc.

(2) Product supply (including repair parts) is not available after production is discontinued.

3. Overseas service

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

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

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

5. Changes in product specifications

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

TRADEMARKS

Microsoft, Windows, Windows Vista, Windows NT, Windows XP, Windows Server, Visio, Excel, PowerPoint, Visual Basic, Visual C++, and Access are either registered trademarks or trademarks of Microsoft Corporation in the United States, Japan, and other countries.
Intel, Pentium, and Celeron are either registered trademarks or trademarks of Intel Corporation in the United States and other countries.
Ethernet is a registered trademark of Xerox Corp.
The SD and SDHC logos are either registered trademarks or trademarks of SD-3C, LLC.
All other company names and product names used in this manual are either trademarks or registered trademarks of their respective companies.





SH(NA)-080956ENG-D(1406)MEE
MODEL: SLMP-R-E
MODEL CODE: 13JV23

MITSUBISHI ELECTRIC CORPORATION

HEAD OFFICE : TOKYO BUILDING, 2-7-3 MARUNOUCHI, CHIYODA-KU, TOKYO 100-8310, JAPAN
NAGOYA WORKS : 1-14 , YADA-MINAMI 5-CHOME , HIGASHI-KU, NAGOYA , JAPAN

When exported from Japan, this manual does not require application to the Ministry of Economy, Trade and Industry for service transaction permission.

Specifications subject to change without notice.