

Mitsubishi Programmable Controller

MELSEC iQ-R
series

MELSEC L
series

MELSEC Q
series

MELSEC Communication Protocol Reference Manual

- RJ71C24
- RJ71C24-R2
- RJ71C24-R4
- LJ71C24
- LJ71C24-R2
- LJ71E71-100
- QJ71C24N
- QJ71C24N-R2
- QJ71C24N-R4
- QJ71C24
- QJ71C24-R2
- QJ71E71-100
- QJ71E71-B5
- QJ71E71-B2

SAFETY PRECAUTIONS

(Read these precautions before using this product.)

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

Note that these precautions apply only to this product. For the safety precautions of the programmable controller system, please read the User's Manual for the CPU module used.

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

[Operating precautions]

CAUTION

- When changing data and operating status, and modifying program of the running programmable controller from an external device such as a personal computer connected to an intelligent function module, read relevant manuals carefully and ensure the safety before operation. Incorrect change or modification may cause system malfunction, damage to the machines, or accidents.
 - Do not power off the programmable controller or reset the CPU module while the setting values in the buffer memory are being written to the flash ROM in the module. Doing so will make the data in the flash ROM undefined. The values need to be set in the buffer memory and written to the flash ROM again. Doing so also can cause malfunction or failure of the module.
-

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 controller.

This manual describes the supported devices, access range, communication procedure, and message format required for using MELSEC communication protocols listed below.

Before using this product, thoroughly read this manual and the related manuals to develop full familiarity with the functions and performance of the programmable controller to ensure correct use.

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

Relevant products

RJ71C24, RJ71C24-R2, RJ71C24-R4,

LJ71C24, LJ71C24-R2, LJ71E71-100,

QJ71C24N, QJ71C24N-R2, QJ71C24N-R4, QJ71C24, QJ71C24-R2, QJ71E71-100, QJ71E71-B5, QJ71E71-B2

CONTENTS

| | |
|---|----|
| SAFETY PRECAUTIONS | 1 |
| CONDITIONS OF USE FOR THE PRODUCT | 3 |
| INTRODUCTION | 4 |
| RELEVANT MANUALS | 13 |
| TERMS | 14 |

PART 1 MELSEC COMMUNICATION PROTOCOL

| | |
|--|-----------|
| CHAPTER 1 ABILITY OF MC PROTOCOL | 16 |
| 1.1 Purposes | 16 |
| 1.2 Features | 17 |
| CHAPTER 2 SUPPORTED DEVICES AND ACCESS RANGES | 18 |
| 2.1 Supported Devices | 18 |
| 2.2 Accessible Range and Accessible Modules | 18 |
| CHAPTER 3 COMMUNICATION PROCEDURE | 20 |
| 3.1 Features of Communication Process | 20 |
| 3.2 Considerations | 21 |
| Considerations when connecting C24 | 23 |
| Considerations when connection E71 | 24 |

PART 2 MESSAGE FORMATS

| | |
|--|-----------|
| CHAPTER 4 MESSAGES OF SERIAL COMMUNICATION MODULE | 26 |
| 4.1 Types and Purposes of Messages | 26 |
| Formats and codes | 26 |
| Frame | 26 |
| 4.2 Message Formats of Each Protocol | 27 |
| Format 1 | 27 |
| Format 2 | 28 |
| Format 3 | 29 |
| Format 4 | 30 |
| Format 5 | 31 |
| 4.3 Details of Setting Data | 32 |
| Control code | 32 |
| Number of data bytes | 33 |
| Block number | 34 |
| Frame ID No | 34 |
| Sum check code | 34 |
| End code | 36 |
| Error code | 36 |
| CHAPTER 5 MESSAGES OF ETHERNET INTERFACE MODULE | 37 |
| 5.1 Types and Purposes of Messages | 37 |
| Code | 37 |

| | |
|---|-----------|
| Frame | 38 |
| 5.2 Message Format | 39 |
| 5.3 Details of Setting Data..... | 40 |
| Header..... | 40 |
| Subheader..... | 40 |
| Request data length and response data length..... | 41 |
| Monitoring timer..... | 41 |
| End code..... | 42 |
| Error information | 42 |

CHAPTER 6 ACCESS ROUTE SETTINGS 43

| | |
|--|-----------|
| 6.1 Accessible Ranges and Settable Data for Each Frame | 43 |
| 4C frame | 43 |
| 3C frame | 44 |
| 2C frame | 44 |
| 1C frame | 45 |
| 4E frame, 3E frame | 46 |
| 1E Frame..... | 47 |
| 6.2 Detail of data to be set | 48 |
| Station No..... | 48 |
| Network No., PC No..... | 50 |
| Request destination module I/O No., request destination module station No..... | 53 |
| Self-station No..... | 56 |

PART 3 COMMAND

CHAPTER 7 COMMANDS AND FUNCTIONS 58

| | |
|-------------------------------------|-----------|
| 7.1 Command List | 58 |
| Commands for 4C/3C/4E/3E frame..... | 59 |
| Commands for 2C frame | 62 |
| Commands for 1C/1E frame | 62 |

CHAPTER 8 DEVICE ACCESS 63

| | |
|--|-----------|
| 8.1 Data to be Specified in Commands | 63 |
| Devices | 63 |
| Device code list..... | 66 |
| Number of device points | 67 |
| Access points | 68 |
| Number of blocks..... | 69 |
| Read data, write data | 70 |
| Set/reset | 76 |
| Monitor condition specification | 77 |
| 8.2 Batch Read and Write | 84 |
| Batch read in word units (command: 0401)..... | 84 |
| Batch read in bit units (command: 0401)..... | 87 |
| Batch write in word units (command: 1401) | 89 |
| Batch write in bit units (command: 1401) | 92 |
| 8.3 Random Read and Write..... | 94 |
| Random read in word units (command: 0403) | 94 |
| Random write in word units (test) (command: 1402)..... | 101 |

| | |
|--|------------|
| Random write in bit units (test) (command: 1402) | 105 |
| 8.4 Batch Read and Write Multiple Blocks | 107 |
| Batch read multiple blocks (command: 0406) | 107 |
| Batch write multiple blocks (command: 1406) | 111 |
| 8.5 Device Memory Monitor | 115 |
| Monitoring procedure | 115 |
| Register monitor data (command: 0801) | 115 |
| Monitor (command: 0802) | 116 |
| CHAPTER 9 LABEL ACCESS | 119 |
| 9.1 Data to be Specified in Commands | 119 |
| Labels | 119 |
| Abbreviation specification of label | 124 |
| Points | 126 |
| Data type ID | 127 |
| Data length, unit specification | 128 |
| Read data, write data | 130 |
| 9.2 Batch Read and Write | 131 |
| Batch read array type labels (command: 041A) | 131 |
| Batch write array type labels (command: 141A) | 137 |
| 9.3 Random Read and Write | 141 |
| Random read labels (command: 041C) | 141 |
| Random write labels (command: 141B) | 144 |
| CHAPTER 10 BUFFER MEMORY ACCESS | 147 |
| 10.1 Buffer Memory | 147 |
| Data to be specified in commands | 147 |
| Batch read (command: 0613) | 149 |
| Batch write (command: 1613) | 151 |
| 10.2 Intelligent Function Module | 153 |
| Accessible modules | 153 |
| Data to be specified in commands | 154 |
| Batch read (command: 0601) | 157 |
| Batch write (command: 1601) | 159 |
| CHAPTER 11 CONTROL MODULE OPERATION | 161 |
| 11.1 Data to be specified in commands | 161 |
| Mode | 161 |
| Clear mode | 161 |
| Model name and model code | 162 |
| Remote password | 164 |
| Loopback data | 165 |
| Communication error information | 166 |
| 11.2 Remote Operation | 167 |
| Remote RUN (command: 1001) | 167 |
| Remote STOP (command: 1002) | 169 |
| Remote PAUSE (command: 1003) | 170 |
| Remote latch clear (command: 1005) | 171 |
| Remote RESET (command: 1006) | 172 |
| Read CPU model name (command: 0101) | 173 |

| | | |
|--------------------------------|--|------------|
| 11.3 | Remote Password | 174 |
| | Execution procedure | 174 |
| | Unlock (command: 1630) | 175 |
| | Lock (command: 1631) | 177 |
| 11.4 | Loopback Test | 179 |
| | Loopback test (command: 0619) | 179 |
| 11.5 | Clear Error Information | 181 |
| | Turn indicator LED OFF, initialize error code (command: 1617) | 181 |
| | Turn COM.ERR. LED OFF (command: 1617) | 183 |
| CHAPTER 12 FILE CONTROL | | 184 |
| 12.1 | Execution Procedure | 184 |
| | Procedure to read information from all files in directory (folder) | 184 |
| | Procedure to read files | 185 |
| | Procedure to overwrite files | 185 |
| | Procedure to create new file and write data | 186 |
| | Procedure to delete files | 187 |
| | Procedure to copy files | 187 |
| | Procedure to modify file creation date and time | 187 |
| 12.2 | Considerations | 188 |
| 12.3 | Data to be specified in commands | 189 |
| | Password | 189 |
| | Drive No. | 192 |
| | File No. | 193 |
| | Number of files | 194 |
| | Number of characters | 194 |
| | Directory specification | 194 |
| | File name specification | 196 |
| | Attribute | 199 |
| | Creation date and time (last edit date and time) | 200 |
| | File size | 201 |
| | File pointer No. | 202 |
| | Offset address | 202 |
| | Number of bytes | 203 |
| | Read data, write data | 203 |
| | Open mode | 204 |
| | Close type | 204 |
| 12.4 | File Check. | 205 |
| | Read directory/file information (command: 1810) | 205 |
| | Search directory/file information (command: 1811) | 211 |
| 12.5 | File Creation and Deletion | 214 |
| | Create new file (command: 1820) | 214 |
| | Delete file (command: 1822) | 217 |
| | Copy file (command: 1824) | 220 |
| 12.6 | File Modification | 224 |
| | Modify file attribute (command: 1825) | 224 |
| | Modify file creation date and time (command: 1826) | 227 |
| | Open file (command: 1827) | 230 |
| | Read file (command: 1828) | 234 |
| | Write to file (command: 1829) | 236 |

| | |
|---|------------|
| Close file (command: 182A) | 238 |
| CHAPTER 13 SERIAL COMMUNICATION MODULE DEDICATED COMMANDS | 240 |
| 13.1 User Frame | 240 |
| Data to be specified in command | 241 |
| Read registered data (command: 0610) | 243 |
| Register data (command: 1610) | 245 |
| Delete registered data (command: 1610) | 247 |
| 13.2 Global Function | 249 |
| Global signal ON/OFF (command: 1618) | 250 |
| 13.3 Transmission sequence initialization function | 252 |
| Initialize transmission sequence (command: 1615) | 252 |
| 13.4 Mode Switching Function | 253 |
| Data to be specified in command | 253 |
| Switch mode (command: 1612) | 257 |
| 13.5 Programmable controller CPU monitoring function | 260 |
| Data to be specified in command | 260 |
| Register (command: 0630) | 263 |
| Deregister (command: 0631) | 273 |
| 13.6 On-demand function | 274 |
| Settings for using the on-demand function | 274 |
| Execution procedure | 275 |
| Execution timing | 276 |
| On-demand (command: 2101) | 277 |

PART 4 COMPATIBILITY WITH QnA SERIES

| | |
|--|------------|
| CHAPTER 14 MELSEC-QnA SERIES SUPPORTED SPECIFICATIONS | 280 |
| 14.1 Frames and Commands that can be used | 280 |
| 14.2 Accessible Modules | 280 |
| 14.3 Considerations | 281 |
| CHAPTER 15 QnACPU DEDICATED COMMANDS | 282 |
| 15.1 QnACPU Dedicated Commands List | 282 |
| 15.2 Defragmentation of Drive Memory | 283 |
| Data to be specified in commands | 284 |
| Read drive memory usage status (command: 0205) | 287 |
| Defragment drive memory (command: 1207) | 289 |
| 15.3 File Control | 291 |
| Considerations for file control | 291 |
| Data to be specified in commands | 292 |
| File control execution procedure for the QnACPU | 300 |
| Read file information table without header statement (command: 0201) | 302 |
| Read file information table with header statement (command: 0202) | 305 |
| Read file presence (Search file) (command: 0203) | 308 |
| Read file No. usage status (command: 0204) | 310 |
| Read file (command: 0206) | 312 |
| Register and deregister file locks (command: 0808) | 315 |
| Create new file (Register file name) (command: 1202) | 317 |
| Write to file (command: 1203) | 319 |

| | |
|---|-----|
| Modify file information (command: 1204) | 323 |
| Delete file (command: 1205) | 329 |
| Copy file (command: 1206) | 331 |

PART 5 COMPATIBILITY WITH A SERIES

| | |
|---|------------|
| CHAPTER 16 MELSEC-A SERIES SUPPORTED SPECIFICATIONS | 336 |
| 16.1 Frames and Commands that can be Used | 336 |
| 16.2 Accessible modules | 336 |
| 16.3 Considerations | 337 |
| CHAPTER 17 COMMUNICATING USING 1C FRAMES | 338 |
| 17.1 Message Format | 338 |
| 17.2 Details of Setting Data..... | 340 |
| Command | 340 |
| Message wait..... | 341 |
| Character area..... | 341 |
| Error code | 342 |
| 17.3 Command and Function Lists for 1C Frame..... | 343 |
| 17.4 Device Memory Read and Write..... | 344 |
| Considerations..... | 344 |
| Data to be specified in command | 345 |
| Batch read (bit units) (command: BR, JR)..... | 348 |
| Batch read (word units) (command: WR, QR)..... | 350 |
| Batch write (bit units) (command: BW, JW)..... | 352 |
| Batch write (word units) (command: WW, QW)..... | 354 |
| Test (random write) (bit units) (command: BT, JT)..... | 356 |
| Test (random write) (word units) (command: WT, QT)..... | 358 |
| Monitor (Command: BM, JM, WM, QM, MB, MJ, MN, MQ)..... | 360 |
| 17.5 Read and Write Extended File Register..... | 365 |
| Considerations for reading and writing extended file register | 365 |
| Specification method for extended file register | 366 |
| Data to be specified in command | 368 |
| Batch read (command: ER)..... | 369 |
| Batch write (command: EW) | 370 |
| Test (random write) (command: ET)..... | 371 |
| Monitor (command: EM, ME)..... | 372 |
| Direct read (command: NR)..... | 375 |
| Direct write (command: NW)..... | 376 |
| 17.6 Read and write Buffer Memory of Special Function Module..... | 377 |
| Data to be specified in command | 377 |
| Accessible modules..... | 379 |
| Batch read (command: TR)..... | 380 |
| Batch write (command: TW) | 382 |
| 17.7 Loopback Test | 383 |
| Loopback test (Command: TT) | 383 |
| CHAPTER 18 COMMUNICATING USING 1E FRAMES | 385 |
| 18.1 Message Format | 385 |
| 18.2 Detail of Setting Data..... | 385 |

| | |
|---|------------|
| Header | 386 |
| Subheader | 386 |
| PC No. | 387 |
| ACPU monitoring timer | 388 |
| End code, Abnormal code | 389 |
| 18.3 Commands and Function List for 1E Frame | 390 |
| 18.4 Read and Write Device Memory | 391 |
| Considerations | 391 |
| Data to be specified in command | 391 |
| Batch read in bit units (command: 00) | 397 |
| Batch read in word units (command: 01) | 399 |
| Batch write in bit units (command: 02) | 401 |
| Batch write in word units (command: 03) | 403 |
| Test in bit units (random write) (command: 04) | 404 |
| Test in word units (random write) (command: 05) | 406 |
| Monitor device memory (command: 06, 07, 08, 09) | 408 |
| 18.5 Read and Write Extended File Register | 412 |
| Data to be specified in command | 412 |
| Batch read (command: 17) | 413 |
| Batch write (command: 18) | 415 |
| Test (random write) (command: 19) | 417 |
| Monitor extended file registers (command: 1A, 1B) | 419 |
| Direct read (command: 3B) | 422 |
| Direct write (command: 3C) | 424 |
| 18.6 Read and Write Buffer Memory of Special Function Module | 426 |
| Data to be specified in command | 426 |
| Accessible modules | 427 |
| Batch read (command: 0E) | 428 |
| Batch write (command: 0F) | 430 |

APPENDIX 432

| | |
|---|------------|
| Appendix 1 Read/Write by Device Extension Specification | 432 |
| Accessing link direct devices | 434 |
| Accessing module access devices | 436 |
| Accessing CPU buffer memory access device | 438 |
| Access for index modification | 440 |
| Accessing devices for indirect specification | 445 |
| Appendix 2 Processing Time | 448 |
| Time chart and communication time of C24 transmission sequence | 448 |
| Transmission time when accessing other stations | 450 |
| Number of scans required for processing | 453 |
| Appendix 3 Compatibility with Multiple CPU Systems | 455 |
| When RCPU is configured in the multiple CPU system | 455 |
| When QCPU is configured in the multiple CPU system | 456 |
| Appendix 4 Applicable Commands for Online Program Change | 457 |
| Commands that cannot be executed during RUN | 457 |
| Commands that enable/disable online change | 457 |
| Setting method for writing data to CPU during RUN | 458 |
| Appendix 5 Number of Processing per One Communication | 459 |
| Commands for 4C/3C/4E/3E frame | 459 |

| | |
|--|------------|
| Commands for 2C frame | 463 |
| Commands for 1C frame | 463 |
| Commands for 1E frame | 464 |
| Appendix 6 Accessible Modules for Each Command..... | 465 |
| Commands for 4C/3C/4E/3E frame..... | 465 |
| Commands for 2C frame | 468 |
| Commands for 1C frame | 469 |
| Commands for 1E frame | 470 |
| Appendix 7 Setting Examples..... | 471 |
| Setting examples of message for serial communication module..... | 471 |
| Setting examples of message for Ethernet interface module | 475 |
| Setting examples of access route | 479 |
| INDEX | 484 |
| REVISIONS..... | 486 |
| WARRANTY | 487 |
| TRADEMARKS | 488 |

RELEVANT MANUALS

| Manual name [manual number] | Description | Available form |
|--|--|-------------------------|
| MELSEC Communication Protocol Reference Manual [SH-080008](this manual) | Explains the specifications, access ranges, message protocols, and functions of MELSEC communication protocol. | e-Manual EPUB PDF |



e-Manual refers to the Mitsubishi FA electronic book manuals that can be browsed using a dedicated tool.
e-Manual has the following features:

- Required information can be cross-searched in multiple manuals.
- Other manuals can be accessed from the links in the manual.
- The hardware specifications of each part can be found from the product figures.
- Pages that users often browse can be bookmarked.

User's manuals for the module

■Serial communication module

| Manual name [manual number] | Description | Available form |
|--|---|---------------------------------------|
| MELSEC iQ-R Serial Communication Module User's Manual(Application) [SH-081251ENG] | Explains the functions, input/output signals, buffer memory, parameter settings, and trouble shooting of serial communication module. | Print book e-Manual EPUB PDF |
| MELSEC iQ-R Serial Communication Module User's Manual(Startup) [SH-081250ENG] | Explains the specifications, functions, procedures prior to operation, system configurations, wring, and data communication examples of serial communication module. | Print book e-Manual EPUB PDF |
| MELSEC-L Serial Communication Module User's Manual (Basic) [SH-080894ENG] | Explains the overview of the module and describes the applicable system configuration, the specifications, the procedures prior to operations, the basic methods of communicating with the external device, maintenance and inspection, and the troubleshooting of serial communication module. | Print book e-Manual EPUB PDF |
| Q Corresponding Serial Communication Module User's Manual (Basic) [SH-080006] | Explains the overview of the module and describes the applicable system configuration, the specifications, the procedures prior to operations, the basic methods of communicating with the external device, maintenance and inspection, and the troubleshooting of serial communication module. | Print book PDF |
| MELSEC-Q/L Serial Communication Module User's Manual (Application) [SH-080007] | Explains the information on how to perform data communication with external devices using serial communication module's special functions. | Print book e-Manual EPUB PDF |

■Ethernet interface module

| Manual name [manual number] | Description | Available form |
|--|--|---------------------------------------|
| MELSEC-L Ethernet Interface Module User's Manual (Basic) [SH-081105ENG] | Explains the information on the specifications of Ethernet interface module, the procedures for data communications with external devices, circuit connection (open/close), fixed buffer exchange, random access buffer exchange, and the troubleshooting. | Print book e-Manual EPUB PDF |
| Q Corresponding Ethernet Interface Module User's Manual (Basic) [SH-080009] | Explains the information on the specifications of Ethernet interface module, the procedures for data communications with external devices, circuit connection (open/close), fixed buffer exchange, random access buffer exchange, and the troubleshooting. | Print book PDF |
| MELSEC-Q/L Ethernet Interface Module User's Manual (Web function) [SH-080180] | Explains how to use the Web function of Ethernet interface module. | Print book e-Manual EPUB PDF |

TERMS

This manual uses the terms listed in the following table unless otherwise noted.

| Terms | Description |
|-----------------------------------|--|
| ACPU | A generic term for MELSEC-A series CPU modules |
| Buffer memory | Memory for intelligent function modules to store setting values and monitor values. |
| Built-in Ethernet port CPU module | A generic term for MELSEC-Q series and MELSEC-L series with built-in Ethernet port |
| C24 | Another term for serial communication modules |
| CC-Link IE | A generic term for CC-Link IE Controller Network and CC-Link IE Field Network |
| Connected station (host station) | Connected station (host station) indicates a station directly connected to external devices |
| Control CPU | A CPU module that controls each module. In a multiple CPU system, a control CPU can be set for each module. |
| CPU module | A generic term for MELSEC programmable controller CPU |
| Device | Supported devices and internal devices (X, Y, W, etc.) of the CPU modules |
| E71 | Another term for Ethernet interface modules |
| Engineering tool | A tool for setting, programming, debugging, and maintaining programmable controllers. The tool indicates GX Developer, GX Works2, and GX Works3 etc. For the supported tools, refer to the following manual.  The user's manual for the module used or the MELSEC iQ-R Module Configuration Manual |
| External device | A device which sends request message to the supported devices (such as a personal computer, HIM, measuring instrument, ID units, barcode readers, regulators, and C24s) |
| Intelligent function module | A generic term for MELSEC iQ-R series modules, MELSEC-Q series modules, and MELSEC-L series modules that have functions other than input or output, such as A/D or D/A converter modules |
| LCPU | A generic term for MELSEC-L series CPU modules |
| Link device | Internal devices (LX/LY/LB/LW/RX/RY/RWr/RWw) of the network modules |
| MC protocol | An abbreviation for MELSEC communication protocol Protocols to access supported devices or programmable controller connected to supported devices from external devices. |
| Module access device | A generic term for module device access of MELSEC iQ-R series and intelligent function module devices of MELSEC-Q/L series |
| Multidrop connection | A connection when connecting a personal computer and multiple external devices or C24s in 1:n or m:n basis using RS-422/485 interface |
| Network module | A generic term for MELSEC programmable controller which can be connected to Ethernet, CC-Link IE Field Network, CC-Link IE Controller Network, MELSECNET/H, and MELSECNET/10. |
| Other station | Other station indicates a station connected to the connected station (host station) on the network. |
| QCPU | A generic term for MELSEC-Q series CPU modules |
| QnACPU | A generic term for MELSEC-QnA series CPU modules |
| RCPU | A generic term for MELSEC iQ-R series CPU module |
| Relay station | A station that relays data link to other station with mounting more than one network modules on one programmable controller. |
| Response message | A processing result message sent from SLMP compatible devices in response to the request message |
| Request message | A processing request message sent from external devices to the supported devices |
| Special function module | A generic term for MELSEC A series modules and MELSEC-QnA series modules that have functions other than input or output, such as A/D or D/A converter modules |
| Supported device | A generic term for devices which receive MC protocol messages |
| User frame | A data name used when registering the fixed format part in a message to be transmitted between an external device and a serial communication module, and using it for data transmission and reception. (The content of data in a user frame must be the same as the specifications of the external device.) Register the order (transmission control code, C24 station No., sum check, fixed data, etc.) of the head and end part of a message to be transmitted to serial communication module. This function is used for the on-demand function of MC protocol and the data send/receive function by nonprocedural protocol. |

PART 1

MELSEC COMMUNICATION PROTOCOL

MELSEC communication protocol (hereinafter abbreviated as MC protocol) is a communication protocol for MELSEC programmable controller used when accessing programmable controller from an external device via C24 or E71. This part explains the overview and basic operations of MC protocol.

1 ABILITY OF MC PROTOCOL

2 SUPPORTED DEVICES AND ACCESS RANGES

3 COMMUNICATION PROCEDURE

1 ABILITY OF MC PROTOCOL

This chapter explains the purposes and features of the MC protocol.

1.1 Purposes

MC protocol performs data communication to control a programmable controller system on the external devices (such as a personal computer, GOT).

Reading and writing data

By reading/writing data to/from the device memory of the CPU module and buffer memory of the intelligent function modules, the following operations can be performed:

■Reading data

Operation monitoring, data analysis, production control, etc. of the CPU module can be performed on the external device.

■Writing data

Production instructions can be issued from the external device.

Reading and writing files

By reading/writing files such as programs and parameters stored in the CPU module, the following operations can be performed:

■Reading files

File management for the CPU modules of the connected station (host station) and other station can be performed on the external device.

■Writing files

Execution programs can be modified (replaced) by writing file data stored in the external device to the programmable controller CPU as necessary.

Remote control of CPU module

The CPU module can be remotely controlled from the external device by performing remote RUN/STOP/PAUSE/Latch Clear/RESET operations.

Monitoring CPU module

The status of CPU module and data in the device memory can be sent to an external device at constant intervals, upon the occurrence of a mechanical error, or when certain conditions are satisfied.

Data transmission from CPU module to external device (On-demand function)

The emergency data that is required to notify to external devices can be sent from the CPU module.

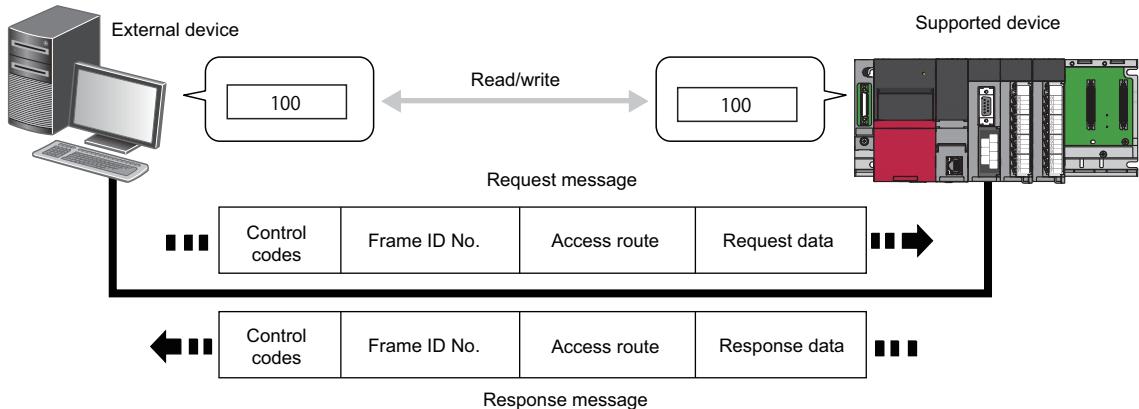
1.2 Features

Communication from external device without using sequential programs

The programmable controller transmits data in accordance with the commands from an external device. Thus, a program for data communication is not required for CPU module. (When using on-demand functions with C24, a sequence program for data communication from CPU module is required.)

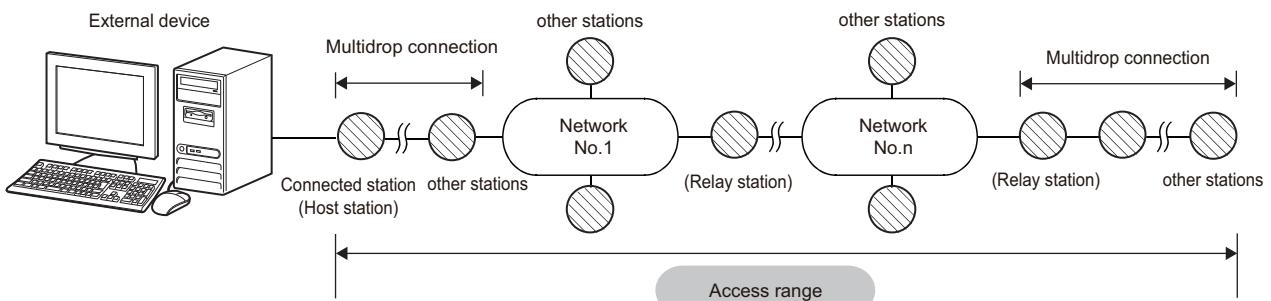
Communication protocols of C24/E71

The message formats and control procedures for an external device to access a programmable controller are defined for each supported device.



Accessible via various network

By using MC protocol, accessing other station via various network can be performed seamlessly.



2 SUPPORTED DEVICES AND ACCESS RANGES

This chapter explains the supported modules and accessible modules.

2.1 Supported Devices

Communications using the MC protocol can be performed as an external device if a device can incorporate application programs and send/receive data using the control procedures of the MC protocol.

This manual explains MC protocol communication when connecting the devices shown below and external devices.

| Type | Series | Model name | Reference |
|-----------------------------|--------------------|---|---|
| Serial communication module | MELSEC iQ-R series | RJ71C24, RJ71C24-R2, RJ71C24-R4 | Page 26 MESSAGES OF SERIAL COMMUNICATION MODULE |
| | MELSEC-L series | LJ71C24, LJ71C24-R2 | |
| | MELSEC-Q series | QJ71C24N, QJ71C24N-R2, QJ71C24N-R4, QJ71C24, QJ71C24-R2 | |
| Ethernet interface module | MELSEC-L series | LJ71E71-100 | Page 37 MESSAGES OF ETHERNET INTERFACE MODULE |
| | MELSEC-Q series | QJ71E71-100, QJ71E71-B5, QJ71E71-B2 | |

2.2 Accessible Range and Accessible Modules

A CPU module and other station on the network can be accessed via a supported device.

Accessible range

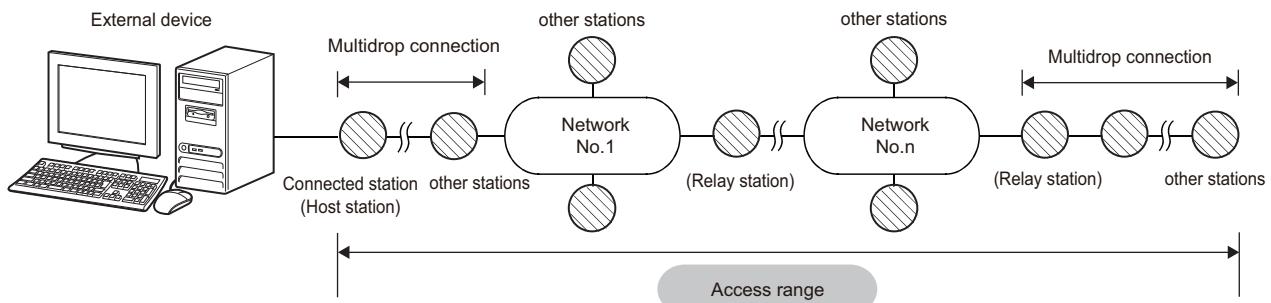
The following devices are accessible.

- Supported devices that are connected to the external device directly (connected station)
- Other stations on the same network with the supported devices^{*1}
- Other stations on the other network that are connected to other station in the same network with the supported device.^{*1}

^{*1} Other station on which network No. and station No. is set, and a multidrop connection station can be accessed.

The accessible range differs in types (frames) of a message format.

The following figure shows the maximum range that can be accessed.



For the accessible range of each message, refer to the following sections.

| Message formats of serial communication modules | Message formats of Ethernet interface modules |
|---|---|
| <ul style="list-style-type: none">Page 43 Accessible range of 4C framePage 44 Accessible range of 3C framePage 44 Accessible range of 2C framePage 45 Accessible range of 1C frame | <ul style="list-style-type: none">Page 46 Accessible range of 4E frame, 3E framePage 47 Accessible range of 1E frame |

For details on the accessible range of programmable controllers of other stations on a network system, refer to the manual for the network system used.

The multiple CPU system is supported. For details on the multiple CPU system, refer to the following section.

☞ Page 455 Compatibility with Multiple CPU Systems

Accessible modules

The following modules in the access range can be accessed.

However, the commands that can be used have some restrictions. (☞ Page 465 Accessible Modules for Each Command)

For the access to A series and QnA series modules, refer to the following sections.

- For QnA series: ☞ Page 280 MELSEC-QnA SERIES SUPPORTED SPECIFICATIONS
- For A series: ☞ Page 336 MELSEC-A SERIES SUPPORTED SPECIFICATIONS

For the restrictions for model name and version of the supported module, refer to the manual of each module.

■Connectable modules

The following modules can be connected from an external device with the serial communication.

- Serial communication module
- CPU modules that have serial communication function

For the serial communication functions, refer to the user's manual of the CPU module used.

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

☞ QnUCPU User's Manual (Function Explanation, Program Fundamentals)

The following modules can be connected from an external device with the Ethernet communication.

- Ethernet interface module
- Built-in Ethernet CPU module

For Ethernet built-in CPU modules, refer to the user's manual of the CPU module used.

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

☞ QnUCPU User's Manual (Communication via Built-in Ethernet Port)

■Accessible modules via supported device

The following modules are accessible from the connected station (host station) or other station.

- CPU module
- MELSECNET/H remote I/O station
- CC-Link IE Field Network head module
- Intelligent function module

■Modules that can be relayed between networks

The networks that can be relayed by setting the network No. and station No., and the devices that can be relayed are shown below.

| Network | Module type | Model name |
|-------------------------------|---|--|
| CC-Link IE Field Network | CC-Link IE Field Network master/local-equipped module | RJ71GF11-T2, RJ71EN71 (when using the CC-Link IE Field Network function) |
| | CC-Link IE Field Network master/local module | LJ71GF11-T2, QJ71GF11-T2, QS0J71GF11-T2 |
| CC-Link IE Controller Network | CC-Link IE Controller Network-equipped module | RJ71GP21-SX |
| | CC-Link IE Controller Network module | QJ71GP21-SX, QJ71GP21S-SX |
| MELSECNET/H | MELSECNET/H module | QJ71LP21, QJ71LP21-25, QJ71LP21S-25, QJ71LP21G, QJ71BR11, QJ71NT11B |
| Ethernet | Ethernet interface module | RJ71EN71, LJ71E71-100, QJ71E71-100, QJ71E71-B5, QJ71E71-B2, QJ71E71 |

In the circumstances that multiple network modules with the same network No. are mounted on the station with C24/E71, the access to other stations is performed via the network module mounted on the slot that has the lowest base unit number when the network No. is specified.



The following networks are accessible.

- Ethernet (Setting the network No. and station No. are required.)
- CC-Link IE Controller Network
- CC-Link IE Field Network
- MELSECNET/H

Up to 8 connection targets (relay stations: 7 stations) can be accessed.

3 COMMUNICATION PROCEDURE

This chapter explains the considerations when performing communication from an external devices to the programmable controller system using MC protocol.

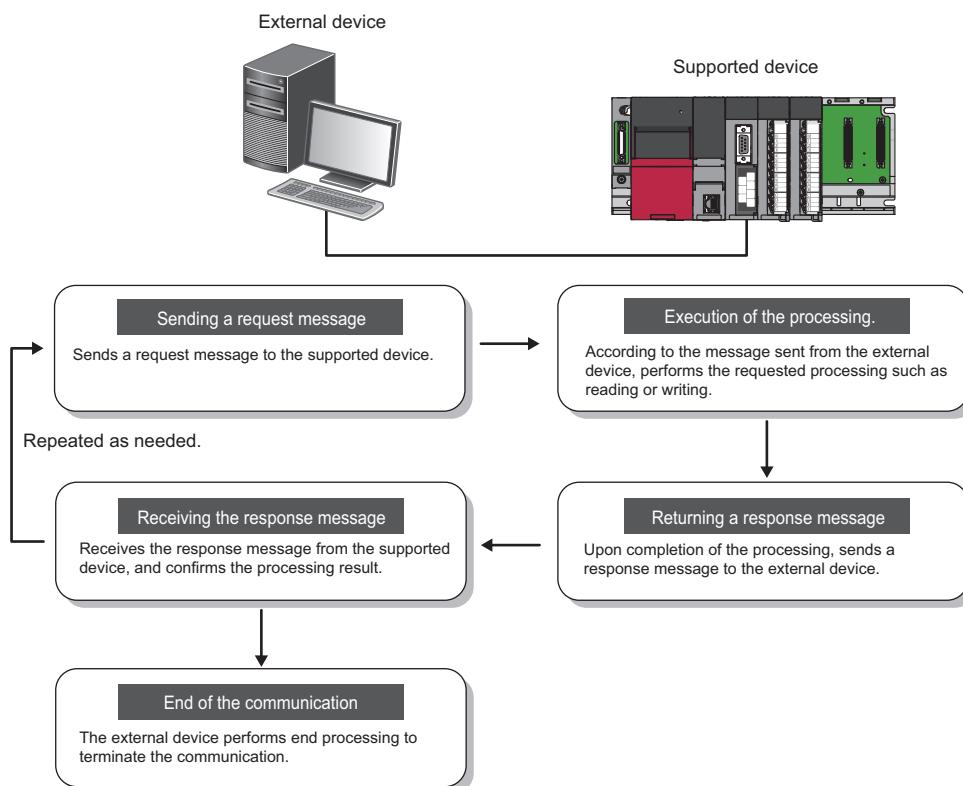
3.1 Features of Communication Process

The following shows the features of data communication using MC protocol.

Request messages and response messages

There are two types of messages in MC protocol; request message and response message.

The request message sent from the external device is processed in the CPU module via MC protocol supported devices. The processing result is returned as a response message to the external device.



Half-duplex communication

Data communication by MC protocol is performed with half-duplex communication.*1

When accessing CPU module, send the next command message after receiving the response message against the previously sent command message from the CPU module. (The command messages cannot be sent until the reception of the response message is completed.)

When the system between external devices and CPU module is configured with an m:n connection, the next command message cannot be sent until data communication between either of the external devices and CPU modules is completed.

*1 When using the on-demand function using C24, full-duplex communication can be performed.

3.2 Considerations

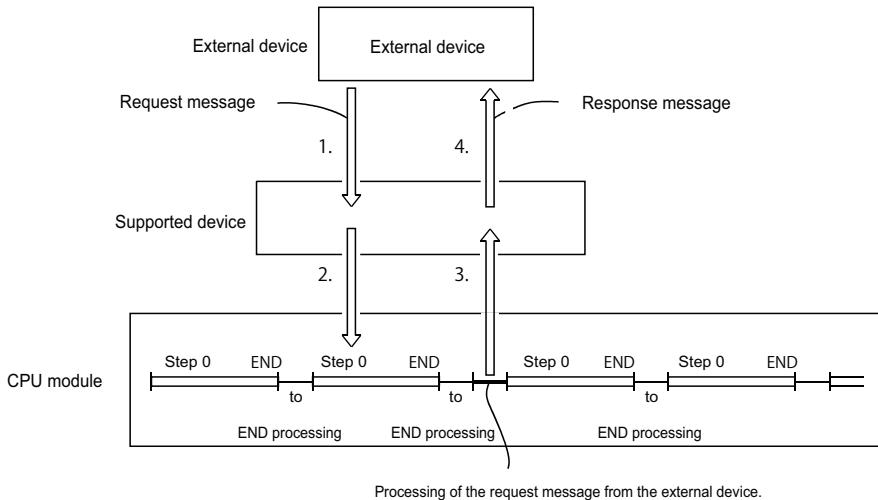
The following are the considerations when performing data communication.

When accessing CPU module

The following are the considerations when accessing a CPU module from an external device via a supported device.

■ Processing timing of CPU module

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



1. Send a request message from the external device to a supported device.
2. When the request message is received from the external device to the target device, a data read request or write request is issued to the CPU module.
3. In accordance with the request issued from the external device during END processing from the external device, CPU modules reads/writes data and then sends the processing result back to the supported device.
4. Once the supported device receives a processing result from the CPU module, send a response message including the processing result to the external device.

■ Read or write data while CPU module is in RUN

- Due to process the request from the external device, scan time of the CPU module will be extended. If the extension of the scan time affects the control, access the CPU module several times with less number of points.
- Before writing, check whether the online change is enabled (such as whether the system protection is unlocked or not) on the CPU module side.

■ When the access target CPU module is protected by system protection

An error occurs at the access target, 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 COM instruction by program.
- Ensure 1 to 100 ms of service processing time using "Service Processing Setting" of Engineering tool.

When accessing MELSECNET/H or MELSECNET/10 remote I/O station

The following are the considerations when accessing a MELSECNET/H remote I/O station or MELSECNET/10 remote I/O station from external devices.

■Considerations for supported frames

1E frame cannot be used for communication.

■Functions that can be used

The functions that can be used for remote I/O station are as follows:

| Function | MELSECNET/H remote I/O station | MELSECNET/10 remote I/O station |
|---|--------------------------------|---------------------------------|
| Read/write device memory | ○ | — |
| Read/write buffer memory | ○ | — |
| Read/write intelligent function module | ○ | ○ |
| Programmable controller CPU monitoring function (only when C24 is mounted) *1 | ○ | — |

*1 The monitoring target devices can be registered within the device range of MELSECNET/H remote I/O station.

Only reading from/writing to the buffer memory of intelligent function module is available for MELSECNET/10 remote I/O stations supported by QnA/A series.

■When writing data

When writing data to the devices of the remote I/O station or intelligent function module mounted on the remote I/O station, set the setting of online program change to allow with Engineering tool.

( Page 458 Setting method for writing data to CPU during RUN)

■Accessible range of other station

Accessing MELSECNET/H remote master station and MELSECNET/H remote I/O station can be performed from E71 with MELSECNET/H remote I/O station mounted.

Other station cannot be accessed via a remote I/O station. (A remote I/O station does not work as a relay station.)

Considerations when connecting C24

The following shows the considerations when performing data communication between C24.

3

Conditions that C24 transmission sequence becomes initial status

The transmission sequence of C24 becomes initial status under the following conditions.

- When turning the power ON, changing the status of reset switch on the front of the CPU, and switching modes.
- When the transmission of a response message for the command message reception is completed.
- When a transmission sequence initialization request is received.
- When the CD signal turns OFF while performing data communication by setting the "CD terminal check enable" in the full-duplex communication of RS-232.

Abnormal response from C24

An abnormal response to the external device using MC protocol is issued when an error is detected in the request sent to the connected station (host station). Therefore, an abnormal response may be issued while sending request from the external device in the full-duplex communication.

Replacement of other station CPU modules to communicate with

C24 imports and retains the information of other station CPU modules after started up.

When replacing the other station CPU module to perform data communication after starting up C24, reboot the C24 if the model name of the CPU module is changed. (Reset the programmable controller of the connected station (host station)/CPU reset).

Framing error on external device

A framing error may occur on the external device when nothing is sent from C24 to the external device via RS-422/485.

 Q Corresponding Serial Communication Module User's Manual (Basic)

 MELSEC-L Serial Communication Module User's Manual (Basic)

 MELSEC iQ-R Serial Communication Module User's Manual (Application)

Skip reading data until C24 sends either STX, ACK, or NAK on the external device.

Before the data communication, check the C24 interface specification written in the following manuals.

 Q Corresponding Serial Communication Module User's Manual (Basic)

 MELSEC-L Serial Communication Module User's Manual (Basic)

 MELSEC iQ-R Serial Communication Module User's Manual (Startup)

Installation of multiple C24s

When multiple external devices that are connected to each C24 request access to the CPU module at the same time, the CPU module decides the order of access.

The access priority order cannot be set by user.

Considerations when connection E71

Send of request message

Before sending a request message, check that the supported device is ready to receive the request message.

When sending multiple request messages

For 4E frame, send the request message with appending "Serial No." to the subheader on the external device. By appending "Serial No.", the send source can be identified when multiple request messages have been sent.

When sending request message continuously

When sending request messages consecutively without waiting for the reception of the response message using 4E frame, make sure that the number of command messages shown below are not exceeded.

| Name | Model name | TCP/UDP | Applicable number of commands for processing per one connection*1 |
|---------------------------|-----------------------------|---------|---|
| Ethernet interface module | QJ71E71-100, LJ71E71-100 | TCP | (6144 ÷ Message size (byte)) |
| | | UDP | 1 + (57 ÷ Number of connections to be used) |
| | QJ71E71-B2, QJ71E71-B5 | TCP | (5120 ÷ Message size (byte)) |
| | | UDP | 1 + (10 ÷ Number of connections to be used) |

*1 If the calculation result became decimal, the number after the decimal points will be rounded off to positive number.

Any of the following symptoms occurs when the exceeded number of commands are sent. In such case, decrease the frequency of request message transmission.

- For TCP/IP, the receive buffer memory of supported device will be full. Since the window size becomes '0', the transmission of request messages will be stopped temporarily until enough receive buffer space becomes available.
- For UDP/IP, an error may occur in the supported device, or response messages may not be returned from the supported device.

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

If the response message is not returned from the supported device, resend the request message from the external device after the specified time set with "Monitoring timer" of the request message is elapsed.

Replacement of devices

After replacing a device on Ethernet due to failure, the devices may not communicate properly because of the change of the MAC address. (When replaced with the device that has the same IP address)

When a device in the Ethernet network is replaced, restart all devices in the network.

PART 2**MESSAGE FORMATS**

This part explains the message format of MC protocol.

[4 MESSAGES OF SERIAL COMMUNICATION MODULE](#)

[5 MESSAGES OF ETHERNET INTERFACE MODULE](#)

[6 ACCESS ROUTE SETTINGS](#)

4 MESSAGES OF SERIAL COMMUNICATION MODULE

This section explains the specifications of the messages of MC protocol and access range when connecting with serial communication from an external device.

4.1 Types and Purposes of Messages

The messages of MC protocol can be classified as shown in the following table depending on the supported device and its intended purpose.

Formats and codes

There are five formats for the message that can be used for serial communication module.

| Setting value | Format | Code of communication data | Remarks | Reference |
|---------------|----------|----------------------------|---|------------------|
| 1 | Format 1 | ASCII code | — | Page 27 Format 1 |
| 2 | Format 2 | ASCII code | Format with block number appended | Page 28 Format 2 |
| 3 | Format 3 | ASCII code | Format enclosed with STX and ETX | Page 29 Format 3 |
| 4 | Format 4 | ASCII code | Format with CR and LF appended at the end | Page 30 Format 4 |
| 5 | Format 5 | Binary code | Can be used by 4C frame. | Page 31 Format 5 |

Set the format with the communication protocol setting of Engineering tool.



Communication using binary code shorten the communication time since the amount of communication data is reduced by approximately half as compared to the one using ASCII code.

Frame

This section explains the types and purposes of the frames (data communication messages) used by the external device to access the supported devices using MC protocol.

The frames for serial communication modules are as follows:

| Frame | Features and purposes | Compatible message format | Format |
|----------|--|--|----------------|
| 4C frame | Accessible from external devices with the maximum access range. | Dedicated protocols for MELSEC-QnA series serial communication modules (QnA extension frame). | Formats 1 to 5 |
| 3C frame | These message formats are simplified compared to the 4C frame. | Dedicated protocols for MELSEC-QnA series serial communication modules (QnA frame). | Formats 1 to 4 |
| 2C frame | Data communication software for MELSEC-QnA series programmable controllers can be used. | Dedicated protocols for MELSEC-QnA series serial communication modules (QnA simplified frame). | |
| 1C frame | These frames have the same message structures as when accessing the CPU module using an MELSEC-A series computer link module. Data communication software for MELSEC-A series programmable controllers can be used. | Dedicated protocols for MELSEC-A series computer link modules | |

4.2 Message Formats of Each Protocol

This section explains the message format and setting data per each format.

Format 1

Message format

■Request message

| | | | | |
|----------------------------|--------------|--------------|--------------|----------------|
| Control code ENQ 05H | Frame ID No. | Access route | Request data | Sum check code |
| Sum check range | | | | |

4

■Response message (Normal completion: Response data)

| | | | | | |
|----------------------------|--------------|--------------|---------------|----------------------------|----------------|
| Control code STX 02H | Frame ID No. | Access route | Response data | Control code ETX 03H | Sum check code |
| Sum check range | | | | | |

■Response message (Normal completion: No response data)

| | | |
|----------------------------|--------------|--------------|
| Control code ACK 06H | Frame ID No. | Access route |
|----------------------------|--------------|--------------|

■Response message (Abnormal completion)

| | | | |
|----------------------------|--------------|--------------|------------|
| Control code NAK 15H | Frame ID No. | Access route | Error code |
|----------------------------|--------------|--------------|------------|

Setting data

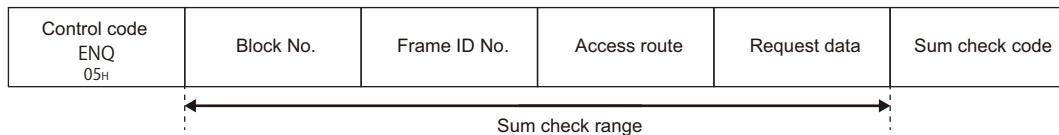
Set the following items.

| Item | Description | Reference |
|---|--|--------------------------------|
| Control code (ENQ, STX, ACK, NAK, ETX) | A code is defined for control. | Page 32 Control code |
| Frame ID No. | Specify the frame to be used. | Page 34 Frame ID No. |
| Access route | Specify the access route. | Page 43 ACCESS ROUTE SETTINGS |
| Request data | Set the command that indicates the request content. Refer to "Request data" rows of each command. | Page 58 COMMANDS AND FUNCTIONS |
| Response data | Store the read data for the command. Refer to "Response data" rows of each command. | |
| Sum check code | The value of the lower one byte (8 bits) of the additional result regarding the data in the sum check target range as a binary data. | Page 34 Sum check code |
| Error code | Error code indicates the content of occurred error. | Page 36 Error code |

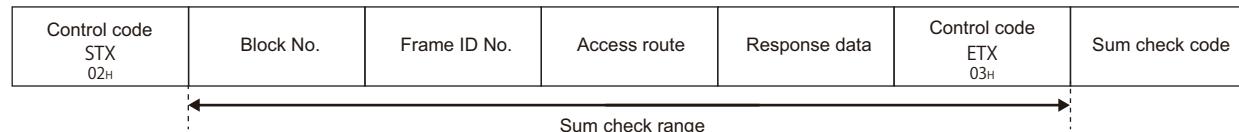
Format 2

Message format

■Request message



■Response message (Normal completion: Response data)



■Response message (Normal completion: No response data)



■Response message (Abnormal completion)



Setting data

Set the following items.

| Item | Description | Reference |
|---|--|--------------------------------|
| Control code (ENQ, STX, ACK, NAK, ETX) | A code is defined for control. | Page 32 Control code |
| Block number | This can set arbitrarily in the range of '00H' to 'FFH'. It is used for data defragmentation. | Page 34 Block number |
| Frame ID No. | Specify the frame to be used. | Page 34 Frame ID No. |
| Access route | Specify the access route. | Page 43 ACCESS ROUTE SETTINGS |
| Request data | Set the command that indicates the request content. Refer to "Request data" rows of each command. | Page 58 COMMANDS AND FUNCTIONS |
| Response data | Store the read data for the command. Refer to "Response data" rows of each command. | |
| Sum check code | The value of the lower one byte (8 bits) of the additional result regarding the data in the sum check target range as a binary data. | Page 34 Sum check code |
| Error code | Error code indicates the content of occurred error. | Page 36 Error code |

Format 3

Message format

■Request message

| | | | | | |
|----------------------------|--------------|--------------|--------------|----------------------------|----------------|
| Control code STX 02H | Frame ID No. | Access route | Request data | Control code ETX 03H | Sum check code |
| ← Sum check range → | | | | | |

■Response message (Normal completion: Response data)

| | | | | | | |
|----------------------------|--------------|--------------|----------|---------------|----------------------------|----------------|
| Control code STX 02H | Frame ID No. | Access route | End code | Response data | Control code ETX 03H | Sum check code |
| ← Sum check range → | | | | | | |

■Response message (Normal completion: No response data)

| | | | | |
|----------------------------|--------------|--------------|----------|----------------------------|
| Control code STX 02H | Frame ID No. | Access route | End code | Control code ETX 03H |
|----------------------------|--------------|--------------|----------|----------------------------|

■Response message (Abnormal completion)

| | | | | | |
|----------------------------|--------------|--------------|----------|------------|----------------------------|
| Control code STX 02H | Frame ID No. | Access route | End code | Error code | Control code ETX 03H |
|----------------------------|--------------|--------------|----------|------------|----------------------------|

Setting data

Set the following items.

| Item | Description | Reference |
|----------------------------|---|--------------------------------|
| Control code (STX, ETX) | A code is defined for control. | Page 32 Control code |
| Frame ID No. | Specify the frame to be used. | Page 34 Frame ID No. |
| Access route | Specify the access route. | Page 43 ACCESS ROUTE SETTINGS |
| Request data | Set the command that indicates the request content. Refer to "Request data" rows of each command. | Page 58 COMMANDS AND FUNCTIONS |
| Response data | Store the read data for the command. Refer to "Response data" rows of each command. | |
| Sum check code | The value of the lower one byte (8 bits) of the additional result regarding the data in the sum check target range as a binary data. | Page 34 Sum check code |
| End code | Indicates that the processing result is a normal completion or abnormal completion. • 4C/3C/2C frame: QACK (normal), QNAK (abnormal) • 1C frame: GG (normal completion), NN (abnormal completion) | Page 36 End code |
| Error code | Error code indicates the content of occurred error. | Page 36 Error code |

Format 4

Message format

■Request message

| | | | | | | |
|----------------------------|--------------|--------------|--------------|----------------|---------------------------|-----------|
| Control code ENQ 05H | Frame ID No. | Access route | Request data | Sum check code | Control code CR 0DH | LF 0AH |
| Sum check range | | | | | | |

■Response message (Normal completion: Response data)

| | | | | | | | |
|----------------------------|--------------|--------------|---------------|----------------------------|----------------|---------------------------|-----------|
| Control code STX 02H | Frame ID No. | Access route | Response data | Control code ETX 03H | Sum check code | Control code CR 0DH | LF 0AH |
| Sum check range | | | | | | | |

■Response message (Normal completion: No response data)

| | | | | |
|----------------------------|--------------|--------------|---------------------------|-----------|
| Control code ACK 06H | Frame ID No. | Access route | Control code CR 0DH | LF 0AH |
|----------------------------|--------------|--------------|---------------------------|-----------|

■Response message (Abnormal completion)

| | | | | | |
|----------------------------|--------------|--------------|------------|---------------------------|-----------|
| Control code NAK 15H | Frame ID No. | Access route | Error code | Control code CR 0DH | LF 0AH |
|----------------------------|--------------|--------------|------------|---------------------------|-----------|

Setting data

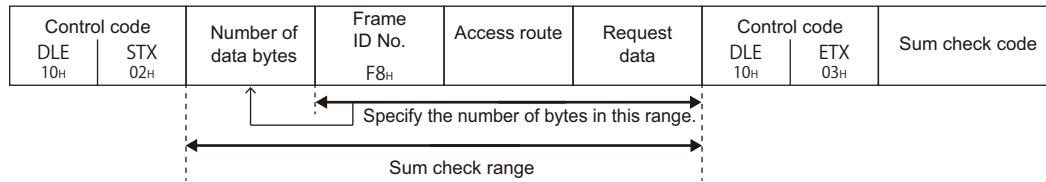
Set the following items.

| Item | Description | Reference |
|---|--|--------------------------------|
| Control code (ENQ, STX, ACK, NAK, ETX, CR, LF) | A code is defined for control. | Page 32 Control code |
| Frame ID No. | Specify the frame to be used. | Page 34 Frame ID No. |
| Access route | Specify the access route. | Page 43 ACCESS ROUTE SETTINGS |
| Request data | Set the command that indicates the request content. Refer to "Request data" rows of each command. | Page 58 COMMANDS AND FUNCTIONS |
| Response data | Store the read data for the command. Refer to "Response data" rows of each command. | |
| Sum check code | The value of the lower one byte (8 bits) of the additional result regarding the data in the sum check target range as a binary data. | Page 34 Sum check code |
| Error code | Error code indicates the content of occurred error. | Page 36 Error code |

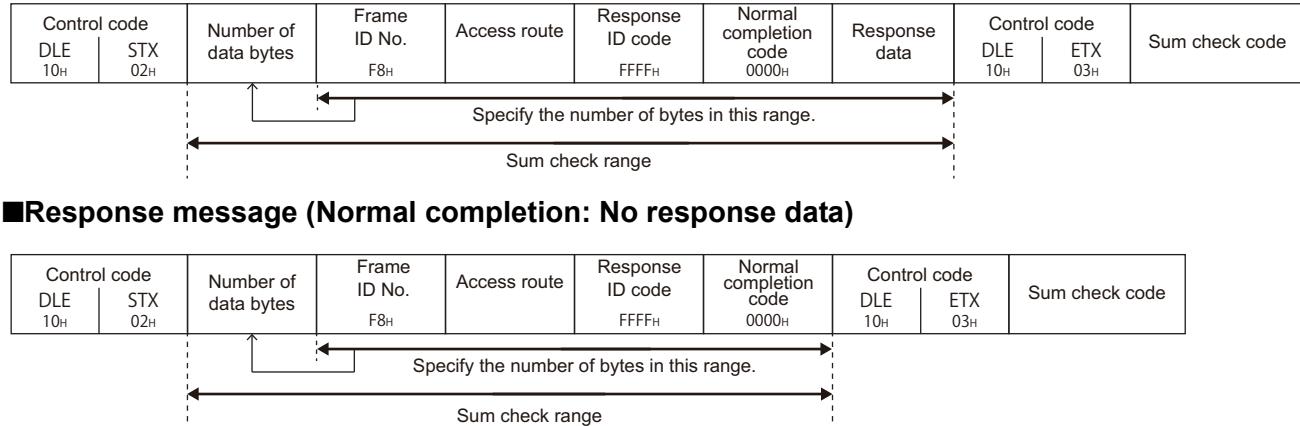
Format 5

Message format

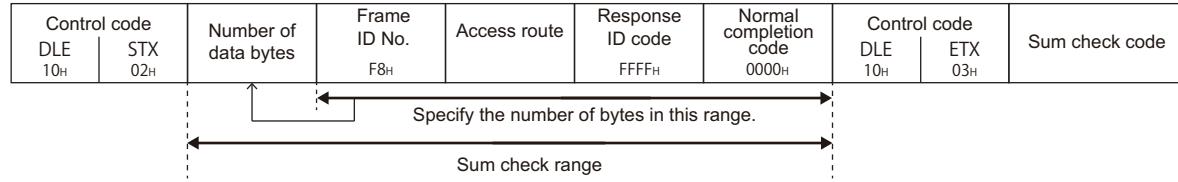
■Request message



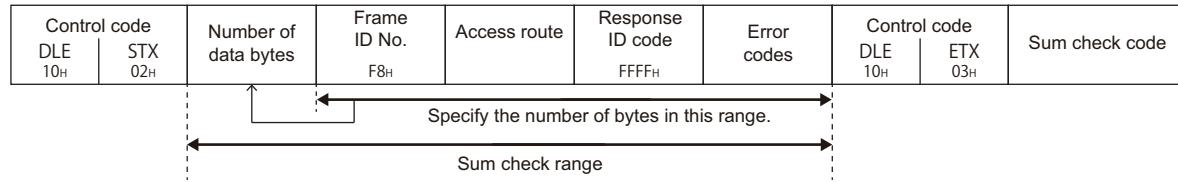
■Response message (Normal completion: Response data)



■Response message (Normal completion: No response data)



■Response message (Abnormal completion)



Setting data

Set the following items.

| Item | Description | Reference |
|---------------------------------|--|--------------------------------|
| Control code (DLE, STX, ETX) | A code is defined for control. | Page 32 Control code |
| Number of data bytes | A number of bytes from a frame ID No. to control code (DLE, ETX). | Page 33 Number of data bytes |
| Frame ID No. | Specify the frame to be used. | Page 34 Frame ID No. |
| Access route | Specify the access route. | Page 43 ACCESS ROUTE SETTINGS |
| Request data | Set the command that indicates the request content. Refer to "Request data" rows of each command. | Page 58 COMMANDS AND FUNCTIONS |
| Response data | Store the read data for the command. Refer to "Response data" rows of each command. | |
| Sum check code | The value of the lower one byte (8 bits) of the additional result regarding the data in the sum check target range as a binary data. | Page 34 Sum check code |
| Response ID code | This indicates a response message. The 2-byte numerical value, 'FFFFH' is stored. | — |
| Normal completion code | This indicates the processing is completed normally. The 2-byte value, '0000H' is stored. | — |
| Error code | Error code indicates the content of occurred error. | Page 36 Error code |

4.3 Details of Setting Data

This section explains how to specify the common data items and their content in each message.

Control code

Control code is a data that has special meaning (such as head data of a message) for C24 transmission control.

Control code used in a message (format 1 to format 4) in ASCII code

The control code used for a message in ASCII code (format 1 to format 4) is shown in the following table.

| Symbol name | Description | Code (hexadecimal) |
|-------------|----------------------|--------------------|
| STX | Start of Text | 02H |
| ETX | End of Text | 03H |
| EOT | End of Transmission | 04H |
| ENQ | Enquiry | 05H |
| ACK | Acknowledge | 06H |
| LF | Line Feed | 0AH |
| CL | Clear | 0CH |
| CR | Carriage Return | 0DH |
| NAK | Negative Acknowledge | 15H |

■EOT(04H), CL(0CH)

EOT and CL are codes for initializing the transmission sequence for data communications in ASCII code using the MC protocol and for placing C24 into wait state to receive commands from an external device.

The transmission sequence is initialized with the command (command code: 1615) when binary code (format 5) is used.

When performing the following at an external device, send the EOT/CL to the C24 depending on the format used.

- Canceling a read/write request by command previously sent. (If a write request is issued, the write request cannot be canceled when the data has already written to the CPU module.)
- Placing C24 into the wait state to receive commands before commands are sent.
- Placing C24 into the state where it has been started up when data communication cannot be performed normally.

The message structure when sending EOT, CL is shown below.

Only the following data is sent. The station No. and PC No. are not required.

| Format | EOT | CL |
|----------------|--------------------------|-------------------------|
| Formats 1 to 3 | EOT 04H | CL 0CH |
| Format 4 | EOT CR LF 04H 0DH 0AH | CL CR LF 0CH 0DH 0AH |

When C24 receives EOT or CL, it proceeds as follows.

- C24 terminates any read/write processing performed to the CPU module upon request from the external device. In this case, C24 does not send a response message to the command previously received.
- C24 initializes the transmission sequence using the MC protocol on the interface side and placing C24 into wait state to receive commands from an external device.
- C24 does not send a response message to the EOT or CL reception. (It does not send anything to external devices.)
- When it receives EOT or CL while the on-demand function (data transmission function from the CPU module to external devices) is being performed, C24 terminates to transmit the on-demand data to external devices. ( Page 274 On-demand function)

Control code used in a message (format 5) in binary code

The control code used for a message in binary code (format 5) is shown in the table below.

| Symbol name | Description | Code (hexadecimal) |
|-------------|------------------|--------------------|
| STX | Start of Text | 02H |
| ETX | End of Text | 03H |
| DLE | Data Link Escape | 10H |

■Additional code (10H)

The additional code is added to distinguish the data when the control code DLE (10H) is the same as the setting data in the frame 5.

When '10H' is included in the data from "Number of data bytes" and "Request data" in the request message, the additional code '10H' is added in front of the data.

When '10H' is included in the data from "Number of data bytes" and "Response data" in the response message, the additional code '10H' is added.

('10H' is transmitted as '10H' + '10H'.)



Calculate the following value except for the additional code.

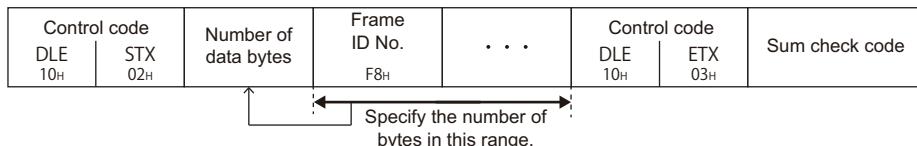
- Number of data bytes (setting item of format 5)
- Sum check code

Number of data bytes

A number of data bytes indicates the total number of bytes from the frame ID No. to control code.

Range

Calculate the data in the range from frame ID No. before DLE (10H) except for the additional code. (☞ Page 33 Additional code (10H))



Setting method

Set the data in binary code (format 5) at data communication.

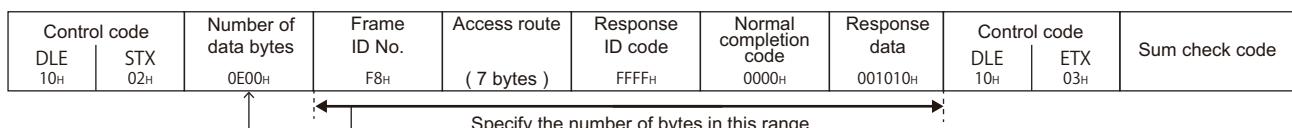
Send 2-byte numerical value from the lower byte (L: bits 0 to 7).

Ex.

Response message (Normal completion: Response data)

- Frame ID No.: 1 byte
- Access route: 7 byte
- Response ID code, normal completion code: 4 bytes
- Response data: 2 bytes + additional code (10H) 1 byte

Number of data bytes = 1 + 7 + 4 + 2 = 14 (0EH)



Block number

Block number is an arbitrary number defined by an external device and used for data defragmentation.

Block number converts data to 2-digit (hexadecimal) ASCII code within the range of '00H' to 'FFH' and sent them from the upper digits.

C24 only checks if the block number is specified within the correct range. It does not check whether the block numbers are sent in order.

Frame ID No.

Specify the frame to be used.

| Type | Setting value |
|----------|------------------|
| 4C frame | F8 |
| 3C frame | F9 |
| 2C frame | FB |
| 1C frame | — (Not required) |

Setting method

■Data communication in ASCII code

Convert the numerical value to 2-digit ASCII code (hexadecimal), and send it from the upper digits.

■Data communication in binary code

Send 1-byte numerical value.

Ex.

For 4C frame (F8)

| ASCII code | Binary code |
|------------------|-------------|
| F 8 46H , 38H | F8H |

Sum check code

Set the sum check code when performing sum check.

For sum check code, set the value to be calculated from the data with the range of sum check for error detection.

Sum check

Sum check is a function for detecting error when data changes while data transmission.

Set the sum check existence by Engineering tool.

■When sum check code is set to "Exist"

Attach a sum check code to the request message.

C24 checks the sum check code. The sum check code is added to the response message.

■When sum check code is set to "None"

The sum check code is not required for the request message.

C24 does not check the sum check code. The sum check code is not added to the response message.

Sum check range

The sum check range of each message format is as follows:

| Format | Message structure | Reference |
|----------------------|---|--|
| Format 1 to format 3 | <p>Control code Sum check code Sum check range</p> | Page 27 Format 1 Page 28 Format 2 Page 29 Format 3 |
| Format 4 | <p>Control code Sum check code Control code CR 0DH LF 0AH Sum check range</p> | Page 30 Format 4 |
| Format 5 | <p>Control code Control code DLE 10H ETX 03H Sum check code Sum check range</p> | Page 31 Format 5 |

Calculation of a sum check code

For sum check code, set the numerical values of the lower 1 byte (8 bits) of the added result (sum) as binary data within the sum check range.

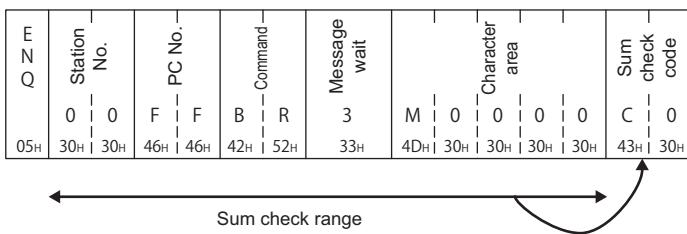
Calculate sum check code except for the additional code. (☞ Page 33 Additional code (10H))

Ex.

In the following case of 1C frame format 1, the sum check code will be 'C0'.

Formula: 30H + 30H + 46H + 46H + 42H + 52H + 33H + 4DH + 30H + 30H + 30H = 2C0H

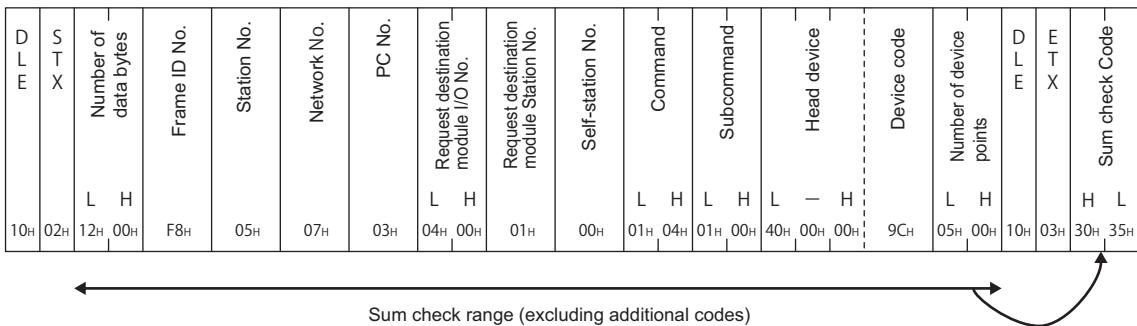
Sum check code: 'C0' (ASCII code 43H, 30H)



In the following case of 4C frame format 5, the sum check code will be '05'.

Formula: 12H + 00H + F8H + 05H + 07H + 03H + 04H + 00H + 01H + 00H + 01H + 04H + 01H + 00H + 40H + 00H + 00H + 9CH + 05H + 00H = 205H

Sum check code: '05' (ASCII code 30H, 35H)



Setting method

■Data communication in ASCII code

Convert the numerical value to 2-digit ASCII code (hexadecimal), and send it from the upper digits.

■Data communication in binary code

The same as the data communication in ASCII code, use the numerical value converted to the 2 digit ASCII code (hexadecimal).

Send 2-byte numerical value from the upper byte (H: bits 8 to 15).

Ex.

Sum check code: '05' (ASCII code 30H, 35H)

ASCII code, binary code

| | |
|-----|-----|
| 0 | 5 |
| 30H | 35H |

End code

Indicates that the processing result is a normal completion or abnormal completion.

The following fixed value is stored.

| Processing result | 4C frame, 3C frame, 2C frame | 1C frame | | | | | | | | | | | | |
|---------------------|--|----------|-----|---|---|-----|-----|-----|-----|--|---|---|-----|-----|
| Normal completion | <table border="1"><tr><td>Q</td><td>A</td><td>C</td><td>K</td></tr><tr><td>51H</td><td>41H</td><td>43H</td><td>48H</td></tr></table> | Q | A | C | K | 51H | 41H | 43H | 48H | <table border="1"><tr><td>G</td><td>G</td></tr><tr><td>47H</td><td>47H</td></tr></table> | G | G | 47H | 47H |
| Q | A | C | K | | | | | | | | | | | |
| 51H | 41H | 43H | 48H | | | | | | | | | | | |
| G | G | | | | | | | | | | | | | |
| 47H | 47H | | | | | | | | | | | | | |
| Abnormal completion | <table border="1"><tr><td>Q</td><td>N</td><td>A</td><td>K</td></tr><tr><td>51H</td><td>4EH</td><td>41H</td><td>48H</td></tr></table> | Q | N | A | K | 51H | 4EH | 41H | 48H | <table border="1"><tr><td>N</td><td>N</td></tr><tr><td>4EH</td><td>4EH</td></tr></table> | N | N | 4EH | 4EH |
| Q | N | A | K | | | | | | | | | | | |
| 51H | 4EH | 41H | 48H | | | | | | | | | | | |
| N | N | | | | | | | | | | | | | |
| 4EH | 4EH | | | | | | | | | | | | | |

Error code

Error code indicates the content of occurred error.

If more than one error occurs at the same time, the error code detected first is returned.

For the content of error code and its corrective action, refer to the user's manual of the module used.

Q Corresponding Serial Communication Module User's Manual (Basic)

MELSEC-L Serial Communication Module User's Manual (Basic)

MELSEC iQ-R Serial Communication Module User's Manual (Application)

Setting method

■Data communication in ASCII code

Convert the numerical value to 4-digit ASCII code (hexadecimal) and send it from upper digits.

■Data communication in binary code

Send 2-byte numerical value from the lower byte (L: bits 0 to 7).

Ex.

When error code 7151H is returned

| ASCII code | Binary code | | | | | | | | | | |
|--|-------------|-----|-----|---|-----|-----|-----|-----|---|-----|-----|
| <table border="1"><tr><td>7</td><td>1</td><td>5</td><td>1</td></tr><tr><td>37H</td><td>31H</td><td>35H</td><td>31H</td></tr></table> | 7 | 1 | 5 | 1 | 37H | 31H | 35H | 31H | <table border="1"><tr><td>51H</td><td>71H</td></tr></table> | 51H | 71H |
| 7 | 1 | 5 | 1 | | | | | | | | |
| 37H | 31H | 35H | 31H | | | | | | | | |
| 51H | 71H | | | | | | | | | | |

For the error code of 1C frame, refer to the following section.

Page 342 Error code

5 MESSAGES OF ETHERNET INTERFACE MODULE

This section explains the specifications of the messages and access range of MC protocol when connecting with Ethernet communication from an external device.

5.1 Types and Purposes of Messages

The messages of MC protocol can be classified as shown in the following table depending on the supported device and its intended purpose.

Code

ASCII code and binary code are available.

Set the operation settings with Engineering tool.

5



Communication using binary code shorten the communication time since the amount of communication data is reduced by approximately half as compared to the one using ASCII code.

Data storage order

The data size and the storing order of values for data in each item vary between ASCII code and binary code.

■Data communication using ASCII code

Data is stored in order from the upper byte to the lower byte.

■Data communication using binary code

Data is stored in order from the lower byte to the upper byte.

Ex.

Subheader of 4E frame request message (serial No. is '1234')

| ASCII code | Binary code | | | | | | | | | | | | | | | |
|---|-----------------------|-----------------------|---------|-----------------------|-----------------------|-----------------------|---------------|---------------|--|--|-----------|-----------|-----------|---------------|---------------|--|
| <p style="text-align: center;">Serial number</p> <table border="1"><tr><td>5 4 0 0</td><td>1 2 3 4</td><td>0 0 0 0</td></tr><tr><td>35H , 34H , 30H , 30H</td><td>31H , 32H , 33H , 34H</td><td>30H , 30H , 30H , 30H</td></tr><tr><td>(Fixed value)</td><td>(Fixed value)</td><td></td></tr></table> | 5 4 0 0 | 1 2 3 4 | 0 0 0 0 | 35H , 34H , 30H , 30H | 31H , 32H , 33H , 34H | 30H , 30H , 30H , 30H | (Fixed value) | (Fixed value) | | <p style="text-align: center;">Serial number</p> <table border="1"><tr><td>54H , 00H</td><td>34H , 12H</td><td>00H , 00H</td></tr><tr><td>(Fixed value)</td><td>(Fixed value)</td><td></td></tr></table> | 54H , 00H | 34H , 12H | 00H , 00H | (Fixed value) | (Fixed value) | |
| 5 4 0 0 | 1 2 3 4 | 0 0 0 0 | | | | | | | | | | | | | | |
| 35H , 34H , 30H , 30H | 31H , 32H , 33H , 34H | 30H , 30H , 30H , 30H | | | | | | | | | | | | | | |
| (Fixed value) | (Fixed value) | | | | | | | | | | | | | | | |
| 54H , 00H | 34H , 12H | 00H , 00H | | | | | | | | | | | | | | |
| (Fixed value) | (Fixed value) | | | | | | | | | | | | | | | |

Frame

This section explains the types and purposes of the frames (data communication messages) used by the external device to access the supported devices using MC protocol.

The frames for Ethernet interface modules are as follows:

| Frame | Features and purposes | Compatible message format | Correspondence code |
|----------|--|--|---------------------------|
| 4E frame | A message format that a "Serial No." (arbitrary number for message identification) is added to 3E frame. By appending a "Serial No.", the send source can be identified when multiple request messages have been sent. | Message formats for SLMP | ASCII code Binary code |
| 3E frame | These frames have the same message structures as when accessing the CPU module using MELSEC-QnA series Ethernet interface module. Data communication software for MELSEC-QnA series programmable controllers can be used. | Message formats for SLMP Message formats for MELSEC-QnA series Ethernet interface modules | ASCII code Binary code |
| 1E frame | These frames have the same message structures as when accessing the CPU module using an MELSEC-A series Ethernet interface module. Data communication software for MELSEC-A series programmable controllers can be used. | Message formats for MELSEC-A series Ethernet interface module | ASCII code Binary code |

5.2 Message Format

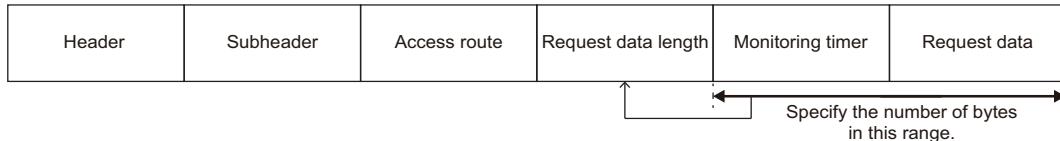
This section explains the message format and setting data for 4E frame and 3E frames.

For the message format for 1E frame, refer to the following section.

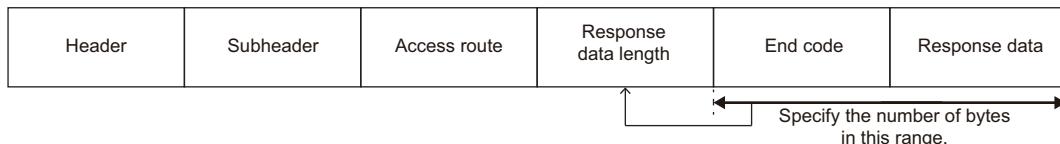
☞ Page 385 Message Format

Message format

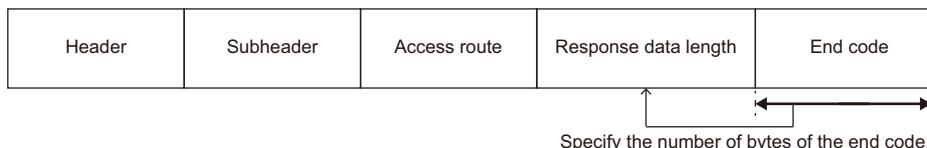
■Request message



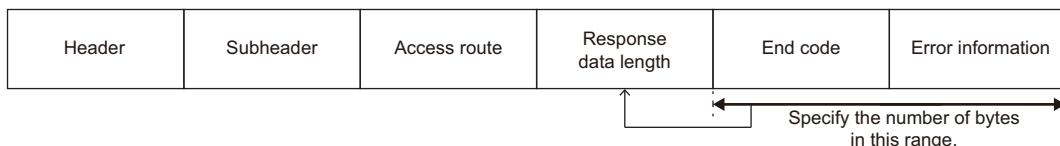
■Response message (At normal completion: Response data)



■Response message (Normal completion: No response data)



■Response message (Normal completion: Response data)



5

Setting data

Set the following items.

| Item | Description | Reference |
|----------------------|---|--|
| Header | A header of Ethernet. Normally, it is added automatically. | Page 40 Header |
| Subheader | The value to be set according to type of message is defined. • 4E frame: Set a serial No. • 3E frame: Fixed value (Request message '5000', Response message 'D000') | Page 40 Subheader |
| Access route | Specify the access route. | Page 43 ACCESS ROUTE SETTINGS |
| Request data length | Specify the data length from the monitoring timer to the request data. | Page 41 Request data length and response data length |
| Monitoring timer | Set the wait time up to the completion of reading and writing processing. | Page 41 Monitoring timer |
| Request data | For the request data, set the command that indicates the request content. Refer to "Request data" rows of each command. | Page 58 COMMANDS AND FUNCTIONS |
| Response data length | The data length from an end code to a response data (at normal completion) or an error information (at abnormal completion) is stored. | Page 41 Request data length and response data length |
| Response data | For the response data, store the read data for the command at normal completion. Refer to "Response data" rows of each command. | Page 58 COMMANDS AND FUNCTIONS |
| End code | The command processing result is stored. | Page 42 End code |
| Error information | Store the information of a station on which an error occurred and information of a command. | Page 42 Error information |

5.3 Details of Setting Data

This section explains how to specify the common data items and their content in each message.

For the setting data with 1E frame, refer to the following section.

☞ Page 385 Detail of Setting Data

Header

A header for TCP/IP and UDP/IP. A header of a request message is added on the external device side and sent. Normally, it is added automatically by an external device. A header for a response message is set automatically by E71.

Subheader

The value to be set according to type of message is defined.

Setting method for 4E frame

Set the fixed value (request message: '5400', response message: 'D400') and a serial No. (0000H to FFFFH).

A serial No. is an arbitrary number that is added on the external device side for message recognition. When a request message is sent with a serial No. added, the same serial No. is added to the response message. Use a serial No. when transmitting more than one request messages from an external device to the same supported device.



Serial No. added on the external device side must be managed at the external device side.

■Data communication using ASCII code

It's 12 bytes in total.

A fixed value is set by 4-digit ASCII code.

For the serial No., convert the numerical value to 4-digit ASCII code (hexadecimal) and send it from upper digits.

The 4-byte '0' (30H) is inserted after the serial No.

■Data communication using binary code

It's 6 bytes in total.

A fixed value is set in 2 bytes.

For the serial No., send 2-byte numerical value from the lower byte (L: bits 0 to 7).

The 2-byte '0' (00H) is inserted after the serial No.

Ex.

Request message (serial No. '1234')

| ASCII code | Binary code | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|-------------|-----|-----|--------|-----|-----|-----|-----|-----|-----|-----|---|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|---------------|--|--|--|--------|--|--|--|--|--|--|--|--|-----|-----|-----|-----|-----|-----|---------------|--|--|--|--------|--|
| <p style="text-align: center;">Serial number</p> <table border="1"><tr><td>5</td><td>4</td><td>0</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>0</td><td>0</td><td>0</td><td>0</td></tr><tr><td>35H</td><td>34H</td><td>30H</td><td>30H</td><td>31H</td><td>32H</td><td>33H</td><td>34H</td><td>30H</td><td>30H</td><td>30H</td><td>30H</td></tr><tr><td>(Fixed value)</td><td></td><td></td><td></td><td>(Free)</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr></table> | 5 | 4 | 0 | 0 | 1 | 2 | 3 | 4 | 0 | 0 | 0 | 0 | 35H | 34H | 30H | 30H | 31H | 32H | 33H | 34H | 30H | 30H | 30H | 30H | (Fixed value) | | | | (Free) | | | | | | | | <p style="text-align: center;">Serial number</p> <table border="1"><tr><td>54H</td><td>00H</td><td>34H</td><td>12H</td><td>00H</td><td>00H</td></tr><tr><td>(Fixed value)</td><td></td><td></td><td></td><td>(Free)</td><td></td></tr></table> | 54H | 00H | 34H | 12H | 00H | 00H | (Fixed value) | | | | (Free) | |
| 5 | 4 | 0 | 0 | 1 | 2 | 3 | 4 | 0 | 0 | 0 | 0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 35H | 34H | 30H | 30H | 31H | 32H | 33H | 34H | 30H | 30H | 30H | 30H | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| (Fixed value) | | | | (Free) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 54H | 00H | 34H | 12H | 00H | 00H | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| (Fixed value) | | | | (Free) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

Setting method for 3E frame

Set the fixed value (request message: '5000', response message: 'D000').

Request message

| ASCII code | Binary code | | | | | | | | | | |
|--|-------------|-----|-----|---|-----|-----|-----|-----|---|-----|-----|
| <table border="1"><tr><td>5</td><td>0</td><td>0</td><td>0</td></tr><tr><td>35H</td><td>30H</td><td>30H</td><td>30H</td></tr></table> | 5 | 0 | 0 | 0 | 35H | 30H | 30H | 30H | <table border="1"><tr><td>50H</td><td>00H</td></tr></table> | 50H | 00H |
| 5 | 0 | 0 | 0 | | | | | | | | |
| 35H | 30H | 30H | 30H | | | | | | | | |
| 50H | 00H | | | | | | | | | | |

Response message

| ASCII code | Binary code |
|----------------------------|-------------|
| D 0 0 0 44H 30H 30H 30H | D0H 00H |

Request data length and response data length

For the request data length, specify the data length from the monitoring timer to the request data.

For the response data length, the data length from an end code to a response data (for normal completion) or an error information (for abnormal completion) is stored.

Setting method

Specify the data length in hexadecimal. (Unit: byte)

■Data communication using ASCII code

Convert the numerical value to 4-digit ASCII code (hexadecimal) and send it from upper digits.

■Data communication using binary code

Send 2-byte numerical value from the lower byte (L: bits 0 to 7).

Ex.

The data length is 24 bytes

| ASCII code | Binary code |
|----------------------------|-------------|
| 0 0 1 8 30H 30H 31H 38H | 18H 00H |

Monitoring timer

Set the wait time up to the completion of reading and writing processing.

Set the wait time from when E71 on the connection station requests processing to the access target to when the response is returned.

- 0000H (0): Wait infinitely (Waits until a processing is completed.)
- 0001H to FFFFH (1 to 65535): Waiting time (unit: 250 ms)

To perform normal data communication, using the timer within the setting range in the table below is recommended depending on the communication destination.

| Access target | Monitoring timer |
|----------------------------------|-------------------------------|
| Connected station (host station) | 0001H to 0028H (0.25s to 10s) |
| Other station | 0002H to 00F0H (0.5s to 60s) |

Setting method

■Data communication using ASCII code

Convert the numerical value to 4-digit ASCII code (hexadecimal) and send it from upper digits.

■Data communication using binary code

Send 2-byte numerical value from the lower byte (L: bits 0 to 7).

Ex.

When specifying 10H (16 × 250 ms = 4 seconds) for the monitoring timer

| ASCII code | Binary code |
|----------------------------|-------------|
| 0 0 1 0 30H 30H 31H 30H | 10H 00H |

End code

The command processing result is stored.

At normal completion, '0' is stored.

At abnormal completion, an error code of the access target is stored.

Error code indicates the content of occurred error.

If more than one error occurs at the same time, the error code detected first is returned.

For the content of error code and its corrective action, refer to the user's manual of the module used.

QCPU User's Manual (Hardware Design, Maintenance and Inspection)

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

Q Corresponding Ethernet Interface Module User's Manual (Basic)

MELSEC-L Ethernet Interface Module User's Manual (Basic)

MELSEC iQ-R Serial Communication Module User's Manual (Application)

Setting method

■Data communication using ASCII code

Convert the numerical value to 4-digit ASCII code (hexadecimal) and send it from upper digits.

■Data communication using binary code

Send 2-byte numerical value from the lower byte (L: bits 0 to 7).

Ex.

Normal completion

| ASCII code | Binary code |
|----------------------------------|-------------|
| 0 0 0 0 30H , 30H , 30H , 30H | 00H , 00H |

Error code C051H

| ASCII code | Binary code |
|----------------------------------|-------------|
| C 0 5 1 43H , 30H , 35H , 31H | 51H , C0H |

Error information

Store the information of a station on which an error occurred and information of a command.

| | | |
|--------------|---------|------------|
| Access route | Command | Subcommand |
|--------------|---------|------------|

- Access route: The information of a station which sent an error response is stored. It may differ from the contents of a request message.
- Command, subcommand: The command and the subcommand when an error occurred are stored.

6 ACCESS ROUTE SETTINGS

This chapter explains the accessible range of each frame of MC protocol and data to specify the access target.

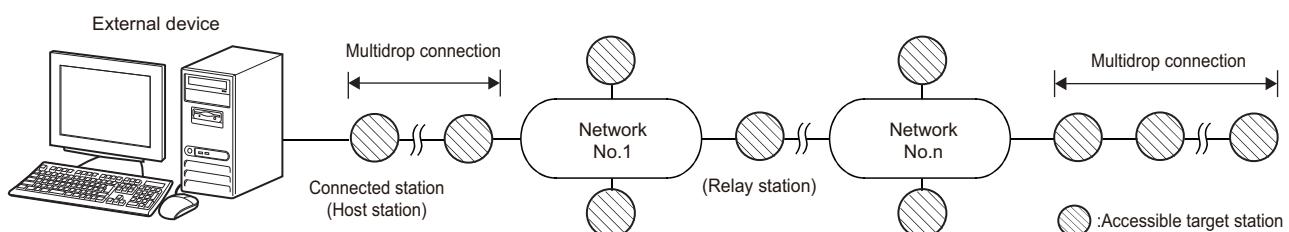
6.1 Accessible Ranges and Settable Data for Each Frame

The accessible range of each frame and the data items to set an access route are as shown below.

4C frame

Accessible range of 4C frame

The following ranges can be accessed.



4C frame is supported by multiple CPU system. ([Page 455 Compatibility with Multiple CPU Systems](#))

Message format (Setting example for accessing connected station (host station))

■Data communication in ASCII code (Format 1 to Format 4)

| Station No. | Network No. | Request destination module I/O No. | Request destination module station No. | Self-station No. |
|-------------|-------------|------------------------------------|--|------------------|
| 0 0 | 0 0 | F F | 0 3 F F | 0 0 0 0 |
| 30H 30H | 30H 30H | 46H 46H | 30H 33H 46H 46H | 30H 30H 30H 30H |

■Data communication in binary code (Format 5)

| Network No. | Request destination module I/O No. | Request destination module station No. |
|-------------|------------------------------------|--|
| Station No. | PC No. | Self-station No. |
| 00H | 00H FFH | FFH 03H 00H 00H |

Data to be set

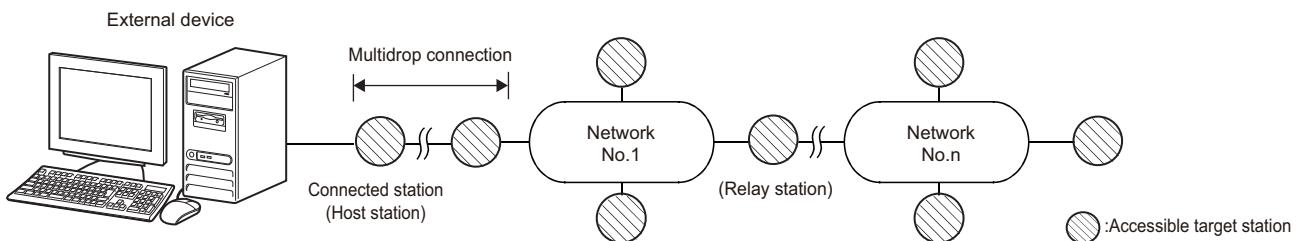
Set the following items.

| Item | Description | Reference |
|--|---|--|
| Station No. | Specify the station to be connected from an external device. | Page 48 Station No. |
| Network No. | Specify the network No. of an access target. | Page 50 Network No., PC No. |
| PC No. | Specify the network module station No. of an access target. | |
| Request destination module I/O No. | <ul style="list-style-type: none">When accessing a multidrop connection station via network, specify the start input/output number of a multidrop connection source module.Specify the CPU module of the multiple CPU system and redundant CPU system. | Page 53 Request destination module I/O No., request destination module station No. |
| Request destination module station No. | When accessing a multidrop connection station via network, specify the station No. of an access target module. | |
| Self-station No. | At the time of m:n multidrop connection, specify the station No. of a request source external device. | Page 56 Self-station No. |

3C frame

Accessible range of 3C frame

The following ranges can be accessed.



Message format (Setting example for accessing connected station (host station))

Station No. Network No. PC No. Self-station No.

| | | | |
|-----|-----|-----|-----|
| 0 0 | 0 0 | F F | 0 0 |
| 30H | 30H | 46H | 30H |

Data to be set

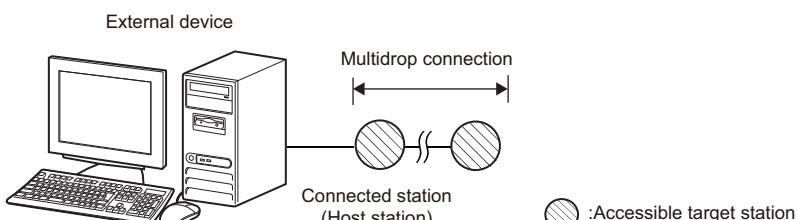
Set the following items.

| Item | Description | Reference |
|------------------|---|-----------------------------|
| Station No. | Specify the station to be connected from an external device. | Page 48 Station No. |
| Network No. | Specify the network No. of an access target. | Page 50 Network No., PC No. |
| PC No. | Specify the network module station No. of an access target. | |
| Self-station No. | At the time of m:n multidrop connection, specify the station No. of a request source external device. | Page 56 Self-station No. |

2C frame

Accessible range of 2C frame

The following ranges can be accessed.



Message format (Setting example for accessing connected station (host station))

Station No. Self-station No.

| | |
|-----|-----|
| 0 0 | 0 0 |
| 30H | 30H |

Data to be set

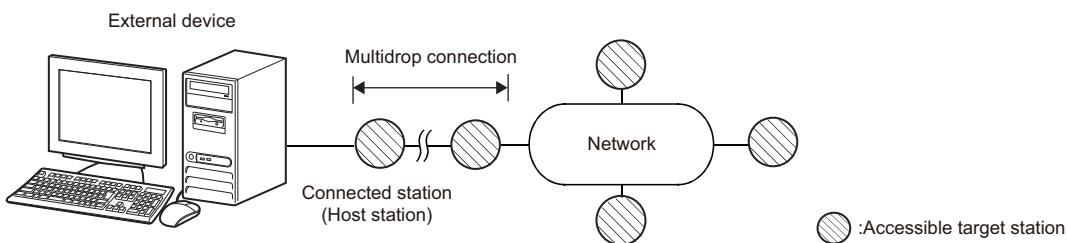
Set the following items.

| Item | Description | Reference |
|------------------|---|--------------------------|
| Station No. | Specify the station to be connected from an external device. | Page 48 Station No. |
| Self-station No. | At the time of m:n multidrop connection, specify the station No. of a request source external device. | Page 56 Self-station No. |

1C frame

Accessible range of 1C frame

The following ranges can be accessed.



When accessing a device, only the applicable device range for MELSEC-A series module can be accessed.

(Page 346 Accessible device range)

Message format (Setting example for accessing connected station (host station))

Station No. PC No.

| | |
|---------|---------|
| 0 0 | F F |
| 30H 30H | 46H 46H |

6

Data to be set

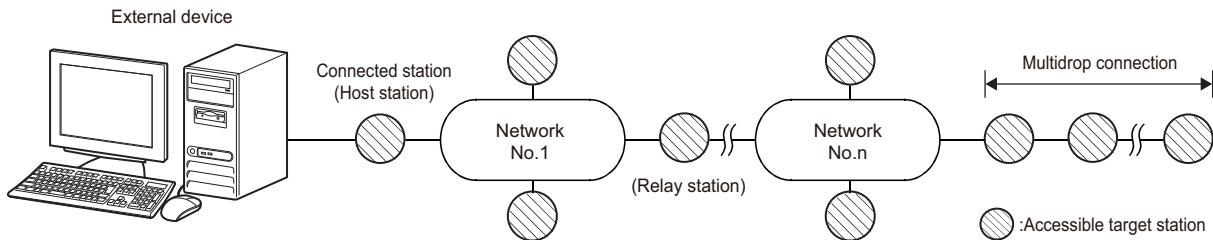
Set the following items.

| Item | Description | Reference |
|-------------|--|-----------------------------|
| Station No. | Specify the station to be connected from an external device. | Page 48 Station No. |
| PC No. | Specify the network module station No. of an access target. | Page 50 Network No., PC No. |

4E frame, 3E frame

Accessible range of 4E frame, 3E frame

The following ranges can be accessed.



4C frame is supported by multiple CPU system. ([Page 455 Compatibility with Multiple CPU Systems](#))

Message format (Setting example for accessing connected station (host station))

■Data communication in ASCII code

| Network No. | Request destination module I/O No. | Request destination module station No. |
|-------------|------------------------------------|--|
| 0 0 | F F | 0 3 F F 0 0 |
| 30H 30H | 46H 46H | 30H 33H 46H 46H 30H 30H |

■Data communication in binary code

| Network No. | Request destination module I/O No. | Request destination module station No. |
|---------------------|------------------------------------|--|
| PC No. | | |
| 00H FFH FFH 03H 00H | | |

Data to be set

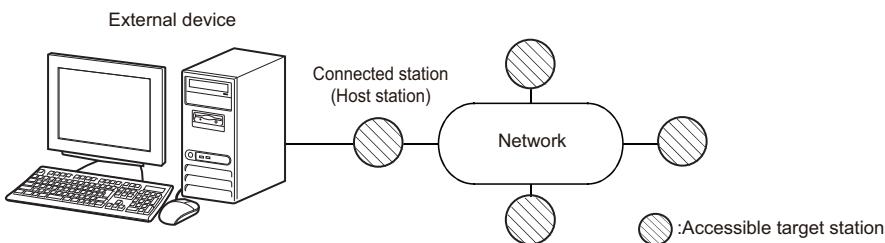
Set the following items.

| Item | Description | Reference |
|--|---|--|
| Network No. | Specify the network No. of an access target. | Page 50 Network No., PC No. |
| PC No. | Specify the network module station No. of an access target. | |
| Request destination module I/O No. | <ul style="list-style-type: none">When accessing a multidrop connection station, specify the start input/output number of a multidrop connection source module.Specify the CPU module of the multiple CPU system and redundant CPU system. | Page 53 Request destination module I/O No., request destination module station No. |
| Request destination module station No. | When accessing a multidrop connection station, specify the station No. of an access target module. | |

1E Frame

Accessible range of 1E frame

The following ranges can be accessed.



When accessing a device, only the applicable device range for MELSEC-A series module can be accessed.

(Page 393 Accessible device range)

Message format (Setting example for accessing connected station (host station))

■Data communication in ASCII code

6

PC No.

| | |
|-----|-----|
| F | F |
| 46H | 46H |

■Data communication in binary code

PC No.

| |
|-----|
| FFH |
|-----|

Data to be set

Set the following items.

| Item | Description | Reference |
|--------|---|-----------------------------|
| PC No. | Specify the network module station No. of an access target. | Page 50 Network No., PC No. |

6.2 Detail of data to be set

This section explains the content and specification method of the data items to set the access route.

○: Necessary, —: Unnecessary

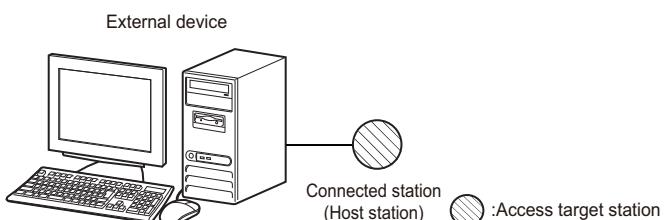
| Item | Frames for C24 | | | | Frames for E71 | | | Reference |
|--|----------------|----|----|----|----------------|----|----|--|
| | 4C | 3C | 2C | 1C | 4E | 3E | 1E | |
| Station No. | ○ | ○ | ○ | ○ | — | — | — | Page 48 Station No. |
| Network No. | ○ | ○ | — | — | ○ | ○ | — | Page 50 Network No., PC No. |
| PC No. | | | | ○ | ○ | ○ | ○ | |
| Request destination module I/O No. | ○ | — | — | — | ○ | ○ | — | Page 53 Request destination module I/O No., request destination module station No. |
| Request destination module station No. | | | | | | | | |
| Self-station No. | ○ | ○ | ○ | — | — | — | — | Page 56 Self-station No. |

Station No.

Specify the station accessed from an external device.

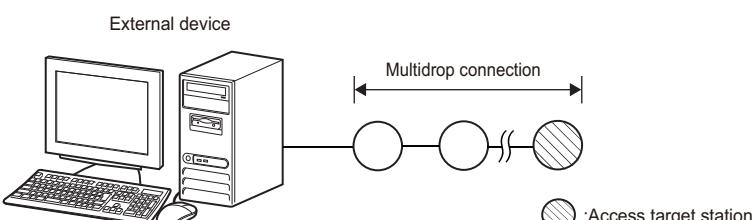
Accessing connected station (host station)

Specify '0' when accessing connected station (host station).



Accessing multidrop connection station

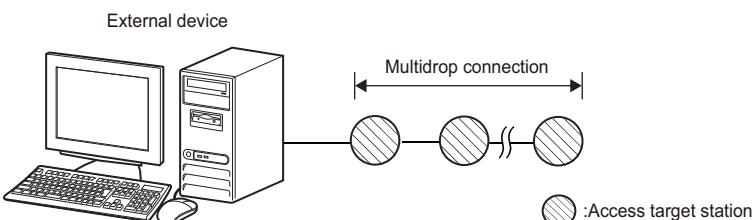
Specify the station No. of an access target station from '0' to '31' (00H to 1FH) when connecting with the multidrop connection.



When accessing all stations connected with the multidrop connection with the global function

Specify 'FF' (FFH) when turning ON/OFF the global signal to all station connected with the multidrop connection using the global function. By specifying 0 to 31 (00H to 1FH), X1A/X1B turns ON only on the specified station, and does not turn ON on the other stations.

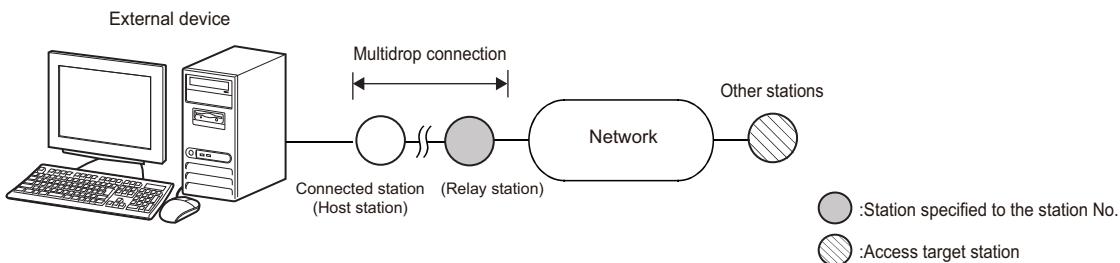
(Page 249 Global Function)



Accessing other stations via network

Specify the station No. from 0 to 31 (00H to 1FH) of a station that relays multidrop connection and network when accessing other stations via network.

Specify '0' when accessing other stations via network without the multidrop connection.



Setting method

■Data communication in ASCII code

Convert the numerical value to 2-digit ASCII code (hexadecimal), and send it from the upper digits.

■Data communication in binary code

Send 1-byte numerical value.

6

Ex.

When the station No. setting for C24 to be accessed is '5'

| ASCII code | Binary code |
|----------------|-------------|
| 0 5 30H 35H | 05H |

When accessing all station connected with the multidrop connection using the global function

| ASCII code | Binary code |
|----------------|-------------|
| F F 46H 46H | FFH |



The station No. of the serial communication module can be checked by using the following parameters of Engineering tool.

- GX Works2: "Station Number Setting" in "Switch Setting"
- GX Works3: "Station Number Settings" in "Module Parameter"

Network No., PC No.

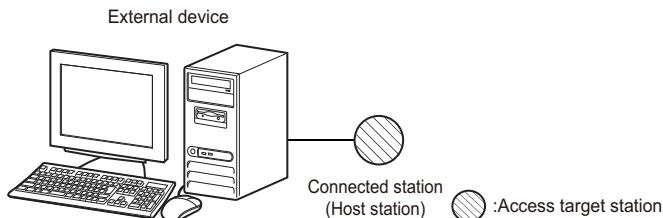
Specify the network No. and station No. that are set with the parameters for the access target network module.
Specify a fixed value when accessing the connection station.

Point

Specify the network No. with the value shown below.
Specifying improper value may result in no response returned.

Accessing connected station (host station)

Specify '0' for the network No., and 'FF' for the PC No.



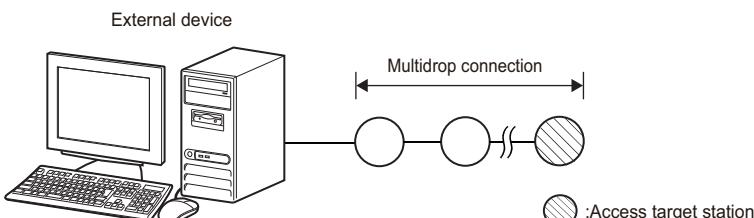
When using the on-demand function

Specify '0' for the network No., and 'FE' for the PC No.

(Page 274 On-demand function)

Accessing multidrop connection station

Specify '0' for the network No., and 'FF' for the PC No.



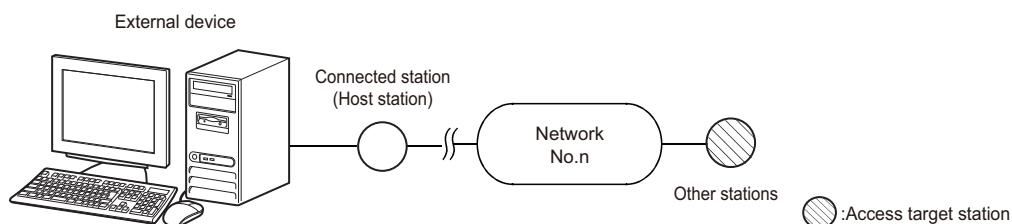
Accessing other stations via network

Specify the network No. and station No. of an access target.

| Access target | Network No. | PC No. |
|--|--|-----------------------|
| Other station of which station No. is set | 01H to EFH (1 to 239) Stations with network No.240 to 255 are not accessible. | 01H to 78H (1 to 120) |
| Specified control station/master station ^{*1} | | 7DH |
| Current control station/Master station ^{*2} | | 7EH |

*1 Access the station set as a control station/master station by parameters.

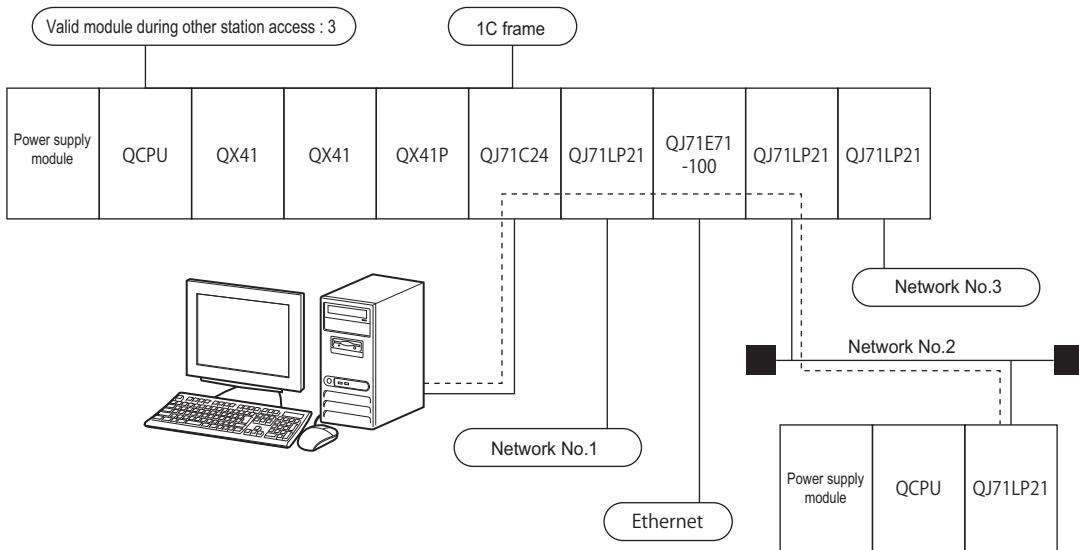
*2 Access the station which is operating as a control station/master station.



■When accessing with the "Valid Module During Other Station Access" setting

1C frame and 1E frame do not have the setting of network No.

When specifying the network of access target is required because more than one network module is mounted on the connection station, set the "Valid Module During Other Station Access" with Engineering tool.



6

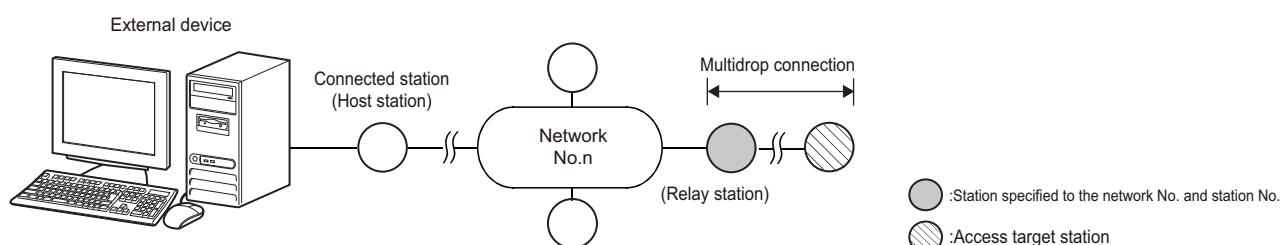
When accessing in accordance with the setting of "Valid Module During Other Station Access" using a frame with the network No. set is desired, specify 'FEH' (254) to the network No.

When accessing other station via C24/E71 mounted on MELSECNET/H remote I/O station, the access to the other station specified with the PC No. of MELSECNET/H remote I/O station is available by specifying 'FEH' to the network No.

Accessing multidrop connection station via network

Specify the network No. and station No. of a station relaying the network routed through and multidrop connection station.

| Access target | Network No. | PC No. |
|--|--|-----------------------|
| Multidrop connection station via network | 01H to EFH (1 to 239) Stations with network No.240 to 255 are not accessible. | 01H to 78H (1 to 120) |



Setting method

■Data communication in ASCII code

Convert the numerical value to 2-digit ASCII code (hexadecimal), and send it from the upper digits.

■Data communication in binary code

Send 1-byte numerical value.

Ex.

Accessing connected station (host station) or multidrop connection station

| ASCII code | Binary code | | | | | | |
|--|-------------|-----|---------|---------|---|-----|-----|
| Network No. PC No. <table border="1"><tr><td>0 0</td><td>F F</td></tr><tr><td>30H 30H</td><td>46H 46H</td></tr></table> | 0 0 | F F | 30H 30H | 46H 46H | Network No. PC No. <table border="1"><tr><td>00H</td><td>FFH</td></tr></table> | 00H | FFH |
| 0 0 | F F | | | | | | |
| 30H 30H | 46H 46H | | | | | | |
| 00H | FFH | | | | | | |

When accessing other station of which network No. is '2' and station No. is '3'

| ASCII code | Binary code | | | | | | |
|--|-------------|-----|---------|---------|---|-----|-----|
| Network No. PC No. <table border="1"><tr><td>0 2</td><td>0 3</td></tr><tr><td>30H 32H</td><td>30H 33H</td></tr></table> | 0 2 | 0 3 | 30H 32H | 30H 33H | Network No. PC No. <table border="1"><tr><td>02H</td><td>03H</td></tr></table> | 02H | 03H |
| 0 2 | 0 3 | | | | | | |
| 30H 32H | 30H 33H | | | | | | |
| 02H | 03H | | | | | | |



The network No. and station No. of the network module can be checked by using the following parameters of Engineering tool.

- GX Works2: "Network Parameter"
- GX Works3: "Module Parameter"

The network No. and the station No. of the network module are normally set in decimal. However, the network No. and the PC No. are set in hexadecimal.

Request destination module I/O No., request destination module station No.

Specify these numbers when an access target is as shown below.

- Multidrop connection station
- CPU module on multiple CPU system
- CPU module on redundant system

Specify the fixed value when the access target is other than those listed above.

| Request destination module I/O No. | Request destination module station No. |
|------------------------------------|--|
| 03FFH | 00H |

Accessing multidrop connection station

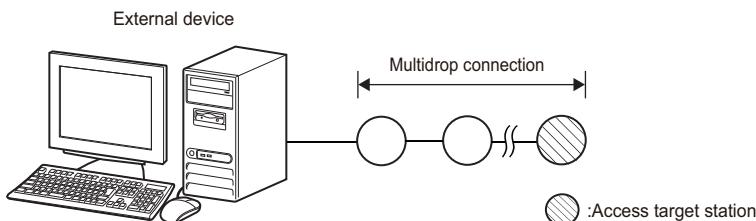
When connecting to an access target with a direct multidrop connection, it can be accessed by specifying a station No.

(Page 48 Station No.).

For the request destination module I/O No. and the request destination module station No., specify the fixed value.

| Request destination module I/O No. | Request destination module station No. |
|------------------------------------|--|
| 03FFH | 00H |

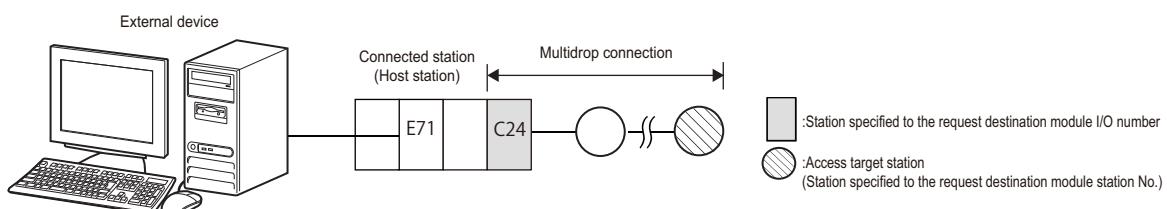
6



■For 4E frame and 3E frame

When accessing a multidrop connection station with the frames (4E frame, 3E frame) for Ethernet interface module, specify the start input/output number of a multidrop connection source module (relay station) and the station No. of an access target module.

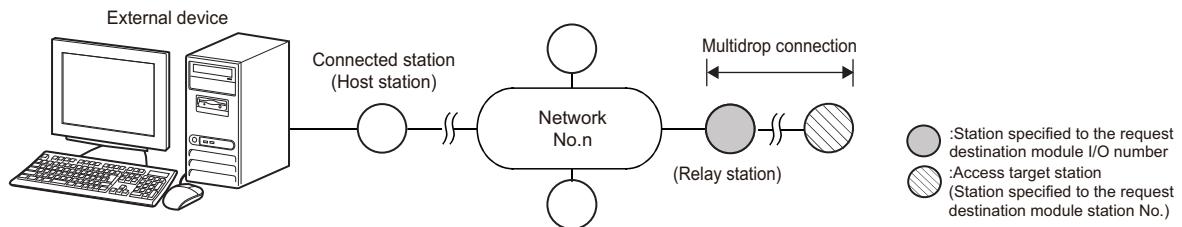
| Request destination module I/O No. | Request destination module station No. |
|---|--|
| 0000H to 01FFH: Values obtained by dividing the start input/output number by 16 | 00H to 1FH (0 to 31): Station No. |



Accessing multidrop connection station via network

Specify the start input/output number of a multidrop connection source module (relay station) and the station No. of an access target.

| Request destination module I/O No. | Request destination module station No. |
|---|--|
| 0000H to 01FFH: Values obtained by dividing the start input/output number by 16 | 00H to 1FH (0 to 31): Station No. |



Accessing multiple CPU system, redundant system

Specify the access target CPU module with the request destination module I/O No. Specify the fixed value (00H) for the station No.

| Access target CPU module | Request destination module I/O No. | Request destination module station No. |
|--------------------------|------------------------------------|--|
| Multiple CPU system | Control CPU | 03FFH |
| | Non-control CPU | 03E0H |
| | | 03E1H |
| | | 03E2H |
| | | 03E3H |
| Redundant system | Control system CPU | 03D0H |
| | Standby system CPU | 03D1H |
| | System A CPU | 03D2H |
| | System B CPU | 03D3H |

Setting method

For the request destination module I/O No., specify the value obtained by dividing the start input/output number assigned to the module by 16 in 4 digits (hexadecimal).

■Data communication in ASCII code

For the request destination module I/O No., convert the numerical value to 4-digit ASCII code (hexadecimal) and send it from the upper digits.

For the request destination module station No., convert the numerical value to 2-digit ASCII code (hexadecimal), and send it from the upper digits.

■Data communication in binary code

For the request destination module I/O No., the 2-byte value is sent from the lower byte (L: bit 0 to 7).

For the request destination module station No., the 1-byte value is sent.

Ex.

Accessing connected station (host station)

| ASCII code | Binary code | | | | | | | | |
|---|-------------|-----|-----------------|---------|---|-----|-----|-----|-----|
| Request destination module I/O No. <table border="1"> <tr> <td>0 3 F F</td> <td>0 0</td> </tr> <tr> <td>30H 33H 46H 46H</td> <td>30H 30H</td> </tr> </table> | 0 3 F F | 0 0 | 30H 33H 46H 46H | 30H 30H | Request destination module I/O No. <table border="1"> <tr> <td>FFH</td> <td>03H</td> </tr> <tr> <td>00H</td> <td>00H</td> </tr> </table> | FFH | 03H | 00H | 00H |
| 0 3 F F | 0 0 | | | | | | | | |
| 30H 33H 46H 46H | 30H 30H | | | | | | | | |
| FFH | 03H | | | | | | | | |
| 00H | 00H | | | | | | | | |

Accessing multidrop connection station via network

- Start input/output number: 0080H (input/output signal: 0080H to 009FH)
- Request destination module I/O No.: 0008H
- Station No.: 5

| ASCII code | Binary code | | | | | | | | |
|---|-------------|-----|-----------------|---------|---|-----|-----|-----|-----|
| Request destination module I/O No. <table border="1"> <tr> <td>0 0 0 8</td> <td>0 5</td> </tr> <tr> <td>30H 30H 30H 38H</td> <td>30H 35H</td> </tr> </table> | 0 0 0 8 | 0 5 | 30H 30H 30H 38H | 30H 35H | Request destination module I/O No. <table border="1"> <tr> <td>08H</td> <td>00H</td> </tr> <tr> <td>05H</td> <td>00H</td> </tr> </table> | 08H | 00H | 05H | 00H |
| 0 0 0 8 | 0 5 | | | | | | | | |
| 30H 30H 30H 38H | 30H 35H | | | | | | | | |
| 08H | 00H | | | | | | | | |
| 05H | 00H | | | | | | | | |



The station No. of the serial communication module can be checked by using the following parameters of Engineering tool.

- GX Works2: "Station Number Setting" in "Switch Setting"
- GX Works3: "Station Number Settings" in "Module Parameter"

When accessing non-control CPU (multiple CPU No.2) on multiple CPU system

| ASCII code | Binary code | | | | | | | | |
|---|-------------|-----|-----------------|---------|---|-----|-----|-----|-----|
| Request destination module I/O No. <table border="1"> <tr> <td>0 3 E 1</td> <td>0 0</td> </tr> <tr> <td>30H 33H 45H 31H</td> <td>30H 30H</td> </tr> </table> | 0 3 E 1 | 0 0 | 30H 33H 45H 31H | 30H 30H | Request destination module I/O No. <table border="1"> <tr> <td>E1H</td> <td>03H</td> </tr> <tr> <td>00H</td> <td>00H</td> </tr> </table> | E1H | 03H | 00H | 00H |
| 0 3 E 1 | 0 0 | | | | | | | | |
| 30H 33H 45H 31H | 30H 30H | | | | | | | | |
| E1H | 03H | | | | | | | | |
| 00H | 00H | | | | | | | | |

Self-station No.

Specify this when more than one external device (m stations) and more than one C24s (n stations) are connected with the multidrop connection.

Specify the fixed value (00H) for any cases other than multidrop connection in a m:n basis.

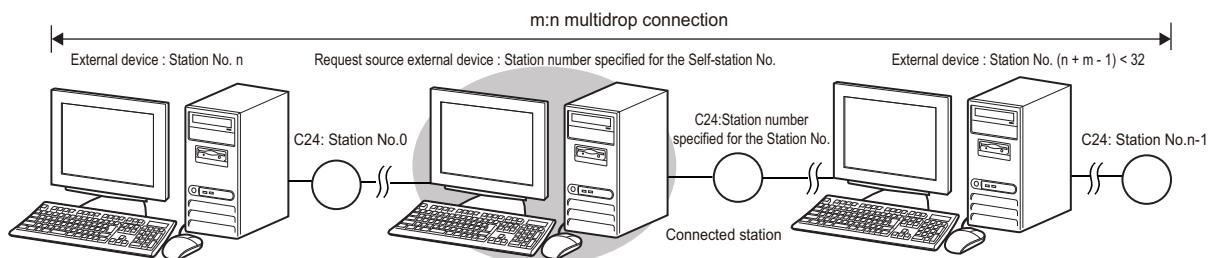
When external devices are connected with the m:n multidrop connection

Specify the station No. of request source external device, 0 to 31 (00H to 1FH).

For the station No. (m stations) of external devices, the value which is not set to C24 (n stations) of the multidrop connection is used.

(The total of 'm' and 'n' is up to 32 stations.)

- Station No. of a request source external device: Specify it to the self-station No.
- Station No. of the connected station C24: Specify it to the station No. ([Page 48 Station No.](#))



Setting method

■Data communication in ASCII code

Convert the numerical value to 2-digit ASCII code (hexadecimal), and send it from the upper digits.

■Data communication in binary code

Send 1-byte numerical value.

Ex.

For connection other than m:n multidrop connection

| ASCII code | Binary code |
|----------------|-------------|
| 0 0 30H 30H | 00H |

For accessing with the m:n multidrop connection

Station No. assigned to a request source external device: 31 (1FH)

| ASCII code | Binary code |
|----------------|-------------|
| 1 F 31H 46H | 1FH |

PART 3 COMMAND

This part explains the functions that can be specified by a message of MC protocol and the message format of request data and message data of each command.

[7 COMMANDS AND FUNCTIONS](#)

[8 DEVICE ACCESS](#)

[9 LABEL ACCESS](#)

[10 BUFFER MEMORY ACCESS](#)

[11 CONTROL MODULE OPERATION](#)

[12 FILE CONTROL](#)

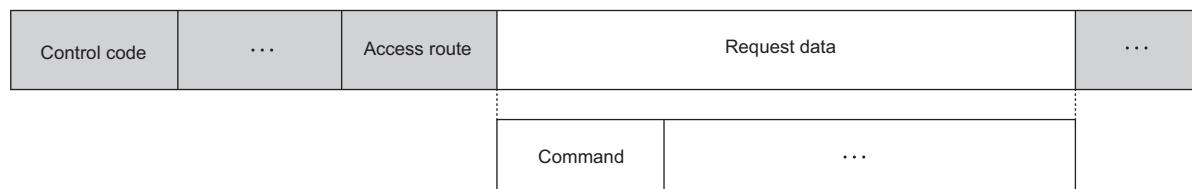
[13 SERIAL COMMUNICATION MODULE DEDICATED COMMANDS](#)

7 COMMANDS AND FUNCTIONS

This chapter explains the commands of MC protocol.

The functions of a message is defined by each command. The message format for request data and response data varies with commands. Depending on the type of frame to be used, the specific value is assigned to a command. The value of command is specified at the head of a request data.

Request message



Response message



The explanation of each command in Part 3, the message format of request data and response data are explained.

For the message formats other than request data and response data, refer to the following sections.

☞ Page 26 MESSAGES OF SERIAL COMMUNICATION MODULE

☞ Page 37 MESSAGES OF ETHERNET INTERFACE MODULE

7.1 Command List

The following shows the list of commands.



There are some commands that cannot be executed while the CPU module is in RUN. Refer to the following section.

☞ Page 457 Applicable Commands for Online Program Change



For details on the number of points processed per communication and modules can be accessed by each command, refer to the following sections.

☞ Page 459 Number of Processing per One Communication

☞ Page 465 Accessible Modules for Each Command

Commands for 4C/3C/4E/3E frame

The following shows the commands for 4C/3C/4E/3E frame.

For 4C/3C/4E/3E frame, specify subcommands in the request message as well.

Device access

| Function | Command | Description | Subcommand | | |
|--------------------------------------|-----------------------------------|-------------|--|---|--|
| Batch read and write | Batch read in word units | 0401 | Read values from devices in word units. Read the values in batch with specifying the consecutive device points. | 0000 0080 0002 0082 | |
| | | | | For MELSEC-Q/L series For MELSEC iQ-R series | |
| | Batch read in bit units | | Read values from devices in bit units. Read the values in batch with specifying the consecutive device points. | 0001 0081 0003 0083 | |
| | | | | For MELSEC-Q/L series For MELSEC iQ-R series | |
| | Batch write in word units | 1401 | Write values to devices in word units. Write the consecutive devices in batch with specifying the consecutive device points. | 0000 0080 0002 0082 | |
| | | | | For MELSEC-Q/L series For MELSEC iQ-R series | |
| | Batch write in bit units | | Write values to devices in bit units. Write the consecutive devices in batch with specifying the consecutive device points. | 0001 0081 0003 0083 | |
| | | | | For MELSEC-Q/L series For MELSEC iQ-R series | |
| Random read and write | Random read in word units | 0403 | Read values from devices in word or double-word units. Read device values with specifying device numbers. Discontinuous device numbers can be specified. | 0000 0080 0002 0082 | |
| | | | | For MELSEC-Q/L series For MELSEC iQ-R series | |
| | Random write in word units (test) | 1402 | Read values randomly in word units by specifying the monitor condition. The read timing can be changed. | 0040 00C0 | |
| | | | | Monitor condition specified | |
| | Random write in bit units (test) | | Write values to devices in word or double-word units. Write device values with specifying device numbers. Discontinuous device numbers can be specified. | 0000 0080 0002 0082 | |
| | | | | For MELSEC-Q/L series For MELSEC iQ-R series | |
| Batch read and write multiple blocks | Batch read multiple blocks | 0406 | Read values for specified multiple blocks by handling consecutive word devices or bit devices as one block. Each block can be specified with discontinuous device numbers. | 0000 0080 0002 0082 | |
| | Batch write multiple blocks | 1406 | Write values for specified multiple blocks by handling consecutive word devices or bit devices as one block. Each block can be specified with discontinuous device numbers. | 0000 0080 0002 0082 | |
| | Register monitor data | 0801 | Register the devices to be monitored. | 0000 0080 0002 0082 | |
| | | | Perform monitor data registration by specifying monitor conditions. The read timing can be changed. | 0040 00C0 | |
| Monitor device memory | Monitor | 0802 | Read the values of registered devices. | 0000 | |
| | | | | — | |

Label access

| Function | | Command | Description | Subcommand | |
|-----------------------|-------------------------------|---------|---|------------|------------------------|
| Batch read and write | Batch read array type labels | 041A | Read the values from array type labels. Read the values in batch with specifying the consecutive array elements. Specify the array type labels or array type elements of structure type labels. | 0000 | For MELSEC iQ-R series |
| | Batch write array type labels | 141A | Write the values to array type labels. Write the values in batch with specifying the consecutive array elements. Specify the array type labels or array type elements of structure type labels. | 0000 | For MELSEC iQ-R series |
| Random read and write | Random read labels | 041C | Read values with specifying multiple labels. | 0000 | For MELSEC iQ-R series |
| | Random write labels | 141B | Write values with specifying multiple labels. | 0000 | For MELSEC iQ-R series |

Buffer memory access

| Function | | Command | Description | Subcommand | |
|-----------------------------|-------------|---------|--|------------|---|
| Buffer memory | Batch read | 0613 | Read data from the buffer memory of the host station (supported device). | 0000 | — |
| | Batch write | 1613 | Write data to the buffer memory of the host station (supported device). | 0000 | — |
| Intelligent function module | Batch read | 0601 | Read intelligent function module buffer memory data. | 0000 | — |
| | Batch write | 1601 | Write intelligent function module buffer memory data. | 0000 | — |

Module control

| Function | | Command | Description | Subcommand | |
|-------------------------|---|---------|---|------------|---|
| Remote control | Remote RUN | 1001 | Perform remote RUN to the access target module. | 0000 | — |
| | Remote STOP | 1002 | Perform remote STOP to the access target module. | 0000 | — |
| | Remote PAUSE | 1003 | Perform remote PAUSE to the access target module. | 0000 | — |
| | Remote latch clear | 1005 | Perform remote latch clear to the access target module. | 0000 | — |
| | Remote RESET | 1006 | Perform remote RESET to the access target module. | 0000 | — |
| | Read CPU model name | 0101 | Read model name and model code from the access target module. | 0000 | — |
| Remote password | Unlock | 1630 | Specify a remote password to enable communications with other devices. (Change a device from the locked state to unlocked state.) | 0000 | — |
| | Lock | 1631 | Specify a remote password to disable communications with other devices. (Change a device from the unlocked state to locked state.) | 0000 | — |
| Loopback test | | 0619 | Test to check whether communications between external device and connection station operate normally. | 0000 | — |
| Clear error information | Turn indicator LED OFF, initialize communication error information/error code | 1617 | Turn OFF the error LED, and initialize communication error information and error code. | 000□ | — |

File control

| Function | Command | Description | Subcommand | |
|------------------------------------|-----------------------------------|--|--------------------------------|--------------------------------|
| File check | Read directory/file information | 1810 For the specified storage destination file, read the file name, file creation date and time (last edit date and time) etc. | 0000 For MELSEC-Q/L series | 0040 For MELSEC iQ-R series |
| | Search directory/file information | 1811 Read the file No. of the specified file. | 0000 For MELSEC-Q/L series | 0040 For MELSEC iQ-R series |
| File creation and deletion | Create new file | 1820 Create a file with specifying its size. (Reserve a storage area for the specified file.) | 0000 For MELSEC-Q/L series | 0040 For MELSEC iQ-R series |
| | Delete file | 1822 Delete a file. | 0000 For MELSEC-Q series | 0004 For MELSEC-L series |
| Copy file | 1824 | Copy a file. | 0000 For MELSEC-Q series | 0040 For MELSEC iQ-R series |
| | | | 0004 For MELSEC-L series | |
| File modification | Modify file attribute | 1825 Change the file attribute. | 0000 For MELSEC-Q series | 0040 For MELSEC-L series |
| | | | 0040 For MELSEC iQ-R series | |
| Modify file creation date and time | 1826 | Modify the file creation date and time (last edit date and time). | 0000 For MELSEC-Q/L series | 0040 For MELSEC iQ-R series |
| | | | 0040 For MELSEC iQ-R series | |
| Open file | 1827 | Open a file to lock it so that the contents of the file cannot be modified by other devices. | 0000 For MELSEC-Q series | 0004 For MELSEC-L series |
| | | | 0040 For MELSEC iQ-R series | |
| Read file | 1828 | Read a file content. | 0000 | — |
| Write to file | 1829 | Write content to a file. | 0000 | — |
| Close file | 182A | Close a file to unlock it by opening file (command: 1827). | 0000 | — |



For the QnACPU dedicated commands, refer to the following section.

☞ Page 282 QnACPU Dedicated Commands List

Serial communication dedicated commands

| Function | | Command | Description | | | Subcommand | |
|--|------------------------|---------|--|--|--------------|------------|--|
| User frame | Read registered data | 0610 | Read the contents of registered user frames. | | 0000 | — | |
| | Register data | 1610 | Register user frames to C24. | | 0000 | — | |
| | Delete registered data | | Delete the registered user frame. | | 0001 | — | |
| Global | | 1618 | Turn ON/OFF the global signal (X1A/X1B). This can be performed to a connected station and multidrop connection station. | | 0000 0001 | — | |
| Initialize transmission sequence | | 1615 | Terminate a current processing request, and place C24 into the wait state to receive commands. This can be performed to a connected station and multidrop connection station. | | 0000 | — | |
| Switch mode | | 1612 | Switch the operation mode and transmission specifications of the specified interface. This can be performed to a connected station and multidrop connection station. | | 0000 | — | |
| Programmable controller CPU monitoring | Register | 0630 | Register the conditions to monitor and start the programmable controller CPU monitoring. | | 0000 | — | |
| | Deregister | 0631 | End the programmable controller CPU monitoring. | | 0000 | — | |
| On-demand | | 2101 | Issue a transmission request to C24 from CPU module, and transmit data to external devices. | | — | — | |

Commands for 2C frame

The following shows the commands for 2C frame.

The commands for 4C/3C frame are equivalent to the following device access commands and subcommands.

| Function | 2C frame command | 4C/3C frame | | Description |
|-----------------------|-----------------------------------|-------------|------------|--|
| | | command | subcommand | |
| Batch read and write | Batch read in bit units | 1 | 0401 | 0001 Read values from devices in bit units. Read the values in batch with specifying the consecutive device points. |
| | Batch read in word units | | | 0000 Read values from devices in word units. Read the values in batch with specifying the consecutive device points. |
| | Batch write in bit units | 3 | 1401 | 0001 Write values to devices in bit units. Write the consecutive devices in batch with specifying the consecutive device points. |
| | Batch write in word units | | | 0000 Write values to devices in word units. Write the consecutive devices in batch with specifying the consecutive device points. |
| Random read and write | Random read in word units | 5 | 0403 | 0000 Read values from devices in word or double-word units. Read device values with specifying device numbers. Discontinuous device numbers can be specified. |
| | Random write in bit units (test) | | | 0001 Write values to devices in bit units. Write device values with specifying device numbers. Discontinuous device numbers can be specified. |
| | Random write in word units (test) | 6 | 1402 | 0000 Write values to devices in word or double-word units. Write device values with specifying device numbers. Discontinuous device numbers can be specified. |
| Monitor device memory | Register monitor data | 8 | | 0000 Register the devices to be monitored. |
| | Monitor | 9 | 0802 | 0000 Read the values of registered devices. |

Commands for 1C/1E frame

For the commands for 1C/1E frame, refer to the following sections.

- ☞ Page 343 Command and Function Lists for 1C Frame
- ☞ Page 390 Commands and Function List for 1E Frame

8 DEVICE ACCESS

This chapter explains the commands to read and write devices.

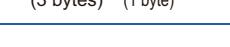
8.1 Data to be Specified in Commands

This section explains the contents and specification methods for data items which are set in each command related to device access.

Devices

Specify the device to be accessed by a device code and a device number.

- The data order differs between ASCII code or binary code.
 - The data size to be set differs between MELSEC-Q/L series subcommands (subcommand: 0000, 0001) and MELSEC iQ-R series subcommands (subcommand: 0002, 0003).

| Subcommand type | ASCII code | Binary code |
|------------------------|---|--|
| For MELSEC-Q/L series | Device code Device number  (2 digits) (6 digits) | Device number Device code  (3 bytes) (1 byte) |
| For MELSEC iQ-R series | Device code Device number  (4 digits) (8 digits) | Device number Device code  (4 bytes) (2 bytes) |



When accessing any of the following devices, use the device extension specification (subcommand: 008□).

- Link direct device
 - Module access device
 - CPU buffer memory access device

For the message format for device extension specification, refer to the following section.

☞ Page 432 Read/Write by Device Extension Specification

Device codes

Specify the device name to be accessed.

Specify the device within the range of the access target module.

For the values of each device code, refer to the following section.

☞ Page 66 Device code list

■Data communication using ASCII code

Convert the numerical value to 2-digit or 4-digit ASCII code (hexadecimal), and send it from the upper digits.

- For MELSEC-Q/L series: 2-digit ASCII code
- For MELSEC iQ-R series: 4-digit ASCII code

The '*' in a device code can also be specified with a space (code: 20H).

■Data communication using binary code

Send the 1-byte or 2-byte numerical value from the lower byte (L: bits 0 to 7).

- For MELSEC-Q/L series: 1 byte
- For MELSEC iQ-R series: 2 bytes

Ex.

For input (X)

| Subcommand type | ASCII code | Binary code | | | | | | | | | | |
|------------------------|--|-------------|-----|-----|-----|---|-----|-----|-----|---|-----|-----|
| For MELSEC-Q/L series | <table border="1"><tr><td>X</td><td>*</td></tr><tr><td>58H</td><td>2AH</td></tr></table> | X | * | 58H | 2AH | <table border="1"><tr><td>9CH</td></tr></table> | 9CH | | | | | |
| X | * | | | | | | | | | | | |
| 58H | 2AH | | | | | | | | | | | |
| 9CH | | | | | | | | | | | | |
| For MELSEC iQ-R series | <table border="1"><tr><td>X</td><td>*</td><td>*</td><td>*</td></tr><tr><td>58H</td><td>2AH</td><td>2AH</td><td>2AH</td></tr></table> | X | * | * | * | 58H | 2AH | 2AH | 2AH | <table border="1"><tr><td>9CH</td><td>00H</td></tr></table> | 9CH | 00H |
| X | * | * | * | | | | | | | | | |
| 58H | 2AH | 2AH | 2AH | | | | | | | | | |
| 9CH | 00H | | | | | | | | | | | |

Device number

Specify the number of device to be accessed.

Specify the device number within the range of the access target module.

■Data communication using ASCII code

Convert the numerical value to 6-digit or 8-digit ASCII code, and sent it from the upper digits.

Specify the device number in decimal or hexadecimal, depending on the device type. (☞ Page 66 Device code list)

- For MELSEC-Q/L series: 6-digit ASCII code
- For MELSEC iQ-R series: 8-digit ASCII code (10 digits at device extension specification)

The '0' in the upper digits can also be specified with a space (code: 20H).

■Data communication using binary code

Send the 3-byte or 4-byte numerical value in order from the lower byte (L: bit 0 to 7).

For a device of which device number is in decimal, convert it to hexadecimal and specify.

- For MELSEC-Q/L series: 3 bytes^{*1}
- For MELSEC iQ-R series: 4 bytes^{*1}

^{*1} For C24, the additional code may be added. (☞ Page 33 Additional code (10H))

Ex.

For input (X) 1234 (a device of which device number is in hexadecimal)

| Subcommand type | ASCII code | Binary code | | | | | | | | | | | | | | | | | | | | |
|------------------------|---|-------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|---|-----|-----|-----|---|-----|-----|-----|-----|
| For MELSEC-Q/L series | <table border="1"> <tr> <td>0</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td> </tr> <tr> <td>30H</td><td>30H</td><td>31H</td><td>32H</td><td>33H</td><td>34H</td> </tr> </table> | 0 | 0 | 1 | 2 | 3 | 4 | 30H | 30H | 31H | 32H | 33H | 34H | <table border="1"> <tr> <td>34H</td><td>12H</td><td>00H</td> </tr> </table> | 34H | 12H | 00H | | | | | |
| 0 | 0 | 1 | 2 | 3 | 4 | | | | | | | | | | | | | | | | | |
| 30H | 30H | 31H | 32H | 33H | 34H | | | | | | | | | | | | | | | | | |
| 34H | 12H | 00H | | | | | | | | | | | | | | | | | | | | |
| For MELSEC iQ-R series | <table border="1"> <tr> <td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td> </tr> <tr> <td>30H</td><td>30H</td><td>30H</td><td>30H</td><td>31H</td><td>32H</td><td>33H</td><td>34H</td> </tr> </table> | 0 | 0 | 0 | 0 | 1 | 2 | 3 | 4 | 30H | 30H | 30H | 30H | 31H | 32H | 33H | 34H | <table border="1"> <tr> <td>34H</td><td>12H</td><td>00H</td><td>00H</td> </tr> </table> | 34H | 12H | 00H | 00H |
| 0 | 0 | 0 | 0 | 1 | 2 | 3 | 4 | | | | | | | | | | | | | | | |
| 30H | 30H | 30H | 30H | 31H | 32H | 33H | 34H | | | | | | | | | | | | | | | |
| 34H | 12H | 00H | 00H | | | | | | | | | | | | | | | | | | | |

For internal relay (M) 1234 (a device of which device number is in decimal)

For binary code, convert the device number to hexadecimal. '1234' (decimal) → '4D2' (hexadecimal)

| Subcommand type | ASCII code | Binary code | | | | | | | | | | | | | | | | | | | | |
|------------------------|---|-------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|---|-----|-----|-----|---|-----|-----|-----|-----|
| For MELSEC-Q/L series | <table border="1"> <tr> <td>0</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td> </tr> <tr> <td>30H</td><td>30H</td><td>31H</td><td>32H</td><td>33H</td><td>34H</td> </tr> </table> | 0 | 0 | 1 | 2 | 3 | 4 | 30H | 30H | 31H | 32H | 33H | 34H | <table border="1"> <tr> <td>D2H</td><td>04H</td><td>00H</td> </tr> </table> | D2H | 04H | 00H | | | | | |
| 0 | 0 | 1 | 2 | 3 | 4 | | | | | | | | | | | | | | | | | |
| 30H | 30H | 31H | 32H | 33H | 34H | | | | | | | | | | | | | | | | | |
| D2H | 04H | 00H | | | | | | | | | | | | | | | | | | | | |
| For MELSEC iQ-R series | <table border="1"> <tr> <td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td> </tr> <tr> <td>30H</td><td>30H</td><td>30H</td><td>30H</td><td>31H</td><td>32H</td><td>33H</td><td>34H</td> </tr> </table> | 0 | 0 | 0 | 0 | 1 | 2 | 3 | 4 | 30H | 30H | 30H | 30H | 31H | 32H | 33H | 34H | <table border="1"> <tr> <td>D2H</td><td>04H</td><td>00H</td><td>00H</td> </tr> </table> | D2H | 04H | 00H | 00H |
| 0 | 0 | 0 | 0 | 1 | 2 | 3 | 4 | | | | | | | | | | | | | | | |
| 30H | 30H | 30H | 30H | 31H | 32H | 33H | 34H | | | | | | | | | | | | | | | |
| D2H | 04H | 00H | 00H | | | | | | | | | | | | | | | | | | | |

For internal relay (M) 16 (with additional code)

For C24 binary code, specify '10H' as '10H + 10H'. (☞ Page 33 Additional code (10H))

| Subcommand type | Binary code (For C24) | Binary code (For E71) | | | | | | | | | |
|------------------------|---|-----------------------|-----|-----|-----|---|--|-----|-----|-----|-----|
| For MELSEC-Q/L series | <table border="1"> <tr> <td>DLE</td> <td>10H</td> <td>00H</td> <td>00H</td> </tr> </table> | DLE | 10H | 00H | 00H | <table border="1"> <tr> <td>10H</td> <td>00H</td> <td>00H</td> </tr> </table> | 10H | 00H | 00H | | |
| DLE | 10H | 00H | 00H | | | | | | | | |
| 10H | 00H | 00H | | | | | | | | | |
| For MELSEC iQ-R series | <table border="1"> <tr> <td>DLE</td> <td>10H</td> <td>00H</td> <td>00H</td> <td>00H</td> </tr> </table> | DLE | 10H | 00H | 00H | 00H | <table border="1"> <tr> <td>10H</td> <td>00H</td> <td>00H</td> <td>00H</td> </tr> </table> | 10H | 00H | 00H | 00H |
| DLE | 10H | 00H | 00H | 00H | | | | | | | |
| 10H | 00H | 00H | 00H | | | | | | | | |

Device code list

The following shows the device code of each device and the notation of device number (decimal/hexadecimal).

The data to be set differs between MELSEC-Q/L series commands (subcommand: 0000, 0001) and MELSEC iQ-R series subcommand (0002, 0003).



- The available device type and device range are in accordance with the device specifications of access target module. Specify the device that can be used for the access target module.
- Accessing a local device is not available.
- Devices not listed in this section cannot be specified by the command for device access of MC protocol.
- When accessing a device that cannot be specified, create a program etc. to copy a value and store the value temporarily in the device that can be specified and access it.
- When a device can be assigned to a standard global label in GX Works3, even the device, to which a device code cannot be specified, can be accessed by specifying the label name. (☞ Page 119 LABEL ACCESS)

—: Inaccessible

| Device | | | | For MELSEC-Q/L series | | For MELSEC iQ-R series | | |
|--|-----------------------------|--------|--|-----------------------|-------|------------------------|-------|--------|
| Device name | | Symbol | Type | Notation | ASCII | Binary | ASCII | Binary |
| Special relay | | SM | Bit | Decimal | SM | 91H | SM** | 0091H |
| Special register | | SD | Word | Decimal | SD | A9H | SD** | 00A9H |
| Input | | X | | Hexadecimal | X* | 9CH | X*** | 009CH |
| Output | | Y | | Hexadecimal | Y* | 9DH | Y*** | 009DH |
| Internal relay | | M | | Decimal | M* | 90H | M*** | 0090H |
| Latch relay | | L | | Decimal | L* | 92H | L*** | 0092H |
| Annunciator | | F | | Decimal | F* | 93H | F*** | 0093H |
| Edge relay | | V | | Decimal | V* | 94H | V*** | 0094H |
| Link relay | | B | | Hexadecimal | B* | A0H | B*** | 00A0H |
| Data register | | D | Word | Decimal | D* | A8H | D*** | 00A8H |
| Link register | | W | | Hexadecimal | W* | B4H | W*** | 00B4H |
| Timer | Contact | TS | Bit | Decimal | TS | C1H | TS** | 00C1H |
| | Coil | TC | | | TC | C0H | TC** | 00C0H |
| | Current value | TN | Word | | TN | C2H | TN** | 00C2H |
| Retentive timer | Contact | STS | Bit | Decimal | SS | C7H | STS* | 00C7H |
| | Coil | STC | | | SC | C6H | STC* | 00C6H |
| | Current value | STN | Word | | SN | C8H | STN* | 00C8H |
| Counter | Contact | CS | Bit | Decimal | CS | C4H | CS** | 00C4H |
| | Coil | CC | | | CC | C3H | CC** | 00C3H |
| | Current value | CN | Word | | CN | C5H | CN** | 00C5H |
| Link special relay | | SB | Bit | Hexadecimal | SB | A1H | SB** | 00A1H |
| Link special register | | SW | Word | Hexadecimal | SW | B5H | SW** | 00B5H |
| Direct access input | | DX | Bit | Hexadecimal | DX | A2H | DX** | 00A2H |
| Direct access output | | DY | | | DY | A3H | DY** | 00A3H |
| Index register | Index register | Z | Word | Decimal | Z* | CCH | Z*** | 00CCH |
| | Long index register | LZ | Double word | | — | — | LZ** | 0062H |
| File register* ^{1,2} | Block switching method | R | Word | Decimal | R* | AFH | R*** | 00AFH |
| | Serial number access method | ZR | | Hexadecimal | ZR | B0H | ZR** | 00B0H |
| Extended data register* ³ | | D | Word | Decimal | D* | A8H | — | — |
| Extended link register* ³ | | W | Word | Hexadecimal | W* | B4H | — | — |
| Refresh data register | | RD | Word | Decimal | — | — | RD** | 002CH |
| Network No. specified device Link direct device | | J□\□ | ☞ Page 434 Accessing link direct devices | | | | | |

| Device | | | | For MELSEC-Q/L series | | For MELSEC iQ-R series | |
|---------------------------------|---------|--|----------|-----------------------|--------|------------------------|--------|
| Device name | Symbol | Type | Notation | ASCII | Binary | ASCII | Binary |
| I/O No. specified device | U | ↗ Page 436 Accessing module access devices ↗ Page 438 Accessing CPU buffer memory access device | | | | | |
| Module access device | U□G | Word | Decimal | G | ABH | G*** | 00ABH |
| CPU buffer memory access device | U3E□IG | Word | Decimal | — | — | G** | 00ABH |
| | U3E□IHG | Word | Decimal | — | — | HG** | 002EH |

- *1 The file register specified to "Use File Register of Each Program" in "CPU Parameter" or "PLC parameter" of the CPU module cannot be accessed from external devices.
- *2 If the file register of the CPU module is consist of multiple blocks, use the device code of the serial number access method.
To specify the file register with the serial number access method, refer to the manual of CPU module.
- *3 If the access target CPU module does not support the access to the extended data register D65536 or later, and the extended link register W10000 or later, replace the extended data register to the file register (ZR) and specify again. For the replacement method, refer to the manual of Q/LCPU module.

Number of device points

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

Setting method

■Data communication using ASCII code

Convert the numerical value to 4-digit ASCII code (hexadecimal), and send it from the upper digits.

Use capitalized code for alphabetical letter.

■Data communication using binary code

Send the 2-byte numerical value^{*1} in order from the lower byte (L: bit 0 to 7).

*1 For C24, the additional code may be added. (↗ Page 33 Additional code (10H))

8

Ex.

For 5 points and 20 points

| Number of device points | ASCII code | Binary code | | | | | | | | | | |
|-------------------------|--|-------------|-----|---|---|-----|-----|-----|-----|--|-----|-----|
| 5 points | <table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td>0</td><td>0</td><td>0</td><td>5</td> </tr> <tr> <td>30H</td><td>30H</td><td>30H</td><td>35H</td> </tr> </table> | 0 | 0 | 0 | 5 | 30H | 30H | 30H | 35H | <table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td>05H</td><td>00H</td> </tr> </table> | 05H | 00H |
| 0 | 0 | 0 | 5 | | | | | | | | | |
| 30H | 30H | 30H | 35H | | | | | | | | | |
| 05H | 00H | | | | | | | | | | | |
| 20 points | <table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td>0</td><td>0</td><td>1</td><td>4</td> </tr> <tr> <td>30H</td><td>30H</td><td>31H</td><td>34H</td> </tr> </table> | 0 | 0 | 1 | 4 | 30H | 30H | 31H | 34H | <table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td>14H</td><td>00H</td> </tr> </table> | 14H | 00H |
| 0 | 0 | 1 | 4 | | | | | | | | | |
| 30H | 30H | 31H | 34H | | | | | | | | | |
| 14H | 00H | | | | | | | | | | | |

Access points

Specify the number of device points to be accessed in word unit, double word unit, or bit unit.

Setting method

■Data communication using ASCII code

Convert the numerical value to 2-digit ASCII code (hexadecimal), and send it from the upper digits.

Use capitalized code for alphabetical letter.

■Data communication using binary code

Send the 1-byte^{*1} numerical value (hexadecimal).

*1 For C24, the additional code may be added. ( Page 33 Additional code (10H))

Ex.

For 5 points and 20 points

| Number of device points | ASCII code | Binary code | | | | | |
|-------------------------|--|-------------|---|-----|-----|---|-----|
| 5 points | <table border="1"><tr><td>0</td><td>5</td></tr><tr><td>30H</td><td>35H</td></tr></table> | 0 | 5 | 30H | 35H | <table border="1"><tr><td>05H</td></tr></table> | 05H |
| 0 | 5 | | | | | | |
| 30H | 35H | | | | | | |
| 05H | | | | | | | |
| 20 points | <table border="1"><tr><td>1</td><td>4</td></tr><tr><td>31H</td><td>34H</td></tr></table> | 1 | 4 | 31H | 34H | <table border="1"><tr><td>14H</td></tr></table> | 14H |
| 1 | 4 | | | | | | |
| 31H | 34H | | | | | | |
| 14H | | | | | | | |

Number of bit access points

Specify the number of device points to be accessed in bit units.

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

Specify the number of device points to be accessed in word unit or double word unit.

Number of blocks

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

Set each number of blocks within the following range.

- Number of word device blocks + Number of bit device blocks ≤ 120

Point

In the following case, calculate it as access point $\times 2$.

- When accessing module of MELSEC iQ-R series by setting device extension specification (subcommand: 008□)

Setting method

■Data communication using ASCII code

Convert the numerical value to 2-digit ASCII code (hexadecimal), and send it from the upper digits.

Use capitalized code for alphabetical letter.

■Data communication using binary code

Send the 1-byte^{*1} numerical value (hexadecimal).

*1 For C24, the additional code may be added. (参照 Page 33 Additional code (10H))

Ex.

For 5 points and 20 points

| Number of device points | ASCII code | Binary code | | | | | |
|-------------------------|--|-------------|---|-----|-----|---|-----|
| 5 points | <table border="1"><tr><td>0</td><td>5</td></tr><tr><td>30H</td><td>35H</td></tr></table> | 0 | 5 | 30H | 35H | <table border="1"><tr><td>05H</td></tr></table> | 05H |
| 0 | 5 | | | | | | |
| 30H | 35H | | | | | | |
| 05H | | | | | | | |
| 20 points | <table border="1"><tr><td>1</td><td>4</td></tr><tr><td>31H</td><td>34H</td></tr></table> | 1 | 4 | 31H | 34H | <table border="1"><tr><td>14H</td></tr></table> | 14H |
| 1 | 4 | | | | | | |
| 31H | 34H | | | | | | |
| 14H | | | | | | | |

8

Number of word device blocks

Specify the number of blocks of the word device.

Number of bit device blocks

Specify the number of blocks of the bit device.

Read data, write data

The read device value is stored for reading, and the data to be written is stored for writing.

The data order differs between bit units or word units.

For bit units

The following shows the data to be read and written in bit units.

■Data communication using ASCII code

The ON/OFF status of each device are represented with single-digit ASCII code.

- For ON: '1' (31H)
- For OFF: '0' (30H)

■Data communication using binary code

Represent the ON/OFF status of each device in 4-bit per 1 point.

- For ON: '1'
- For OFF: '0'

When the number of points is odd number, the lowest 4 bits are set to '0'.

Ex.

When indicating ON/OFF status of five points from M10

| M10 | M11 | M12 | M13 | M14 | | | | | | | | |
|----------------------------------|---|------------|------------|-----------------------|------------|------------|------------|--|--|-----|-----|-----|
| ON | OFF | ON | OFF | ON | | | | | | | | |
| ASCII code | Binary code (For C24) ^{*1} | | | Binary code (For E71) | | | | | | | | |
| 1 0 1 0 1 31H 30H 31H 30H 31H | <table><tr><td>DLE 10H</td><td>DLE 10H</td><td>DLE 10H</td><td>DLE 10H</td><td></td></tr></table> | | | DLE 10H | DLE 10H | DLE 10H | DLE 10H | | <table><tr><td>10H</td><td>10H</td><td>10H</td></tr></table> | 10H | 10H | 10H |
| DLE 10H | DLE 10H | DLE 10H | DLE 10H | | | | | | | | | |
| 10H | 10H | 10H | | | | | | | | | | |

*1 For C24, the additional code may be added. ( Page 33 Additional code (10H))

For word units (16-point unit for bit device)

The following shows the data to be read and written in word units.

When handling data other than bit data, refer to the following section.

Page 75 Considerations for handling real number data and character string data

■Data communication using ASCII code

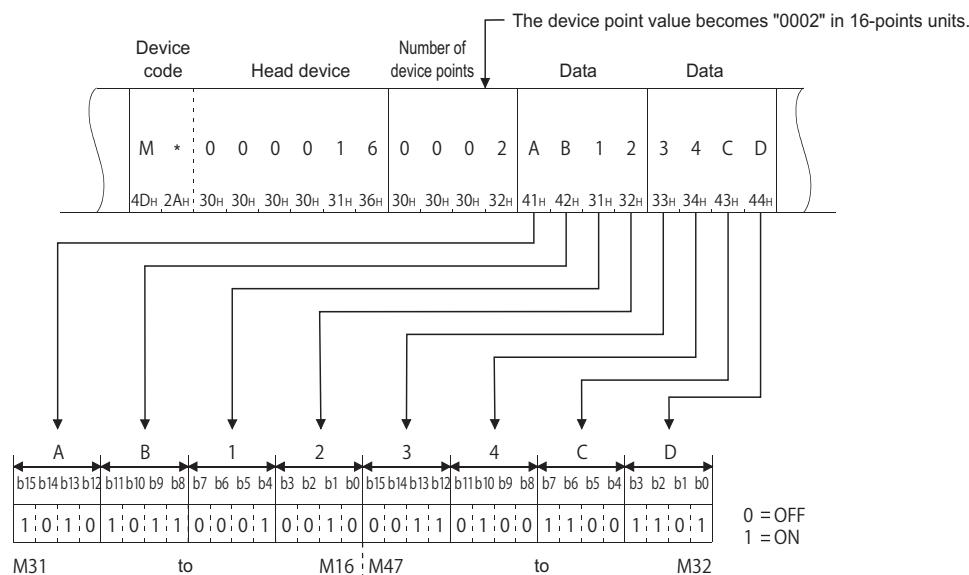
Convert the 1-word(16 points of bit device) numerical value to 4-digit ASCII code (hexadecimal), and send it from the upper digits.

Use capitalized code for alphabetical letter.

The ON/OFF status of bit device is a value of hexadecimal 1-digit in 4-point units.

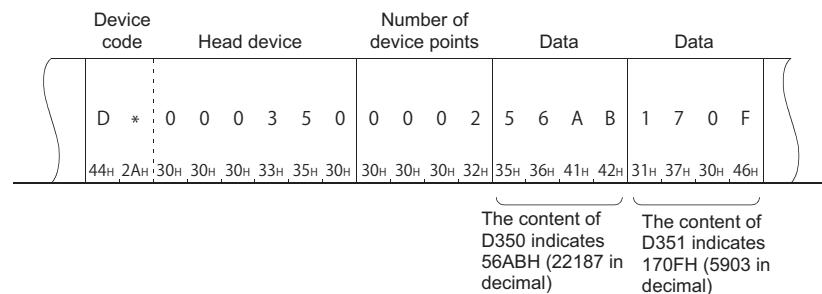
Ex.

When indicating ON/OFF status of 32 points from M16



Ex.

When indicating the stored data of D350 and D351

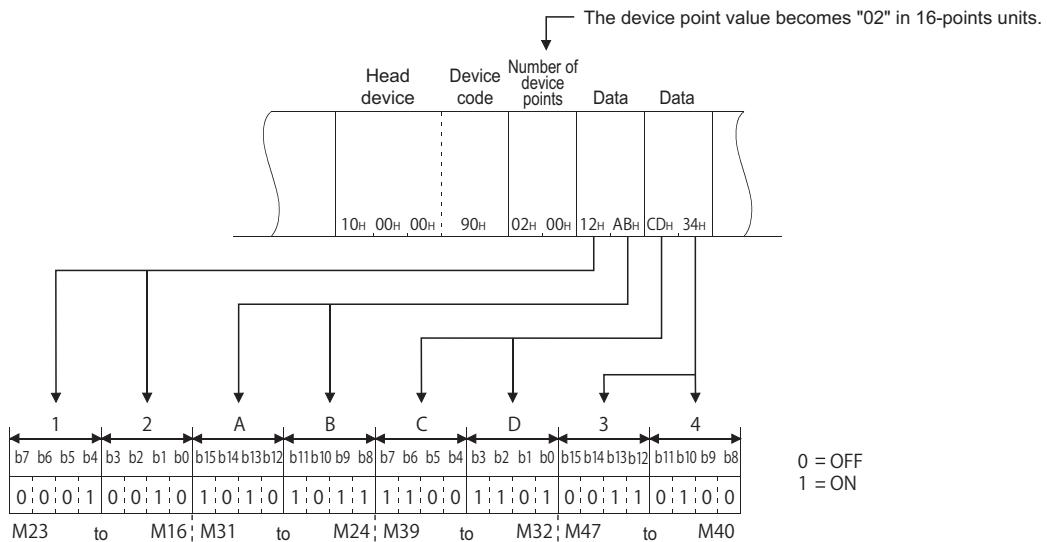


■Data communication using binary code

Send the numerical value in order from the lower byte (L: bit 0 to 7) by handling 16 points unit as 2 bytes.

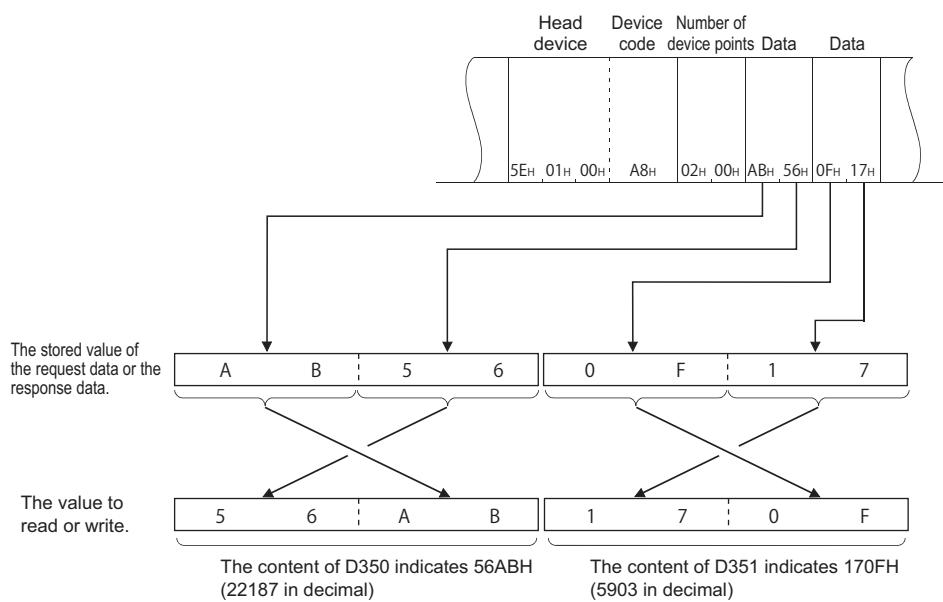
Ex.

When indicating ON/OFF status of 32 points from M16



Ex.

When indicating the stored data of D350 and D351



For double word unit (32-point unit for bit device)

The following shows the data to be read and written in double word units.

■Data communication using ASCII code

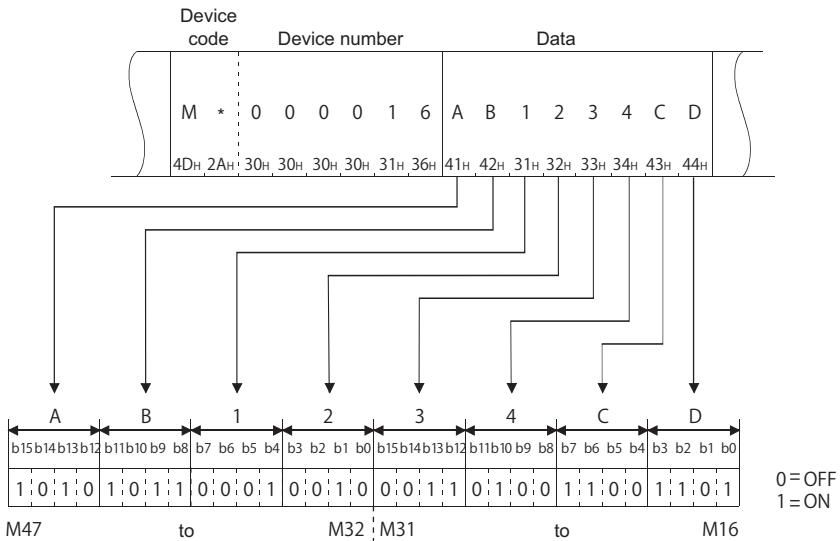
Convert the 2-word numerical value (32 points of bit device) to 8-digit ASCII code (hexadecimal), and send it from the upper digits.

Use capitalized code for alphabetical letter.

The ON/OFF status of the bit device is 1-digit hexadecimal value in 4-point units.

Ex.

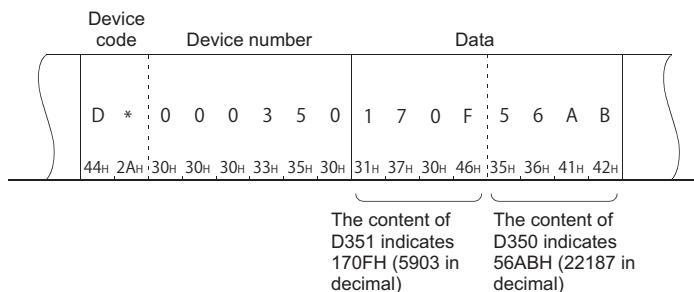
When indicating ON/OFF status of 32 points from M16



8

Ex.

When indicating the stored data of D350 (D351)

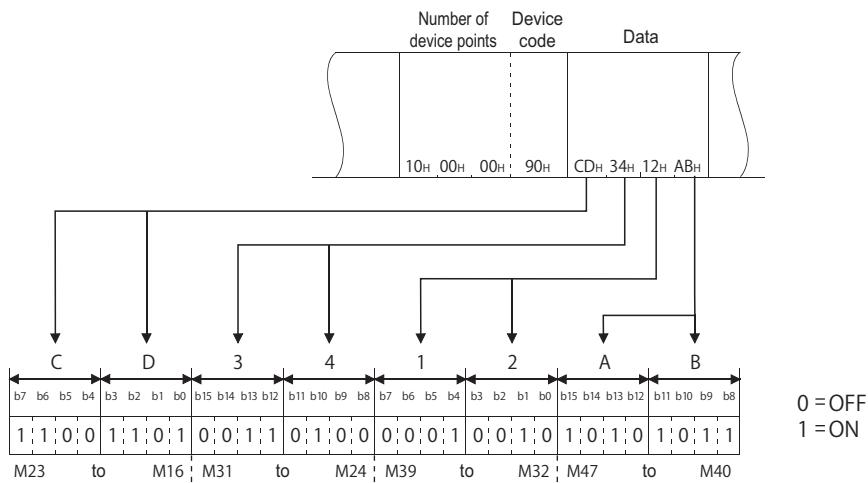


■Data communication using binary code

Send the numerical value in order from the lower byte (L: bit 0 to 7) by handling 32 points unit as 4 bytes.

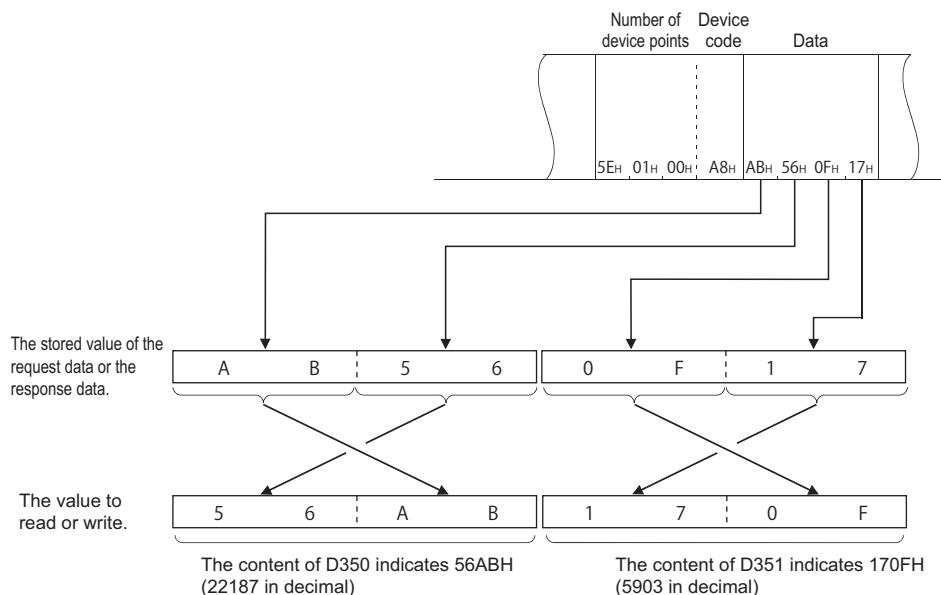
Ex.

When indicating ON/OFF status of 32 points from M16



Ex.

When indicating the stored data of D350 (D351)



Considerations for handling real number data and character string data

The word data and double word data are handled as integer value (16-bit data or 32-bit data).

When data other than integer (real number, character string) is stored in a device, the stored value is read as integer value.

- When real number (0.75) is stored in D0 and D1: D0 = 0000H, D1 = 3F40H
- When character string ('12AB') is stored in D2 and D3: D2 = 3231H, D3 = 4241H

For data to be used as real number or character string data in the instructions of the programmable controller, write it to the device/label according to the defined data specification method. For more details on how to specify data used in instructions, refer to the programming manual of the CPU module used.

■For character string data

The following shows the images how character string data is stored.

| Item | For ASCII code character string | For Unicode character string |
|--|---------------------------------|------------------------------|
| Character string to be stored | 'ABC' | 'ABCD' |
| Character code | '41H', '42H', '43H' | '41H', '42H', '43H', '44H' |
| Image when character string data is stored from D0 | NULL indicates 00H. | NULL indicates 00H. |

Ex.

Write ASCII code character string data used in the instructions which handle character strings to word device

Store the character string ('ABCD') to D0 and D1: D0 = 4241H ('BA'), D1 = 4443H ('DC')

Specify the following data for write data.

| ASCII code | Binary code |
|------------|-------------|
| | |

Point

When communicating ASCII code character string data in ASCII code, data is rearranged every two characters and stored.

Set/reset

Specify the ON/OFF status of bit device.

- For ON: '1'

| Subcommand type | ASCII code | Binary code | | | | | | | | | | |
|------------------------|--|-------------|-----|-----|-----|---|-----|-----|-----|---|-----|-----|
| For MELSEC-Q/L series | <table border="1"><tr><td>0</td><td>1</td></tr><tr><td>30H</td><td>31H</td></tr></table> | 0 | 1 | 30H | 31H | <table border="1"><tr><td>01H</td></tr></table> | 01H | | | | | |
| 0 | 1 | | | | | | | | | | | |
| 30H | 31H | | | | | | | | | | | |
| 01H | | | | | | | | | | | | |
| For MELSEC iQ-R series | <table border="1"><tr><td>0</td><td>0</td><td>0</td><td>1</td></tr><tr><td>30H</td><td>30H</td><td>30H</td><td>31H</td></tr></table> | 0 | 0 | 0 | 1 | 30H | 30H | 30H | 31H | <table border="1"><tr><td>01H</td><td>00H</td></tr></table> | 01H | 00H |
| 0 | 0 | 0 | 1 | | | | | | | | | |
| 30H | 30H | 30H | 31H | | | | | | | | | |
| 01H | 00H | | | | | | | | | | | |

- For OFF: '0'

| Subcommand type | ASCII code | Binary code | | | | | | | | | | |
|------------------------|--|-------------|-----|-----|-----|---|-----|-----|-----|---|-----|-----|
| For MELSEC-Q/L series | <table border="1"><tr><td>0</td><td>0</td></tr><tr><td>30H</td><td>30H</td></tr></table> | 0 | 0 | 30H | 30H | <table border="1"><tr><td>00H</td></tr></table> | 00H | | | | | |
| 0 | 0 | | | | | | | | | | | |
| 30H | 30H | | | | | | | | | | | |
| 00H | | | | | | | | | | | | |
| For MELSEC iQ-R series | <table border="1"><tr><td>0</td><td>0</td><td>0</td><td>0</td></tr><tr><td>30H</td><td>30H</td><td>30H</td><td>30H</td></tr></table> | 0 | 0 | 0 | 0 | 30H | 30H | 30H | 30H | <table border="1"><tr><td>00H</td><td>00H</td></tr></table> | 00H | 00H |
| 0 | 0 | 0 | 0 | | | | | | | | | |
| 30H | 30H | 30H | 30H | | | | | | | | | |
| 00H | 00H | | | | | | | | | | | |

Monitor condition specification

The following explains the data to be used when specifying monitor conditions by the following commands.

- Random read in word units (command: 0403)
- Register monitor data (command: 0801)

| | | |
|-------------------|---------------------------|-------------------------|
| Monitor condition | Step No. specification | Device specification |
|-------------------|---------------------------|-------------------------|

Monitor condition specification

| Item | Description | | Reference |
|---------------------------|--|----------------------------|--|
| Step No. specification | File specification | File No. | Specify the registration number of file which includes a program to be the conditions in a module. The number for specification can be obtained from the file control command. |
| | | File name | Specify the file name, extension, attribute of a file which include a program to be the conditions. |
| | | Extension | |
| | | Attribute | |
| | SFC specification | SFC pattern | Specify this when a program is SFC. |
| | | SFC block I/O No. | Specify the SFC block No. and SFC step No. which include a step to be the conditions. |
| | | SFC step No. | |
| Step No. | Specify the step No., pointer (P) No., or interrupt pointer (I) No. for a program to be the condition. | | Page 82 Step No. |
| Device specification | Word device value specification | Device | Specify the device to be a condition. |
| | | Mask value | Specify this when detecting arbitrary bit range of word device. |
| | | Monitor condition value | Specify the device value to be a condition. |
| | Bit device value specification | Device | Specify the device to be a condition. |
| | | Monitor condition value | Specify the device value to be a condition. |

Point

The monitor conditions can be specified by following QCPUs.

- Basic model QCPU
- High Performance model QCPU
- Process CPU

To access a module which does not support this function, select the subcommand that does not specify monitor conditions. For more information on the supported modules, refer to the following section.

☞ Page 465 Accessible Modules for Each Command

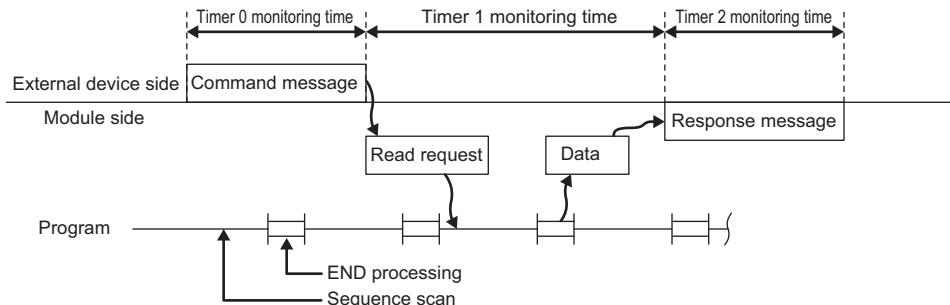
When the subcommands which do not specify monitor conditions are selected, each data for specifying the monitor conditions are not necessary.

Specification of read timing by monitor condition

The read timing can be changed by specifying monitor conditions according to the selection of subcommand.

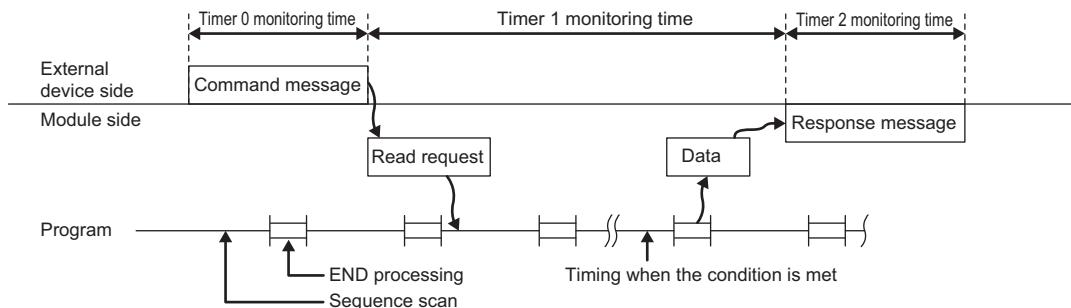
When do not specify monitor condition

Data is read by the END processing after a read request.



When monitor condition is specified

Data is read by the END processing after the specified monitor conditions are satisfied.



Monitoring with multiple conditions to the device memory of the same CPU module cannot be performed at the same time. When this command, to which a monitor condition is specified, is executed while monitor with other conditions is being performed, the command is completed abnormally.

Monitor condition

Following conditions can be specified as "Monitor condition" for read timing.

| Conditions that can be specified | | Condition satisfaction timing |
|----------------------------------|---------------------------------|--|
| Step No. specification | | When specified program step is executed |
| Device specification | Word device value specification | When specified word device value reached the specified value |
| | Bit device value specification | When specified bit device turned ON/OFF |

When the step No. specification and device specification are specified together, the read processing is performed when both of the conditions are satisfied.

Specify the following values according to the conditions.

○: Specified, —: Not specified

| Step No. specification | Device specification | ASCII code | Binary code |
|------------------------|-----------------------------------|----------------------------|-------------|
| ○ | — | 0 1 0 F 30H 31H 30H 46H | 01H 0FH |
| — | ○ Word device value specification | 0 2 0 F 30H 32H 30H 46H | 02H 0FH |
| ○ | | 0 3 0 F 30H 33H 30H 46H | 03H 0FH |
| — | ○ Bit device value specification | 0 4 0 F 30H 34H 30H 46H | 04H 0FH |
| ○ | | 0 5 0 F 30H 35H 30H 46H | 05H 0FH |

Step No. specification

Specify a condition using a step No. of program.

Data is read at the END processing immediately after a step of the specified program is executed.

■When step No. is specified

Specify the following items.

| Item | Description | | | Reference | |
|------------------------|--------------------|--|--|--|--|
| Step No. specification | File specification | File No. | | Specify the registration number of file which includes a program to be the conditions in a module. The number for specification can be obtained from the file control command.*1 | |
| | | File name | | Specify the file name, extension, attribute of a file which includes a program to be the conditions. | |
| | | Extension | | Page 293 File name, extension, and attribute | |
| SFC specification | Attribute | SFC pattern | | | |
| | | SFC block I/O No. | | | |
| | | SFC step No. | | | |
| Step No. | | Specify the step No., pointer (P) No., or interrupt pointer (I) No. for a program to be the condition. | | Page 82 Step No. | |

*1 When 'FFFFH' is specified to the file, the specified file is searched with the file name and extension. In this case, a read and write request from a supported device to the CPU module may be delayed more than one sequence scan time.

■When step No. is not specified

Set the following when a monitor condition without step No. specification is selected.

- File No.: 0
- File name, extension, attribute: Space (20H)
- SFC specification, step No.: 0

ASCII code

| File No. | File name | Extension | Attribute | SFC pattern | Block No. | Step No. | Step No. |
|-------------------------------|--|---------------------------|---------------------------|-------------------------------|-------------------------------|-------------------------------|-------------------------------|
| 0 0 0 0 30H, 30H, 30H, 30H | 20H, 20H, 20H, 20H, 20H, 20H, 20H, 20H 30H, 30H, 30H, 30H, 30H, 30H, 30H, 30H | 20H 30H, 30H, 30H, 30H | 20H 30H, 30H, 30H, 30H | 0 0 0 0 30H, 30H, 30H, 30H | 0 0 0 0 30H, 30H, 30H, 30H | 0 0 0 0 30H, 30H, 30H, 30H | 0 ... 0 30H, 30H, 30H, 30H |
| File specification | | | | SFC specification | | | |

Binary code

| File No. | File name | Extension | Attribute | Step No. | Step No. | Block No. | SFC pattern |
|--|--|---------------------------|---------------------------|--|--|--|--|
| 00H, 00H 20H, 20H, 20H, 20H, 20H, 20H, 20H, 20H 00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H | 20H, 20H, 20H, 20H, 20H, 20H, 20H, 20H 00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H | 20H 00H, 00H, 00H, 00H | 20H 00H, 00H, 00H, 00H | 00H, 00H, 00H, 00H 00H, 00H, 00H, 00H |
| File specification | | | | SFC specification | | | |

Device specification

Specify a condition using a device and its value.

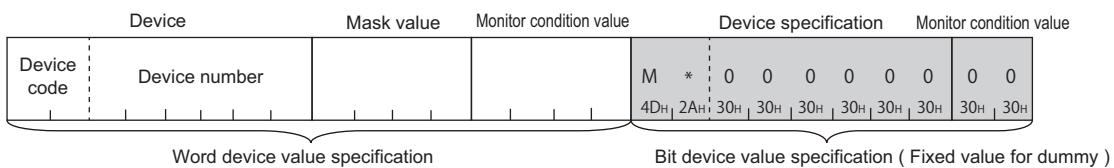
Data is read at the END processing immediately after the specified device reached the specified value.

When word device value is specified

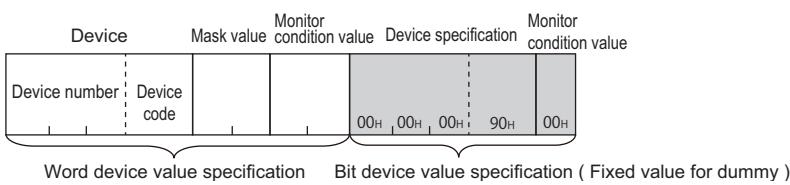
Set the following when a monitor condition which specifies word device value is selected.

| Item | Description | | Reference |
|----------------------|---------------------------------|-------------------------|---|
| Device specification | Word device value specification | Device | Specify the device to be a condition. |
| | | Mask value | Specify this when detecting arbitrary bit range of word device. |
| | | Monitor condition value | Specify the device value to be a condition. |
| | Bit device value specification | Device | Specify an arbitrary device. |
| | | Monitor condition value | Specify the fixed value (0). |

ASCII code



Binary code

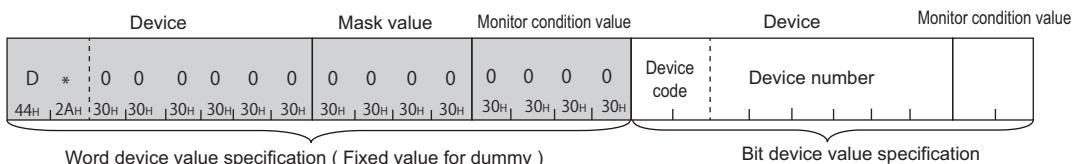


When bit device value is specified

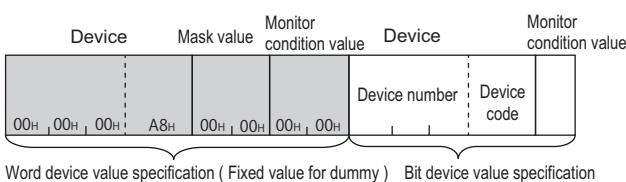
Set the following when a monitor condition which specifies bit device value is selected.

| Item | Description | | Reference |
|----------------------|---------------------------------|-------------------------|--|
| Device specification | Word device value specification | Device | Specify an arbitrary device. |
| | | Mask value | Specify the fixed value (0). |
| | | Monitor condition value | |
| | Bit device value specification | Device | Specify the device to be a condition. |
| | | Monitor condition value | Specify the condition with the following value. 02H: Condition is satisfied at rising (OFF→ON) 04H: Condition is satisfied at falling (ON→OFF) |

ASCII code



Binary code



■When no device is specified

Set the following when monitor condition without device specification is selected.

ASCII code

| | | | | | | | | | | | | | | | | | | | | | | | |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| D | * | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | M | * | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 44H | 2AH | 30H | 2AH | 30H |

Binary code

| | | | | | | | | | | | | | |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| 00H | 00H | 00H | A8H | 00H | 90H | 00H |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|

SFC specification

The block No. and step No. of SFC (MELSAp3) program can be specified as a monitor condition.

Specify the following values.

| Condition | SFC pattern | SFC block No. | SFC step No. |
|----------------------------|-------------|---------------------------|---------------------------|
| Specify an SFC program. | 0003H | 0000H to 013FH (0 to 319) | 0000H to 01FFH (0 to 511) |
| Do not specify SFC program | 0000H | 0000H | 0000H |

■Data communication using ASCII code

Convert the numerical value to 4-digit ASCII code (hexadecimal), and send it from the upper digits.

■Data communication using binary code

Send the 2-byte numerical value^{*1} in order from the lower byte (L: bit 0 to 7).

*1 For C24, the additional code may be added. (参照 Page 33 Additional code (10H))

Ex.

For 0003H

| ASCII code | Binary code |
|----------------------------|-------------|
| 0 0 0 3 30H 30H 30H 33H | 03H 00H |

Step No.

Specify the step No., pointer (P) No., or interrupt pointer (I) No. of the sequence program.

Specify a following 4-byte value.

| Condition | b31 | b30 | b29 to b0 |
|--|----------|-----|--|
| Specify the step No. of a sequence program | 0 | 0 | (Step No. of the arbitrary sequence program) |
| Specify pointer No. | 0 | 1 | (Arbitrary pointer No.) |
| Specify an interrupt pointer No. | 1 | 0 | (Arbitrary interrupt pointer No.) |
| Do not specify | 0000000H | | |

■Data communication using ASCII code

Convert the numerical value to 8-digit ASCII code (hexadecimal), and send it from the upper digits.

■Data communication using binary code

Send the 4-byte numerical value^{*1} in order from the lower byte (L: bit 0 to 7).

*1 For C24, the additional code may be added. (参照 Page 33 Additional code (10H))

Ex.

When specifying the interrupt pointer I28 (8000001CH)

| ASCII code | Binary code |
|--|-----------------|
| 8 0 0 0 0 0 1 C 38H 30H 30H 30H 30H 31H 43H | 1CH 00H 00H 80H |

Mask value, monitor conditions when word device value is specified

Specify the value of word device to be set as a monitor condition.

Arbitrary bit range of word devices can only be specified by specifying mask value.

(Logical AND by each bit of the specified word device data and the designated mask value is compared with the monitor condition value.)

■Data communication using ASCII code

Convert the numerical value to 4-digit ASCII code (hexadecimal), and send it from the upper digits.

■Data communication using binary code

Send the 2-byte numerical value^{*1} in order from the lower byte (L: bit 0 to 7).

*1 For C24, the additional code may be added. (☞ Page 33 Additional code (10H))

Ex.

Specifying when bit 0 to 14 of D0 reaches 1000 (3E8H) as a condition

- Mask value: 7FFFH

| ASCII code | Binary code | | | | | | | | | | |
|---|-------------|-----|-----|---|-----|-----|-----|-----|--|-----|-----|
| <table border="1"> <tr> <td>7</td> <td>F</td> <td>F</td> <td>F</td> </tr> <tr> <td>37H</td> <td>46H</td> <td>46H</td> <td>46H</td> </tr> </table> | 7 | F | F | F | 37H | 46H | 46H | 46H | <table border="1"> <tr> <td>FFH</td> <td>7FH</td> </tr> </table> | FFH | 7FH |
| 7 | F | F | F | | | | | | | | |
| 37H | 46H | 46H | 46H | | | | | | | | |
| FFH | 7FH | | | | | | | | | | |

- Monitor condition value: 03E8H

| ASCII code | Binary code | | | | | | | | | | |
|---|-------------|-----|-----|---|-----|-----|-----|-----|--|-----|-----|
| <table border="1"> <tr> <td>0</td> <td>3</td> <td>E</td> <td>8</td> </tr> <tr> <td>30H</td> <td>33H</td> <td>45H</td> <td>38H</td> </tr> </table> | 0 | 3 | E | 8 | 30H | 33H | 45H | 38H | <table border="1"> <tr> <td>E8H</td> <td>03H</td> </tr> </table> | E8H | 03H |
| 0 | 3 | E | 8 | | | | | | | | |
| 30H | 33H | 45H | 38H | | | | | | | | |
| E8H | 03H | | | | | | | | | | |

Monitor conditions when bit device value is specified

Specify a bit device status change as the monitor condition from either rising (OFF → ON) or falling (ON → OFF).

| Condition | ASCII code | Binary code | | | | | |
|------------------|---|-------------|---|-----|-----|---|-----|
| Rising (OFF→ON) | <table border="1"> <tr> <td>0</td> <td>2</td> </tr> <tr> <td>30H</td> <td>32H</td> </tr> </table> | 0 | 2 | 30H | 32H | <table border="1"> <tr> <td>02H</td> </tr> </table> | 02H |
| 0 | 2 | | | | | | |
| 30H | 32H | | | | | | |
| 02H | | | | | | | |
| Falling (ON→OFF) | <table border="1"> <tr> <td>0</td> <td>4</td> </tr> <tr> <td>30H</td> <td>34H</td> </tr> </table> | 0 | 4 | 30H | 34H | <table border="1"> <tr> <td>04H</td> </tr> </table> | 04H |
| 0 | 4 | | | | | | |
| 30H | 34H | | | | | | |
| 04H | | | | | | | |

8.2 Batch Read and Write

Read or write the values of consecutive devices in batch by specifying the number of device points.

Batch read in word units (command: 0401)

Read values from devices in word units.



When accessing any of the following devices, use the device extension specification (subcommand: 008□).

- Link direct device
- Module access device
- CPU buffer memory access device

For the message format for device extension specification, refer to the following section.

Page 432 Read/Write by Device Extension Specification

Message format

The following shows the message format of the request data and response data of the command.

■Request data

| Command | Subcommand | Head device | Number of device points |
|-------------------|------------|-------------|-------------------------|
| 4C/3C/4E/3E frame | | | |

■Response data

The value of read device is stored in word units. The data order differs between ASCII code or binary code. (Page 70 Read data, write data)

Data specified by request data

■Command

| Frame | ASCII code | Binary code | | | | | | | | | | |
|-------------------|--|-------------|-----|---|---|-----|-----|-----|-----|---|-----|-----|
| 4C/3C/4E/3E frame | <table border="1"><tr><td>0</td><td>4</td><td>0</td><td>1</td></tr><tr><td>30H</td><td>34H</td><td>30H</td><td>31H</td></tr></table> | 0 | 4 | 0 | 1 | 30H | 34H | 30H | 31H | <table border="1"><tr><td>01H</td><td>04H</td></tr></table> | 01H | 04H |
| 0 | 4 | 0 | 1 | | | | | | | | | |
| 30H | 34H | 30H | 31H | | | | | | | | | |
| 01H | 04H | | | | | | | | | | | |
| 2C frame | <table border="1"><tr><td>2</td></tr><tr><td>32H</td></tr></table> | 2 | 32H | — | | | | | | | | |
| 2 | | | | | | | | | | | | |
| 32H | | | | | | | | | | | | |

■Subcommand

| Type | ASCII code | Binary code | | | | | | | | | | |
|------------------------|--|-------------|-----|---|---|-----|-----|-----|-----|---|-----|-----|
| For MELSEC-Q/L series | <table border="1"><tr><td>0</td><td>0</td><td>0</td><td>0</td></tr><tr><td>30H</td><td>30H</td><td>30H</td><td>30H</td></tr></table> | 0 | 0 | 0 | 0 | 30H | 30H | 30H | 30H | <table border="1"><tr><td>00H</td><td>00H</td></tr></table> | 00H | 00H |
| 0 | 0 | 0 | 0 | | | | | | | | | |
| 30H | 30H | 30H | 30H | | | | | | | | | |
| 00H | 00H | | | | | | | | | | | |
| For MELSEC iQ-R series | <table border="1"><tr><td>0</td><td>0</td><td>0</td><td>2</td></tr><tr><td>30H</td><td>30H</td><td>30H</td><td>32H</td></tr></table> | 0 | 0 | 0 | 2 | 30H | 30H | 30H | 32H | <table border="1"><tr><td>02H</td><td>00H</td></tr></table> | 02H | 00H |
| 0 | 0 | 0 | 2 | | | | | | | | | |
| 30H | 30H | 30H | 32H | | | | | | | | | |
| 02H | 00H | | | | | | | | | | | |

For 2C frame, the specification is not required. Functions and specification methods are equivalent to the subcommands for MELSEC-Q/L series.

■Head device

Specify the head device of the consecutive devices. (Page 63 Devices)



Long index register (LZ) cannot be specified.

■Number of device points

Specify the number of device points to be read within the following range in word units. ([Page 67 Number of device points](#))

| Access target | Range | |
|--|-----------------|---------------------------------------|
| | Word device | Bit device |
| MELSEC iQ-R series module MELSEC-Q/L series module | 1 to 960 points | 1 to 960 words (1 to 15360 points) |
| MELSEC-QnA series module Module on other station via MELSEC-QnA series network module | 1 to 480 points | 1 to 480 words (1 to 7680 points) |
| MELSEC-A series module | 1 to 64 points | 1 to 32 words (1 to 512 points) |

Read 16-point bit device by specifying one point of "Number of device points".

Set the head device number for MELSEC-A series module with a multiple of 16.

Communication example (Reading bit device)

Read the value of M100 to M131 (for 2 words). (Subcommand: for MELSEC-Q/L series)

■Data communication using ASCII code

(Request data)

| Subcommand | Device code | Head device number | Number of device points |
|----------------------------------|----------------------------------|--------------------|--|
| 0 4 0 1 30H , 34H , 30H , 31H | 0 0 0 0 30H , 30H , 30H , 30H | M * 4DH , 2AH | 0 0 0 1 0 0 0 2 30H , 30H , 30H , 31H , 30H , 30H , 30H , 32H |

(Response data)

| | |
|--|--|
| 1 2 3 4 31H , 32H , 33H , 34H | 0 0 0 2 30H , 30H , 30H , 32H |
| 1 2 3 4 0 0 0 1 0 0 1 0 0 0 1 1 0 1 0 0 | 0 0 0 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 |

M115 to M100 M131 to M116

0 = OFF
1 = ON

■Data communication using binary code

(Request data)

| Subcommand | Device code | Head device number | Number of device points |
|------------|-------------|--------------------|-------------------------|
| 01H , 04H | 00H , 00H | 64H , 00H , 00H | 90H 02H , 00H |

(Response data)

| |
|--|
| 34H , 12H , 02H , 00H |
| 3 4 1 2 0 2 0 0 0 0 1 1 0 1 0 0 0 0 1 0 0 0 1 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 |

M107 to M100 M115 to M108 M123 to M116 M131 to M124

0 = OFF
1 = ON

Communication example (Reading word device)

Read values of T100 to T102. (Subcommand: for MELSEC-Q/L series)

T100 = 4660 (1234H), T101 = 2 (2H), T102 = 7663 (1DEFH) are stored.

■Data communication using ASCII code

(Request data)

| Subcommand | Device code | Head device number | Number of device points |
|-------------------------------|---|-------------------------------------|---|
| 0 4 0 1 30H, 34H, 30H, 31H | 0 0 0 0 30H, 30H, 30H, 30H 54H, 4EH | T N 30H, 30H, 30H, 31H, 30H, 30H | 0 0 0 1 0 0 30H, 30H, 30H, 31H, 30H, 33H |

(Response data)

| | | |
|-------------------------------|-------------------------------|-------------------------------|
| 1 2 3 4 31H, 32H, 33H, 34H | 0 0 0 2 30H, 30H, 30H, 32H | 1 D E F 31H, 44H, 45H, 46H |
|-------------------------------|-------------------------------|-------------------------------|

T100 T101 T102

■Data communication using binary code

(Request data)

| Subcommand | Device code | Head device number | Number of device points |
|------------|-------------|--------------------|-------------------------|
| 01H, 04H | 00H, 00H | 64H, 00H, 00H | C2H, 03H, 00H |

(Response data)

| | | |
|----------|----------|----------|
| 34H, 12H | 02H, 00H | EFH, 1DH |
|----------|----------|----------|

T100 T101 T102

Batch read in bit units (command: 0401)

Read values from devices in bit units.



When accessing any of the following devices, use the device extension specification (subcommand: 008□).

- Link direct device
- Module access device
- CPU buffer memory access device

For the message format for device extension specification, refer to the following section.

Page 432 Read/Write by Device Extension Specification

Message format

The following shows the message format of the request data and response data of the command.

■Request data

| Command | Subcommand | Head device | Number of device points |
|-------------------|----------------------------|-------------|-------------------------|
| 4C/3C/4E/3E frame | 0 4 0 1 30H 34H 30H 31H | | 01H 04H |

■Response data

The value of read device is stored in bit units. The data order differs between ASCII code or binary code. (Page 70 Read data, write data)

Data specified by request data

■Command

| Frame | ASCII code | Binary code |
|-------------------|----------------------------|-------------|
| 4C/3C/4E/3E frame | 0 4 0 1 30H 34H 30H 31H | 01H 04H |
| 2C frame | 1 31H | — |

■Subcommand

| Type | ASCII code | Binary code |
|------------------------|----------------------------|-------------|
| For MELSEC-Q/L series | 0 0 0 1 30H 30H 30H 31H | 01H 00H |
| For MELSEC iQ-R series | 0 0 0 3 30H 30H 30H 33H | 03H 00H |

For 2C frame, the specification is not required. Functions and specification methods are equivalent to the subcommands for MELSEC-Q/L series.

■Head device

Specify the head device of the consecutive devices. (Page 63 Devices)



Long index register (LZ) cannot be specified.

■Number of device points

Specify the number of device points to be read within the following range. ([Page 67 Number of device points](#))

| Access target | C24 | E71 | |
|--|------------------|------------------|------------------|
| | | ASCII code | Binary code |
| MELSEC iQ-R series module MELSEC-Q/L series module | 1 to 7904 points | 1 to 3584 points | 1 to 7168 points |
| MELSEC-QnA series module Module on other station via MELSEC-QnA series network module | 1 to 3952 points | 1 to 1792 points | 1 to 3584 points |
| MELSEC-A series module | 1 to 256 points | | |

Communication example

Read values of M100 to M107. (Subcommand: for MELSEC-Q/L series)

■Data communication using ASCII code

(Request data)

| Subcommand | Device code | Head device number | Number of device points |
|-------------------------------|-------------------------------|--------------------|---|
| 0 4 0 1 30H, 34H, 30H, 31H | 0 0 0 1 30H, 30H, 30H, 31H | M * 4DH, 2AH | 0 0 0 1 0 0 0 8 30H, 30H, 30H, 31H, 30H, 30H, 30H, 38H |

(Response data)

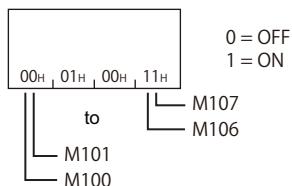
| | |
|---|-------------------|
| 0 0 0 1 0 0 1 1 30H, 30H, 30H, 31H, 30H, 30H, 31H, 31H | 0 = OFF 1 = ON |
| M100 to M107 | |

■Data communication using binary code

(Request data)

| Subcommand | Device code | Head device number | Number of device points |
|------------|-------------|--------------------|-------------------------|
| 01H, 04H | 01H, 00H | 64H, 00H, 00H | 90H, 08H, 00H |

(Response data)



Batch write in word units (command: 1401)

Write values to devices in word units.



When accessing any of the following devices, use the device extension specification (subcommand: 008□).

- Link direct device
- Module access device
- CPU buffer memory access device

For the message format for device extension specification, refer to the following section.

☞ Page 432 Read/Write by Device Extension Specification

Message format

The following shows the message format of the request data and response data of the command.

■Request data

| Command | Subcommand | Head device | Number of device points | Write data |
|---------|------------|-------------|-------------------------|------------|
|---------|------------|-------------|-------------------------|------------|

■Response data

There is no response data for this command.

Data specified by request data

■Command

| Frame | ASCII code | Binary code | | | | | | | | | | |
|-------------------|--|-------------|-----|---|---|-----|-----|-----|-----|---|-----|-----|
| 4C/3C/4E/3E frame | <table border="1"><tr><td>1</td><td>4</td><td>0</td><td>1</td></tr><tr><td>31H</td><td>34H</td><td>30H</td><td>31H</td></tr></table> | 1 | 4 | 0 | 1 | 31H | 34H | 30H | 31H | <table border="1"><tr><td>01H</td><td>14H</td></tr></table> | 01H | 14H |
| 1 | 4 | 0 | 1 | | | | | | | | | |
| 31H | 34H | 30H | 31H | | | | | | | | | |
| 01H | 14H | | | | | | | | | | | |
| 2C frame | <table border="1"><tr><td>4</td></tr><tr><td>34H</td></tr></table> | 4 | 34H | — | | | | | | | | |
| 4 | | | | | | | | | | | | |
| 34H | | | | | | | | | | | | |

■Subcommand

| Type | ASCII code | Binary code | | | | | | | | | | |
|------------------------|--|-------------|-----|---|---|-----|-----|-----|-----|---|-----|-----|
| For MELSEC-Q/L series | <table border="1"><tr><td>0</td><td>0</td><td>0</td><td>0</td></tr><tr><td>30H</td><td>30H</td><td>30H</td><td>30H</td></tr></table> | 0 | 0 | 0 | 0 | 30H | 30H | 30H | 30H | <table border="1"><tr><td>00H</td><td>00H</td></tr></table> | 00H | 00H |
| 0 | 0 | 0 | 0 | | | | | | | | | |
| 30H | 30H | 30H | 30H | | | | | | | | | |
| 00H | 00H | | | | | | | | | | | |
| For MELSEC iQ-R series | <table border="1"><tr><td>0</td><td>0</td><td>0</td><td>2</td></tr><tr><td>30H</td><td>30H</td><td>30H</td><td>32H</td></tr></table> | 0 | 0 | 0 | 2 | 30H | 30H | 30H | 32H | <table border="1"><tr><td>02H</td><td>00H</td></tr></table> | 02H | 00H |
| 0 | 0 | 0 | 2 | | | | | | | | | |
| 30H | 30H | 30H | 32H | | | | | | | | | |
| 02H | 00H | | | | | | | | | | | |

For 2C frame, the specification is not required. Functions and specification methods are equivalent to the subcommands for MELSEC-Q/L series.

■Head device

Specify the head device of the consecutive devices. (☞ Page 63 Devices)



Long index register (LZ) cannot be specified.

■Number of device points

Specify the number of device points to be written within the following range in word units. (☞ Page 67 Number of device points)

| Access target | Range | |
|--|-----------------|---------------------------------------|
| | Word device | Bit device |
| MELSEC iQ-R series module MELSEC-Q/L series module | 1 to 960 points | 1 to 960 words (1 to 15360 points) |
| MELSEC-QnA series module Module on other station via MELSEC-QnA series network module | 1 to 480 points | 1 to 480 words (1 to 7680 points) |
| MELSEC-A series module | 1 to 64 points | 1 to 10 words (1 to 160 points) |

For bit device, read 16-point bit device by specifying one point of "Number of device points".

Set the head device number for MELSEC-A series module with a multiple of 16.

■Write data

Specify the data to be written for the number of device points in hexadecimal. (☞ Page 70 Read data, write data)

Communication example (Writing bit device)

Write the values to M100 to M131 (for 2 words). (Subcommand: for MELSEC-Q/L series)

■Data communication using ASCII code

| Subcommand | Device code | Head device number | Number of device points | Write data |
|------------------------------------|----------------------------|------------------------------------|------------------------------------|------------------------------------|
| 1 4 0 1 31H 34H 30H 31H | 0 0 0 0 30H 30H 30H 30H | M * 4DH 2AH | 0 0 0 1 30H 30H 30H 31H | 0 0 0 2 30H 30H 30H 32H |
| 31H 34H 30H 31H 30H 30H 30H 30H | 4DH 2AH | 30H 30H 30H 31H 30H 30H 30H 32H | 30H 30H 30H 33H 32H 33H 34H 37H | 2 3 4 7 A B 9 6 41H 42H 39H 36H |

| | | | | | | | |
|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|
| 2 | 3 | 4 | 7 | A | B | 9 | 6 |
| 0 0 1 0 0 0 1 1 | 0 1 0 1 0 0 0 1 | 1 1 0 1 0 0 0 1 | 1 1 1 0 1 0 0 1 | 1 1 1 0 1 0 1 0 | 1 0 1 0 1 0 1 1 | 1 1 0 0 1 0 1 1 | 0 1 1 0 1 1 0 0 |

M115 to M100 ; M131 to M116

0 = OFF
1 = ON

■Data communication using binary code

| Subcommand | Device code | Head device number | Number of device points | Write data |
|--|--|--|--|-----------------|
| 01H 14H 00H 00H | 64H 00H 90H 02H | 00H 00H 00H 00H | 47H 23H 96H ABH | |
| 010001110010000111000111001011010101010111 | 010001110010000111000111001011010101010111 | 010001110010000111000111001011010101010111 | 010001110010000111000111001011010101010111 | 4 7 2 3 9 6 A B |

| | | | | | | | |
|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|
| 4 | 7 | 2 | 3 | 9 | 6 | A | B |
| 0 1 0 0 0 1 1 1 | 0 0 1 0 0 0 1 1 | 1 0 0 1 0 1 1 0 | 1 0 0 1 0 1 1 0 | 1 0 1 0 1 0 1 1 | 0 1 0 1 0 1 0 1 | 1 1 0 1 0 1 0 1 | 1 1 0 1 0 1 0 1 |

M107 to M100 ; M115 to M108 ; M123 to M116 ; M131 to M124

0 = OFF
1 = ON

Communication example (Writing word device)

Write '6549' (1995H) to D100, '4610' (1202H) to D101, and '4400' (1130H) to D102. (Subcommand: for MELSEC-Q/L series)

■Data communication using ASCII code

| Subcommand | Device code | Head device number | Number of device points | Write data | | |
|-------------------------------|-------------------------------|--------------------|---|-------------------------------|---|----------------|
| 1 4 0 1 31H, 34H, 30H, 31H | 0 0 0 0 30H, 30H, 30H, 30H | D * 44H, 2AH | 0 0 0 1 0 0 30H, 30H, 30H, 31H, 30H, 30H | 0 0 0 3 30H, 30H, 30H, 33H | 1 9 9 5 1 2 0 2 1 1 3 0 31H, 39H, 39H, 35H, 31H, 32H, 30H, 32H, 31H, 31H, 33H, 30H | D100 D101 D102 |

■Data communication using binary code

| Subcommand | Device code | Head device number | Number of device points | Write data |
|----------------------|----------------------|--|----------------------------|----------------|
| 01H, 14H 00H, 00H | 64H, 00H, 00H A8H | 03H, 00H 95H, 19H, 02H, 12H, 30H, 11H | | D100 D101 D102 |

Batch write in bit units (command: 1401)

Write values to devices in bit units.



When accessing any of the following devices, use the device extension specification (subcommand: 008□).

- Link direct device
- Module access device
- CPU buffer memory access device

For the message format for device extension specification, refer to the following section.

☞ Page 432 Read/Write by Device Extension Specification

Message format

The following shows the message format of the request data and response data of the command.

■Request data

| Command | Subcommand | Head device | Number of device points | Write data |
|---------|------------|-------------|-------------------------|------------|
|---------|------------|-------------|-------------------------|------------|

■Response data

There is no response data for this command.

Data specified by request data

■Command

| Frame | ASCII code | Binary code | | | | | | | | | | |
|-------------------|--|-------------|-----|---|---|-----|-----|-----|-----|---|-----|-----|
| 4C/3C/4E/3E frame | <table border="1"><tr><td>1</td><td>4</td><td>0</td><td>1</td></tr><tr><td>31H</td><td>34H</td><td>30H</td><td>31H</td></tr></table> | 1 | 4 | 0 | 1 | 31H | 34H | 30H | 31H | <table border="1"><tr><td>01H</td><td>14H</td></tr></table> | 01H | 14H |
| 1 | 4 | 0 | 1 | | | | | | | | | |
| 31H | 34H | 30H | 31H | | | | | | | | | |
| 01H | 14H | | | | | | | | | | | |
| 2C frame | <table border="1"><tr><td>3</td></tr><tr><td>33H</td></tr></table> | 3 | 33H | — | | | | | | | | |
| 3 | | | | | | | | | | | | |
| 33H | | | | | | | | | | | | |

■Subcommand

| Type | ASCII code | Binary code | | | | | | | | | | |
|------------------------|--|-------------|-----|---|---|-----|-----|-----|-----|---|-----|-----|
| For MELSEC-Q/L series | <table border="1"><tr><td>0</td><td>0</td><td>0</td><td>1</td></tr><tr><td>30H</td><td>30H</td><td>30H</td><td>31H</td></tr></table> | 0 | 0 | 0 | 1 | 30H | 30H | 30H | 31H | <table border="1"><tr><td>01H</td><td>00H</td></tr></table> | 01H | 00H |
| 0 | 0 | 0 | 1 | | | | | | | | | |
| 30H | 30H | 30H | 31H | | | | | | | | | |
| 01H | 00H | | | | | | | | | | | |
| For MELSEC iQ-R series | <table border="1"><tr><td>0</td><td>0</td><td>0</td><td>3</td></tr><tr><td>30H</td><td>30H</td><td>30H</td><td>33H</td></tr></table> | 0 | 0 | 0 | 3 | 30H | 30H | 30H | 33H | <table border="1"><tr><td>03H</td><td>00H</td></tr></table> | 03H | 00H |
| 0 | 0 | 0 | 3 | | | | | | | | | |
| 30H | 30H | 30H | 33H | | | | | | | | | |
| 03H | 00H | | | | | | | | | | | |

For 2C frame, the specification is not required. Functions and specification methods are equivalent to the subcommands for MELSEC-Q/L series.

■Head device

Specify the head device of the consecutive devices. (☞ Page 63 Devices)



Long index register (LZ) cannot be specified.

■Number of device points

Specify the number of device points to be written within the following range. (☞ Page 67 Number of device points)

| Access target | C24 | E71 | |
|--|------------------|------------------|------------------|
| | | ASCII code | Binary code |
| MELSEC iQ-R series module MELSEC-Q/L series module | 1 to 7904 points | 1 to 3584 points | 1 to 7168 points |
| MELSEC-QnA series module Module on other station via MELSEC-QnA series network module | 1 to 3952 points | 1 to 1792 points | 1 to 3584 points |
| MELSEC-A series module | 1 to 160 points | | |

■Write data

Specify the value to be written to a device for the number equivalent to the specified number of device points. (☞ Page 70 Read data, write data)

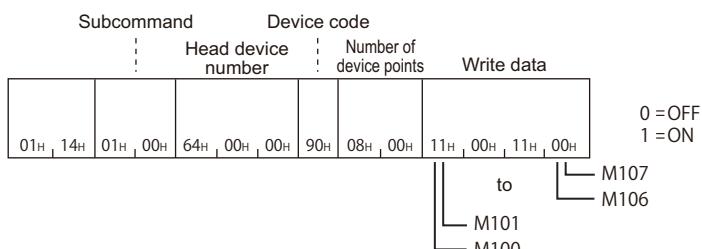
Communication example

Write values to M100 to M107. (Subcommand: for MELSEC-Q/L series)

■Data communication using ASCII code

| Subcommand | Device code | Head device number | Number of device points | Write data | | |
|-------------------------------|-------------------------------|--------------------|---|-------------------------------|---|-----------------------------------|
| 1 4 0 1 31H, 34H, 30H, 31H | 0 0 0 1 30H, 30H, 30H, 31H | M * 4DH, 2AH | 0 0 0 1 0 0 0 30H, 30H, 30H, 31H, 30H, 30H | 0 0 0 8 30H, 30H, 30H, 38H | 1 1 0 0 1 1 0 0 31H, 31H, 30H, 30H, 31H, 31H, 30H, 30H | M100 to M107 0 = OFF 1 = ON |

■Data communication using binary code



8.3 Random Read and Write

Read or write device values by specifying the device numbers. It can be specified with discontinuous device numbers.

Random read in word units (command: 0403)

Read values from devices in word units and double word units. It can be specified with discontinuous device number.



When accessing any of the following devices, use the device extension specification (subcommand: 008□).

- Link direct device
- Module access device
- CPU buffer memory access device

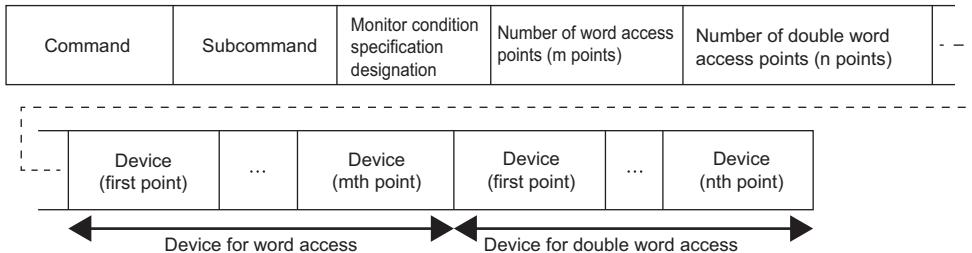
For the message format for device extension specification, refer to the following section.

☞ Page 432 Read/Write by Device Extension Specification

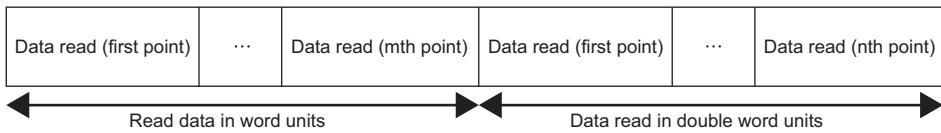
Message format

The following shows the message format of the request data and response data of the command.

■Request data



■Response data



The value of read device is stored in word units and in double word units. The data order differs between ASCII code or binary code. (☞ Page 70 Read data, write data)

Data specified by request data

■Command

| Frame | ASCII code | Binary code | | | | | | | | | | |
|-------------------|--|-------------|-----|---|---|-----|-----|-----|-----|---|-----|-----|
| 4C/3C/4E/3E frame | <table border="1"><tr><td>0</td><td>4</td><td>0</td><td>3</td></tr><tr><td>30H</td><td>34H</td><td>30H</td><td>33H</td></tr></table> | 0 | 4 | 0 | 3 | 30H | 34H | 30H | 33H | <table border="1"><tr><td>03H</td><td>04H</td></tr></table> | 03H | 04H |
| 0 | 4 | 0 | 3 | | | | | | | | | |
| 30H | 34H | 30H | 33H | | | | | | | | | |
| 03H | 04H | | | | | | | | | | | |
| 2C frame | <table border="1"><tr><td>5</td></tr><tr><td>35H</td></tr></table> | 5 | 35H | — | | | | | | | | |
| 5 | | | | | | | | | | | | |
| 35H | | | | | | | | | | | | |

■Subcommand

The read timing can be changed by specifying monitor conditions according to the selection of subcommand. (☞ Page 77 Monitor condition specification)

For 2C frame, the specification is not required. Functions and specification methods are equivalent to the subcommands that do not specify monitor conditions for MELSEC-Q/L series.

When do not specify monitor condition

| Type | ASCII code | Binary code | | | | | | | | | | |
|------------------------|--|-------------|-----|---|---|-----|-----|-----|-----|---|-----|-----|
| For MELSEC-Q/L series | <table border="1"><tr><td>0</td><td>0</td><td>0</td><td>0</td></tr><tr><td>30H</td><td>30H</td><td>30H</td><td>30H</td></tr></table> | 0 | 0 | 0 | 0 | 30H | 30H | 30H | 30H | <table border="1"><tr><td>00H</td><td>00H</td></tr></table> | 00H | 00H |
| 0 | 0 | 0 | 0 | | | | | | | | | |
| 30H | 30H | 30H | 30H | | | | | | | | | |
| 00H | 00H | | | | | | | | | | | |
| For MELSEC iQ-R series | <table border="1"><tr><td>0</td><td>0</td><td>0</td><td>2</td></tr><tr><td>30H</td><td>30H</td><td>30H</td><td>32H</td></tr></table> | 0 | 0 | 0 | 2 | 30H | 30H | 30H | 32H | <table border="1"><tr><td>02H</td><td>00H</td></tr></table> | 02H | 00H |
| 0 | 0 | 0 | 2 | | | | | | | | | |
| 30H | 30H | 30H | 32H | | | | | | | | | |
| 02H | 00H | | | | | | | | | | | |

When monitoring condition is specified

| Type | ASCII code | Binary code | | | | | | | | | | |
|-----------------------|--|-------------|-----|---|---|-----|-----|-----|-----|---|-----|-----|
| For MELSEC-Q/L series | <table border="1"><tr><td>0</td><td>0</td><td>4</td><td>0</td></tr><tr><td>30H</td><td>30H</td><td>34H</td><td>30H</td></tr></table> | 0 | 0 | 4 | 0 | 30H | 30H | 34H | 30H | <table border="1"><tr><td>40H</td><td>00H</td></tr></table> | 40H | 00H |
| 0 | 0 | 4 | 0 | | | | | | | | | |
| 30H | 30H | 34H | 30H | | | | | | | | | |
| 40H | 00H | | | | | | | | | | | |

*1 The access targets to which a monitor conditions can be specified have some restrictions. (☞ Page 465 Accessible Modules for Each Command)



At monitor condition specification, use the subcommand 00C0 for the device extension specification. The message format for device extension specification is the same as that of 008□. Refer to it by substituting 008□ to 00C0.

☞ Page 432 Read/Write by Device Extension Specification

■Monitor condition specification

Specify the conditions for timing to read data. (☞ Page 77 Monitor condition specification)

When do not specify the monitor condition, the specification of this data item is not required.

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

Specify the number of device points to be read within the following range. (☞ Page 68 Access points)

| Access target | Range |
|---|---|
| MELSEC iQ-R series module (subcommand: 0000) MELSEC-Q/L series module (subcommand: 0000) | 1 ≤ Number of word access points + Number of double word access points ≤ 192 points |
| MELSEC iQ-R series module (subcommand: 0002, 008□) MELSEC-Q/L series module (subcommand: 0080) MELSEC-QnA series module Module on other station via MELSEC-QnA series network module | 1 ≤ Number of word access points + Number of double word access points ≤ 96 points |
| MELSEC-A series module | Cannot be used. |

The number of point is specified in the following units depending on device type.

| Device type | Number of word access points | Number of double word access points |
|-------------|------------------------------|-------------------------------------|
| Bit device | 16-point units | 32-point units |
| Word device | 1 word units | 2 word units |



When using subcommand for MELSEC-Q/L series module, calculate it as access points × 2 in the following case.

- When specifying the file register (ZR) of High Performance model QCPU

■Device

Specify the device to be read. ( Page 63 Devices)

Point

The number of points equivalent to the specified 'Number of word access points' and 'Number of double word access points' is specified for 'Device', respectively. When '0' is specified for the access points, this specification is not required.

Communication example (Monitor condition is not specified)

Read values of D0, T0, M100 to M115, X20 to X2F with word access. Read values of D1500 to D1501, Y160 to Y17F, M1111 to M1142 with 4 double word access. (Subcommand: for MELSEC-Q/L series)

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

■Data communication using ASCII code

(Request data)

| Subcommand | Number of word access points | Number of double word access points |
|-------------------------------|-------------------------------|-------------------------------------|
| 0 4 0 3 30H, 34H, 30H, 33H | 0 0 0 0 30H, 30H, 30H, 30H | 0 4 0 3 30H, 34H, 30H, 33H |

Device for word access

| Device code | Device number | Device code | Device number | Device code | Device number | Device code | Device number |
|--|--|--|--|-------------|---------------|-------------|---------------|
| D * 0 0 0 0 0 0 44H, 2AH, 30H, 30H, 30H, 30H, 30H | T N 0 0 0 0 0 0 54H, 4EH, 30H, 30H, 30H, 30H, 30H | M * 0 0 0 1 0 0 4DH, 2AH, 30H, 30H, 31H, 30H, 30H | X * 0 0 0 2 0 0 58H, 2AH, 30H, 30H, 30H, 32H, 30H | | | | |

Device for double word access

| Device code | Device number | Device code | Device number | Device code | Device number |
|---|---|--|---------------|-------------|---------------|
| D * 0 0 1 5 0 0 44H, 2AH, 30H, 30H, 31H, 35H, 30H, 30H | Y * 0 0 0 1 6 0 59H, 2AH, 30H, 30H, 30H, 31H, 36H, 30H | M * 0 0 1 1 1 1 4DH, 2AH, 30H, 30H, 31H, 31H, 31H | | | |

(Response data)

| Read data 1 in word units | Read data 2 in word units | Read data 3 in word units | Read data 4 in word units |
|-------------------------------|-------------------------------|-------------------------------|-------------------------------|
| 1 9 9 5 31H, 39H, 39H, 35H | 1 2 0 2 31H, 32H, 30H, 32H | 2 0 3 0 32H, 30H, 33H, 30H | 4 8 4 9 34H, 38H, 34H, 39H |

D0 T0 M115 to M100 X2F to X20

| Read data 1 in double word units | Read data 2 in double word units | Read data 3 in double word units |
|---|---|--|
| 4 C 5 4 4 F 4 E 34H, 43H, 35H, 34H, 34H, 46H, 34H, 45H | C 3 D E B 9 A F 43H, 33H, 44H, 45H, 42H, 39H, 41H, 46H | B A D D B C B 7 42H, 41H, 44H, 42H, 43H, 42H, 37H |

D1501 D1500 Y17F to Y160 M1142 to M1111

Read data 3 in word units

| | | | |
|---------------------------------|-----------------|---|---|
| 2 | 0 | 3 | 0 |
| 0 0 1 0 0 0 0 0 0 0 1 1 0 0 0 0 | 0 =OFF 1 =ON | | |

M115 to M100

Read data 2 in double word units

| | |
|-----------------------------|-----------------|
| C | F |
| 1 1 0 0 0 0 - - 1 0 1 1 1 1 | 0 =OFF 1 =ON |

Y17F to Y160

Read data 4 in word units

| | | | |
|---------------------------------|-----------------|---|---|
| 4 | 8 | 4 | 9 |
| 0 1 0 0 1 0 0 0 0 1 0 0 1 0 0 1 | 0 =OFF 1 =ON | | |

X2F to X20

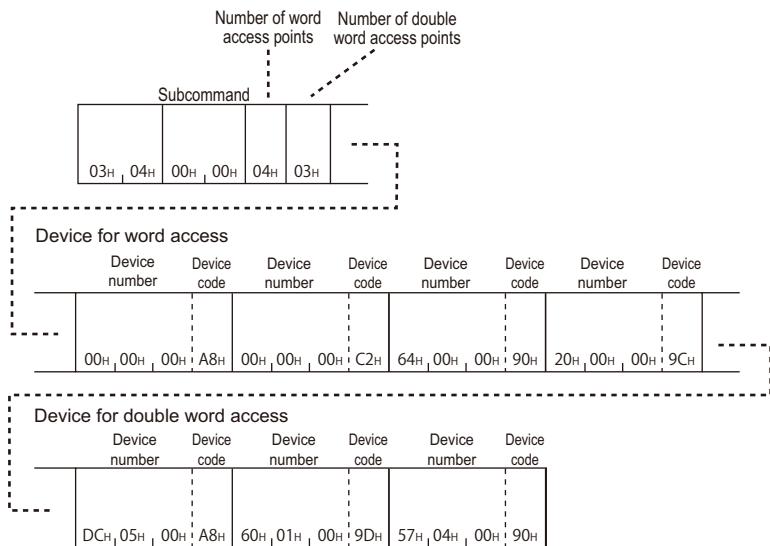
Read data 3 in double word units

| | | |
|-----------------------------------|-----------------|---|
| B | B | 7 |
| 1 0 1 1 1 0 1 - - 0 1 0 1 1 0 1 1 | 0 =OFF 1 =ON | |

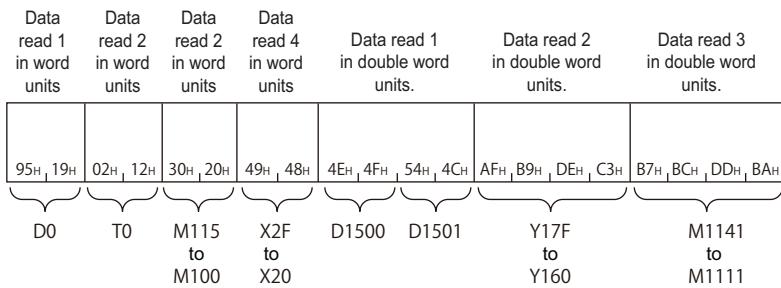
M1142 to M1111

■Data communication using binary code

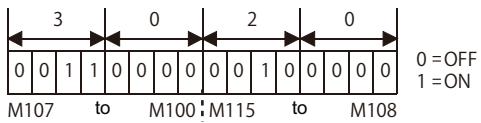
(Request data)



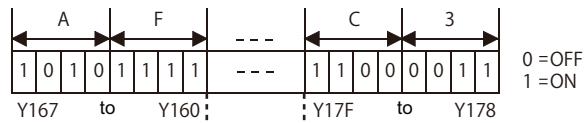
(Response data)



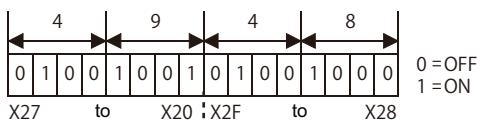
Data read 3 in word units



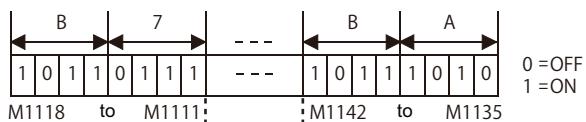
Data read 2 in double word units



Data read 4 in word units



Data read 3 in double word units



Communication example (Monitor condition is specified)

Read values of D0, T0, M100 to M115, X20 to X2F with word access. Read values of D1500 to D1501, Y160 to Y17F, M1111 to M1142 with 4 double word access.

The monitor condition is as follows: When the value of link register (W100) reached '7BH' (123) while the step No.1000 of program file CONB1.QPG is being executed.

■Data communication using ASCII code

(Request data)

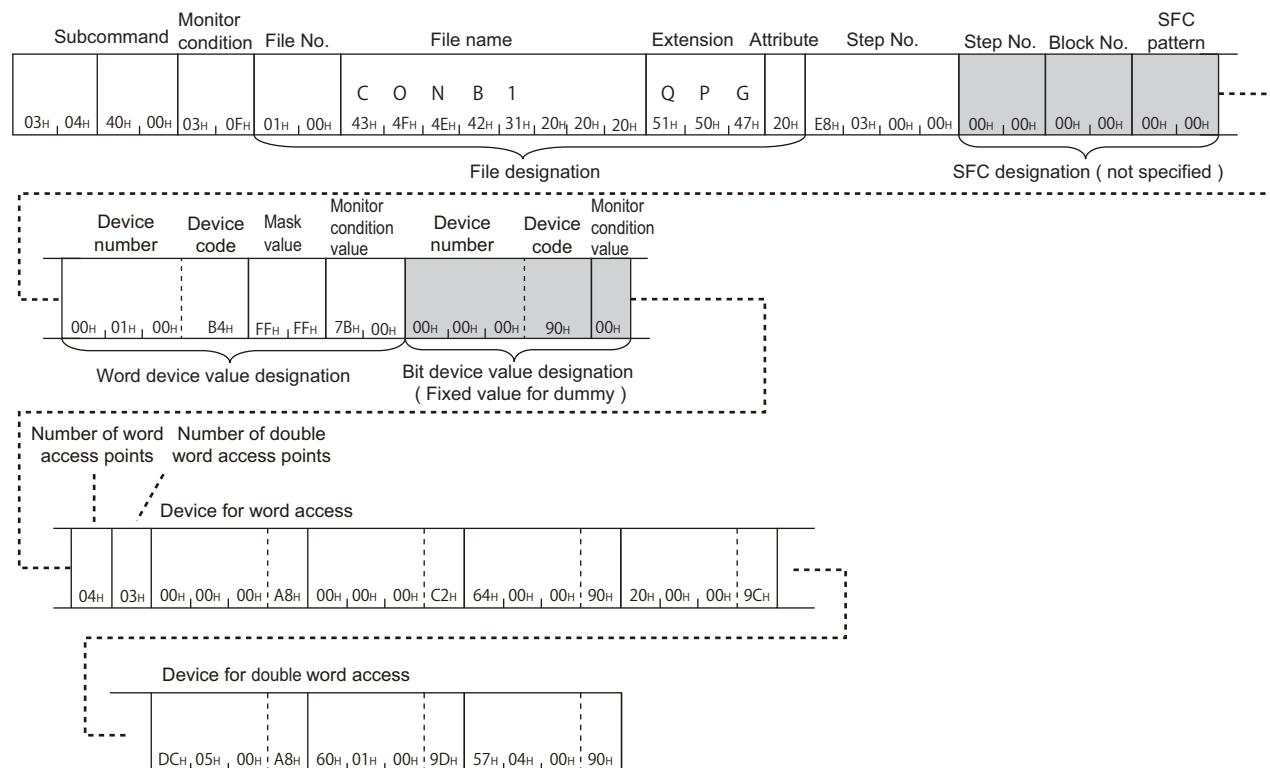
| Subcommand | Monitor condition | File No. | File name | Extension | Attribute | | | | | | | | | | | | | | | |
|---|---|--|--|---|---|---|---|--|--|-------------|---------------|-------------------------------|---|-------------------------------|--|--|--|--|-----------------|--|
| 0 4 0 3 30H, 34H, 30H, 33H | 0 0 4 0 30H, 30H, 34H, 30H | 0 3 30H, 33H | 0 F 30H, 46H | 0 0 0 1 30H, 30H, 30H, 31H | C O N B 1 43H, 4FH, 4EH, 42H, 31H, 20H, 20H, 20H 51H, 50H, 47H, 20H | | | | | | | | | | | | | | | |
| File designation | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | |
| <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>SFC pattern</th> <th>Block No.</th> <th>Step No.</th> <th>Step No.</th> <th colspan="2"></th> </tr> </thead> <tbody> <tr> <td>0 0 0 0 30H, 30H, 30H, 30H</td> <td>0 0 0 0 30H, 30H, 30H, 30H</td> <td>0 0 0 0 30H, 30H, 30H, 30H</td> <td>0 0 0 0 3 E 8 30H, 30H, 30H, 30H, 33H, 45H, 38H</td> <td colspan="2"></td> </tr> </tbody> </table> | | | | | | SFC pattern | Block No. | Step No. | Step No. | | | 0 0 0 0 30H, 30H, 30H, 30H | 0 0 0 0 30H, 30H, 30H, 30H | 0 0 0 0 30H, 30H, 30H, 30H | 0 0 0 0 3 E 8 30H, 30H, 30H, 30H, 33H, 45H, 38H | | | | | |
| SFC pattern | Block No. | Step No. | Step No. | | | | | | | | | | | | | | | | | |
| 0 0 0 0 30H, 30H, 30H, 30H | 0 0 0 0 30H, 30H, 30H, 30H | 0 0 0 0 30H, 30H, 30H, 30H | 0 0 0 0 3 E 8 30H, 30H, 30H, 30H, 33H, 45H, 38H | | | | | | | | | | | | | | | | | |
| <td data-cs="6" data-kind="parent" style="text-align: center;">SFC designation (not specified)</td> <td data-kind="ghost"></td> <td data-kind="ghost"></td> <td data-kind="ghost"></td> <td data-kind="ghost"></td> <td data-kind="ghost"></td> | SFC designation (not specified) | | | | | | | | | | | | | | | | | | | |
| <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>Device code</th> <th>Device number</th> <th>Mask value</th> <th>Monitor condition value</th> <th>Device code</th> <th>Device number</th> <th>Monitor condition value</th> </tr> </thead> <tbody> <tr> <td>W * 0 0 0 1 0 0 57H, 2AH, 30H, 30H, 30H, 31H, 30H, 30H</td> <td>F F F F 46H, 46H, 46H, 46H</td> <td>0 0 7 B 30H, 30H, 37H, 42H</td> <td></td> <td>M * 0 0 0 0 0 0 4DH, 2AH, 30H, 30H, 30H, 30H, 30H</td> <td>0 0 0 0 0 0 30H, 30H, 30H, 30H, 30H</td> <td>0 0 30H, 30H</td> </tr> </tbody> </table> | | | | | | Device code | Device number | Mask value | Monitor condition value | Device code | Device number | Monitor condition value | W * 0 0 0 1 0 0 57H, 2AH, 30H, 30H, 30H, 31H, 30H, 30H | F F F F 46H, 46H, 46H, 46H | 0 0 7 B 30H, 30H, 37H, 42H | | M * 0 0 0 0 0 0 4DH, 2AH, 30H, 30H, 30H, 30H, 30H | 0 0 0 0 0 0 30H, 30H, 30H, 30H, 30H | 0 0 30H, 30H | |
| Device code | Device number | Mask value | Monitor condition value | Device code | Device number | Monitor condition value | | | | | | | | | | | | | | |
| W * 0 0 0 1 0 0 57H, 2AH, 30H, 30H, 30H, 31H, 30H, 30H | F F F F 46H, 46H, 46H, 46H | 0 0 7 B 30H, 30H, 37H, 42H | | M * 0 0 0 0 0 0 4DH, 2AH, 30H, 30H, 30H, 30H, 30H | 0 0 0 0 0 0 30H, 30H, 30H, 30H, 30H | 0 0 30H, 30H | | | | | | | | | | | | | | |
| <td data-cs="3" data-kind="parent" style="text-align: center;">Word device value designation</td> <td data-kind="ghost"></td> <td data-kind="ghost"></td> <td data-cs="3" data-kind="parent" style="text-align: center;">Bit device value designation (Fixed value for dummy)</td> <td data-kind="ghost"></td> <td data-kind="ghost"></td> | Word device value designation | | | Bit device value designation (Fixed value for dummy) | | | | | | | | | | | | | | | | |
| <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>Number of word access points</th> <th>Number of double word access points</th> <th colspan="5"></th> </tr> </thead> <tbody> <tr> <td>0 4 30H, 34H</td> <td>0 3 30H, 33H</td> <td colspan="5"></td> </tr> </tbody> </table> | | | | | | | Number of word access points | Number of double word access points | | | | | | 0 4 30H, 34H | 0 3 30H, 33H | | | | | |
| Number of word access points | Number of double word access points | | | | | | | | | | | | | | | | | | | |
| 0 4 30H, 34H | 0 3 30H, 33H | | | | | | | | | | | | | | | | | | | |
| <p>Device for word access</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tbody> <tr> <td>D * 0 0 0 0 0 0 44H, 2AH, 30H, 30H, 30H, 30H, 30H</td> <td>T N 0 0 0 0 0 0 54H, 4EH, 30H, 30H, 30H, 30H, 30H</td> <td>M * 0 0 0 1 0 0 4DH, 2AH, 30H, 30H, 31H, 30H, 30H</td> <td>X * 0 0 0 0 2 0 58H, 2AH, 30H, 30H, 30H, 32H, 30H</td> </tr> </tbody> </table> | | | | | | D * 0 0 0 0 0 0 44H, 2AH, 30H, 30H, 30H, 30H, 30H | T N 0 0 0 0 0 0 54H, 4EH, 30H, 30H, 30H, 30H, 30H | M * 0 0 0 1 0 0 4DH, 2AH, 30H, 30H, 31H, 30H, 30H | X * 0 0 0 0 2 0 58H, 2AH, 30H, 30H, 30H, 32H, 30H | | | | | | | | | | | |
| D * 0 0 0 0 0 0 44H, 2AH, 30H, 30H, 30H, 30H, 30H | T N 0 0 0 0 0 0 54H, 4EH, 30H, 30H, 30H, 30H, 30H | M * 0 0 0 1 0 0 4DH, 2AH, 30H, 30H, 31H, 30H, 30H | X * 0 0 0 0 2 0 58H, 2AH, 30H, 30H, 30H, 32H, 30H | | | | | | | | | | | | | | | | | |
| <p>Device for double word access</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tbody> <tr> <td>D * 0 0 1 5 0 0 44H, 2AH, 30H, 30H, 31H, 35H, 30H, 30H</td> <td>Y * 0 0 0 1 6 0 59H, 2AH, 30H, 30H, 30H, 31H, 36H, 30H</td> <td>M * 0 0 1 1 1 1 4DH, 2AH, 30H, 30H, 31H, 31H, 31H</td> </tr> </tbody> </table> | | | | | | D * 0 0 1 5 0 0 44H, 2AH, 30H, 30H, 31H, 35H, 30H, 30H | Y * 0 0 0 1 6 0 59H, 2AH, 30H, 30H, 30H, 31H, 36H, 30H | M * 0 0 1 1 1 1 4DH, 2AH, 30H, 30H, 31H, 31H, 31H | | | | | | | | | | | | |
| D * 0 0 1 5 0 0 44H, 2AH, 30H, 30H, 31H, 35H, 30H, 30H | Y * 0 0 0 1 6 0 59H, 2AH, 30H, 30H, 30H, 31H, 36H, 30H | M * 0 0 1 1 1 1 4DH, 2AH, 30H, 30H, 31H, 31H, 31H | | | | | | | | | | | | | | | | | | |

(Response data)

It is the same as the communication example when monitor condition is not specified. ([Page 97 Data communication using ASCII code](#))

■Data communication using binary code

(Request data)



(Response data)

It is the same as the communication example when monitor condition is not specified. (Page 98 Data communication using binary code)

Random write in word units (test) (command: 1402)

Write values to devices in word units and double word units. It can be specified with discontinuous device numbers.



When accessing any of the following devices, use the device extension specification (subcommand: 008□).

- Link direct device
- Module access device
- CPU buffer memory access device

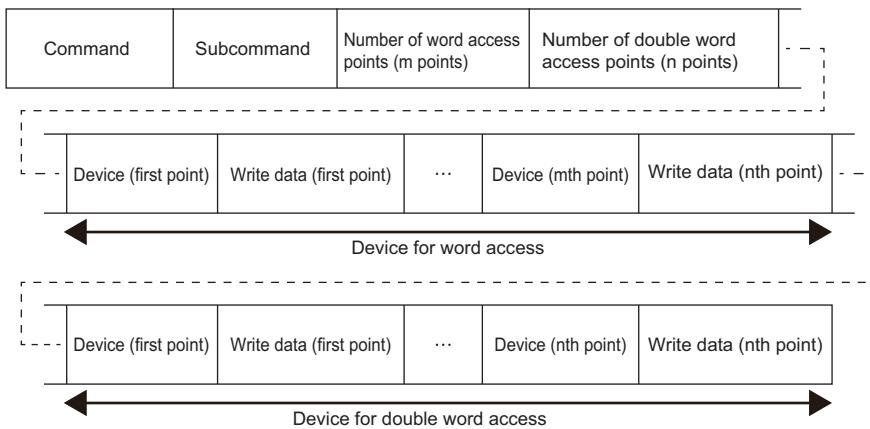
For the message format for device extension specification, refer to the following section.

☞ Page 432 Read/Write by Device Extension Specification

Message format

The following shows the message format of the request data and response data of the command.

■Request data



8

■Response data

There is no response data for this command.

Data specified by request data

■Command

| Frame | ASCII code | Binary code | | | | | | | | | | |
|-------------------|--|-------------|-----|---|---|-----|-----|-----|-----|---|-----|-----|
| 4C/3C/4E/3E frame | <table border="1"><tr><td>1</td><td>4</td><td>0</td><td>2</td></tr><tr><td>31H</td><td>34H</td><td>30H</td><td>32H</td></tr></table> | 1 | 4 | 0 | 2 | 31H | 34H | 30H | 32H | <table border="1"><tr><td>02H</td><td>14H</td></tr></table> | 02H | 14H |
| 1 | 4 | 0 | 2 | | | | | | | | | |
| 31H | 34H | 30H | 32H | | | | | | | | | |
| 02H | 14H | | | | | | | | | | | |
| 2C frame | <table border="1"><tr><td>7</td></tr><tr><td>37H</td></tr></table> | 7 | 37H | — | | | | | | | | |
| 7 | | | | | | | | | | | | |
| 37H | | | | | | | | | | | | |

■Subcommand

| Type | ASCII code | Binary code | | | | | | | | | | |
|------------------------|--|-------------|-----|---|---|-----|-----|-----|-----|---|-----|-----|
| For MELSEC-Q/L series | <table border="1"><tr><td>0</td><td>0</td><td>0</td><td>0</td></tr><tr><td>30H</td><td>30H</td><td>30H</td><td>30H</td></tr></table> | 0 | 0 | 0 | 0 | 30H | 30H | 30H | 30H | <table border="1"><tr><td>00H</td><td>00H</td></tr></table> | 00H | 00H |
| 0 | 0 | 0 | 0 | | | | | | | | | |
| 30H | 30H | 30H | 30H | | | | | | | | | |
| 00H | 00H | | | | | | | | | | | |
| For MELSEC iQ-R series | <table border="1"><tr><td>0</td><td>0</td><td>0</td><td>2</td></tr><tr><td>30H</td><td>30H</td><td>30H</td><td>32H</td></tr></table> | 0 | 0 | 0 | 2 | 30H | 30H | 30H | 32H | <table border="1"><tr><td>02H</td><td>00H</td></tr></table> | 02H | 00H |
| 0 | 0 | 0 | 2 | | | | | | | | | |
| 30H | 30H | 30H | 32H | | | | | | | | | |
| 02H | 00H | | | | | | | | | | | |

For 2C frame, the specification is not required. Functions and specification methods are equivalent to the subcommands for MELSEC-Q/L series.

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

Specify the number of device points to be written within the following range. (☞ Page 68 Access points)

| Access target | Range |
|---|--|
| MELSEC iQ-R series module (subcommand: 0000) MELSEC-Q/L series module (subcommand: 0000) | $1 \leq (\text{Number of word access points} \times 12) + (\text{Number of double word access points} \times 14) \leq 1920$ points |
| MELSEC iQ-R series module (subcommand: 0002, 008□) MELSEC-Q/L series module (subcommand: 0080) MELSEC-QnA series module Module on other station via MELSEC-QnA series network module | $1 \leq (\text{Number of word access points} \times 12) + (\text{Number of double word access points} \times 14) \leq 960$ points |
| MELSEC-A series module | $1 \leq \text{Number of word access points} \leq 10$ points |

The number of point is specified in the following units depending on device type.

| Device type | Number of word access points | Number of word access points |
|-------------|------------------------------|------------------------------|
| Bit device | 16-point units | 32-point units |
| Word device | 1 word units | 2 word units |

■Device

Specify a device to be written. (☞ Page 63 Devices)

Set the head device number with a multiple of 16 for bit device access of MELSEC-A series module.

■Write data

Specify the values to be written to device. (☞ Page 70 Read data, write data)

Specify the write data in hexadecimal.



Specify 'Device' and 'Write data' with number of points which are specified in 'Number of word access points' and 'Number of double word access points'. When the access point is set to '0', the specification is not required.

Communication example

Write values to devices as follows. (Subcommand: for MELSEC-Q/L series)

| Item | Device to be written |
|--------------------|--|
| Word access | D0, D1, M100 to M115, X20 to X2F |
| Double word access | D1500 to D1501, Y160 to Y17F, M1111 to M1142 |

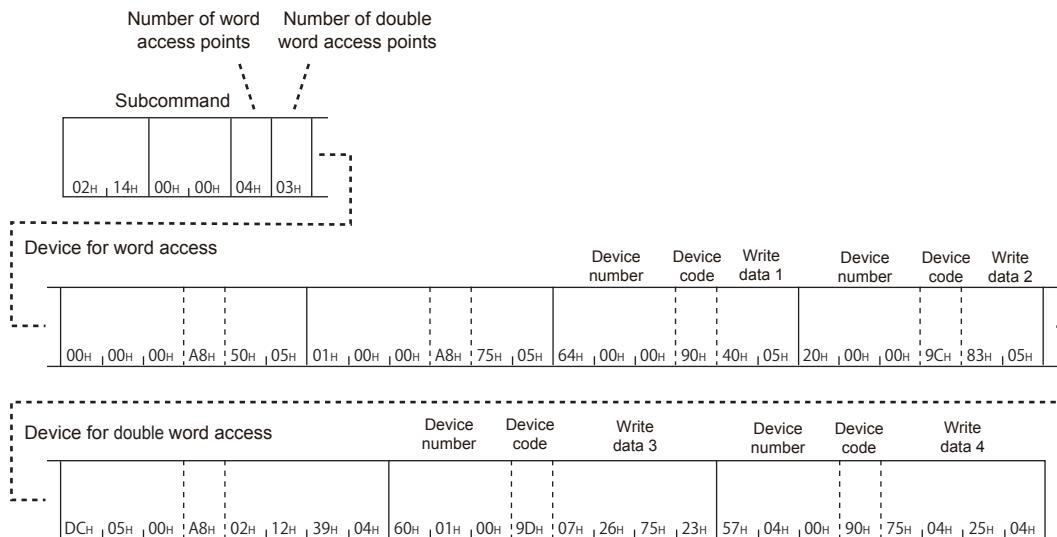
■Data communication using ASCII code

(Request data)

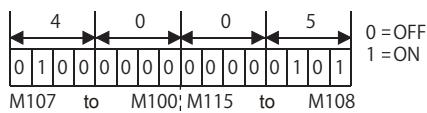
| Subcommand | Number of word access points | Number of double word access points | | | | | | | | | |
|--|--|---------------------------------------|-----------------|-----------------|---------------------------------|---------------------------------------|--|---------------------------------------|-------------------------|--|--|
| 1 4 0 2 31H 34H 30H 32H | 0 0 0 0 30H 30H 30H 30H | 0 4 0 3 30H 34H 30H 33H | | | | | | | | | |
| Device for word access | | | | | | | | | | | |
| <table border="1"> <tr> <td>D * 0 0 0 0 0 0</td> <td>0 5 5 0</td> <td>D * 0 0 0 0 1 0 5 7 5</td> </tr> <tr> <td>44H, 2AH 30H, 30H, 30H, 30H, 30H</td> <td>30H, 35H, 35H, 30H</td> <td>44H, 2AH 30H, 30H, 30H, 30H, 31H</td> </tr> <tr> <td>30H, 30H, 30H, 30H, 30H</td> <td>30H, 35H, 37H, 35H</td> <td></td> </tr> </table> | | | D * 0 0 0 0 0 0 | 0 5 5 0 | D * 0 0 0 0 1 0 5 7 5 | 44H, 2AH 30H, 30H, 30H, 30H, 30H | 30H, 35H, 35H, 30H | 44H, 2AH 30H, 30H, 30H, 30H, 31H | 30H, 30H, 30H, 30H, 30H | 30H, 35H, 37H, 35H | |
| D * 0 0 0 0 0 0 | 0 5 5 0 | D * 0 0 0 0 1 0 5 7 5 | | | | | | | | | |
| 44H, 2AH 30H, 30H, 30H, 30H, 30H | 30H, 35H, 35H, 30H | 44H, 2AH 30H, 30H, 30H, 30H, 31H | | | | | | | | | |
| 30H, 30H, 30H, 30H, 30H | 30H, 35H, 37H, 35H | | | | | | | | | | |
| Device code Device number Write data 1 Device code Device number Write data 2 | | | | | | | | | | | |
| <table border="1"> <tr> <td>M * 0 0 0 1 0 0</td> <td>0 5 4 0</td> <td>X * 0 0 0 0 2 0 0 5 8 3</td> </tr> <tr> <td>4DH, 2AH 30H, 30H, 30H, 31H, 30H</td> <td>30H, 35H, 34H, 30H</td> <td>58H, 2AH 30H, 30H, 30H, 32H, 30H</td> </tr> <tr> <td>30H, 30H, 30H, 30H, 30H</td> <td>30H, 35H, 38H, 33H</td> <td></td> </tr> </table> | | | M * 0 0 0 1 0 0 | 0 5 4 0 | X * 0 0 0 0 2 0 0 5 8 3 | 4DH, 2AH 30H, 30H, 30H, 31H, 30H | 30H, 35H, 34H, 30H | 58H, 2AH 30H, 30H, 30H, 32H, 30H | 30H, 30H, 30H, 30H, 30H | 30H, 35H, 38H, 33H | |
| M * 0 0 0 1 0 0 | 0 5 4 0 | X * 0 0 0 0 2 0 0 5 8 3 | | | | | | | | | |
| 4DH, 2AH 30H, 30H, 30H, 31H, 30H | 30H, 35H, 34H, 30H | 58H, 2AH 30H, 30H, 30H, 32H, 30H | | | | | | | | | |
| 30H, 30H, 30H, 30H, 30H | 30H, 35H, 38H, 33H | | | | | | | | | | |
| Device for double word access | | | | | | | | | | | |
| <table border="1"> <tr> <td>D * 0 0 1 5 0 0</td> <td>0 4 3 9 1 2 0 2</td> <td>Y * 0 0 0 1 6 0 2 3 7 5 2 6 0 7</td> </tr> <tr> <td>44H, 2AH 30H, 30H, 31H, 35H, 30H, 30H</td> <td>30H, 34H, 33H, 39H, 31H, 32H, 30H, 32H</td> <td>59H, 2AH 30H, 30H, 30H, 31H, 36H, 30H</td> </tr> <tr> <td>30H, 30H, 30H, 30H, 30H</td> <td>32H, 33H, 37H, 35H, 32H, 36H, 30H, 37H</td> <td></td> </tr> </table> | | | D * 0 0 1 5 0 0 | 0 4 3 9 1 2 0 2 | Y * 0 0 0 1 6 0 2 3 7 5 2 6 0 7 | 44H, 2AH 30H, 30H, 31H, 35H, 30H, 30H | 30H, 34H, 33H, 39H, 31H, 32H, 30H, 32H | 59H, 2AH 30H, 30H, 30H, 31H, 36H, 30H | 30H, 30H, 30H, 30H, 30H | 32H, 33H, 37H, 35H, 32H, 36H, 30H, 37H | |
| D * 0 0 1 5 0 0 | 0 4 3 9 1 2 0 2 | Y * 0 0 0 1 6 0 2 3 7 5 2 6 0 7 | | | | | | | | | |
| 44H, 2AH 30H, 30H, 31H, 35H, 30H, 30H | 30H, 34H, 33H, 39H, 31H, 32H, 30H, 32H | 59H, 2AH 30H, 30H, 30H, 31H, 36H, 30H | | | | | | | | | |
| 30H, 30H, 30H, 30H, 30H | 32H, 33H, 37H, 35H, 32H, 36H, 30H, 37H | | | | | | | | | | |
| Device code Device number Write data 3 | | | | | | | | | | | |
| <table border="1"> <tr> <td>M * 0 0 1 1 1 1</td> <td>0 4 2 5 0 4 7 5</td> <td></td> </tr> <tr> <td>4DH, 2AH 30H, 30H, 31H, 31H, 31H</td> <td>30H, 34H, 32H, 35H, 30H, 34H, 37H, 35H</td> <td></td> </tr> </table> | | | M * 0 0 1 1 1 1 | 0 4 2 5 0 4 7 5 | | 4DH, 2AH 30H, 30H, 31H, 31H, 31H | 30H, 34H, 32H, 35H, 30H, 34H, 37H, 35H | | | | |
| M * 0 0 1 1 1 1 | 0 4 2 5 0 4 7 5 | | | | | | | | | | |
| 4DH, 2AH 30H, 30H, 31H, 31H, 31H | 30H, 34H, 32H, 35H, 30H, 34H, 37H, 35H | | | | | | | | | | |
| Device code Device number Write data 4 | | | | | | | | | | | |
| <table border="1"> <tr> <td>M * 0 0 1 1 1 1</td> <td>0 4 2 5 0 4 7 5</td> <td></td> </tr> <tr> <td>4DH, 2AH 30H, 30H, 31H, 31H, 31H</td> <td>30H, 34H, 32H, 35H, 30H, 34H, 37H, 35H</td> <td></td> </tr> </table> | | | M * 0 0 1 1 1 1 | 0 4 2 5 0 4 7 5 | | 4DH, 2AH 30H, 30H, 31H, 31H, 31H | 30H, 34H, 32H, 35H, 30H, 34H, 37H, 35H | | | | |
| M * 0 0 1 1 1 1 | 0 4 2 5 0 4 7 5 | | | | | | | | | | |
| 4DH, 2AH 30H, 30H, 31H, 31H, 31H | 30H, 34H, 32H, 35H, 30H, 34H, 37H, 35H | | | | | | | | | | |
| Write data 1 in word units | | | | | | | | | | | |
| <p>0 = OFF 1 = ON</p> | | | | | | | | | | | |
| M115 to M100 | | | | | | | | | | | |
| Write data 3 in double word units | | | | | | | | | | | |
| <p>0 = OFF 1 = ON</p> | | | | | | | | | | | |
| Y17F to Y178; Y167 to Y160 | | | | | | | | | | | |
| Write data 2 in word units | | | | | | | | | | | |
| <p>0 = OFF 1 = ON</p> | | | | | | | | | | | |
| X2F to X20 | | | | | | | | | | | |
| Write data 4 in double word units | | | | | | | | | | | |
| <p>0 = OFF 1 = ON</p> | | | | | | | | | | | |
| M1142 to M1135; M1118 to M1111 | | | | | | | | | | | |

■Data communication using binary code

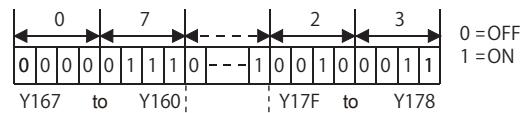
(Request data)



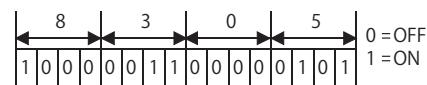
Write data 1 in word units



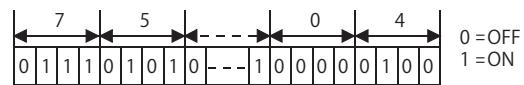
Write data 3 in double word units



Write data 2 in word units



Write data 4 in double word units



Random write in bit units (test) (command: 1402)

Write values to devices in bit units. It can be specified with discontinuous device number.



When accessing any of the following devices, use the device extension specification (subcommand: 008□).

- Link direct device
- Module access device
- CPU buffer memory access device

For the message format for device extension specification, refer to the following section.

☞ Page 432 Read/Write by Device Extension Specification

Message format

The following shows the message format of the request data and response data of the command.

■Request data

| Command | Subcommand | Number of bit access points (n points) | Device (first point) | Set/reset (first point) | ... | Device (nth point) | Set/reset (nth point) |
|---------|------------|--|----------------------|-------------------------|-----|--------------------|-----------------------|
|---------|------------|--|----------------------|-------------------------|-----|--------------------|-----------------------|

■Response data

There is no response data for this command.

Request data

■Command

| Frame | ASCII code | Binary code | | | | | | | | | | |
|-------------------|--|-------------|-----|---|---|-----|-----|-----|-----|---|-----|-----|
| 4C/3C/4E/3E frame | <table border="1"><tr><td>1</td><td>4</td><td>0</td><td>2</td></tr><tr><td>31H</td><td>34H</td><td>30H</td><td>32H</td></tr></table> | 1 | 4 | 0 | 2 | 31H | 34H | 30H | 32H | <table border="1"><tr><td>02H</td><td>14H</td></tr></table> | 02H | 14H |
| 1 | 4 | 0 | 2 | | | | | | | | | |
| 31H | 34H | 30H | 32H | | | | | | | | | |
| 02H | 14H | | | | | | | | | | | |
| 2C frame | <table border="1"><tr><td>6</td></tr><tr><td>36H</td></tr></table> | 6 | 36H | — | | | | | | | | |
| 6 | | | | | | | | | | | | |
| 36H | | | | | | | | | | | | |

■Subcommand

| Type | ASCII code | Binary code | | | | | | | | | | |
|------------------------|--|-------------|-----|---|---|-----|-----|-----|-----|---|-----|-----|
| For MELSEC-Q/L series | <table border="1"><tr><td>0</td><td>0</td><td>0</td><td>1</td></tr><tr><td>30H</td><td>30H</td><td>30H</td><td>31H</td></tr></table> | 0 | 0 | 0 | 1 | 30H | 30H | 30H | 31H | <table border="1"><tr><td>01H</td><td>00H</td></tr></table> | 01H | 00H |
| 0 | 0 | 0 | 1 | | | | | | | | | |
| 30H | 30H | 30H | 31H | | | | | | | | | |
| 01H | 00H | | | | | | | | | | | |
| For MELSEC iQ-R series | <table border="1"><tr><td>0</td><td>0</td><td>0</td><td>3</td></tr><tr><td>30H</td><td>30H</td><td>30H</td><td>33H</td></tr></table> | 0 | 0 | 0 | 3 | 30H | 30H | 30H | 33H | <table border="1"><tr><td>03H</td><td>00H</td></tr></table> | 03H | 00H |
| 0 | 0 | 0 | 3 | | | | | | | | | |
| 30H | 30H | 30H | 33H | | | | | | | | | |
| 03H | 00H | | | | | | | | | | | |

For 2C frame, the specification is not required. Functions and specification methods are equivalent to the subcommands for MELSEC-Q/L series.

■Number of bit access points

Specify the number of device points to be written within the following range. (☞ Page 67 Number of device points)

| Access target | Range |
|---|-----------------|
| MELSEC iQ-R series module (subcommand: 0001) MELSEC-Q/L series module (subcommand: 0001) | 1 to 188 points |
| MELSEC iQ-R series module (subcommand: 0003, 008□) MELSEC-Q/L series module (subcommand: 0081) MELSEC-QnA series module Module on other station via MELSEC-QnA series network module | 1 to 94 points |
| MELSEC-A series module | 1 to 20 points |

■Device

Specify a device to be written. ([Page 63 Devices](#))

Specify a bit device.

■Set/reset

Specify the ON/OFF status of bit devices. ([Page 76 Set/reset](#))



The number of points equivalent to the specified "Number of bit access points" is specified for "Device" and "Set/reset", respectively.

Communication example

Turn M50 OFF, and turn Y2F ON. (Subcommand: for MELSEC-Q/L series)

■Data communication using ASCII code

(Request data)

| Subcommand | Number of bit access points | Device code | Device number | Set/reset | Device code | Device number | Set/reset |
|-------------------------------|-------------------------------|-----------------|-----------------|---|-----------------|-----------------|--|
| 1 4 0 2 31H, 34H, 30H, 32H | 0 0 0 1 30H, 30H, 30H, 31H | 0 2 30H, 32H | M * 4DH, 2AH | 0 0 0 0 5 0 30H, 30H, 30H, 35H, 30H, 30H | 0 0 30H, 30H | Y * 59H, 2AH | 0 0 0 0 2 F 0 1 30H, 30H, 30H, 32H, 46H, 30H, 31H |

■Data communication using binary code

(Request data)

| Subcommand | Number of bit access points | Device code | Set/reset | Device code | Set/reset |
|------------|-----------------------------|--------------------|-----------|---------------|-----------|
| 02H, 14H | 01H, 00H | 02H, 32H, 00H, 00H | 90H, 00H | 2FH, 00H, 00H | 9DH, 01H |

8.4 Batch Read and Write Multiple Blocks

Read or write values for specified multiple blocks by handling consecutive devices as one block.

Batch read multiple blocks (command: 0406)

Read values for specified multiple blocks by handling consecutive word devices or bit devices as one block.

Each block can be specified with discontinuous device numbers.



When communicating with a Universal model QCPU or an LCPU, if other than "Specify service process execution counts" is selected for "Service Processing Setting" of the CPU module, data separation may occur. To avoid data separation, select "Specify service process execution counts".



When accessing any of the following devices, use the device extension specification (subcommand: 008□).

- Link direct device
- Module access device
- CPU buffer memory access device

For the message format for device extension specification, refer to the following section.

Page 432 Read/Write by Device Extension Specification

Message format

The following shows the message format of the request data and response data of the command.

8

■Request data

| | | | | | | |
|--------------------------------------|------------|--|---|---------------------------------------|-----|-------------------------------------|
| Command | Subcommand | Number of word device blocks (m points) | Number of bit device blocks (n points) | Block of word device (first point) | ... | Block of word device (mth point) |
| ----- | | | | | | |
| Block of bit device (first point) | ... | Block of bit device (nth point) | | | | |

■Response data

The value of read device is stored in hexadecimal. The data order differs between ASCII code or binary code. (Page 70 Read data, write data)

| | | | | | |
|---|-----|--|-------------------------|-----|-----------------------|
| Data for the number of word device blocks | | Data for the number of bit device blocks | | | |
| Word device | | Bit device | | | |
| Data in the first block | ... | Data in the mth block | Data in the first block | ... | Data in the nth block |

Data specified by request data

■Command

| ASCII code | Binary code |
|----------------------------|-------------|
| 0 4 0 6 30H 34H 30H 36H | 06H 04H |

■Subcommand

| Type | ASCII code | Binary code | | | | | | | | | | |
|------------------------|--|-------------|-----|---|---|-----|-----|-----|-----|--|-----|-----|
| For MELSEC-Q/L series | <table border="1" style="margin-left: auto; margin-right: auto;"> <tr><td>0</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>30H</td><td>30H</td><td>30H</td><td>30H</td></tr> </table> | 0 | 0 | 0 | 0 | 30H | 30H | 30H | 30H | <table border="1" style="margin-left: auto; margin-right: auto;"> <tr><td>00H</td><td>00H</td></tr> </table> | 00H | 00H |
| 0 | 0 | 0 | 0 | | | | | | | | | |
| 30H | 30H | 30H | 30H | | | | | | | | | |
| 00H | 00H | | | | | | | | | | | |
| For MELSEC iQ-R series | <table border="1" style="margin-left: auto; margin-right: auto;"> <tr><td>0</td><td>0</td><td>0</td><td>2</td></tr> <tr><td>30H</td><td>30H</td><td>30H</td><td>32H</td></tr> </table> | 0 | 0 | 0 | 2 | 30H | 30H | 30H | 32H | <table border="1" style="margin-left: auto; margin-right: auto;"> <tr><td>02H</td><td>00H</td></tr> </table> | 02H | 00H |
| 0 | 0 | 0 | 2 | | | | | | | | | |
| 30H | 30H | 30H | 32H | | | | | | | | | |
| 02H | 00H | | | | | | | | | | | |

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

Specify the number of device blocks to be read in hexadecimal. (☞ Page 69 Number of blocks)

Specify the total number of each block within the following range.

| Access target | Range |
|---|---|
| MELSEC iQ-R series module (subcommand: 0000) MELSEC-Q/L series module (subcommand: 0000) MELSEC-QnA series module | 1≤Number of word device blocks + Number of bit device blocks≤120 points |
| MELSEC iQ-R series module (subcommand: 0002, 008□) MELSEC-Q/L series module (subcommand: 0080) | 1≤Number of word device blocks + Number of bit device blocks≤60 points |
| | |

■Block of word device, block of bit device

Specify the device to be read by handling consecutive devices as one block.



Specify the block with the number of points equivalent to the specified "Number of word device blocks" and "Number of bit device blocks", respectively. When '0' is specified, this specification is not required.

Specify the following items for every block.

| Block (1 point) | |
|-----------------|-------------------------|
| Head device | Number of device points |

- Device: Specify the head device of the consecutive devices. (☞ Page 63 Devices)
- Number of device points: Specify the number of device points to be read. (☞ Page 67 Number of device points)

Specify the total number of device points for each block within the range of 1 to 960.

Word device is 1-word per one point, and bit device is 16-bit for one point.



Use a bit device block when the contact and coil for the following devices are specified.

- Timer
- Retentive timer
- Counter



Long index register (LZ) cannot be specified.

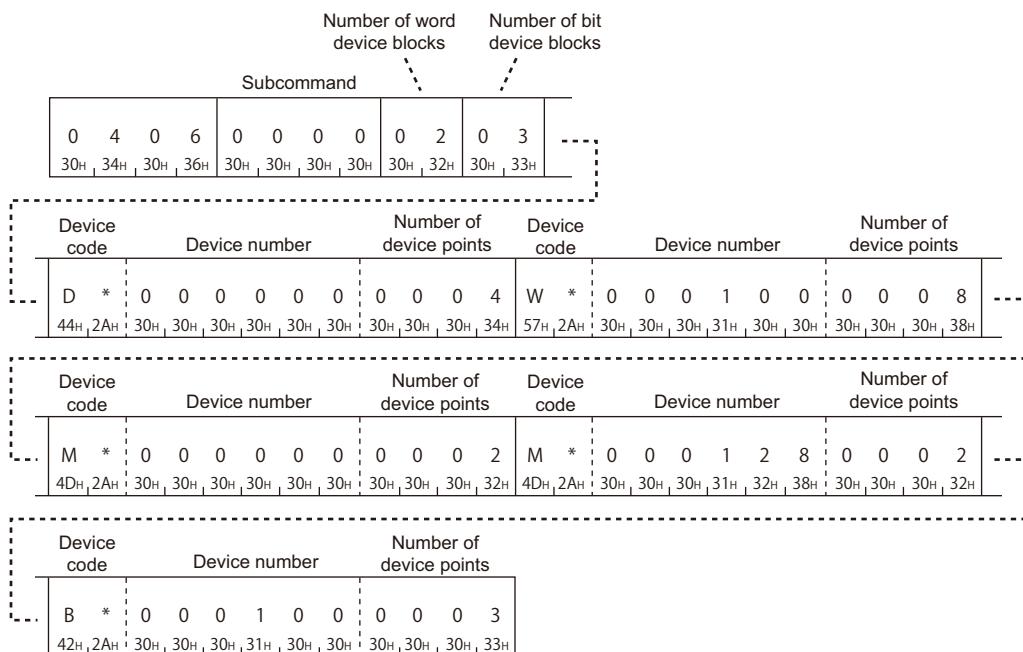
Communication example

Read values from device as follows. (Subcommand: for MELSEC-Q/L series)

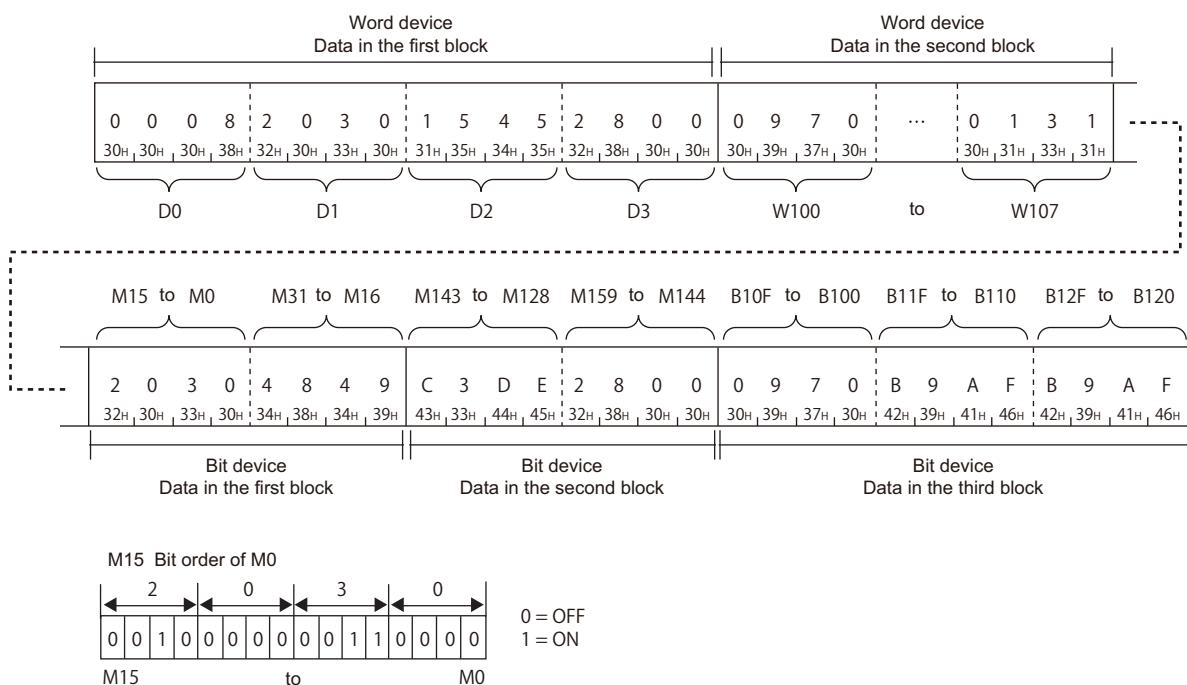
| Item | Content to be read |
|-------------|---|
| Word device | Block 1: D0 to D3 (4 points) Block 2: W100 to W107 (8 points) |
| Bit device | Block 1: M0 to M31 (2 points) Block 2: M128 to M159 (2 points) Block 3: B100 to B12F (3 points) |

■Data communication using ASCII code

(Request data)

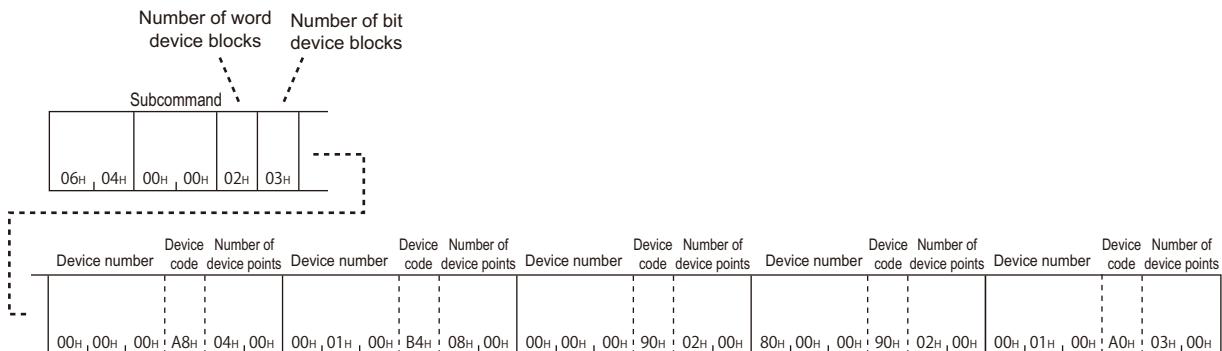


(Response data)

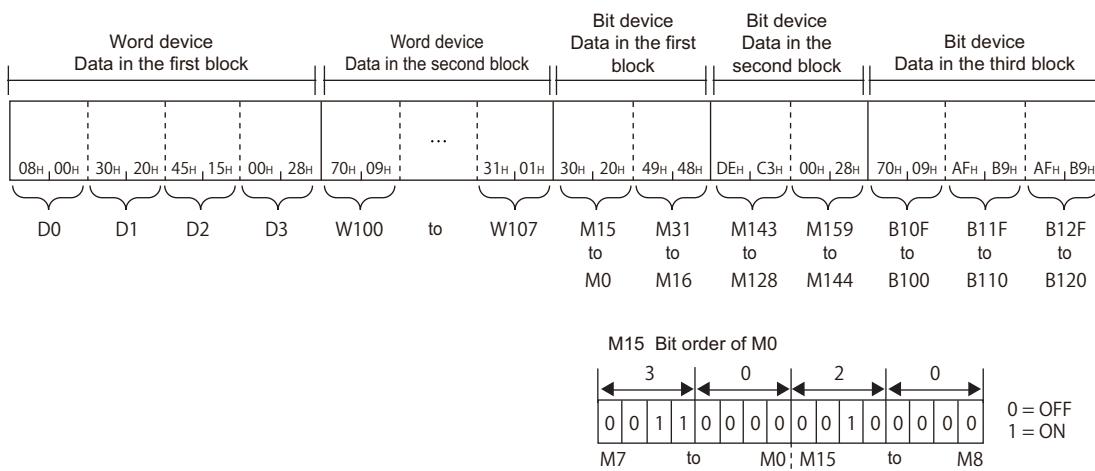


■Data communication using binary code

(Request data)



(Response data)



Batch write multiple blocks (command: 1406)

Write values for specified multiple blocks by handling consecutive word devices or bit devices as one block.
Each block can be specified with discontinuous device numbers.

Restriction

When communicating with a Universal model QCPU or an LCPU, if other than "Specify service process execution counts" is selected for "Service Processing Setting" of the CPU module, data separation may occur. To avoid data separation, select "Specify service process execution counts".

Point

When accessing any of the following devices, use the device extension specification (subcommand: 008□).

- Link direct device
- Module access device
- CPU buffer memory access device

For the message format for device extension specification, refer to the following section.

☞ Page 432 Read/Write by Device Extension Specification

Message format

The following shows the message format of the request data and response data of the command.

Request data

| | | | | | | |
|---------|------------|--|---|---------------------------------------|-----|-------------------------------------|
| Command | Subcommand | Number of word device blocks (m points) | Number of bit device blocks (n points) | Block of word device (first point) | ... | Block of word device (mth point) |
| ----- | | | | | | |
| | | Block of bit device (first point) | ... | Block of bit device (nth point) | | |

Response data

There is no response data for this command.

Data specified by request data

Command

| ASCII code | Binary code |
|----------------------------|-------------|
| 1 4 0 6 31H 34H 30H 36H | 06H 14H |

Subcommand

| Type | ASCII code | Binary code |
|------------------------|----------------------------|-------------|
| For MELSEC-Q/L series | 0 0 0 0 30H 30H 30H 30H | 00H 00H |
| For MELSEC iQ-R series | 0 0 0 2 30H 30H 30H 32H | 02H 00H |

Number of word device blocks, number of bit device blocks

Specify the number of device blocks to be written in hexadecimal. (☞ Page 69 Number of blocks)

Specify the total number of each block within the following range.

| Access target | Range |
|---|---|
| MELSEC iQ-R series module (subcommand: 0000) MELSEC-Q/L series module (subcommand: 0000) MELSEC-QnA series module | 1≤Number of word device blocks + Number of bit device blocks≤120 points |
| MELSEC iQ-R series module (subcommand: 0002, 008□) MELSEC-Q/L series module (subcommand: 0080) | 1≤Number of word device blocks + Number of bit device blocks≤60 points |

■Block of word device, Block of bit device

Specify the device to be written by handling consecutive devices as one block.



Specify the block with the number of points equivalent to the specified "Number of word device blocks" and "Number of bit device blocks", respectively. When '0' is specified, this specification is not required.

Specify the following items for each block.

| Block (1 point) | | |
|-----------------|-------------------------|------------|
| Head device | Number of device points | Write data |

- Device: Specify the head device of the consecutive devices. ([Page 63 Devices](#))
- Number of device points: Specify the number of device points to be written. ([Page 67 Number of device points](#))
- Write data: Specify the data to be written for the number of device points in hexadecimal. ([Page 70 Read data, write data](#))

Specify the number of device points within the following range.

| Subcommand | Range |
|--|--|
| When using MELSEC-Q/L series (0000, 0080) | 1≤(Total number of each block×4) + (Total number of device)≤960 points |
| When using MELSEC iQ-R series (0002, 0082) | 1≤(Total number of each block×9) + (Total number of device)≤960 points |

Word device is 1-word per one point, and bit device is 16-bit for one point.



Use a bit device block when the contact and coil for the following devices are specified.

- Timer
- Retentive timer
- Counter



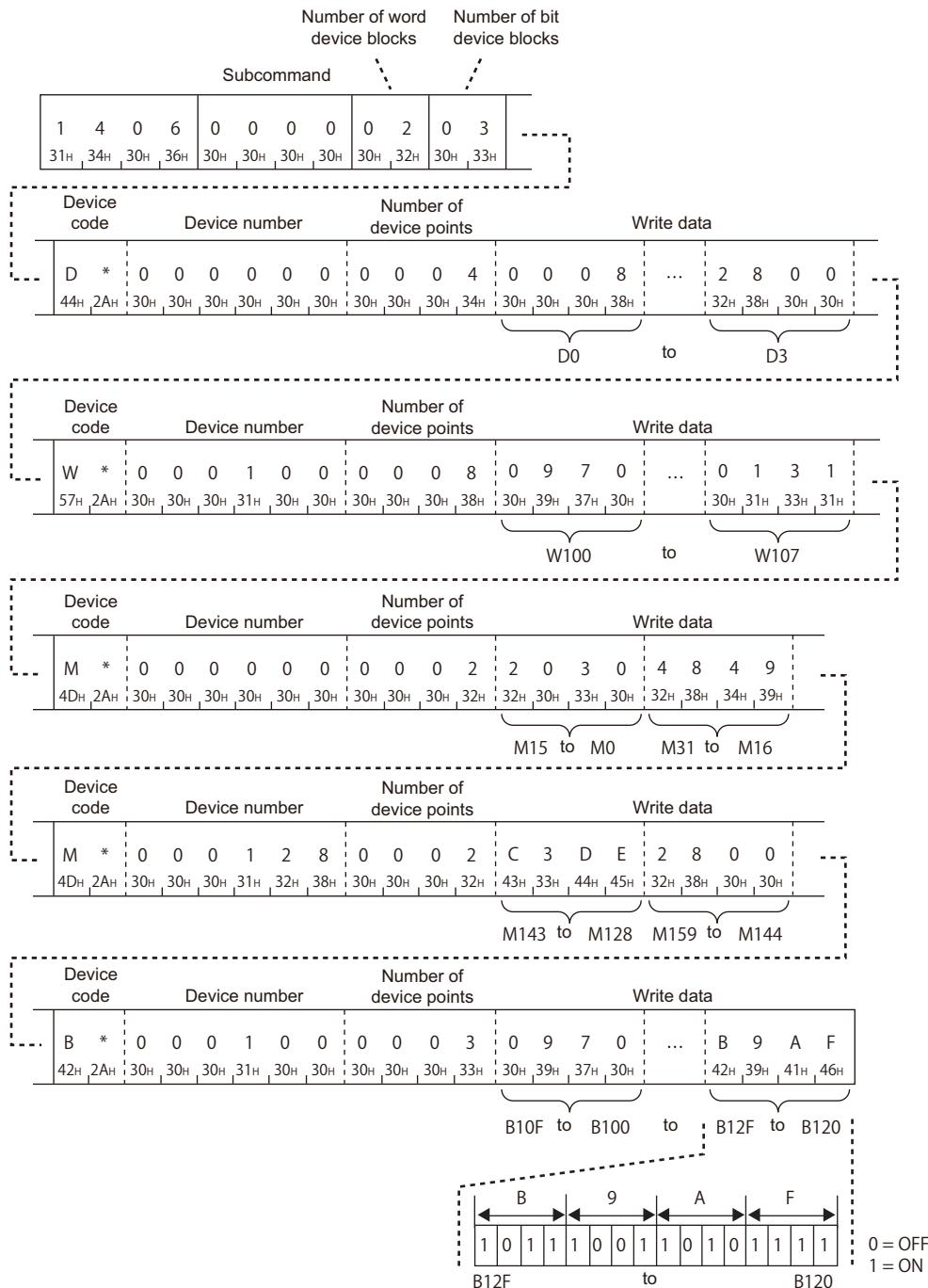
Long index register (LZ) cannot be specified.

Communication example

Write values to devices as follows. (Subcommand: for MELSEC-Q/L series)

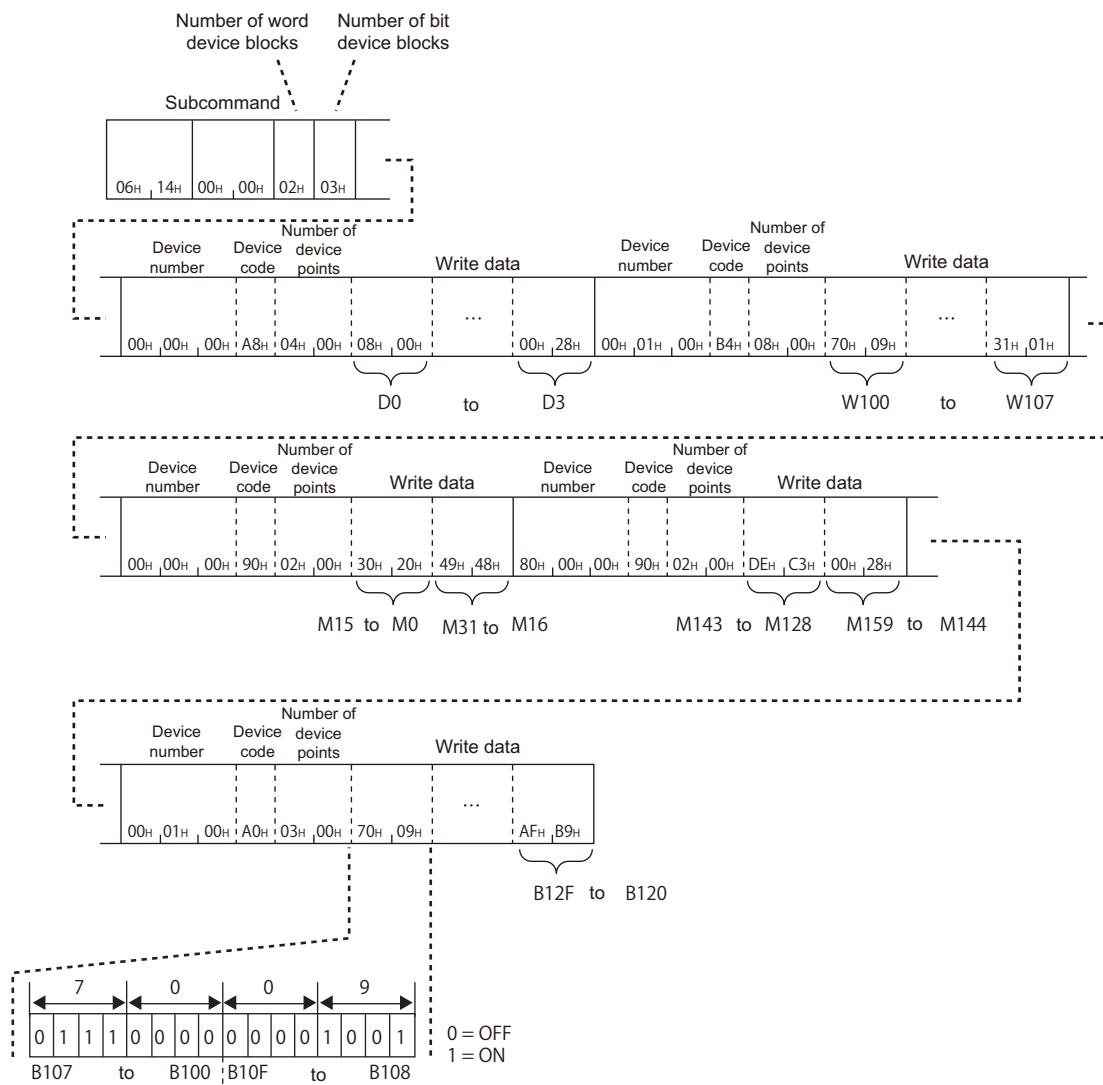
■Data communication using ASCII code

(Request data)



■Data communication using binary code

(Request data)



8.5 Device Memory Monitor

Read the registered device data, and monitor it.

Monitoring procedure

The following shows the procedure to monitor devices.

1. Registration of monitor device

Register a device to be read. (☞ Page 115 Register monitor data (command: 0801))

2. Execution of monitor

Read values from a registered device. (☞ Page 116 Monitor (command: 0802))

3. Registration of monitor devices

Reregister a device when a device to be read is changed. (☞ Page 115 Register monitor data (command: 0801))



- Monitoring with multiple conditions to the device memory of the same CPU module cannot be performed at the same time. If monitor (command: 0802) is executed while monitor with other conditions is being performed, the command is completed abnormally.
- Register devices to be read with the 'register monitor data' (command: 0801) before executing the 'monitor' (command: 0802). If the 'monitor' (command: 0802) is executed without the registration, the command is completed abnormally.
- When the access target module is restarted, the registered content is deleted. Register the devices to be read again with the 'register monitor data' (command: 0801).

8

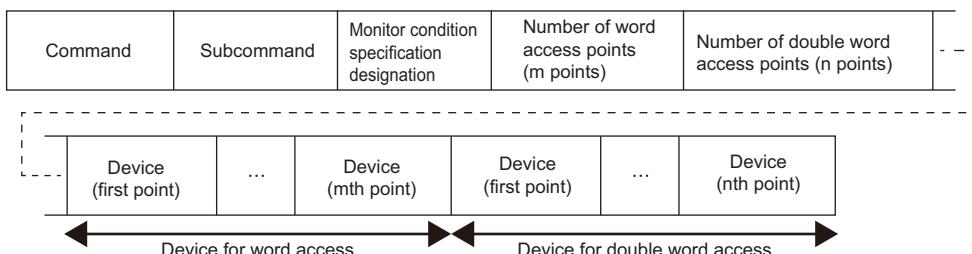
Register monitor data (command: 0801)

Register devices to be monitored.

Message format

The following shows the message format of the request data and response data of the command.

■Request data



■Response data

There is no response data for this command.

Data specified by request data

■Command

| Frame | ASCII code | Binary code | | | | | | | | | | |
|-------------------|--|-------------|-----|---|---|-----|-----|-----|-----|---|-----|-----|
| 4C/3C/4E/3E frame | <table border="1"><tr><td>0</td><td>8</td><td>0</td><td>1</td></tr><tr><td>30H</td><td>38H</td><td>30H</td><td>31H</td></tr></table> | 0 | 8 | 0 | 1 | 30H | 38H | 30H | 31H | <table border="1"><tr><td>01H</td><td>08H</td></tr></table> | 01H | 08H |
| 0 | 8 | 0 | 1 | | | | | | | | | |
| 30H | 38H | 30H | 31H | | | | | | | | | |
| 01H | 08H | | | | | | | | | | | |
| 2C frame | <table border="1"><tr><td>8</td></tr><tr><td>38H</td></tr></table> | 8 | 38H | — | | | | | | | | |
| 8 | | | | | | | | | | | | |
| 38H | | | | | | | | | | | | |

The data other than commands is the same as the data specified by 'random read in word units' (command: 0403).

☞ Page 94 Random read in word units (command: 0403)

Monitor (command: 0802)

Read value of registered device.

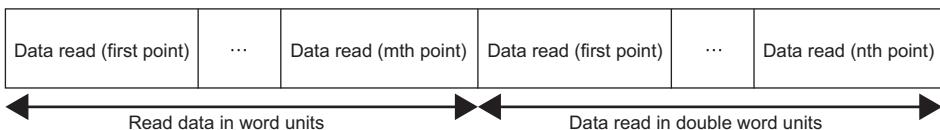
Message format

The following shows the message format of the request data and response data of the command.

■Request data



■Response data



The value of read device is stored in word units and in double word units. The data order differs between ASCII code or binary code. (☞ Page 70 Read data, write data)

Data specified by request data

■Command

| Frame | ASCII code | Binary code | | | | | | | | | | |
|-------------------|--|-------------|-----|---|---|-----|-----|-----|-----|---|-----|-----|
| 4C/3C/4E/3E frame | <table border="1"><tr><td>0</td><td>8</td><td>0</td><td>2</td></tr><tr><td>30H</td><td>38H</td><td>30H</td><td>32H</td></tr></table> | 0 | 8 | 0 | 2 | 30H | 38H | 30H | 32H | <table border="1"><tr><td>02H</td><td>08H</td></tr></table> | 02H | 08H |
| 0 | 8 | 0 | 2 | | | | | | | | | |
| 30H | 38H | 30H | 32H | | | | | | | | | |
| 02H | 08H | | | | | | | | | | | |
| 2C frame | <table border="1"><tr><td>9</td></tr><tr><td>39H</td></tr></table> | 9 | 39H | — | | | | | | | | |
| 9 | | | | | | | | | | | | |
| 39H | | | | | | | | | | | | |

■Subcommand

| ASCII code | Binary code | | | | | | | | | | |
|--|-------------|-----|-----|---|-----|-----|-----|-----|---|-----|-----|
| <table border="1"><tr><td>0</td><td>0</td><td>0</td><td>0</td></tr><tr><td>30H</td><td>30H</td><td>30H</td><td>30H</td></tr></table> | 0 | 0 | 0 | 0 | 30H | 30H | 30H | 30H | <table border="1"><tr><td>00H</td><td>00H</td></tr></table> | 00H | 00H |
| 0 | 0 | 0 | 0 | | | | | | | | |
| 30H | 30H | 30H | 30H | | | | | | | | |
| 00H | 00H | | | | | | | | | | |

For 2C frame, the specification is not required. Functions and specification methods are equivalent to the subcommands for MELSEC-Q/L series.

Communication example

Read the following devices registered by the 'register monitor data' (command: 0801).

- Word access: D0, T0, M100 to M115, X20 to X2F
- Double word access: D1500 to D1501, Y160 to Y17F, M1111 to M1142

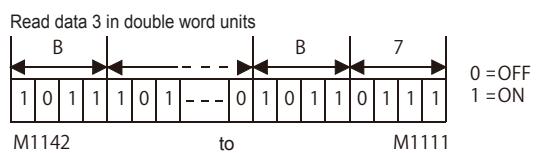
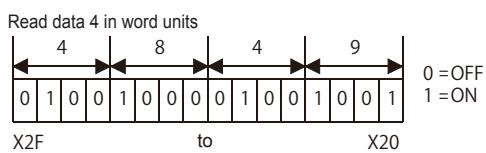
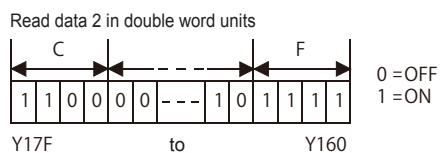
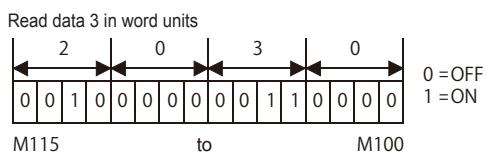
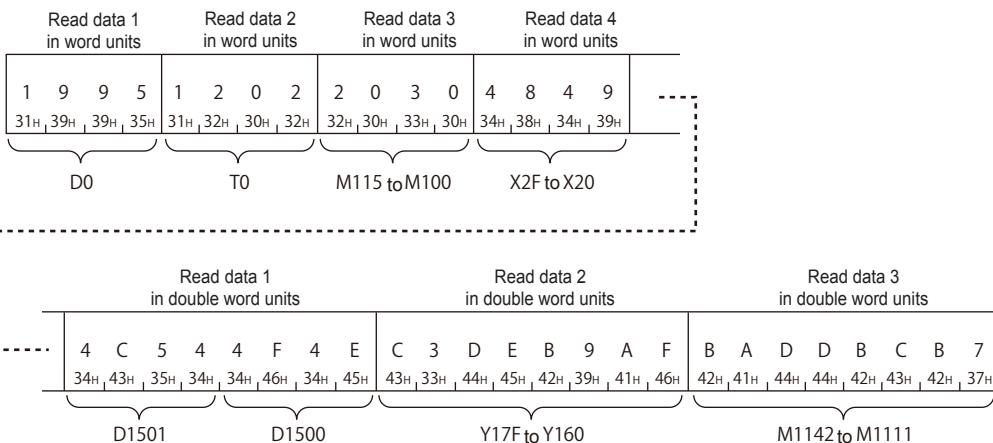
D0 = 6549 (1995H), T0 = 4610 (1202H), D1500 = 20302 (4F4EH), D1501 = 19540 (4C54H) are stored.

■Data communication using ASCII code

(Request data)

| | | | | | | | |
|-----|-----|-----|-----|-----|-----|-----|-----|
| 0 | 8 | 0 | 2 | 0 | 0 | 0 | 0 |
| 30H | 38H | 30H | 32H | 30H | 30H | 30H | 30H |

(Response data)

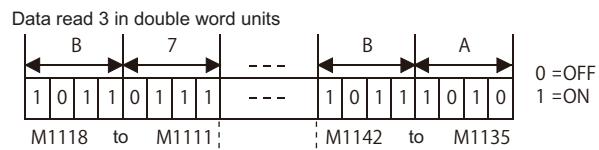
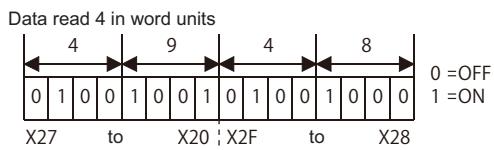
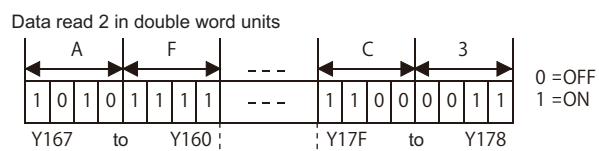
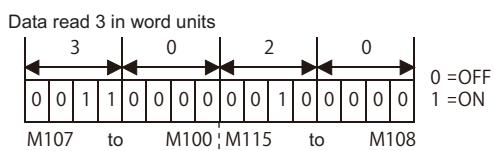
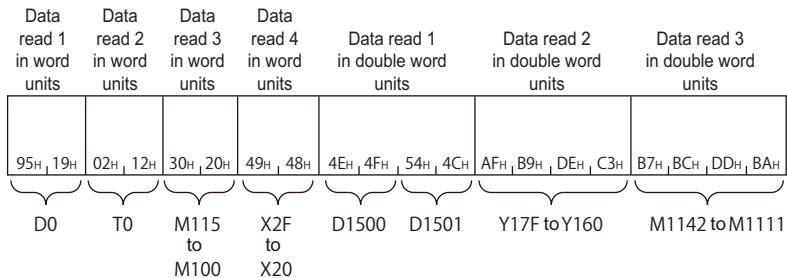


■Data communication using binary code

(Request data)

| | |
|----------|----------|
| | |
| 02H, 08H | 00H, 00H |

(Response data)



9 LABEL ACCESS

9

This section explains the commands to read and write devices using the standard global label of GX Works3. The commands can be used when the connected station and request target is MELSEC iQ-R series module.

Restriction

- Local labels cannot be accessed.
- Safety global labels and standard/safety shared global labels cannot be accessed.
- The global labels set with GX Works2 cannot be accessed.
- Enable "Access from External Device" with the global label setting editor of GX Works3 at the time of label access. (Disabled by default.)

9.1 Data to be Specified in Commands

This section explains the contents and specification methods for data items which are set in each command related to label access.

Labels

Specify the global label name to be accessed.

The "Label name" is specified by variable length. Specify the length of character string by "Label name length". (Null is unnecessary at the end of a label name character string.)

| ASCII code | Binary code |
|---|--|
| Label name length Label name <div style="display: flex; justify-content: space-around;"><div style="text-align: center;">(4 digits)</div><div style="text-align: center;">(Variable length)</div></div> | Label name length Label name <div style="display: flex; justify-content: space-around;"><div style="text-align: center;">(2 bytes)</div><div style="text-align: center;">(Variable length)</div></div> |

Label name length

Specify the number of characters of a label name.

■Data communication using ASCII code

Convert the numerical value to 4-digit ASCII code (hexadecimal), and send it from upper digits.

■Data communication using binary code

Send 2-byte^{*1} numerical values from lower byte (L: bits 0 to 7).

*1 For C24, the additional code may be added. ( Page 33 Additional code (10H))

Ex.

When the number of characters is three.

| ASCII code | Binary code |
|---|-------------|
| 0 0 0 3 30H 30H 30H 33H | 03H 00H |

Label name

Specify the character string of a label name.

■Data communication using ASCII code

Convert the numerical value of UTF-16, which indicates a global label name, to ASCII code (hexadecimal), and send it from the upper digits.

■Data communication using binary code

Send the numerical value of UTF-16, which indicates a global label name from lower byte (L: bit 0 to 7).

Ex.

When 'A'

(UTF-16: 'A'= 0041)

| ASCII code | Binary code |
|----------------------------|-------------|
| 0 0 4 1 30H 30H 34H 41H | 41H 00H |

Labels of the simple data type

Specify the label name as it is.

Bit specification (example: Lbl.3) and digit specification (example: K4Lbl) cannot be used.

Ex.

When the label name is 'Lbl'

| Item | Value of code corresponds to character | | |
|-------------|--|----------|----------|
| File name | L | b | I |
| UTF-16 | 004C | 0062 | 006C |
| ASCII code | 30303443 | 30303632 | 30303643 |
| Binary code | 4C00 | 6200 | 6C00 |

Array type label

Specify the element number with square brackets '[']' after label name. (UTF-16F: '[' = 005B, ']' = 005D)

The element name of array cannot be specified in the square bracket. Specify a numerical value of the element number.

Ex.

When the element number of array name 'Lbl' is '20'

- Label name length: 7H
- Label name character string: Lbl[20]

| Item | Value of code corresponds to character | | | | | | | |
|-------------|--|----------|----------|----------|----------|----------|----------|--|
| File name | L | b | I | [| 2 | 0 |] | |
| UTF-16 | 004C | 0062 | 006C | 005B | 0032 | 0030 | 005D | |
| ASCII code | 30303443 | 30303632 | 30303643 | 30303542 | 30303332 | 30303330 | 30303544 | |
| Binary code | 4C00 | 6200 | 6C00 | 5B00 | 3200 | 3000 | 5D00 | |

■Decimal/hexadecimal notation of element numbers

An element number can be specified in decimal or hexadecimal.

The element numbers can be distinguished by attaching 'k' or 'h' in front of the numerical value. (UTF-16: 'k'= 006B, 'h= 0068)

When only numerical value is specified, it will be handled as a decimal number.

- Decimal: Numerical value only, or attach 'k' in front of the numerical value. (Example: Lbl[10], Lbl[k10])
- Hexadecimal: Attach 'h' in front of the numerical value. (Example: Lbl[h10])

■Two-dimensional array, three-dimensional array,

Up to three-dimensional arrays can be specified.

For two-dimension and three-dimension array, specify an element number by separating with comma (',') in a square bracket.
(UTF-16: ',' = 002C)

Ex.

When the element numbers are 2, 1, 3 of three-dimensional array of array name 'Lbl'

- Label name length: AH
- Label name character string: Lbl[2,1,3]

| Item | Value of code corresponds to character | | | | | | | | | | | |
|-------------|--|----------|----------|----------|----------|----------|----------|----------|----------|----------|--|--|
| File name | L | b | I | [| 2 | , | 1 | , | 3 |] | | |
| UTF-16 | 004C | 0062 | 006C | 005B | 0032 | 002C | 0031 | 002C | 0033 | 005D | | |
| ASCII code | 30303443 | 30303632 | 30303643 | 30303542 | 30303332 | 30303243 | 30303331 | 30303243 | 30303333 | 30303544 | | |
| Binary code | 4C00 | 6200 | 6C00 | 5B00 | 3200 | 2C00 | 3100 | 2C00 | 3300 | 5D00 | | |

Restriction

Two-dimensional array and three-dimensional array bit type labels cannot be specified with the 'batch read/write array type labels' (command: 041A, 141A).

Structure labels

Specify the structure labels by attaching period '.' as an element name. (UTF-16: '.' = 002E)

Specify the element name for the end member. Only specifying a structure name cannot set the whole structure as a target.

Ex.

For the element name 'Data' of the structure name 'Str1'

- Label name length: 9H
- Label name character string: Str1.Data

| Item | Value of code corresponds to character | | | | | | | | | |
|-------------|--|----------|----------|----------|----------|----------|----------|----------|----------|------|
| File name | S | t | r | 1 | . | D | a | t | a | t |
| UTF-16 | 0053 | 0074 | 0072 | 0031 | 002E | 0044 | 0061 | 0074 | 0061 | 0061 |
| ASCII code | 30303533 | 30303734 | 30303732 | 30303331 | 30303245 | 30303434 | 30303631 | 30303734 | 30303631 | |
| Binary code | 5300 | 7400 | 7200 | 3100 | 2E00 | 4400 | 6100 | 7400 | 6100 | 6100 |

■Array member

When the structure member is an array type, specify the array element number with square bracket '[']' as is the case in array type label.

Ex.

When the element name 'Data' of structure name 'Str1', and 'Data' is a two-dimensional array element.

- Label name length: EH
- Label name character string: Str1.Data[1,3]

| Item | Value of code corresponds to character | | | | | | | | | | | | | |
|-------------|--|------|------|------|------|------|------|------|------|------|------|------|------|------|
| File name | S | t | r | 1 | . | D | a | t | a | [| 1 | , | 3 |] |
| UTF-16 | 0053 | 0074 | 0072 | 0031 | 002E | 0044 | 0061 | 0074 | 0061 | 005B | 0031 | 002C | 0033 | 005D |
| ASCII code | 3030 | 3030 | 3030 | 3030 | 3030 | 3030 | 3030 | 3030 | 3030 | 3030 | 3030 | 3030 | 3030 | 3030 |
| Binary code | 5300 | 7400 | 7200 | 3100 | 2E00 | 4400 | 6100 | 7400 | 6100 | 5B00 | 3100 | 2C00 | 3300 | 5D00 |

■Structure member

When the structure member is a structure type, specify the element name by delimiting with period '.' up to the end member.

Ex.

When specifying the member, 'memberB1', of the structure member name 'memberA3' for the structure type label name 'LabelA'

- Label name character string: LabelA.memberA3.memberB1

Restriction

When arbitrary device is assigned, the structure type label with structure type members cannot be specified.

Timer type label, counter type label

The labels of the following data types are handled as a structure with a contact, coil, and current value for its element.

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

Specify the following element name with adding a period '..'. (UTF-16: '.' = 002E)

| Item | Element name | Example when a label name is 'Lbl1' |
|---------------|--------------|-------------------------------------|
| Contact | S | Lbl1.S |
| Coil | C | Lbl1.C |
| Current value | N | Lbl1.N |

Ex.

For the contact of timer type label name 'Lbl1'

- Label name length: 6H
- Label name character string: Lbl1.S

| Item | Value of code corresponds to character | | | | | |
|-------------|--|----------|----------|----------|----------|----------|
| Label name | L | b | I | 1 | . | S |
| UTF-16 | 004C | 0062 | 006C | 0031 | 002E | 0053 |
| ASCII code | 30303443 | 30303632 | 30303643 | 30303331 | 30303245 | 30303533 |
| Binary code | 4C00 | 6200 | 6C00 | 3100 | 2E00 | 5300 |

Restriction 

The labels of which data type is timer, counter, retentive timer, long timer, long counter, or long retentive timer cannot be specified with the 'batch read/write array type labels' (command: 041A, 141A).

Abbreviation specification of label

Abbreviation specification can be used when specifying a structure type label as an access target.

When a label name or a structure member name is specified by the abbreviation specification, the character string specified as a "label name" can be simplified using "%n" (n: offset value).

Ex.

When a structure type label name 'LabelA' and its structure member name 'memberA3' is specified as abbreviation specification, they can be abbreviated as follows.

(LabelA= %1, memberA3= %2)

| Actual label name | Abbreviated label name |
|--------------------------|------------------------|
| LabelA.memberA1 | %1.memberA1 |
| LabelA.memberA2 | %1.memberA2 |
| LabelA.memberA3.memberB1 | %1.%2.memberB1 |
| LabelA.memberA3.memberB2 | %1.%2.memberB2 |

When do not abbreviate labels

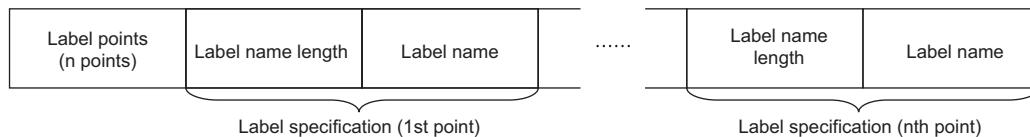
Specify '0'.

| ASCII code | Binary code |
|-------------------------------|-------------|
| 0 0 0 0 30H, 30H, 30H, 30H | 00H, 00H |

When abbreviate labels

Specify the points to be specified and label name.

Label names can be abbreviated ('%1' to '%n') as an offset value (1 to n) in the specified order.



The character strings that include a dot (.) cannot be abbreviated. Abbreviate character string in a label name or member name unit.

The label name and member name of the array type cannot be abbreviated.

Ex.

The example that cannot be abbreviated is as follows:

| Actual label name | String that cannot be abbreviated | String that can be abbreviated |
|-----------------------------|---|----------------------------------|
| LabelA.memberA3.memberB1 | "LabelA.memberA3.memberB1", "LabelA.memberA3" | 'LabelA', 'memberA3', 'memberB1' |
| LabelA.memberA4[1].memberB1 | "memberA4", "memberA4[1]" | "LabelA", "memberB1" |

■Label points

Specify the label points which abbreviate a label names by abbreviation specification. ([Page 126 Points](#))

■Label specification

Specify the following items for each label for the points specified by label points. ([Page 119 Labels](#))

- Label name length: Specify the number of characters of a label name or structure member name.
- Label name: Specify a global label name or a structure member name.

Ex.

When a structure type label name 'LabelA' and its structure member name 'memberA3' is specified as abbreviation specification

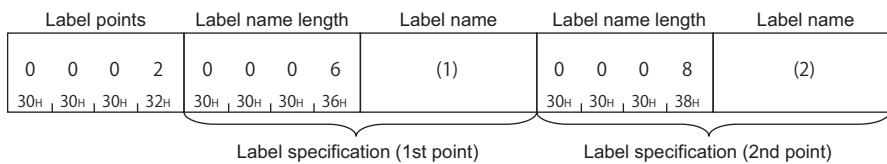
(1) LabelA

| Item | Value of code corresponds to character | | | | | |
|-------------|--|----------|----------|----------|----------|----------|
| Label name | L | a | b | e | I | A |
| UTF-16 | 004C | 0061 | 0062 | 0065 | 006C | 0041 |
| ASCII code | 30303443 | 30303631 | 30303632 | 30303635 | 30303643 | 30303431 |
| Binary code | 4C00 | 6100 | 6200 | 6500 | 6C00 | 4100 |

(2) memberA3

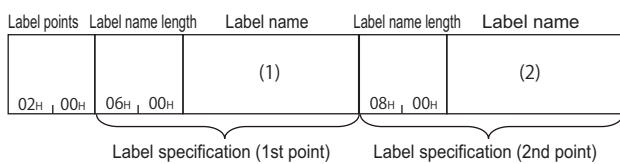
| Item | Value of code corresponds to character | | | | | | | |
|-------------|--|----------|----------|----------|----------|----------|----------|----------|
| Label name | m | e | m | b | e | r | A | 3 |
| UTF-16 | 006D | 0065 | 006D | 0062 | 0065 | 0072 | 0041 | 0033 |
| ASCII code | 30303644 | 30303635 | 30303644 | 30303632 | 30303635 | 30303732 | 30303431 | 30303333 |
| Binary code | 6D00 | 6500 | 6D00 | 6200 | 6500 | 7200 | 4100 | 3300 |

■Data communication using ASCII code



In the figure (1) and (2), set the value of "ASCII code" indicated in the table of "Value of code corresponds to character" of each label name.

■Data communication using binary code



In the figure (1) and (2), set the value of "Binary code" indicated in the table of "Value of code corresponds to character" of each label name.

Points

Specify the number of the data to be read or written.

Setting method

The setting method of each item to specify the number of points is common.

Since data to be transmitted is 1920 bytes at maximum, the maximum number of points which can be specified varies depending on the label name length contained in the data.

■Data communication using ASCII code

Convert the numerical value to 4-digit ASCII code (hexadecimal), and send it from upper digits.

■Data communication using binary code

Send 2-byte^{*1} numerical values from lower byte (L: bits 0 to 7).

*1 For C24, the additional code may be added. (☞ Page 33 Additional code (10H))

Ex.

For 3 points

| ASCII code | Binary code |
|----------------------------------|-------------|
| 0 0 0 3 30H 30H 30H 33H | 03H 00H |

Array points

Specify the number of arrays.

Label points

Specify the number of labels.

Data type ID

The data type of the read label is stored.

When writing data, the data type ID is not specified. The specified Follow the data type of the specified label.

The value of data type ID of each data type is shown below.

| Data type | Data type ID (decimal) | Setting value (hexadecimal) |
|--|---------------------------|--------------------------------|
| Bit | 1 | 01H |
| Word [Unsigned]/Bit String [16-bit] | 2 | 02H |
| Double Word [Unsigned]/Bit String [32-bit] | 3 | 03H |
| Word [Signed] | 4 | 04H |
| Double Word [Signed] | 5 | 05H |
| FLOAT [Single Precision] | 6 | 06H |
| FLOAT [Double Precision] | 7 | 07H |
| Time | 8 | 08H |
| String | 9 | 09H |
| String[Unicode] | 10 | 0AH |
| Pointer | Cannot be specified. | |
| Timer | Contact | 1 |
| Counter | Coil | 01H |
| Retentive timer | Current value | 02H |
| Long timer | Contact | 1 |
| Long counter | Coil | 01H |
| Long retentive timer | Current value | 03H |

For an array type label and a structure type label, the data type of the element from which value is read is stored.

Setting method

■Data communication using ASCII code

Convert the numerical value to 2-digit ASCII code (hexadecimal), and send it from the upper digits.

■Data communication using binary code

Send a 1-byte numerical value.

Ex.

Double Word [Signed]

| ASCII code | Binary code | | | | | |
|---|-------------|---|-----|-----|---|-----|
| <table border="1"> <tr> <td>0</td> <td>5</td> </tr> <tr> <td>30H</td> <td>35H</td> </tr> </table> | 0 | 5 | 30H | 35H | <table border="1"> <tr> <td>05H</td> </tr> </table> | 05H |
| 0 | 5 | | | | | |
| 30H | 35H | | | | | |
| 05H | | | | | | |

Data length, unit specification

Specify the length of data to be read/written.

Unit specification

Specify the units of data length when reading/writing data from/to array in batch.

■Bit specification

The value of data length is handled as the number of bits.

Specify this when the data type of the label is bit.

| ASCII code | Binary code |
|-------------------------------|-------------|
| 0 0 0 0 30H, 30H, 30H, 30H | 00H, 00H |

Read data or write data are stored in 16 point unit of bit devices (2 bytes).

When data length is not a multiple of 16 at the time of bit specification, '0' is stored in an invalid area.

■Byte specification

The value of data length is handled as a number of bytes of read data or write data.

Specify when the data type of a label is other than bit.

| ASCII code | Binary code |
|-------------------------------|-------------|
| 0 0 0 1 30H, 30H, 30H, 31H | 01H, 00H |

Data length

Specify the length of read data or write data.

Specify the following values according to the data type of a label.

Specifying and writing data length which is not suited for the data type of label results in abnormal completion.

| Data type | Batch read and write array type labels | | Random read and write |
|--|--|---|-----------------------|
| | Unit specification | Array data length | |
| Bit | Bit units | 1 × Number of array element | 2 |
| Word [Unsigned]/Bit String [16-bit] | Byte units | 2 × Number of array element | 2 |
| Double Word [Unsigned]/Bit String [32-bit] | | 4 × Number of array element | 4 |
| Word [Signed] | | 2 × Number of array element | 2 |
| Double Word [Signed] | | 4 × Number of array element | 4 |
| FLOAT [Single Precision] | | 4 × Number of array element | 4 |
| FLOAT [Double Precision] | | 8 × Number of array element | 8 |
| Time | | 4 × Number of array element | 4 |
| String | | Depend on the number of character strings (1 per 1 character, including end NULL) ^{*1} | |
| String [Unicode] | | Depend on the number of character strings (2 per 1 character, including end NULL) ^{*2} | |
| Pointer | | Cannot be specified. | |
| Timer | Contact Coil Current value | Cannot be specified. | 2 |
| Counter | | | |
| Retentive timer | | | 2 |
| Long timer | | | |
| Long counter | Contact Coil | | |
| Long retentive timer | Current value | | 4 |

*1 Specify the following values per one array element/label.

Character string length specified to "Data Length of Character String Data Type" with an Engineering tool + 1
(When the value is an odd number, add 1 and specify with an even number.)

*2 Specify the following values per one array element/label.

(Character string length specified to "Data Length of Character String Data Type" with an Engineering tool + 1) × 2

■Data communication using ASCII code

Convert the numerical value to 4-digit ASCII code (hexadecimal), and send it from upper digits.

■Data communication using binary code

Send 2-byte^{*1} numerical values from lower byte (L: bits 0 to 7).

*1 For C24, the additional code may be added. (☞ Page 33 Additional code (10H))

Setting example

■Batch read and write of array

For the bit specification, specify the number of bits to be accessed.

- Unit specification: Bit
- Label: Bit type 3 points
- Array data length: 1 bit × Number of array element (3) = 3 bits

| ASCII code | Binary code |
|-------------------------------|-------------|
| 0 0 0 3 30H, 30H, 30H, 33H | 03H, 00H |

For byte specification, specify the data length of the label to be accessed in byte unit.

- Unit specification: Byte
- Label: Word type 5 points
- Array data length: 2 bytes × (5) = 10 bytes

| ASCII code | Binary code |
|-------------------------------|-------------|
| 0 0 0 A 30H, 30H, 30H, 3AH | 0A, 00H |

■Random read and write

For bit type, data length will be 2 (fixed value).

- Label: Bit type 1 point
- Data length: 2 (fixed)

| ASCII code | Binary code |
|-------------------------------|-------------|
| 0 0 0 2 30H, 30H, 30H, 32H | 02H, 00H |

The character string type [Unicode] will be 2 bytes per one character of UTF-16 code.

- Label: Character string type [Unicode] one point
- "Data Length of Character String Data Type" of an Engineering tool: 32
- Data length: Number of characters (32 + 1) × Data length per one character string (2) = 66 (42H)

| ASCII code | Binary code |
|-------------------------------|-------------|
| 0 0 4 2 30H, 30H, 34H, 32H | 42H, 00H |

Read data, write data

The read data is stored for reading, and the data to be written is stored for writing.

Data is stored with variable length. The length of data is specified by "Data length." (☞ Page 128 Data length)

The storing method of data is same as that of reading or writing data in word units (bit device in 16-points). (☞ Page 70

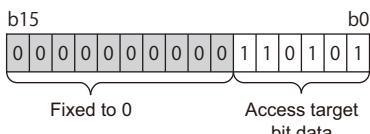
Read data, write data)

For bit type labels

Read data and write data are handled in 16-point units, but '0' is stored in data other than label points to be accessed.

Ex.

For bit type label 6 points



For character string type labels

Store NULL at the end of the valid character string. The data after NULL of the read data will be undefined value.

| Item | Character string type of ASCII code | | Character string type of Unicode |
|--|---|---|---|
| Number of character strings of label | Odd | Even number | — |
| NULL code to be stored to the end | NULL (00H) | NULL (00H)×2 | NULL (0000H) |
| Data length of array element/label per one point | Character string specified to "Data Length of Character String Data Type" + 1 | Character string specified to "Data Length of Character String Data Type" + 2 | (Character string specified to "Data Length of Character String Data Type" + 1)×2 |

Ex.

For read data of 1-point character string type (ASCII code) label

- Label character string: 'ABCD' (4 characters)
- "Data Length of Character String Data Type" of an Engineering tool: 32
- Data length: Number of characters (32 + 1) × Data length per one character string (1) = 34 (22H)

| ASCII code | Binary code | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|-------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|--|--|---|---|---|---|-----|-----|-----|-----|-----|-----|-----|-----|-----|--|
| <table border="1"><tr><td>B</td><td>A</td><td>D</td><td>C</td><td>NUL</td><td>NUL</td><td>...</td></tr><tr><td>4 2</td><td>4 1</td><td>4 4</td><td>4 3</td><td>0 0</td><td>0 0</td><td>...</td></tr><tr><td>34H</td><td>32H</td><td>34H</td><td>34H</td><td>34H</td><td>33H</td><td></td></tr></table> <p>("Data Length of String Data Type"(32)+2)×2 bytes</p> | B | A | D | C | NUL | NUL | ... | 4 2 | 4 1 | 4 4 | 4 3 | 0 0 | 0 0 | ... | 34H | 32H | 34H | 34H | 34H | 33H | | <table border="1"><tr><td>A</td><td>B</td><td>C</td><td>D</td><td>NUL</td><td>NUL</td><td>...</td></tr><tr><td>41H</td><td>42H</td><td>43H</td><td>44H</td><td>00H</td><td>00H</td><td></td></tr></table> <p>"Data Length of String Data Type"(32)+2 bytes</p> | A | B | C | D | NUL | NUL | ... | 41H | 42H | 43H | 44H | 00H | 00H | |
| B | A | D | C | NUL | NUL | ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 4 2 | 4 1 | 4 4 | 4 3 | 0 0 | 0 0 | ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 34H | 32H | 34H | 34H | 34H | 33H | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| A | B | C | D | NUL | NUL | ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 41H | 42H | 43H | 44H | 00H | 00H | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |



When communicating ASCII code character string data in ASCII code, data is rearranged every two characters and stored.

9.2 Batch Read and Write

Read/write data by specifying the continuous element of array in batch.

Batch read array type labels (command: 041A)

Read data by specifying the continuous element of array in batch.

Specify the array type label or array type element of structure label.

The labels other than array type can be specified in one point unit. (Specify the number of array element as '1'.)

Message format

The following shows the message format of the request data and response data of the command.

■Request data

| | | | | | | |
|---------|------------|----------------------------|-------------------------------|--------------------------------------|-----|------------------------------------|
| Command | Subcommand | Array points (n points) | Abbreviation specification | Array specification (first point) | ... | Array specification (nth point) |
|---------|------------|----------------------------|-------------------------------|--------------------------------------|-----|------------------------------------|

■Response data

The read array data is stored for the number of array points which are specified with request data.

| | | | |
|----------------------------|-----------------------------|-----|---------------------------|
| Array points (n points) | Array data (first point) | ... | Array data (nth point) |
|----------------------------|-----------------------------|-----|---------------------------|

Data specified by request data

■Command

| ASCII code | Binary code |
|----------------------------|-------------|
| 0 4 1 A 30H 34H 31H 41H | 1AH 04H |

■Subcommand

| ASCII code | Binary code |
|----------------------------|-------------|
| 0 0 0 0 30H 30H 30H 30H | 00H 00H |

■Array points

Specify the point of array to be read. ( Page 126 Points)

■Abbreviation specification

Specify the label name length and label name to be abbreviated. ( Page 124 Abbreviation specification of label)

When do not abbreviate, specify '0'.

| ASCII code | Binary code |
|----------------------------|-------------|
| 0 0 0 0 30H 30H 30H 30H | 00H 00H |

■Array specification

Specify the details of arrays for number of arrays specified to array points.

| | | | | |
|-------------------|------------|--------------------|--------------|-------------------|
| Label name length | Label name | Unit specification | Fixed values | Array data length |
|-------------------|------------|--------------------|--------------|-------------------|

Array specification (Array one point)

Specify the following items for each array.

- Label name length, label name: Specify the label name and label length of a global label. ([Page 119 Labels](#))
- Unit specification: For the bit type labels, specify the unit specification in bit units (0). As for other than bit type label, specify the unit specification in byte units (1).
- Array data length: Specify the data size of array in the unit specified with "Unit specification." ([Page 128 Data length, unit specification](#))
- Fixed value: Fixed to '0'

| ASCII code | Binary code | | | | | |
|--|-------------|---|-----|-----|---|-----|
| <table border="1"><tr><td>0</td><td>0</td></tr><tr><td>30H</td><td>30H</td></tr></table> | 0 | 0 | 30H | 30H | <table border="1"><tr><td>00H</td></tr></table> | 00H |
| 0 | 0 | | | | | |
| 30H | 30H | | | | | |
| 00H | | | | | | |

Data stored in response data

■Array points

The same data as request data are stored.

■Array data

The read array data is stored for number of arrays which are specified with array point.

| | | | |
|--------------|--------------------|-------------------|-----------|
| Data type ID | Unit specification | Array data length | Read data |
|--------------|--------------------|-------------------|-----------|

Array data (Array one point)

The following items are stored for each array.

- Data type ID: The data type of the label name is stored with the defined ID. ([Page 127 Data type ID](#))
- Unit specification: For the bit type labels, specify the unit specification in bit units (0). As for other than bit type label, specify the unit specification in byte units (1).
- Array data length: Specify the data size of array in the unit specified with "Unit specification." ([Page 128 Data length, unit specification](#))
- Read data: The value of the read label is stored. ([Page 130 Read data, write data](#))

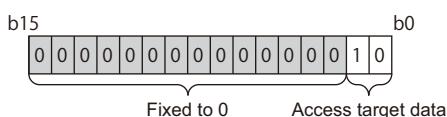


When unit specification is bit specification, read data are stored in 16-bit (2-byte) units.

Communication example (Bit specification)

For one-dimensional array type label 'Lbl', read 2-bit data from Lbl [2].

The value of the read label is as follows:



■Data communication using ASCII code

(Request data)

| Subcommand | Array points | Abbreviation specification | Label name length | Label name | Unit specification (Fixed value) | Array data length |
|-------------------------------|-------------------------------|-------------------------------|-------------------------------|-------------------------------|----------------------------------|--------------------------------------|
| 0 4 1 A 30H, 34H, 31H, 41H | 0 0 0 0 30H, 30H, 30H, 30H | 0 0 0 1 30H, 30H, 30H, 31H | 0 0 0 0 30H, 30H, 30H, 30H | 0 0 0 6 30H, 30H, 30H, 36H | (1) 30H, 30H | 0 0 0 0 2 30H, 30H, 30H, 30H, 32H |

Array specification (Array one point)

In the figure (1), set the value of "ASCII code" indicated in the following table.

(1) Lbl[2]

| Item | Value of code corresponds to character | | | | | |
|------------|--|----------|----------|----------|----------|----------|
| Label name | L | b | I | [| 2 |] |
| UTF-16 | 004C | 0062 | 006C | 005B | 0032 | 005D |
| ASCII code | 30303443 | 30303632 | 30303643 | 30303542 | 30303332 | 30303544 |

(Response data)

| Array points | Data type ID | Unit specification | Array data length | Read data |
|-------------------------------|-----------------|--------------------|-------------------------------|-----------------|
| 0 0 0 1 30H, 30H, 30H, 31H | 0 1 30H, 31H | 0 0 30H, 30H | 0 0 0 2 30H, 30H, 30H, 32H | 0 2 30H, 32H |

Array data (Array one point)

■Data communication using binary code

(Request data)

| Subcommand | Array points | Abbreviation specification | Label name length | Label name | Unit specification | (Fixed value) | Array data length |
|------------|--------------|----------------------------|-------------------|-------------|--|--------------------|-------------------|
| 1AH, 04H | 00H, 00H | 01H, 00H | 00H, 00H | L b I [2] | 4CH, 00H, 62H, 00H, 6CH, 00H, 5BH, 00H, 32H, 00H, 5DH, 00H | 00H, 00H, 02H, 00H | |

Array specification (Array one point)

(Response data)

| Data type ID | Array points | Unit specification | Array data length | Read data |
|--------------|--------------|--------------------|-------------------|-----------|
| | 01H, 00H | 01H, 00H | 02H, 00H | 02H, 00H |

Array data (Array one point)

Communication example (Byte specification)

For one-dimensional array type label 'Lbl', read 5-word data from Lbl [2].

■Data communication using ASCII code

(Request data)

| Subcommand | Array points | Abbreviation specification | Label name length | Label name | Unit specification | (Fixed value) | Array data length |
|-------------------------------|-------------------------------|-------------------------------|-------------------------------|-------------------------------|--------------------|---------------|----------------------------------|
| 0 4 1 A 30H, 34H, 31H, 41H | 0 0 0 0 30H, 30H, 30H, 30H | 0 0 0 1 30H, 30H, 30H, 31H | 0 0 0 0 30H, 30H, 30H, 30H | 0 0 0 6 30H, 30H, 30H, 36H | (1) 30H | 0 1 31H | 0 0 0 0 A 30H 30H 30H 30H 41H |

Array specification (Array one point)

In the figure (1), set the value of "ASCII code" indicated in the following table.

(1) Lbl[2]

| Item | Value of code corresponds to character | | | | | |
|------------|--|----------|----------|----------|----------|----------|
| Label name | L | b | I | [| 2 |] |
| UTF-16 | 004C | 0062 | 006C | 005B | 0032 | 005D |
| ASCII code | 30303443 | 30303632 | 30303643 | 30303542 | 30303332 | 30303544 |

(Response data)

| Array points | Data type ID | Unit specification | Array data length | Read data |
|-------------------------------|-----------------|--------------------|-------------------------------|---|
| 0 0 0 1 30H, 30H, 30H, 31H | 0 2 30H, 33H | 0 1 30H, 31H | 0 0 0 A 30H, 30H, 30H, 41H | 0 0 4 4 0 0 6 1 0 0 7 4 0 0 6 1 0 0 3 1 30H, 30H, 34H, 34H, 30H, 30H, 36H, 31H, 30H, 30H, 37H, 34H, 30H, 30H, 36H, 31H, 30H, 30H, 33H, 31H |

Array data (Array one point)

■Data communication using binary code

(Request data)

| Subcommand | Array points | Abbreviation specification | Label name length | Label name | (Fixed value) | Unit specification | Array data length |
|------------|--------------|----------------------------|-------------------|-------------|--|--------------------|-------------------|
| 1AH, 04H | 00H, 00H | 01H, 00H | 00H, 00H | L b I [2] | 4CH, 00H, 62H, 00H, 6CH, 00H, 5BH, 00H, 32H, 00H, 5DH, 00H | 01H | 00H, 0AH, 00H |

Array specification (Array one point)

(Response data)

| Unit specification | Array points | Data type ID | Array data length | Read data |
|--------------------|--------------|--------------|-------------------|--|
| 01H, 00H | 02H | 01H | 0AH, 00H | 44H, 00H, 61H, 00H, 74H, 00H, 61H, 00H, 31H, 00H |

Array data (Array one point)

Communication example (Abbreviate with structure type array)

Read the following data from the structure label 'Typ1', which has the array type element.

- 8 bytes from Typ1.led[2]
- 4 bytes from Typ1.No[1]

The notation of each label when using abbreviation specification (Typ1= %1) is as follows.

(1)Typ1

| Item | Value of code corresponds to character | | | |
|-------------|--|----------|----------|----------|
| Label name | T | y | p | 1 |
| UTF-16 | 0054 | 0079 | 0070 | 0031 |
| ASCII code | 30303534 | 30303739 | 30303730 | 30303331 |
| Binary code | 5400 | 7900 | 7000 | 3100 |

(2)Typ1.led[2]

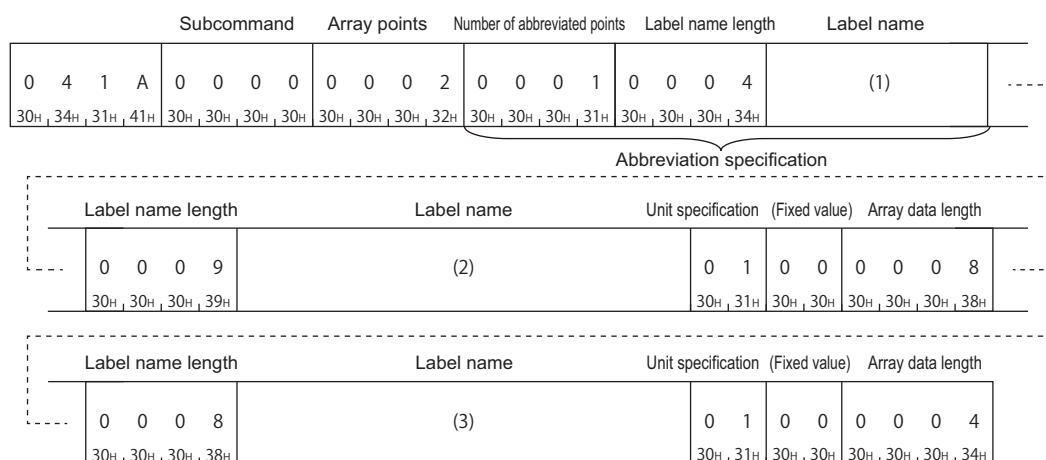
| Item | Value of code corresponds to character | | | | | | | | |
|----------------------|--|----------|----------|----------|----------|----------|----------|----------|----------|
| Abbreviated notation | % | 1 | . | I | e | d | [| 2 |] |
| UTF-16 | 0025 | 0031 | 002E | 006C | 0065 | 0064 | 005B | 0032 | 005D |
| ASCII code | 30303235 | 30303331 | 30303245 | 30303643 | 30303635 | 30303634 | 30303542 | 30303332 | 30303544 |
| Binary code | 2500 | 3100 | 2E00 | 6C00 | 6500 | 6400 | 5B00 | 3200 | 5D00 |

(3)Typ1.No[1]

| Item | Value of code corresponds to character | | | | | | | | |
|----------------------|--|----------|----------|----------|----------|----------|----------|----------|--|
| Abbreviated notation | % | 1 | . | N | o | [| 1 |] | |
| UTF-16 | 0025 | 0031 | 002E | 004E | 006F | 005B | 0031 | 005D | |
| ASCII code | 30303235 | 30303331 | 30303245 | 30303445 | 30303646 | 30303542 | 30303331 | 30303544 | |
| Binary code | 2500 | 3100 | 2E00 | 4E00 | 6F00 | 5B00 | 3100 | 5D00 | |

■Data communication using ASCII code

(Request data)



In the figure (1) to (3), set the value of "ASCII code" indicated in the table of "Value of code corresponds to character" of each label.

(Response data)

■ Data communication using binary code

(Request data)

| Subcommand | Array points | Number of abbreviated points | Label name length | Label name |
|----------------------------|--------------------|------------------------------|-------------------|-------------------|
| 1AH, 04H | 00H, 00H | 02H, 00H | 01H, 00H | 04H, 00H |
| (1) | | | | |
| Abbreviation specification | | | | |
| Label name length | Label name | Unit specification | Array data length | Label name length |
| 09H, 00H | (2) | 01H, 00H | 08H, 00H | 0B, 00H |
| (3) | | | | |
| (Fixed value) | | | | |
| Label name | Unit specification | Array data length | (Fixed value) | |
| 01H, 00H | 04H, 00H | | (4) | |

In the figure (1) to (3), set the value of "Binary code" indicated in the table of "Value of code corresponds to character" of each label.

(Response data)

| Unit specification | | | | Unit specification | | | |
|-----------------------|--------------|-------------------|-----------|--|-------------------|-----------|----------|
| Array points | Data type ID | Array data length | Read data | Data type ID | Array data length | Read data | |
| | | | | | | | |
| 02H, 00H | 03H | 01H | 08H, 00H | 31H, 00H, 32H, 00H, 33H, 00H, 34H, 00H | 03H | 01H | 04H, 00H |
| | | | | 30H, 00H, 31H, 00H | | | |
| Read data of Typ1.led | | | | Read data of Typ1.No | | | |

Batch write array type labels (command: 141A)

Write the values in batch with specifying the consecutive array elements.

Specify the array type labels or array type elements of structure type labels.

The labels other than array type can be specified in one point unit. (Specify the number of array element as '1'.)

Message format

The following shows the message format of the request data and response data of the command.

■Request data

| Command | Subcommand | Array points (n points) | Abbreviation specification | Array specification (first point) | ... | Array specification (nth point) |
|---------|------------|----------------------------|-------------------------------|---|-----|---------------------------------------|
|---------|------------|----------------------------|-------------------------------|---|-----|---------------------------------------|

■Response data

There is no response data for this command.

Data specified by request data

■Command

| ASCII code | Binary code |
|----------------------------------|-------------|
| 1 4 1 A 31H , 34H , 31H , 41H | 1AH , 14H |

■Subcommand

| ASCII code | Binary code |
|----------------------------------|-------------|
| 0 0 0 0 30H , 30H , 30H , 30H | 00H , 00H |

■Array points

Specify the point of array to be written. (☞ Page 126 Points)

■Abbreviation specification

Specify the label name length and label name to be abbreviated. (☞ Page 124 Abbreviation specification of label)

When do not abbreviate, specify '0'.

| ASCII code | Binary code |
|----------------------------------|-------------|
| 0 0 0 0 30H , 30H , 30H , 30H | 00H , 00H |

■Array specification

Specify the details of arrays for number of arrays specified to array points.

| Label name length | Label name | Unit specification | Fixed values | Array data length | Write data |
|---------------------------------------|------------|-----------------------|-----------------|-------------------|------------|
| Array specification (Array one point) | | | | | |

Specify the following items for each array.

- Label name length, label name: Specify the label name and label length of a global label. (☞ Page 119 Labels)
- Unit specification: For the bit type labels, specify the unit specification in bit units (0). As for other than bit type label, specify the unit specification in byte units (1).
- Array data length: Specify the data size of array in the unit specified by "Unit specification." (☞ Page 128 Data length, unit specification)
- Write data: Specify the value to be written. (☞ Page 130 Read data, write data)

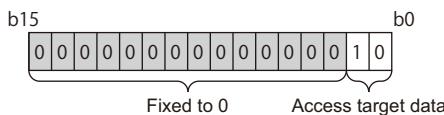
- Fixed value: '0'.

| ASCII code | Binary code |
|---|--|
|  0 0 30H 30H |  00H |

Communication example (Bit specification)

For one-dimensional array type label 'Lbl', write 2-bit data from Lbl [2].

The value of the label to be written is as follows:



■Data communication using ASCII code

(Request data)

In the figure (1), set the value of "ASCII code" indicated in the following table.

(1) Lb|[2]

| Item | Value of code corresponds to character | | | | | |
|------------|--|----------|----------|----------|----------|----------|
| Label name | L | b | I | [| 2 |] |
| UTF-16 | 004C | 0062 | 006C | 005B | 0032 | 005D |
| ASCII code | 30303443 | 30303632 | 30303643 | 30303542 | 30303332 | 30303544 |

■Data communication using binary code

(Request data)

| Subcommand | Array points | Abbreviation specification | Label name length | (Fixed value) | | | | | | | | |
|------------|--------------|----------------------------|-------------------|---------------|----------|----------|----------|--------------------|----------|-------------------|-----|------------|
| | | | | Label name | | | | Unit specification | | Array data length | | Write data |
| 1AH, 14H | 00H, 00H | 01H, 00H | 00H, 00H | L | b | I | [| 2 |] | 00H | 00H | 02H, 00H |
| | | | | 06H, 00H | 4CH, 00H | 62H, 00H | 6CH, 00H | 5BH, 00H | 32H, 00H | 5DH, 00H | | 02H, 00H |

Communication example (Byte specification)

For one-dimensional array type label 'Lbl', write 5-word data from Lbl [2].

■Data communication using ASCII code

(Request data)

| Subcommand | Array points | Abbreviation specification | Label name length | Label name | Unit specification (Fixed value) |
|-------------------------------|-------------------------------|-------------------------------|-------------------------------|-------------------------------|---|
| 1 4 1 A 31H, 34H, 31H, 41H | 0 0 0 0 30H, 30H, 30H, 30H | 0 0 0 1 30H, 30H, 30H, 31H | 0 0 0 0 30H, 30H, 30H, 30H | 0 0 0 6 30H, 30H, 30H, 36H | (1) 30H, 31H 0 1 0 0 30H, 30H, 30H |

| Array data length | | Write data | | | | | | | | | |
|-------------------------------|--|------------|--|--|--|--|--|--|--|--|--|
| 0 0 0 A 30H, 30H, 30H, 41H | 0 0 4 4 0 0 6 1 0 0 7 4 0 0 6 1 0 0 3 1 30H, 30H, 34H, 34H, 30H, 30H, 36H, 31H, 30H, 30H, 37H, 34H, 30H, 30H, 36H, 31H, 30H, 33H, 31H | | | | | | | | | | |

In the figure (1), set the value of "ASCII code" indicated in the following table.

(1) Lbl[2]

| Item | Value of code corresponds to character | | | | | |
|------------|--|----------|----------|----------|----------|----------|
| Label name | L | b | I | [| 2 |] |
| UTF-16 | 004C | 0062 | 006C | 005B | 0032 | 005D |
| ASCII code | 30303443 | 30303632 | 30303643 | 30303542 | 30303332 | 30303544 |

■Data communication using binary code

(Request data)

| Subcommand | Array points | Abbreviation specification | Label name length | Label name |
|----------------------|----------------------|----------------------------|---|------------|
| 1AH, 14H 00H, 00H | 01H, 00H 00H, 00H | 06H, 00H 4CH, 00H | L b I [2] 62H, 00H, 6CH, 00H, 5BH, 00H, 32H, 00H, 5DH, 00H | - |

| Unit specification | Array data length | Write data | | | | | | | | | |
|---------------------|----------------------|----------------------|----------------------|--|--|--|--|--|--|--|--|
| 01H 00H 0AH, 00H | 44H, 00H 61H, 00H | 74H, 00H 61H, 00H | 31H, 00H 31H, 00H | | | | | | | | |

Communication example (Abbreviate with structure type array)

Write the following data to three-dimensional structure label 'Typ1', which has the array type element.

- 8 bytes from Typ1.led[5]
- 4 bytes from Typ1.No[7]

The notation of each label when using abbreviation specification (Typ1= %1) is as follows.

(1) Typ1

| Item | Value of code corresponds to character | | | |
|-------------|--|----------|----------|----------|
| Label name | T | y | p | 1 |
| UTF-16 | 0054 | 0079 | 0070 | 0031 |
| ASCII code | 30303534 | 30303739 | 30303730 | 30303331 |
| Binary code | 5400 | 7900 | 7000 | 3100 |

(2) Typ1.led[5]

| Item | Value of code corresponds to character | | | | | | | | | |
|----------------------|--|----------|----------|----------|----------|----------|----------|----------|----------|--|
| Abbreviated notation | % | 1 | . | I | e | d | [| 5 |] | |
| UTF-16 | 0025 | 0031 | 002E | 006C | 0065 | 0064 | 005B | 0035 | 005D | |
| ASCII code | 30303235 | 30303331 | 30303245 | 30303643 | 30303635 | 30303634 | 30303542 | 30303335 | 30303544 | |
| Binary code | 2500 | 3100 | 2E00 | 6C00 | 6500 | 6400 | 5B00 | 3500 | 5D00 | |

(3) Typ1.No[7]

| Item | Value of code corresponds to character | | | | | | | |
|----------------------|--|----------|----------|----------|----------|----------|----------|----------|
| Abbreviated notation | % | 1 | . | N | o | [| 7 |] |
| UTF-16 | 0025 | 0031 | 002E | 004E | 006F | 005B | 0037 | 005D |
| ASCII code | 30303235 | 30303331 | 30303245 | 30303445 | 30303646 | 30303542 | 30303337 | 30303544 |
| Binary code | 2500 | 3100 | 2E00 | 4E00 | 6F00 | 5B00 | 3700 | 5D00 |

■ Data communication using ASCII code

(Request data)

| Subcommand | Array points | Number of abbreviated points | Label name length | Label name | | | | | | |
|--|---------------------------------|---|-------------------|-------------|----------------------------------|-------------------|-------------------|---------------|---------------------------------|---|
| 1 4 1 A | 0 0 0 0 | 0 0 0 2 | 0 0 0 1 | 0 0 0 4 (1) | | | | | | |
| 31H, 34H, 31H, 41H 30H, 30H, 30H, 30H 30H, 30H, 30H, 32H 30H, 30H, 30H, 31H 30H, 30H, 30H, 34H | | | | | | | | | | |
| Abbreviation specification (%1) | | | | | | | | | | |
| <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <th>Label name length</th> <th>Label name</th> </tr> <tr> <td>0 0 0 9</td> <td>(2)</td> </tr> <tr> <td>30H, 30H, 30H, 39H</td> <td></td> </tr> </table> | | | | | Label name length | Label name | 0 0 0 9 | (2) | 30H, 30H, 30H, 39H | |
| Label name length | Label name | | | | | | | | | |
| 0 0 0 9 | (2) | | | | | | | | | |
| 30H, 30H, 30H, 39H | | | | | | | | | | |
| <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <th>Unit specification (Fixed value)</th> <th>Array data length</th> <th>Write data of (2)</th> </tr> <tr> <td>0 1 0 0 0 0 8</td> <td>1 2 3 4 5 6 7 8 9 A B C D E F 0</td> <td>31H, 31H, 30H, 30H, 30H, 38H 31H, 32H, 33H, 34H, 35H, 36H, 37H, 38H, 39H, 41H, 42H, 43H, 44H, 45H, 46H, 30H</td> </tr> </table> | | | | | Unit specification (Fixed value) | Array data length | Write data of (2) | 0 1 0 0 0 0 8 | 1 2 3 4 5 6 7 8 9 A B C D E F 0 | 31H, 31H, 30H, 30H, 30H, 38H 31H, 32H, 33H, 34H, 35H, 36H, 37H, 38H, 39H, 41H, 42H, 43H, 44H, 45H, 46H, 30H |
| Unit specification (Fixed value) | Array data length | Write data of (2) | | | | | | | | |
| 0 1 0 0 0 0 8 | 1 2 3 4 5 6 7 8 9 A B C D E F 0 | 31H, 31H, 30H, 30H, 30H, 38H 31H, 32H, 33H, 34H, 35H, 36H, 37H, 38H, 39H, 41H, 42H, 43H, 44H, 45H, 46H, 30H | | | | | | | | |
| <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <th>Label name length</th> <th>Label name</th> </tr> <tr> <td>0 0 0 8</td> <td>(3)</td> </tr> <tr> <td>30H, 30H, 30H, 38H</td> <td></td> </tr> </table> | | | | | Label name length | Label name | 0 0 0 8 | (3) | 30H, 30H, 30H, 38H | |
| Label name length | Label name | | | | | | | | | |
| 0 0 0 8 | (3) | | | | | | | | | |
| 30H, 30H, 30H, 38H | | | | | | | | | | |
| <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <th>Unit specification (Fixed value)</th> <th>Array data length</th> <th>Write data of (3)</th> </tr> <tr> <td>0 1 0 0 0 0 4</td> <td>1 2 3 4 5 6 7 8</td> <td>30H, 31H, 30H, 30H, 30H, 34H 31H, 32H, 33H, 34H, 35H, 36H, 37H, 38H</td> </tr> </table> | | | | | Unit specification (Fixed value) | Array data length | Write data of (3) | 0 1 0 0 0 0 4 | 1 2 3 4 5 6 7 8 | 30H, 31H, 30H, 30H, 30H, 34H 31H, 32H, 33H, 34H, 35H, 36H, 37H, 38H |
| Unit specification (Fixed value) | Array data length | Write data of (3) | | | | | | | | |
| 0 1 0 0 0 0 4 | 1 2 3 4 5 6 7 8 | 30H, 31H, 30H, 30H, 30H, 34H 31H, 32H, 33H, 34H, 35H, 36H, 37H, 38H | | | | | | | | |

In the figure (1) to (3), set the value of "ASCII code" indicated in the table of "Value of code corresponding to character" of each label name.

■ Data communication using binary code

(Request data)

| Subcommand | Array points | Number of abbreviated points | Label name length | Label name | | | | | | | | | | | | |
|---|--------------|------------------------------|--------------------|-------------------|---------------------------------------|------------|---------------|--------------------|-------------------|-------------------|----------|-----|-----|-----|----------|---------------------------------------|
| | | | | (1) | | | | | | | | | | | | |
| 1AH, 14H 00H, 00H 02H, 00H 01H, 00H 04H, 00H | | | | | | | | | | | | | | | | |
| Abbreviation specification (%1) | | | | | | | | | | | | | | | | |
| <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <th>Label name length</th> <th>Label name</th> <th>(Fixed value)</th> <th>Unit specification</th> <th>Array data length</th> <th>Write data of (2)</th> </tr> <tr> <td>09H, 00H</td> <td>(2)</td> <td>01H</td> <td>00H</td> <td>08H, 00H</td> <td>34H, 12H, 78H, 56H, BC, 9AH, F0H, DEH</td> </tr> </table> | | | | | Label name length | Label name | (Fixed value) | Unit specification | Array data length | Write data of (2) | 09H, 00H | (2) | 01H | 00H | 08H, 00H | 34H, 12H, 78H, 56H, BC, 9AH, F0H, DEH |
| Label name length | Label name | (Fixed value) | Unit specification | Array data length | Write data of (2) | | | | | | | | | | | |
| 09H, 00H | (2) | 01H | 00H | 08H, 00H | 34H, 12H, 78H, 56H, BC, 9AH, F0H, DEH | | | | | | | | | | | |
| <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <th>Label name length</th> <th>Label name</th> <th>(Fixed value)</th> <th>Unit specification</th> <th>Array data length</th> <th>Write data of (3)</th> </tr> <tr> <td>08H, 00H</td> <td>(3)</td> <td>01H</td> <td>00H</td> <td>04H, 00H</td> <td>34H, 12H, 78H, 56H</td> </tr> </table> | | | | | Label name length | Label name | (Fixed value) | Unit specification | Array data length | Write data of (3) | 08H, 00H | (3) | 01H | 00H | 04H, 00H | 34H, 12H, 78H, 56H |
| Label name length | Label name | (Fixed value) | Unit specification | Array data length | Write data of (3) | | | | | | | | | | | |
| 08H, 00H | (3) | 01H | 00H | 04H, 00H | 34H, 12H, 78H, 56H | | | | | | | | | | | |

In the figure (1) to (3), set the value of "Binary code" indicated in the table of "Value of code corresponding to character" of each label name.

9.3 Random Read and Write

Specify a label and read/write value in one point unit. When reading and writing data in batch by specifying continuous elements of array, use batch read and write command. (☞ Page 131 Batch Read and Write)

Random read labels (command: 041C)

Read value in one point units by specifying multiple labels.

Message format

The following shows the message format of the request data and response data of the command.

■Request data

| | | | | | | |
|---------|------------|----------------------------|-------------------------------|------------------------------------|-----|------------------------------------|
| Command | Subcommand | Label points (n points) | Abbreviation specification | Label specification (1st point) | ... | Label specification (nth point) |
|---------|------------|----------------------------|-------------------------------|------------------------------------|-----|------------------------------------|

■Response data

The data of read label is stored for the number of label points specified with request data.

| | | | |
|----------------------------|---------------------------|-----|---------------------------|
| Label points (n points) | Label data (1st point) | ... | Label data (nth point) |
|----------------------------|---------------------------|-----|---------------------------|

Data specified by request data

■Command

| ASCII code | Binary code |
|----------------------------|-------------|
| 0 4 1 C 30H 34H 31H 43H | 1CH 04H |

■Subcommand

| ASCII code | Binary code |
|----------------------------|-------------|
| 0 0 0 0 30H 30H 30H 30H | 00H 00H |

■Label points

Specify the point of label to read. (☞ Page 126 Points)

■Abbreviation specification

Specify the label name length and label name to be abbreviated. (☞ Page 124 Abbreviation specification of label)
When do not abbreviate, specify '0'.

| ASCII code | Binary code |
|----------------------------|-------------|
| 0 0 0 0 30H 30H 30H 30H | 00H 00H |

■Label specification

Specify the label name and label length of global label for the specified number of label points. (☞ Page 119 Labels)
For a structure type label or array type label, specify the data of each element.

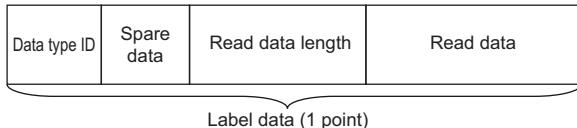
Data stored in response data

■Label points

The same data as request data are stored.

■Label data

The data of read label for the specified number of label points is stored.



The following items are stored for each label.

- Data type ID: The data type of label name is stored with the defined ID. (☞ Page 127 Data type ID)
- Read data length: Specify the read data size in byte units. (☞ Page 128 Data length, unit specification)
- Read data: The value of read label is stored in the format of "Data type ID". (☞ Page 130 Read data, write data)
- Spare data: The 2-byte of system data for data communication with ASCII code, and the 1-byte^{*1} of system data for data communication with binary code is stored.

*1 For C24, the additional code may be added. (☞ Page 33 Additional code (10H))

Communication example

Read data from three labels.

- Lbl1 (bit type) = 1 (ON)
- Lbl2.Lbl[2] (word type array element of structure label) = 0031H
- Lbl3 (word type) = 0001H

The notation of each label when using abbreviation specification (Lbl2 = %1) is as follows.

(1) Lbl2 (abbreviation specification)

| Item | Value of code corresponds to character | | | |
|-------------|--|----------|----------|----------|
| Label name | L | b | I | 2 |
| UTF-16 | 004C | 0062 | 006C | 0032 |
| ASCII code | 30303443 | 30303632 | 30303643 | 30303332 |
| Binary code | 4C00 | 6200 | 6C00 | 3200 |

(2) Lbl1

| Item | Value of code corresponds to character | | | |
|-------------|--|----------|----------|----------|
| Label name | L | b | I | 1 |
| UTF-16 | 004C | 0062 | 006C | 0031 |
| ASCII code | 30303443 | 30303632 | 30303643 | 30303331 |
| Binary code | 4C00 | 6200 | 6C00 | 3100 |

(3) Lbl2.Lbl[2]

| Item | Value of code corresponds to character | | | | | | | | | |
|----------------------|--|----------|----------|----------|----------|----------|----------|----------|----------|--|
| Abbreviated notation | % | 1 | . | L | b | I | [| 2 |] | |
| UTF-16 | 0025 | 0031 | 002E | 004C | 0062 | 006C | 005B | 0032 | 005D | |
| ASCII code | 30303235 | 30303331 | 30303245 | 30303443 | 30303632 | 30303643 | 30303542 | 30303332 | 30303544 | |
| Binary code | 2500 | 3100 | 2E00 | 4C00 | 6200 | 6C00 | 5B00 | 3200 | 5D00 | |

(4) Lbl3

| Item | Value of code corresponds to character | | | |
|-------------|--|----------|----------|----------|
| Label name | L | b | I | 3 |
| UTF-16 | 004C | 0062 | 006C | 0033 |
| ASCII code | 30303443 | 30303632 | 30303643 | 30303333 |
| Binary code | 4C00 | 6200 | 6C00 | 3300 |

■Data communication using ASCII code

(Request data)

| Subcommand | Label points | Number of abbreviated points | Label name length | Label name |
|--------------------|--------------------|------------------------------|--------------------|----------------------|
| 0 4 1 C | 0 0 0 0 | 0 0 0 3 | 0 0 0 1 | 0 0 0 4 (1) ----- |
| 30H, 34H, 31H, 43H | 30H, 30H, 30H, 30H | 30H, 30H, 30H, 33H | 30H, 30H, 30H, 31H | 30H, 30H, 30H, 34H |

| Label name length | Label name | Label name length | Label name | Label name length | Label name |
|--------------------|------------|-------------------------------|------------|-------------------------------|------------|
| 0 0 0 4 ----- | (2) | 0 0 0 9 30H, 30H, 30H, 39H | (3) | 0 0 0 4 30H, 30H, 30H, 34H | (4) |
| 30H, 30H, 30H, 34H | | | | | |

In the figure (1) to (4), set the value of "ASCII code" indicated in the table of "Value of code corresponding to character" of each label name.

(Response data)

| Label points | Data type ID | Spare data | Read data length | Read data | ----- |
|-------------------------------|-------------------------------|-------------------------------|-------------------------------|-------------------------------|-----------------|
| 0 0 0 3 30H, 30H, 30H, 33H | 0 1 30H, 31H | 0 0 30H, 30H | 0 0 0 2 30H, 30H, 30H, 32H | 0 0 0 1 30H, 30H, 30H, 31H | ----- |
| Label data (1st point) | | | | | |
| Data type ID | Spare data | Read data length | Read data | Data type ID | Spare data |
| 0 2 30H, 32H | 0 0 30H, 30H | 0 0 0 2 30H, 30H, 30H, 32H | 0 0 3 1 30H, 30H, 33H, 31H | 0 2 30H, 32H | 0 0 30H, 30H |
| Label data (2nd point) | | | Label data (3rd point) | | |
| Read data length | Read data | ----- | Read data length | Read data | ----- |
| 0 0 0 2 30H, 30H, 30H, 32H | 0 0 0 1 30H, 30H, 30H, 31H | | 0 0 0 2 30H, 30H, 30H, 32H | 0 0 0 1 30H, 30H, 30H, 31H | |

■Data communication using binary code

(Request data)

| Subcommand | Label points | Number of abbreviated points | Label name length | Label name | ----- |
|------------|--------------|------------------------------|-------------------|-----------------------|-------|
| 1CH, 04H | 00H, 00H | 03H, 00H | 01H, 00H | 04H, 00H (1) ----- | |
| | | | | | |

| Label name length | Label name | Label name length | Label name | Label name length | Label name |
|-------------------|------------|-------------------|------------|-------------------|------------|
| 04H, 00H ----- | (2) | 09H, 00H | (3) | 04H, 00H | (4) |
| 04H, 00H | | | | | |

In the figure (1) to (4), set the value of "Binary code" indicated in the table of "Value of code corresponding to character" of each label name.

(Response data)

| Data type ID | | | Data type ID | | | Data type ID | | | |
|------------------------|------------|------------------|------------------------|------------|------------------|------------------------|------------|------------------|-----------|
| Label points | Spare data | Read data length | Read data | Spare data | Read data length | Read data | Spare data | Read data length | Read data |
| 03H, 00H | 01H, 00H | 02H, 00H | 01H, 00H | 02H, 00H | 02H, 00H | 31H, 00H | 02H, 00H | 02H, 00H | 01H, 00H |
| Label data (1st point) | | | Label data (2nd point) | | | Label data (3rd point) | | | |
| | | | | | | | | | |

Random write labels (command: 141B)

Write value in one point units by specifying multiple labels.

Message format

The following shows the message format of the request data and response data of the command.

■Request data

| Command | Subcommand | Label points (n points) | Abbreviation specification | Label specification (1st point) | ... | Label specification (nth point) |
|---------|------------|----------------------------|-------------------------------|---------------------------------------|-----|------------------------------------|
|---------|------------|----------------------------|-------------------------------|---------------------------------------|-----|------------------------------------|

■Response data

There is no response data for this command.

Data specified by request data

■Command

| ASCII code | Binary code |
|----------------------------------|-------------|
| 1 4 1 B 31H , 34H , 31H , 42H | 1BH , 14H |

■Subcommand

| ASCII code | Binary code |
|----------------------------------|-------------|
| 0 0 0 0 30H , 30H , 30H , 30H | 00H , 00H |

■Label points

Specify the number of label points to be written. ([Page 126 Points](#))

■Abbreviation specification

Specify the label name length and label name to be abbreviated. ([Page 119 Labels](#))

When do not abbreviate, specify '0'.

| ASCII code | Binary code |
|----------------------------------|-------------|
| 0 0 0 0 30H , 30H , 30H , 30H | 00H , 00H |

■Label specification

Specify the label name and write data for the specified number of label points.

| | | | |
|-------------------|------------|-------------------|------------|
| Label name length | Label name | Write data length | Write data |
|-------------------|------------|-------------------|------------|

Label specification (1 point)

Specify the following items for each label.

- Label name length, label name: Specify the label name and label length of a global label. ([Page 119 Labels](#))
- Write data length: Specify the read data size in byte unit. ([Page 128 Data length, unit specification](#))
- Write data: Store the values of labels to be written. ([Page 130 Read data, write data](#))

Communication example

Write data to three labels.

- Lbl1 (bit type) = 1 (ON)
- Lbl2.Lbl[2] (word type array element of structure label) = 0031H
- Lbl3 (word type) = 0001H

The notation of each label when using abbreviation specification (Lbl2 = %1) is as follows.

(1) Lbl2 (abbreviation specification)

| Item | Value of code corresponds to character | | | |
|-------------|--|----------|----------|----------|
| Label name | L | b | I | 2 |
| UTF-16 | 004C | 0062 | 006C | 0032 |
| ASCII code | 30303443 | 30303632 | 30303643 | 30303332 |
| Binary code | 4C00 | 6200 | 6C00 | 3200 |

(2) Lbl1

| Item | Value of code corresponds to character | | | |
|-------------|--|----------|----------|----------|
| Label name | L | b | I | 1 |
| UTF-16 | 004C | 0062 | 006C | 0031 |
| ASCII code | 30303443 | 30303632 | 30303643 | 30303331 |
| Binary code | 4C00 | 6200 | 6C00 | 3100 |

(3) Lbl2.Lbl[2]

| Item | Value of code corresponds to character | | | | | | | | | |
|----------------------|--|----------|----------|----------|----------|----------|----------|----------|----------|--|
| Abbreviated notation | % | 1 | . | L | b | I | [| 2 |] | |
| UTF-16 | 0025 | 0031 | 002E | 004C | 0062 | 006C | 005B | 0032 | 005D | |
| ASCII code | 30303235 | 30303331 | 30303245 | 30303443 | 30303632 | 30303643 | 30303542 | 30303332 | 30303544 | |
| Binary code | 2500 | 3100 | 2E00 | 4C00 | 6200 | 6C00 | 5B00 | 3200 | 5D00 | |

(4) Lbl3

| Item | Value of code corresponds to character | | | |
|-------------|--|----------|----------|----------|
| Label name | L | b | I | 3 |
| UTF-16 | 004C | 0062 | 006C | 0033 |
| ASCII code | 30303443 | 30303632 | 30303643 | 30303333 |
| Binary code | 4C00 | 6200 | 6C00 | 3300 |

■Data communication using ASCII code

(Request data)

| Subcommand | Label points | Number of abbreviated points | Label name length | Label name |
|-------------------------------|-------------------------------|-------------------------------|-------------------------------|--------------------------------------|
| 1 4 1 B 31H, 34H, 31H, 42H | 0 0 0 0 30H, 30H, 30H, 30H | 0 0 0 3 30H, 30H, 30H, 33H | 0 0 0 1 30H, 30H, 30H, 31H | 0 0 0 4 30H, 30H, 30H, 34H (1) |

| Label name length | Label name | Write data length | Write data |
|-------------------------------|------------|-------------------------------|-------------------------------|
| 0 0 0 4 30H, 30H, 30H, 34H | (2) | 0 0 0 2 30H, 30H, 30H, 32H | 0 0 0 1 30H, 30H, 30H, 31H |

| Label name length | Label name | Write data length | Write data |
|-------------------------------|------------|-------------------------------|-------------------------------|
| 0 0 0 9 30H, 30H, 30H, 39H | (3) | 0 0 0 2 30H, 30H, 30H, 32H | 0 0 3 1 30H, 30H, 33H, 31H |

| Label name length | Label name | Write data length | Write data |
|-------------------------------|------------|-------------------------------|-------------------------------|
| 0 0 0 4 30H, 30H, 30H, 34H | (4) | 0 0 0 2 30H, 30H, 30H, 32H | 0 0 0 1 30H, 30H, 30H, 31H |

In the figure (1) to (4), set the value of "ASCII code" indicated in the table of "Value of code corresponding to character" of each label name.

■Data communication using binary code

(Request data)

| Subcommand | Label points | Number of abbreviated points | Label name length | Label name |
|------------|--------------|------------------------------|-------------------|-----------------|
| 1BH, 14H | 00H, 00H | 03H, 00H | 01H, 00H | 04H, 00H (1) |

| Label name length | Label name | Write data length | Write data |
|-------------------|------------|-------------------|------------|
| 04H, 00H | (2) | 02H, 00H | 01H, 00H |

| Label name length | Label name | Write data length | Write data |
|-------------------|------------|-------------------|------------|
| 09H, 00H | (3) | 02H, 00H | 31H, 00H |

| Label name length | Label name | Write data length | Write data |
|-------------------|------------|-------------------|------------|
| 04H, 00H | (4) | 02H, 00H | 01H, 00H |

In the figure (1) to (4), set the value of "Binary code" indicated in the table of "Value of code corresponding to character" of each label name.

10 BUFFER MEMORY ACCESS

10

This chapter explains the commands which read and write the buffer memory.

Point

The buffer memory can be accessed with device access function using module access device (Un\G).

☞ Page 436 Accessing module access devices

☞ Page 63 DEVICE ACCESS

10.1 Buffer Memory

This section explains the command which reads and writes data to the buffer memory of the supported device connected to the external device.

Restriction

The command can only be used for C24 (including multidrop connection station) and E71 connected to an external device. It cannot be used via network.

Point

This command is processed by C24/E71 connected to the CPU module without waiting for the END process.

Data to be specified in commands

This section explains the contents and specification methods for data items which are set in each command related to the access to the host station (supported device) buffer memory.

Start address

Specify the start address of the buffer memory to be read/written.

■Data communication using ASCII code

Convert the numerical value to 8-digit (hexadecimal) ASCII code and send it from the upper digits.

■Data communication in binary code

Send 4-byte numerical values from the lower byte (L: bits 0 to 7).

Ex.

When the head area address is 1E1H

| ASCII code | Binary code |
|--|-----------------|
| 0 0 0 0 0 1 E 1 30H 30H 30H 30H 31H 45H 31H | E1H 01H 00H 00H |

Word length

Specify the word length of the buffer memory to be read/written.

■Data communication using ASCII code

Convert the numerical value to 4-digit ASCII code (hexadecimal) and send it from upper digits.

■Data communication in binary code

Send 2-byte numerical values from lower byte (L: bits 0 to 7).

Ex.

For 5 words and 20 words

| Number of device points | ASCII code | Binary code | | | | | | | | | | |
|-------------------------|--|-------------|-----|---|---|-----|-----|-----|-----|---|-----|-----|
| 5 words | <table border="1"><tr><td>0</td><td>0</td><td>0</td><td>5</td></tr><tr><td>30H</td><td>30H</td><td>30H</td><td>35H</td></tr></table> | 0 | 0 | 0 | 5 | 30H | 30H | 30H | 35H | <table border="1"><tr><td>05H</td><td>00H</td></tr></table> | 05H | 00H |
| 0 | 0 | 0 | 5 | | | | | | | | | |
| 30H | 30H | 30H | 35H | | | | | | | | | |
| 05H | 00H | | | | | | | | | | | |
| 20 words | <table border="1"><tr><td>0</td><td>0</td><td>1</td><td>4</td></tr><tr><td>30H</td><td>30H</td><td>31H</td><td>34H</td></tr></table> | 0 | 0 | 1 | 4 | 30H | 30H | 31H | 34H | <table border="1"><tr><td>14H</td><td>00H</td></tr></table> | 14H | 00H |
| 0 | 0 | 1 | 4 | | | | | | | | | |
| 30H | 30H | 31H | 34H | | | | | | | | | |
| 14H | 00H | | | | | | | | | | | |

Read data, write data

The read buffer memory value is stored for reading, and the data to be written is stored for writing.

This function reads/writes data in word unit.

■Data communication using ASCII code

Convert the numerical value to 4-digit ASCII code (hexadecimal) and send it from upper digits.

■Data communication in binary code

Send 2-byte numerical values from lower byte (L: bits 0 to 7).

Ex.

When the data for one buffer memory address is 09C1H

| ASCII code | Binary code | | | | | | | | | | |
|--|-------------|-----|-----|---|-----|-----|-----|-----|---|-----|-----|
| <table border="1"><tr><td>0</td><td>9</td><td>C</td><td>1</td></tr><tr><td>30H</td><td>39H</td><td>43H</td><td>31H</td></tr></table> | 0 | 9 | C | 1 | 30H | 39H | 43H | 31H | <table border="1"><tr><td>C1H</td><td>09H</td></tr></table> | C1H | 09H |
| 0 | 9 | C | 1 | | | | | | | | |
| 30H | 39H | 43H | 31H | | | | | | | | |
| C1H | 09H | | | | | | | | | | |

Batch read (command: 0613)

Read data from the buffer memory of the host station (supported device).

Message format

The following shows the message format of the request data and response data of the command.

■Request data

| Command | Subcommand | Start address | Word length |
|---------|------------|---------------|-------------|
| 0613H | 00H | 30H | 01H |

■Response data

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

☞ Page 148 Read data, write data

Data specified by request data

■Command

| ASCII code | Binary code |
|----------------------------|-------------|
| 0 6 1 3 30H 36H 31H 33H | 13H 06H |

■Subcommand

| ASCII code | Binary code |
|----------------------------|-------------|
| 0 0 0 0 30H 30H 30H 30H | 00H 00H |

■Start address

Specify the buffer memory start address to be read. (☞ Page 147 Start address)

■Word length

Specify the word length of the buffer memory to be read. (☞ Page 148 Word length)

Specification range: 1H to 1E0H (480)



Specify the access range within the range of buffer memory.

(Start address + Word length -1) ≤ Buffer memory range

Communication example

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

■Data communication using ASCII code

(Request data)

| Start address | | | | Word length | | |
|---------------|-----|-----|-----|-------------|-----|-----|
| 0 | 6 | 1 | 3 | 0 | 0 | 0 |
| 30H | 36H | 31H | 33H | 30H | 30H | 30H |

(Response data)

| Read data 1 | Read data 2 | Read data 10 |
|-----------------|-----------------|-----------------|
| 0 5 0 0 | 0 9 C 1 | ... 0 0 C 8 |
| 30H 35H 30H 30H | 30H 39H 43H 31H | 30H 30H 43H 38H |

Values of address 78H =0500H
Values of address 79H =09C1H
Values of address 81H =00C8H

■Data communication in binary code

(Request data)

| Start address | | Word length | |
|---------------|-----|-------------|-----|
| 13H | 06H | 00H | 00H |

(Response data)

| Read data 1 | Read data 2 | Read data 10 |
|------------------------------|------------------------------|------------------------------|
| 00H 05H | C1H 09H | ... C8H 00H |
| Values of address 78H =0500H | Values of address 79H =09C1H | Values of address 81H =00C8H |

Batch write (command: 1613)

Write data to the buffer memory of the host station (supported device).

Restriction

Do not write any data in "System area" or "Write-protect area" in the buffer memory.

Writing data to the "System area" or "Write-protect area" may cause malfunction of the programmable controller system.

Message format

The following shows the message format of the request data and response data of the command.

■Request data

| | | | | | | |
|---------|------------|---------------|-------------|--------------|-----|--------------|
| Command | Subcommand | Start address | Word length | Write data 1 | ... | Write data n |
|---------|------------|---------------|-------------|--------------|-----|--------------|

■Response data

There is no response data for this command.

Data specified by request data

■Command

| ASCII code | Binary code |
|----------------------------------|-------------|
| 1 6 1 3 31H , 36H , 31H , 33H | 13H , 16H |

■Subcommand

| ASCII code | Binary code |
|----------------------------------|-------------|
| 0 0 0 0 30H , 30H , 30H , 30H | 00H , 00H |

■Start address

Specify the buffer memory start address to be written. (☞ Page 147 Start address)

■Word length

Specify the word length of the buffer memory to be written. (☞ Page 148 Word length)

Specification range: 1H to 1E0H(480)

■Write data

Specify the data to be written in the buffer memory. (☞ Page 148 Read data, write data)

Point

Specify the access range within the range of buffer memory.

(Start address + Word length -1) ≤ Buffer memory range

Communication example

Write the value to buffer memory addresses from 2680H to 2683H (9856 to 9859).

■Data communication using ASCII code

(Request data)

| Start address | Word length | Write data 1 | Write data 4 |
|--------------------------------|----------------------------|--|----------------------------------|
| 1 6 1 3 31H 36H 31H 33H | 0 0 0 0 30H 30H 30H 30H | 0 0 0 2 6 8 0 30H 30H 30H 32H 36H 38H 30H | 0 0 0 4 30H 30H 30H 34H |
| Values of address 2680H =2000H | | | 2 0 0 0 0 ... 32H 30H 30H 30H |
| Values of address 2683H =0H | | | 0 0 0 0 30H 30H 30H 30H |

■Data communication in binary code

(Request data)

| Start address | Word length | Write data 1 | Write data 4 |
|--------------------------------|-------------|-----------------|-----------------|
| 13H 16H | 00H 00H | 80H 26H 00H 00H | 04H 00H 00H 20H |
| Values of address 2680H =2000H | | | ... |
| Values of address 2683H =0H | | | 00H 00H |

10.2 Intelligent Function Module

The section explains the commands to read from/write to the buffer memory of an intelligent function module.

Accessible modules

The following shows the accessible intelligent functional modules to buffer memory.

10

Accessing buffer memory using module access device (Un\G)

The intelligent function modules, which can be accessed the module access device (Un\G), can be accessed by the device access function.

☞ Page 436 Accessing module access devices

☞ Page 63 DEVICE ACCESS

Accessing buffer memory with calculating a start address

Use the command (0601, 1601) shown in this section to access the following devices.

| Module | Model | Additional values when calculating start address |
|---|---|--|
| Load cell input module | Q61LD | 2000H |
| Loop control module | Q62HLC | 10000H |
| Analog—digital converter module | Q62AD-DGH, Q64AD, Q64AD-GH, Q66AD-DG, Q68AD-G, Q68ADV, Q68ADI | 1008H |
| Digital—analog converter module | Q62DA, Q62DA-FG, Q62DAN, Q64DA, Q64DAN, Q66DA-G, Q68DAV, Q68DAI, Q68DAVN, Q68DAIN | 1008H |
| Analog input/output module | Q64AD2DA | 2000H |
| Temperature control module | Q64TCTT, Q64TCRT, Q64TCTTBW, Q64TCRTBW | 1000H |
| Temperature input module (function version B) | Q64TD, Q64RD | 2000H |
| Temperature input module (function version C) | Q64TD, Q64TDV-GH, Q64RD, Q64RD-G | 8000H |
| Channel isolated thermocouple input module | Q68TD-G-H01, Q68TD-G-H02 | 1008H |
| Channel isolated RTD input module | Q68RD3-G | 1008H |
| ID Interface module | QD35ID1, QD35ID2 | 4000H |
| Intelligent communication module | QD51, QD51-R24 | 10000H |
| Channel isolated pulse input module | QD60P8-G | 2000H |
| High-speed counter module | QD62, QD62E, QD62D | 3CH |
| Multichannel high-speed counter module | QD63P6 | 2000H |
| 4Mpps capable high-speed counter module | QD64D2 | 2000H |
| Positioning module | QD70P4, QD70P8, QD70D4, QD70D8, QD72P3C3 QD75P1, QD75P2, QD75P4, QD75D1, QD75D2, QD75D4, QD75M1, QD75M2, QD75M4, QD75MH1, QD75MH2, QD75MH4 | 5000H 10000H |
| High speed data logger module | QD81DL96 | 10000H |
| CC-Link system master/local module | QJ61BT11, QJ61BT11N | 10000H |
| CC-Link/LT master module | QJ61CL12 | 01B4H |
| Serial communication module | QJ71C24N, QJ71C24N-R2, QJ71C24N-R4, QJ71C24, QJ71C24-R2 | 10000H |
| AS-i master module | QJ71AS92 | 10000H |
| Ethernet interface module | QJ71E71-100, QJ71E71-B5, QJ71E71-B2 | 10000H |
| FL-net (OPCN-2) interface module | QJ71FL71-T, QJ71FL71-B2, QJ71FL71-B5, QJ71FL71-T-F01, QJ71FL71-B2-F01, QJ71FL71-B5-F01 | 10000H |
| MODBUS interface module | QJ71MB91, QJ71MT91 | 10000H |
| MES interface module | QJ71MES96 | 10000H |
| Web server module | QJ71WS96 | 10000H |
| PROFIBUS-DP Interface module | QJ71PB92D | 10000H |
| PROFIBUS-DP Master module | QJ71PB92V | 10000H |
| PROFIBUS-DP Slave module | QJ71PB93D | 10004H |

By using the command (0601, 1601), MELSEC-QnA series special function modules can be accessed.

| Module | Model | Additional values when calculating start address |
|------------------------------------|--|--|
| CC-Link system master/local module | AJ61QBT11, A1SJ61QBT11 | 2000H |
| Ethernet interface module | AJ71QE71, AJ71QE71-B5, A1SJ71QE71-B2, A1SJ71QE71-B5 | 4000H |
| Serial communication module | AJ71QC24, AJ71QC24-R2, AJ71QC24-4, AJ71QC24N, AJ71QC24N-R2, AJ71QC24N-R4, A1SJ71QC24, A1SJ71QC24-R2, A1SJ71QC24N, A1SJ71QC24N-R2 | 4000H |

When accessing buffer memory of MELSEC-A series special function modules, use the command of 1C/1E frame.

- 1C frame: [Page 377 Read and write Buffer Memory of Special Function Module](#)
- 1E frame: [Page 426 Read and Write Buffer Memory of Special Function Module](#)

Data to be specified in commands

This section explains the contents and specification methods for data items which are set in each command related to the access to the intelligent function module buffer memory.

Start address

Specify the start address of the buffer memory to be read/written.

■Calculation method

To access the buffer memory of the intelligent function module which consisted of word units by byte unit, specify a start address calculated by byte unit.

Calculate the start address as follows:

Start address = (Buffer memory address $\times 2$) + Additional value of a module

For the arbitrary additional value of the module, refer to the following section.

[Page 153 Accessing buffer memory with calculating a start address](#)

Ex.

When specifying Q62DA buffer memory address 18H

$(18H \times 2) + 1008H = 30H + 1008H = 1038H$

■Data communication using ASCII code

Convert the numerical value to 8-digit (hexadecimal) ASCII code and send it from the upper digits.

■Data communication in binary code

Send 4-byte numerical values from the lower byte (L: bits 0 to 7).

Ex.

When the start address is 1038H

| ASCII code | Binary code |
|--|-----------------------|
| 0 0 0 0 1 0 3 8 30H , 30H , 30H , 30H , 31H , 30H , 33H , 38H | 38H , 10H , 00H , 00H |

Number of bytes

Specify the number of bytes of the data to be read/written.

■Calculation method

The buffer memory for intelligent function module consists of two bytes (one word) for one area. Calculate the number of bytes by 2-byte per data for one buffer memory address.

$$\text{Number of bytes} = (\text{Number of buffer memory address} \times 2)$$

Ex.

When accessing buffer memory address 160 to 161 (A0H to A1H)

$$(161 - 160 + 1) \times 2 = 2 \times 2 = 4 \text{ bytes}$$

■Data communication using ASCII code

Convert the numerical value to 4-digit ASCII code (hexadecimal) and send it from upper digits.

Use capitalized code for alphabetical letter.

■Data communication in binary code

Send 2-byte numerical values from lower byte (L: bits 0 to 7).

Ex.

For 20 bytes

| Number of device points | ASCII code | Binary code | | | | | | | | | | |
|-------------------------|--|-------------|-----|---|---|-----|-----|-----|-----|---|-----|-----|
| 20 words | <table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td>0</td><td>0</td><td>1</td><td>4</td></tr> <tr> <td>30H</td><td>30H</td><td>31H</td><td>34H</td></tr> </table> | 0 | 0 | 1 | 4 | 30H | 30H | 31H | 34H | <table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td>14H</td><td>00H</td></tr> </table> | 14H | 00H |
| 0 | 0 | 1 | 4 | | | | | | | | | |
| 30H | 30H | 31H | 34H | | | | | | | | | |
| 14H | 00H | | | | | | | | | | | |

Module number

Specify the start input/output number of an intelligent function module to be accessed.

For the module number, specify the value obtained by dividing the start input/output number by 16 in 4 digits (hexadecimal).

■When the number of occupied slots are 2

Specify the value obtained by adding 1 to the module number for the following modules.

| Module type | Model |
|----------------------------|----------------------|
| Temperature control module | Q64TCTTBW, Q64TCRTBW |
| Positioning module | QD70D4, QD70D8 |

■Data communication using ASCII code

Convert the numerical value to 4-digit ASCII code (hexadecimal) and send it from upper digits.

Use capitalized code for alphabetical letter.

■Data communication in binary code

Send 2-byte numerical values from lower byte (L: bits 0 to 7).

Ex.

When accessing the positioning module whose start input/output number is 0080H

| Module number | ASCII code | Binary code | | | | | | | | | | |
|---|--|-------------|-----|---|---|-----|-----|-----|-----|---|-----|-----|
| For QD70P4: 0008H | <table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td>0</td><td>0</td><td>0</td><td>8</td></tr> <tr> <td>30H</td><td>30H</td><td>30H</td><td>38H</td></tr> </table> | 0 | 0 | 0 | 8 | 30H | 30H | 30H | 38H | <table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td>08H</td><td>00H</td></tr> </table> | 08H | 00H |
| 0 | 0 | 0 | 8 | | | | | | | | | |
| 30H | 30H | 30H | 38H | | | | | | | | | |
| 08H | 00H | | | | | | | | | | | |
| For QD70D4 (occupied slots are 2): 0008H + 1 = 0009H | <table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td>0</td><td>0</td><td>0</td><td>9</td></tr> <tr> <td>30H</td><td>30H</td><td>30H</td><td>39H</td></tr> </table> | 0 | 0 | 0 | 9 | 30H | 30H | 30H | 39H | <table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td>09H</td><td>00H</td></tr> </table> | 09H | 00H |
| 0 | 0 | 0 | 9 | | | | | | | | | |
| 30H | 30H | 30H | 39H | | | | | | | | | |
| 09H | 00H | | | | | | | | | | | |

■Intelligent function module number of MELSECNET/H remote I/O station

Intelligent function module number of MELSECNET/H remote I/O station is the upper 2 digits of the last number represented with 3-digit number of the following "Input/output signal based on the remote I/O station".

Specify this with "Input/output signal based on the remote I/O station" regardless of the contents of common parameters set in the master station of MELSECNET/H remote I/O network.

| | | | | | | | Intelligent function module No. "04H" |
|--|--------------------|--------------------|----------------------|-----------------------------------|--------------------|------------------|---------------------------------------|
| | | | | | | | -----► |
| Input/output signal based on the remote I/O station | Y 00 to 1F | Y 20 to 2F | X/Y 30 to 4F | Y 50 to 6F | Y 70 to 8F | | |
| Power supply module | AJ72 LP25 | Output module | Output module | Intelligent function module | Output module | Output module | |
| Remote I/O station, station 1 | | 32 points | 16 points | 32 points | 16 points | 32 points | |
| Input/output signal by common parameters | Y 400 to 41F | Y 420 to 42F | X/Y 430 to 44F | Y 450 to 46F | Y 470 to 48F | | |

Read data, write data

The read buffer memory value is stored for reading, and the data to be written is stored for writing.

This function reads/writes data in byte unit.

■Data communication using ASCII code

Handle a word data for one buffer memory address as 2-byte data.

Convert the value to 2-digit (hexadecimal) ASCII code per one byte and send it from the upper digit.

■Data communication in binary code

Handle a word data for one buffer memory address as 2-byte data.

Send 2-byte numerical values from lower byte (L: bits 0 to 7).

Ex.

When the data for one buffer memory address is 09C1H

| ASCII code | Binary code |
|----------------------------|-------------|
| C 1 0 9 43H 31H 30H 39H | C1H 09H |

When reading data from the following buffer memory addresses, 0H to 2H (additional value of the start address: 10000H)

| Buffer memory | | | Read 6 bytes from address 0H | | | Read 4 bytes from address 1H | | |
|---------------|-------------|-----------|------------------------------|-----------------------|-------------|------------------------------|-----------------------|-------------|
| Address | Stored data | | Start address | Read data (byte unit) | | Start address | Read data (byte unit) | |
| | Word unit | Byte unit | | ASCII code | Binary code | | ASCII code | Binary code |
| 0H | 0003H | 03H | 10000H | 30H, 33H | 03H | — | 30H, 31H | 01H |
| | | 00H | | 30H, 30H | 00H | | | |
| 1H | 0001H | 01H | 10002H | 30H, 31H | 01H | 30H, 30H | 00H | 12H |
| | | 00H | | 30H, 30H | 00H | | | |
| 2H | 0012H | 12H | | 31H, 32H | 12H | | | |
| | | 00H | | 30H, 30H | 00H | | | |

Batch read (command: 0601)

Read the data from the buffer memory of an intelligent function module.

Message format

The following shows the message format of the request data and response data of the command.

■Request data

| | | | | |
|---------|------------|---------------|-----------------|------------|
| Command | Subcommand | Start address | Number of bytes | Module No. |
|---------|------------|---------------|-----------------|------------|

■Response data

The value read from the buffer memory is stored.

☞ Page 156 Read data, write data

Data specified by request data

■Command

| ASCII code | Binary code | | | | | | | | | | |
|---|-------------|-----|-----|---|-----|-----|-----|-----|---|-----|-----|
| <table border="1"> <tr> <td>0</td><td>6</td><td>0</td><td>1</td> </tr> <tr> <td>30H</td><td>36H</td><td>30H</td><td>31H</td> </tr> </table> | 0 | 6 | 0 | 1 | 30H | 36H | 30H | 31H | <table border="1"> <tr> <td>01H</td><td>06H</td> </tr> </table> | 01H | 06H |
| 0 | 6 | 0 | 1 | | | | | | | | |
| 30H | 36H | 30H | 31H | | | | | | | | |
| 01H | 06H | | | | | | | | | | |

■Subcommand

| ASCII code | Binary code | | | | | | | | | | |
|---|-------------|-----|-----|---|-----|-----|-----|-----|---|-----|-----|
| <table border="1"> <tr> <td>0</td><td>0</td><td>0</td><td>0</td> </tr> <tr> <td>30H</td><td>30H</td><td>30H</td><td>30H</td> </tr> </table> | 0 | 0 | 0 | 0 | 30H | 30H | 30H | 30H | <table border="1"> <tr> <td>00H</td><td>00H</td> </tr> </table> | 00H | 00H |
| 0 | 0 | 0 | 0 | | | | | | | | |
| 30H | 30H | 30H | 30H | | | | | | | | |
| 00H | 00H | | | | | | | | | | |

■Start address

Specify the buffer memory start address to be read. (☞ Page 154 Start address)

■Number of bytes

Specify the number of bytes of the buffer memory to be read. (☞ Page 155 Number of bytes)

Specification range: 2H to 780H (1920)

■Module number

Specify the intelligent function module to be read. (☞ Page 155 Module number)



Specify the access range within the range of buffer memory.

Communication example

Read the data of the buffer memory addresses 1H to 2H of Q62DA whose input/output signal is from 30H to 4FH (module No.: 03H).

| Buffer memory of Q62DA | | | Data for command |
|------------------------|-------------------|-------------------------|-----------------------|
| Address | Name | Stored data (word unit) | Read data (byte unit) |
| 1H | CH1 digital value | 0001H | 01H |
| | | | 00H |
| 2H | CH2 digital value | 0012H | 12H |
| | | | 00H |

■Data communication using ASCII code

(Request data)

| Start address | Number of bytes | Module No. |
|-------------------------------|-------------------------------|-------------------------------|
| 0 6 0 1 30H, 36H, 30H, 31H | 0 0 0 0 30H, 30H, 30H, 30H | 0 0 0 1 30H, 30H, 30H, 31H |

(Response data)

| Read data | | | |
|----------------------|-----------------|----------------------|-----------------|
| 0 1 30H, 31H | 0 0 30H, 30H | 1 2 31H, 32H | 0 0 30H, 30H |
| Values of address 1H | | Values of address 2H | |

=0001H =0012H

■Data communication in binary code

(Request data)

| Start address | Number of bytes | Module No. |
|---------------|-----------------|--------------------|
| 01H, 06H | 00H, 00H | 0AH, 10H, 00H, 00H |

(Response data)

| Read data | | | |
|----------------------|-----|----------------------|-----|
| 01H | 00H | 12H | 00H |
| Values of address 1H | | Values of address 2H | |

=0001H =0012H

Batch write (command: 1601)

Write data to the buffer memory of an intelligent function module.

Restriction

Do not write any data in "System area" or "Write-protect area" in the buffer memory.

Writing data to the "System area" or "Write-protect area" may cause malfunction of the programmable controller system.

Message format

The following shows the message format of the request data and response data of the command.

■Request data

| Command | Subcommand | Start address | Number of bytes | Module No. | Write data |
|---------|------------|---------------|-----------------|------------|------------|
| | | | | | |

■Response data

There is no response data for this command.

Data specified by request data

■Command

| ASCII code | Binary code | | | | | | | | | | |
|---|-------------|-----|-----|---|-----|-----|-----|-----|--|-----|-----|
| <table border="1"> <tr> <td>1</td> <td>6</td> <td>0</td> <td>1</td> </tr> <tr> <td>31H</td> <td>36H</td> <td>30H</td> <td>31H</td> </tr> </table> | 1 | 6 | 0 | 1 | 31H | 36H | 30H | 31H | <table border="1"> <tr> <td>01H</td> <td>16H</td> </tr> </table> | 01H | 16H |
| 1 | 6 | 0 | 1 | | | | | | | | |
| 31H | 36H | 30H | 31H | | | | | | | | |
| 01H | 16H | | | | | | | | | | |

■Subcommand

| ASCII code | Binary code | | | | | | | | | | |
|---|-------------|-----|-----|---|-----|-----|-----|-----|--|-----|-----|
| <table border="1"> <tr> <td>0</td> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>30H</td> <td>30H</td> <td>30H</td> <td>30H</td> </tr> </table> | 0 | 0 | 0 | 0 | 30H | 30H | 30H | 30H | <table border="1"> <tr> <td>00H</td> <td>00H</td> </tr> </table> | 00H | 00H |
| 0 | 0 | 0 | 0 | | | | | | | | |
| 30H | 30H | 30H | 30H | | | | | | | | |
| 00H | 00H | | | | | | | | | | |

■Start address

Specify the buffer memory start address to be written. ([Page 154 Start address](#))

■Number of bytes

Specify the number of bytes of the buffer memory to be written. ([Page 155 Number of bytes](#))

Specification range: 2H to 780H (1920)

■Module number

Specify the intelligent function module to be written. ([Page 155 Module number](#))

■Write data

Specify the data to be written in the buffer memory. ([Page 156 Read data, write data](#))

Point

Specify the access range within the range of buffer memory.

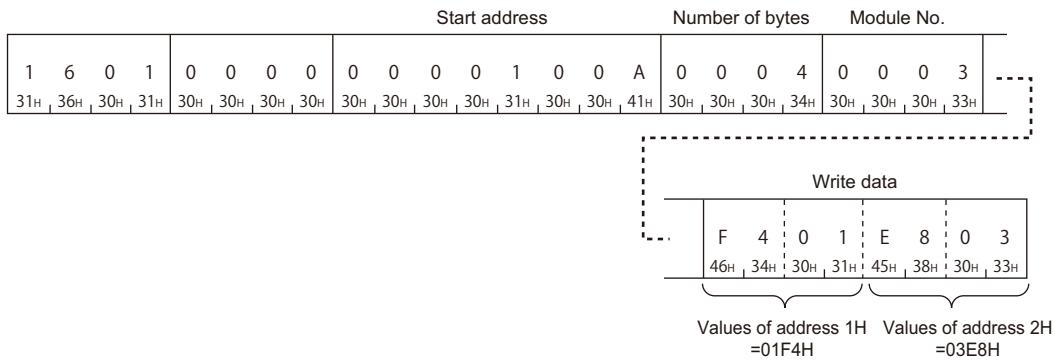
Communication example

Write data to the buffer memory addresses from 1H to 2H of Q62DA whose input/output signals are from 30H to 4FH (module No.: 03H).

| Buffer memory of Q62DA | | | Data for command |
|------------------------|-------------------|-------------------------|------------------------|
| Address | Name | Stored data (word unit) | Write data (byte unit) |
| 1H | CH1 digital value | 01F4H | F4H |
| | | | 01H |
| 2H | CH2 digital value | 03E8H | E8H |
| | | | 03H |

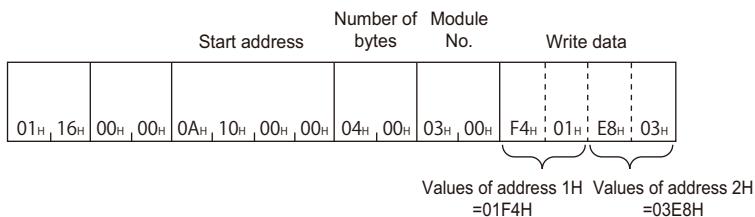
■Data communication using ASCII code

(Request data)



■Data communication in binary code

(Request data)



11 CONTROL MODULE OPERATION

This chapter explains the commands for changing operation status and performing test using the functions of the module.

11.1 Data to be specified in commands

This section explains the contents and specification methods for data items which are set in each command related to module control.

11

Mode

Select the operation when the request target module is already in remote operation by other device.

Setting method

■Data communication using ASCII code

Convert the numerical value to 4-digit ASCII code (hexadecimal) and send it from upper digits.

■Data communication using binary code

Send 2-byte numerical values from lower byte (L: bits 0 to 7).

| Process | | ASCII code | Binary code | | | | | | | | | | |
|-------------------------|--|--|-------------|---|---|---|-----|-----|-----|-----|---|-----|-----|
| Do not execute forcibly | Do not apply remote RUN/remote PAUSE while remote STOP/remote PAUSE is applied from other external device. | <table border="1"><tr><td>0</td><td>0</td><td>0</td><td>1</td></tr><tr><td>30H</td><td>30H</td><td>30H</td><td>31H</td></tr></table> | 0 | 0 | 0 | 1 | 30H | 30H | 30H | 31H | <table border="1"><tr><td>01H</td><td>00H</td></tr></table> | 01H | 00H |
| 0 | 0 | 0 | 1 | | | | | | | | | | |
| 30H | 30H | 30H | 31H | | | | | | | | | | |
| 01H | 00H | | | | | | | | | | | | |
| Execute forcibly | Apply remote RUN/remote PAUSE while remote STOP/remote PAUSE is applied from other external device. | <table border="1"><tr><td>0</td><td>0</td><td>0</td><td>3</td></tr><tr><td>30H</td><td>30H</td><td>30H</td><td>33H</td></tr></table> | 0 | 0 | 0 | 3 | 30H | 30H | 30H | 33H | <table border="1"><tr><td>03H</td><td>00H</td></tr></table> | 03H | 00H |
| 0 | 0 | 0 | 3 | | | | | | | | | | |
| 30H | 30H | 30H | 33H | | | | | | | | | | |
| 03H | 00H | | | | | | | | | | | | |

Clear mode

Select the range of the device memory to be cleared by the initialization processing at remote RUN.

When the device initial values are set, they will be reflected after the process with the selected clear mode.

Setting method

■Data communication using ASCII code

Convert the numerical value to 2-digit ASCII code (hexadecimal), and send it from the upper digits.

■Data communication using binary code

Send a 1-byte numerical value.

| Process | | ASCII code | Binary code | | | | | |
|----------------------------------|---|--|-------------|---|-----|-----|---|-----|
| Do not clear | Do not clear device memory. | <table border="1"><tr><td>0</td><td>0</td></tr><tr><td>30H</td><td>30H</td></tr></table> | 0 | 0 | 30H | 30H | <table border="1"><tr><td>00H</td></tr></table> | 00H |
| 0 | 0 | | | | | | | |
| 30H | 30H | | | | | | | |
| 00H | | | | | | | | |
| Clear outside the range of latch | Clear the device memory outside the range of latch. | <table border="1"><tr><td>0</td><td>1</td></tr><tr><td>30H</td><td>31H</td></tr></table> | 0 | 1 | 30H | 31H | <table border="1"><tr><td>01H</td></tr></table> | 01H |
| 0 | 1 | | | | | | | |
| 30H | 31H | | | | | | | |
| 01H | | | | | | | | |
| All clear | Clear all device memory including the range of latch. | <table border="1"><tr><td>0</td><td>2</td></tr><tr><td>30H</td><td>32H</td></tr></table> | 0 | 2 | 30H | 32H | <table border="1"><tr><td>02H</td></tr></table> | 02H |
| 0 | 2 | | | | | | | |
| 30H | 32H | | | | | | | |
| 02H | | | | | | | | |

Model name and model code

The model name and model code of the access target module.

Model name

The character string of a model name is stored in 16-digit ASCII code.

If the read model name is less than 16 characters, a space (20H) is stored for the shortage of the characters.

The model name is stored in ASCII code while communicating in binary code.

Ex.

For Q02HCPU

ASCII code, binary code

| |
|---|
| Q 0 2 H C P U |
| 51H , 30H , 32H , 48H , 43H , 50H , 55H , 20H |

Model code

The model code of the module is stored.

■Data communication using ASCII code

Convert the numerical value to 4-digit ASCII code (hexadecimal) and send it from upper digits.

■Data communication using binary code

Send 2-byte numerical values from lower byte (L: bits 0 to 7).

Ex.

For Q02HCPU (41H)

| ASCII code | Binary code |
|---|-------------|
| 0 0 4 1 30H , 30H , 34H , 31H | 41H , 00H |

Model name and model code list

The following shows the list of model names and model codes.

■MELSEC iQ-R series

| Model name | Model code |
|--------------------|---------------------|
| RCPU ^{*1} | 0360H ^{*1} |
| R04CPU | 4800H |
| R08CPU | 4801H |
| R16CPU | 4802H |
| R32CPU | 4803H |
| R120CPU | 4804H |

*1 When accessing RCPU from the connected station other than MELSEC iQ-R series, the common model name 'RCPU' and model code '0360H' is stored.

■MELSEC-L series

| Model name | Model code |
|-----------------------|------------|
| L02SCPU, L02SCPU-P | 0543H |
| L02CPU, L02CPU-P | 0541H |
| L06CPU, L06CPU-P | 0544H |
| L26CPU, L26CPU-P | 0545H |
| L26CPU-BT, L26CPU-PBT | 0542H |
| LJ72GF15-T2 | 0641H |

■MELSEC-Q series

| Model name | Model code |
|---------------------------|------------|
| Q00JCPU | 0250H |
| Q00CPU | 0251H |
| Q01CPU | 0252H |
| Q02CPU, Q02HCPU, Q02PHCPU | 0041H |
| Q06HCPU, Q06PHCPU | 0042H |
| Q12HCPU, Q12PHCPU | 0043H |
| Q25HCPU, Q25PHCPU | 0044H |
| Q12PRHCPU | 004BH |
| Q25PRHCPU | 004CH |
| Q00UJCPU | 0260H |
| Q00UCPU | 0261H |
| Q01UCPU | 0262H |
| Q02UCPU | 0263H |
| Q03UDCPU, Q03UDECPU | 0268H |
| Q03UDVCPU | 0366H |
| Q04UDHCPU, Q04UDEHCPU | 0269H |
| Q04UDVCPU | 0367H |
| Q06UDHCPU, Q06UDEHCPU | 026AH |
| Q06UDVCPU | 0368H |
| Q10UDHCPU, Q10UDEHCPU | 0266H |
| Q13UDHCPU, Q13UDEHCPU | 026BH |
| Q13UDVCPU | 036AH |
| Q20UDHCPU, Q20UDEHCPU | 0267H |
| Q26UDHCPU, Q26UDEHCPU | 026CH |
| Q26UDVCPU | 036CH |
| Q50UDEHCPU | 026DH |
| Q100UDEHCPU | 026EH |
| QS001CPU | 0230H |

Remote password

Remote password length

Specify the number of characters of remote password.

- For MELSEC-Q/L series modules, the character string is fixed to 4 characters.
- For MELSEC iQ-R series modules, specify the number of character string of the remote password (6 to 32 characters).

■Data communication using ASCII code

Convert the numerical value to 4-digit ASCII code (hexadecimal) and send it from upper digits.

■Data communication using binary code

Send 2-byte^{*1} numerical values from lower byte (L: bits 0 to 7).

- *1 For C24, the additional code may be added. (参照 Page 33 Additional code (10H))

Ex.

For 4 characters, 32 characters

| Access target | ASCII code | Binary code | | | | | | | | | | |
|--|--|-------------|-----|---|---|-----|-----|-----|-----|---|-----|-----|
| MELSEC-Q/L series module (4 characters fixed) | <table border="1"><tr><td>0</td><td>0</td><td>0</td><td>4</td></tr><tr><td>30H</td><td>30H</td><td>30H</td><td>34H</td></tr></table> | 0 | 0 | 0 | 4 | 30H | 30H | 30H | 34H | <table border="1"><tr><td>04H</td><td>00H</td></tr></table> | 04H | 00H |
| 0 | 0 | 0 | 4 | | | | | | | | | |
| 30H | 30H | 30H | 34H | | | | | | | | | |
| 04H | 00H | | | | | | | | | | | |
| MELSEC iQ-R series module (32 characters) | <table border="1"><tr><td>0</td><td>0</td><td>2</td><td>0</td></tr><tr><td>30H</td><td>30H</td><td>32H</td><td>30H</td></tr></table> | 0 | 0 | 2 | 0 | 30H | 30H | 32H | 30H | <table border="1"><tr><td>20H</td><td>00H</td></tr></table> | 20H | 00H |
| 0 | 0 | 2 | 0 | | | | | | | | | |
| 30H | 30H | 32H | 30H | | | | | | | | | |
| 20H | 00H | | | | | | | | | | | |

Remote password

The remote password is set by Engineering tool.

Specify the remote password in ASCII code while communicating in binary code.

Ex.

When password is 'ABCDEF'

ASCII code, binary code

| | | | | | |
|-----|-----|-----|-----|-----|-----|
| A | B | C | D | E | F |
| 41H | 42H | 43H | 44H | 45H | 46H |

Loopback data

Loopback data length

Number of bytes of loopback data.

■Data communication using ASCII code

Convert the numerical value to 4-digit ASCII code (hexadecimal) and send it from upper digits.

■Data communication using binary code

Send 2-byte^{*1} numerical values from lower byte (L: bits 0 to 7).

*1 For C24, the additional code may be added. (参照 Page 33 Additional code (10H))

Ex.

For 5 bytes

| ASCII code | Binary code |
|----------------------------|-------------|
| 0 0 0 5 30H 30H 30H 35H | 05H 00H |

Loopback data

The following characters can be used.

- 0 to 9 (30H to 39H)
- A to F (41H to 46H)

Specify the loopback data in ASCII code while communicating in binary code.

Ex.

For 'ABCDEF'

| ASCII code, binary code |
|--|
| A B C D E F 41H 42H 43H 44H 45H 46H |

Communication error information

A data to specify the communication error information to be initialized.

The data is equivalent to the buffer memory '0H', '1H' of MELSEC-Q/L series C24.

Specify the value (16-bit integer) in bit unit. The communication error information and its corresponding bit are as follows.

| Item | b15 | b14 | b13 to b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---------------------------------------|---------|---------|-----------|-----|-----|-----|-----|-----|-----|-----|---------|
| Communication error information (CH1) | — | — | — | NEU | ACK | NAK | C/N | P/S | PRO | SIO | SD.WAIT |
| Communication error information (CH2) | CH1.ERR | CH2.ERR | | | | | | | | | |

The communication error information to be initialized can be specified by turning the corresponding bit ON (1).

When the command is executed, the following corresponding bit of the buffer memory turns OFF (0).

"LED lighting status, communication error status" (Buffer memory 513 (201H), 514 (202H))

For MELSEC iQ-R series C24, specify '0'.

Setting method

■Data communication using ASCII code

Convert the numerical value to 4-digit ASCII code (hexadecimal) and send it from upper digits.

■Data communication using binary code

Send 2-byte *1 numerical values from lower byte (L: bits 0 to 7).

*1 For C24, the additional code may be added. ( Page 33 Additional code (10H))

Ex.

When initializing C/N, P/S, PRO, and SIO (bit 1 to 4) with MELSEC-Q/L series C24.

| Item | ASCII code | Binary code | | | | | | | | | | |
|---------------------------------------|--|-------------|-----|---|---|-----|-----|-----|-----|---|----|----|
| Communication error information (CH1) | <table border="1"><tr><td>0</td><td>0</td><td>1</td><td>E</td></tr><tr><td>30H</td><td>30H</td><td>31H</td><td>45H</td></tr></table> | 0 | 0 | 1 | E | 30H | 30H | 31H | 45H | <table border="1"><tr><td>1E</td><td>00</td></tr></table> | 1E | 00 |
| 0 | 0 | 1 | E | | | | | | | | | |
| 30H | 30H | 31H | 45H | | | | | | | | | |
| 1E | 00 | | | | | | | | | | | |
| Communication error information (CH2) | <table border="1"><tr><td>0</td><td>0</td><td>0</td><td>0</td></tr><tr><td>30H</td><td>30H</td><td>30H</td><td>30H</td></tr></table> | 0 | 0 | 0 | 0 | 30H | 30H | 30H | 30H | <table border="1"><tr><td>00</td><td>00</td></tr></table> | 00 | 00 |
| 0 | 0 | 0 | 0 | | | | | | | | | |
| 30H | 30H | 30H | 30H | | | | | | | | | |
| 00 | 00 | | | | | | | | | | | |

Ex.

For MELSEC iQ-R series

| Item | ASCII code | Binary code | | | | | | | | | | |
|---------------------------------------|--|-------------|-----|---|---|-----|-----|-----|-----|---|----|----|
| Communication error information (CH1) | <table border="1"><tr><td>0</td><td>0</td><td>0</td><td>0</td></tr><tr><td>30H</td><td>30H</td><td>30H</td><td>30H</td></tr></table> | 0 | 0 | 0 | 0 | 30H | 30H | 30H | 30H | <table border="1"><tr><td>00</td><td>00</td></tr></table> | 00 | 00 |
| 0 | 0 | 0 | 0 | | | | | | | | | |
| 30H | 30H | 30H | 30H | | | | | | | | | |
| 00 | 00 | | | | | | | | | | | |
| Communication error information (CH2) | <table border="1"><tr><td>0</td><td>0</td><td>0</td><td>0</td></tr><tr><td>30H</td><td>30H</td><td>30H</td><td>30H</td></tr></table> | 0 | 0 | 0 | 0 | 30H | 30H | 30H | 30H | <table border="1"><tr><td>00</td><td>00</td></tr></table> | 00 | 00 |
| 0 | 0 | 0 | 0 | | | | | | | | | |
| 30H | 30H | 30H | 30H | | | | | | | | | |
| 00 | 00 | | | | | | | | | | | |

11.2 Remote Operation

Change the operation status of CPU module.

For the remote operation function, refer to the manual of each CPU module.

Point

- When powering OFF to ON or resetting the access target CPU after applying remote RUN/STOP/PAUSE, the information of the remote operation will be cancelled. After powering OFF to ON or resetting CPU, the CPU operates with the status of the switch on the CPU module.
- When the system protection of the access target module is enabled, the remote operation cannot be performed and an error response will be returned. Disable the system protection of the CPU module.
- For E71, the communication using UDP/IP is recommended. For the communication using TCP/IP, re-establishment of the connection is required because the connection is disconnected at resetting CPU module.
- One communication of command provides remote operation for 1 station.

Remote RUN (command: 1001)

Perform remote RUN for the access target module.

Restriction

The command can be executed when the switch of the access target module is RUN. In the STOP status, the command is completed normally, however, the access target CPU will not be in RUN status.

Message format

The following shows the message format of the request data and response data of the command.

■Request data

| Command | Subcommand | Mode | Clear mode | Fixed values |
|---------|------------|------|------------|--------------|
| | | | | |

■Response data

There is no response data for this command.

Data specified by request data

■Command

| ASCII code | Binary code (for C24)*1 | Binary code (for E71) | | | | | | | | | | | | | | | | |
|---|-------------------------|-----------------------|-----|---|-----|-----|-----|-----|---|-----|-----|-----|-----|-----|-----|--|-----|-----|
| <table border="1"> <tr> <td>1</td><td>0</td><td>0</td><td>1</td></tr> <tr> <td>31H</td><td>30H</td><td>30H</td><td>31H</td></tr> </table> | 1 | 0 | 0 | 1 | 31H | 30H | 30H | 31H | <table border="1"> <tr> <td>01H</td><td>DLE</td><td>10H</td></tr> <tr> <td>01H</td><td>10H</td><td>10H</td></tr> </table> | 01H | DLE | 10H | 01H | 10H | 10H | <table border="1"> <tr> <td>01H</td><td>10H</td></tr> </table> | 01H | 10H |
| 1 | 0 | 0 | 1 | | | | | | | | | | | | | | | |
| 31H | 30H | 30H | 31H | | | | | | | | | | | | | | | |
| 01H | DLE | 10H | | | | | | | | | | | | | | | | |
| 01H | 10H | 10H | | | | | | | | | | | | | | | | |
| 01H | 10H | | | | | | | | | | | | | | | | | |

*1 For C24, the additional code is added. (☞ Page 33 Additional code (10H))

■Subcommand

| ASCII code | Binary code | | | | | | | | | | |
|---|-------------|-----|-----|---|-----|-----|-----|-----|--|-----|-----|
| <table border="1"> <tr> <td>0</td><td>0</td><td>0</td><td>0</td></tr> <tr> <td>30H</td><td>30H</td><td>30H</td><td>30H</td></tr> </table> | 0 | 0 | 0 | 0 | 30H | 30H | 30H | 30H | <table border="1"> <tr> <td>00H</td><td>00H</td></tr> </table> | 00H | 00H |
| 0 | 0 | 0 | 0 | | | | | | | | |
| 30H | 30H | 30H | 30H | | | | | | | | |
| 00H | 00H | | | | | | | | | | |

■Mode

Select the operation when the request target module is already in remote operation by other device. (☞ Page 161 Mode)

- 0001H: Do not execute forcibly
- 0003H: Execute forcibly

■Clear mode

Select the range of the device memory to be cleared by the initialization processing at remote RUN. (☞ Page 161 Clear mode)

- 00H: Do not clear
- 01H: Clear only outside the latch range
- 02H: All clear

■Fixed value

Fixed to '0'.

| ASCII code | Binary code | | | | | |
|---|-------------|---|-----|-----|--|-----|
| <table border="1"> <tr> <td>0</td><td>0</td></tr> <tr> <td>30H</td><td>30H</td></tr> </table> | 0 | 0 | 30H | 30H | <table border="1"> <tr> <td>00H</td></tr> </table> | 00H |
| 0 | 0 | | | | | |
| 30H | 30H | | | | | |
| 00H | | | | | | |

Communication example

Remote RUN is performed when the mode is set to "Do not execute forcibly", and the clear mode is set to "All clear".

■Data communication using ASCII code

(Request data)

| Mode | Clear mode | | | | | | | | | | | | | | | | |
|---|------------|-----|-----|---|-----|-----|-----|-----|---|---|---|---|---|-----|-----|-----|-----|
| <table border="1"> <tr> <td>1</td><td>0</td><td>0</td><td>1</td></tr> <tr> <td>31H</td><td>30H</td><td>30H</td><td>31H</td></tr> </table> | 1 | 0 | 0 | 1 | 31H | 30H | 30H | 31H | <table border="1"> <tr> <td>0</td><td>0</td><td>0</td><td>0</td></tr> <tr> <td>30H</td><td>30H</td><td>30H</td><td>30H</td></tr> </table> | 0 | 0 | 0 | 0 | 30H | 30H | 30H | 30H |
| 1 | 0 | 0 | 1 | | | | | | | | | | | | | | |
| 31H | 30H | 30H | 31H | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 0 | | | | | | | | | | | | | | |
| 30H | 30H | 30H | 30H | | | | | | | | | | | | | | |

| | | | | | | | |
|-----|-----|-----|-----|-----|-----|-----|-----|
| 0 | 0 | 0 | 1 | 0 | 2 | 0 | 0 |
| 31H | 30H | 30H | 31H | 30H | 30H | 30H | 30H |

■Data communication using binary code

(Request data)

| Mode | Clear mode | | | | |
|--|------------|-----|--|-----|-----|
| <table border="1"> <tr> <td>01H</td><td>10H</td></tr> </table> | 01H | 10H | <table border="1"> <tr> <td>00H</td><td>00H</td></tr> </table> | 00H | 00H |
| 01H | 10H | | | | |
| 00H | 00H | | | | |

| | | | | | | | |
|-----|-----|-----|-----|-----|-----|-----|-----|
| 01H | 10H | 00H | 00H | 01H | 00H | 02H | 00H |
| 01H | 10H | 00H | 00H | 01H | 00H | 02H | 00H |

Remote STOP (command: 1002)

Perform remote STOP for the access target module.

Message format

The following shows the message format of the request data and response data of the command.

■Request data

| | | |
|---------|------------|--------------|
| Command | Subcommand | Fixed values |
|---------|------------|--------------|

■Response data

There is no response data for this command.

Data specified by request data

■Command

| ASCII code | Binary code (for C24)*1 | Binary code (for E71) | | | | | | | | | | | | | | | | | | |
|---|-------------------------|-----------------------|-----|---|-----|-----|-----|-----|---|-----|-----|-----|--|-----|--|---|-----|-----|--|--|
| <table border="1"> <tr> <td>1</td><td>0</td><td>0</td><td>2</td> </tr> <tr> <td>31H</td><td>30H</td><td>30H</td><td>32H</td> </tr> </table> | 1 | 0 | 0 | 2 | 31H | 30H | 30H | 32H | <table border="1"> <tr> <td>02H</td><td>DLE</td><td>10H</td> </tr> <tr> <td></td><td>10H</td><td></td> </tr> </table> | 02H | DLE | 10H | | 10H | | <table border="1"> <tr> <td>02H</td><td>10H</td> </tr> <tr> <td></td><td></td> </tr> </table> | 02H | 10H | | |
| 1 | 0 | 0 | 2 | | | | | | | | | | | | | | | | | |
| 31H | 30H | 30H | 32H | | | | | | | | | | | | | | | | | |
| 02H | DLE | 10H | | | | | | | | | | | | | | | | | | |
| | 10H | | | | | | | | | | | | | | | | | | | |
| 02H | 10H | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | |

*1 For C24, the additional code is added. (☞ Page 33 Additional code (10H))

■Subcommand

| ASCII code | Binary code | | | | | | | | | | | | |
|---|-------------|-----|-----|---|-----|-----|-----|-----|---|-----|-----|--|--|
| <table border="1"> <tr> <td>0</td><td>0</td><td>0</td><td>0</td> </tr> <tr> <td>30H</td><td>30H</td><td>30H</td><td>30H</td> </tr> </table> | 0 | 0 | 0 | 0 | 30H | 30H | 30H | 30H | <table border="1"> <tr> <td>00H</td><td>00H</td> </tr> <tr> <td></td><td></td> </tr> </table> | 00H | 00H | | |
| 0 | 0 | 0 | 0 | | | | | | | | | | |
| 30H | 30H | 30H | 30H | | | | | | | | | | |
| 00H | 00H | | | | | | | | | | | | |
| | | | | | | | | | | | | | |

■Fixed value

The value is '0001H'.

Communication example

Perform remote STOP.

■Data communication using ASCII code

(Request data)

| | | | | | | | | | | | |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| 1 | 0 | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 31H | 30H | 30H | 32H | 30H | 31H |

■Data communication using binary code

(Request data)

| | | | | | |
|-----|-----|-----|-----|-----|-----|
| 02H | 10H | 00H | 00H | 01H | 00H |
|-----|-----|-----|-----|-----|-----|

Remote PAUSE (command: 1003)

Perform remote PAUSE for the access target module.



The command can be executed when the switch of the access target module is RUN. In the STOP status, the command is completed normally, however, the access target CPU will not be in PAUSE status.

Message format

The following shows the message format of the request data and response data of the command.

■Request data

| | | |
|---------|------------|------|
| Command | Subcommand | Mode |
|---------|------------|------|

■Response data

There is no response data for this command.

Data specified by request data

■Command

| ASCII code | Binary code (for C24) ^{*1} | Binary code (for E71) | | | | | | | | | | | | | | | | |
|--|-------------------------------------|-----------------------|-----|---|-----|-----|-----|-----|--|--|-----|--|-----|-----|-----|---|-----|-----|
| <table border="1"><tr><td>1</td><td>0</td><td>0</td><td>3</td></tr><tr><td>31H</td><td>30H</td><td>30H</td><td>33H</td></tr></table> | 1 | 0 | 0 | 3 | 31H | 30H | 30H | 33H | <table border="1"><tr><td></td><td>DLE</td><td></td></tr><tr><td>03H</td><td>10H</td><td>10H</td></tr></table> | | DLE | | 03H | 10H | 10H | <table border="1"><tr><td>03H</td><td>10H</td></tr></table> | 03H | 10H |
| 1 | 0 | 0 | 3 | | | | | | | | | | | | | | | |
| 31H | 30H | 30H | 33H | | | | | | | | | | | | | | | |
| | DLE | | | | | | | | | | | | | | | | | |
| 03H | 10H | 10H | | | | | | | | | | | | | | | | |
| 03H | 10H | | | | | | | | | | | | | | | | | |

*1 For C24, the additional code is added. (参照 Page 33 Additional code (10H))

■Subcommand

| ASCII code | Binary code | | | | | | | | | | |
|--|-------------|-----|-----|---|-----|-----|-----|-----|---|-----|-----|
| <table border="1"><tr><td>0</td><td>0</td><td>0</td><td>0</td></tr><tr><td>30H</td><td>30H</td><td>30H</td><td>30H</td></tr></table> | 0 | 0 | 0 | 0 | 30H | 30H | 30H | 30H | <table border="1"><tr><td>00H</td><td>00H</td></tr></table> | 00H | 00H |
| 0 | 0 | 0 | 0 | | | | | | | | |
| 30H | 30H | 30H | 30H | | | | | | | | |
| 00H | 00H | | | | | | | | | | |

■Mode

Select the operation when the request target module is already in remote operation by other device. (参照 Page 161 Mode)

- 0001H: Do not execute forcibly
- 0003H: Execute forcibly

Communication example

Remote PAUSE is performed when the mode is set to "Do not execute forcibly".

■Data communication using ASCII code

(Request data)

| | | | | | | | | | | | |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|---|
| 1 | 0 | 0 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 31H | 30H | 30H | 33H | 30H | 30H | 30H | 30H | 30H | 30H | 31H | |

■Data communication using binary code

(Request data)

| | | | | | |
|-----|-----|-----|-----|-----|-----|
| | | | | | |
| 03H | 10H | 00H | 00H | 01H | 00H |

Remote latch clear (command: 1005)

Perform remote latch clear for the access target module.



- Execute the command after changing the status of the access target module to STOP.
- The remote latch clear cannot be performed when the access target CPU is in remote STOP or remote PAUSE by other devices. The command will be terminated abnormally. Clear the remote STOP or remote PAUSE before executing the command.

Message format

The following shows the message format of the request data and response data of the command.

■Request data

| Command | Subcommand | Fixed values |
|----------|------------|----------------------|
| 1 31H | 0 30H | 0 30H 5 35H |

■Response data

There is no response data for this command.

Data specified by request data

■Command

| ASCII code | Binary code (for C24)*1 | Binary code (for E71) |
|----------------------------|-------------------------|-----------------------|
| 1 0 0 5 31H 30H 30H 35H | 05H DLE 10H 10H | 05H 10H |

*1 For C24, the additional code is added. (参照 Page 33 Additional code (10H))

■Subcommand

| ASCII code | Binary code |
|----------------------------|-------------|
| 0 0 0 0 30H 30H 30H 30H | 00H 00H |

■Fixed value

The value is '0001H'.

Communication example

Perform remote latch clear.

■Data communication using ASCII code

(Request data)

| | | | | | | | | | | | |
|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|
| 1 31H | 0 30H | 0 30H | 5 35H | 0 30H | 1 31H |
|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|

■Data communication using binary code

(Request data)

| | | |
|------------|------------|------------|
| 05H 00H | 10H 00H | 01H 00H |
|------------|------------|------------|

Remote RESET (command: 1006)

Perform remote RESET for the access target module.



- Execute the command after changing the status of the access target module to STOP. If the CPU module is stopped due to the error, the command can be executed even when the switch of the CPU module is in the position of RUN,
- If a remote RESET operation enable/disable setting exists in the access target parameter, set it to enable.
- Remote RESET may not be performed due to the hardware error of the access target device.
- When performing remote RESET, the response message may not be returned because the access target CPU is reset.

Message format

The following shows the message format of the request data and response data of the command.

■Request data

| Command | Subcommand | Fixed values |
|----------------------------|------------|--------------|
| 1 0 0 6 31H 30H 30H 36H | | |

■Response data

There is no response data for this command.

Data specified by request data

■Command

| ASCII code | Binary code (for C24) ^{*1} | Binary code (for E71) |
|----------------------------|-------------------------------------|-----------------------|
| 1 0 0 6 31H 30H 30H 36H | 06H DLE 10H 10H | 06H 10H |

*1 For C24, the additional code is added. (参照 Page 33 Additional code (10H))

■Subcommand

| ASCII code | Binary code |
|----------------------------|-------------|
| 0 0 0 0 30H 30H 30H 30H | 00H 00H |

■Fixed value

The value is '0001H'.

Communication example

Perform remote RESET.

■Data communication using ASCII code

(Request data)

| | | |
|----------------------------|----------------------------|----------------------------|
| 1 0 0 6 31H 30H 30H 36H | 0 0 0 0 30H 30H 30H 30H | 0 0 0 1 30H 30H 30H 31H |
|----------------------------|----------------------------|----------------------------|

■Data communication using binary code

(Request data)

| | | |
|---------|---------|---------|
| 06H 10H | 00H 00H | 01H 00H |
|---------|---------|---------|

Read CPU model name (command: 0101)

Read model name and model code from the access target module.

Message format

The following shows the message format of the request data and response data of the command.

■ Request data

| Command | Subcommand |
|---------|------------|
|---------|------------|

11

■ Response data

Model name and model code are stored. (☞ Page 162 Model name and model code)



- Discriminate the model name of the CPU with the model code.

Data specified by request data

■ Command

| ASCII code | Binary code | | | | | | | | | | |
|--|-------------|-----|-----|---|-----|-----|-----|-----|---|-----|-----|
| <table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td>0</td><td>1</td><td>0</td><td>1</td></tr> <tr> <td>30H</td><td>31H</td><td>30H</td><td>31H</td></tr> </table> | 0 | 1 | 0 | 1 | 30H | 31H | 30H | 31H | <table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td>01H</td><td>01H</td></tr> </table> | 01H | 01H |
| 0 | 1 | 0 | 1 | | | | | | | | |
| 30H | 31H | 30H | 31H | | | | | | | | |
| 01H | 01H | | | | | | | | | | |

■ Subcommand

| ASCII code | Binary code | | | | | | | | | | |
|--|-------------|-----|-----|---|-----|-----|-----|-----|---|-----|-----|
| <table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td>0</td><td>0</td><td>0</td><td>0</td></tr> <tr> <td>30H</td><td>30H</td><td>30H</td><td>30H</td></tr> </table> | 0 | 0 | 0 | 0 | 30H | 30H | 30H | 30H | <table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td>00H</td><td>00H</td></tr> </table> | 00H | 00H |
| 0 | 0 | 0 | 0 | | | | | | | | |
| 30H | 30H | 30H | 30H | | | | | | | | |
| 00H | 00H | | | | | | | | | | |

Communication example

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

■ Data communication using ASCII code

(Request data)

| | | | | | | | |
|--------|--------|--------|--------|--------|--------|--------|--------|
| 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 |
| 30_H | 31_H | 30_H | 31_H | 30_H | 30_H | 30_H | 30_H |

(Response data)

| | |
|---|------------------|
| Q 0 2 U C P U | 0 2 6 3 |
| 51 _H . 30 _H . 32 _H . 55 _H . 43 _H . 50 _H . 55 _H . 20 _H . 30 _H . 32 _H . 36 _H . 33 _H | |

■ Data communication using binary code

Data.com
(Request data)

01_H + 01_H 00_H + 00_H

(Response data)

11.3 Remote Password

This section explains the commands that unlock or lock the remote password.

For details on the remote password, refer to the manuals of access target CPU or CPU module.

Restriction

The command can only be used for C24 (including multidrop connection station) and E71 connected to the external device. It cannot be used via network.

For the modem connection, access route must be set as the same route as that of the connected station (host station).

( Page 43 ACCESS ROUTE SETTINGS)

Execution procedure

The communication with the module in which the remote password is set, follow the procedure shown below.

1. Line connection by modem (C24)/open processing for connection (E71)
2. Access permission (unlock processing) ( Page 175 Unlock (command: 1630))
3. Access processing
Perform data communication by various commands of MC protocol.
4. Access prohibition (lock processing) ( Page 177 Lock (command: 1631))
For C24 or TCP/IP communication of E71, the lock processing is performed automatically at modem disconnection/close processing.
5. Line disconnection/close processing of connection

Point

All the commands received when the remote password is locked will be an error response. Perform data communication after the unlock processing.

Unlock (command: 1630)

Specify the remote password to unlock the module. (The module can communicate.)

Restriction

If the incorrect password is entered several times, the password is locked out and cannot be cleared for a while.

Point

When the command is sent to the unlocked module, the unlock status is not changed. (The password verification is not performed.)

Message format

The following shows the message format of the request data and response data of the command.

■Request data

| Command | Subcommand | Remote password length | Remote password |
|---------|------------|------------------------|-----------------|
| | | | |

■Response data

There is no response data for this command.

Data specified by request data

■Command

| ASCII code | Binary code |
|----------------------------|-------------|
| 1 6 3 0 31H 36H 33H 30H | 30H 16H |

■Subcommand

| ASCII code | Binary code |
|----------------------------|-------------|
| 0 0 0 0 30H 30H 30H 30H | 00H 00H |

■Remote password length

Specify the number of characters of remote password. ([Page 164 Remote password length](#))

■Remote password

Specify the set remote password. ([Page 164 Remote password](#))

Communication example (for MELSEC-Q series)

Unlock the MELSEC-Q/L series module in which the following remote password has been set.

- Remote password: '1234' (4 characters)

■Data communication using ASCII code

(Request data)

| Subcommand | Remote password length | Remote password |
|-------------------------------|-------------------------------|-------------------------------|
| 1 6 3 0 31H, 36H, 33H, 30H | 0 0 0 0 30H, 30H, 30H, 30H | 0 0 0 4 30H, 30H, 30H, 34H |

1 2 3 4
31H, 32H, 33H, 34H

■Data communication using binary code

(Request data)

| Subcommand | Remote password length | Remote password |
|------------|------------------------|-----------------|
| 30H, 16H | 00H, 00H | 04H, 00H |

1 2 3 4
31H, 32H, 33H, 34H

Communication example (for the files of MELSEC iQ-R series)

Unlock the MELSEC iQ-R series module in which the following remote password has been set.

- Remote password: 'abcdefghijklmnoprstuvwxyz' (26 characters)

■Data communication using ASCII code

(Request data)

| Subcommand | Remote password length | Remote password |
|-------------------------------|-------------------------------|-------------------------------|
| 1 6 3 0 31H, 36H, 33H, 30H | 0 0 0 0 30H, 30H, 30H, 30H | 0 0 1 A 30H, 30H, 31H, 41H |

Remote password

| |
|--|
| a b c d e f g h i j k l m n o p q r s t u v w x y z |
| 61H, 62H, 63H, 64H, 65H, 66H, 67H, 68H, 69H, 6AH, 6BH, 6CH, 6DH, 6EH, 6FH, 70H, 71H, 72H, 73H, 74H, 75H, 76H, 77H, 78H, 79H, 7AH |

■Data communication using binary code

(Request data)

| Subcommand | Remote password length | Remote password |
|------------|------------------------|-----------------|
| 30H, 16H | 00H, 00H | 1AH, 00H |

Remote password

| |
|--|
| a b c d e f g h i j k l m n o p q r s t u v w x y z |
| 61H, 62H, 63H, 64H, 65H, 66H, 67H, 68H, 69H, 6AH, 6BH, 6CH, 6DH, 6EH, 6FH, 70H, 71H, 72H, 73H, 74H, 75H, 76H, 77H, 78H, 79H, 7AH |

Lock (command: 1631)

Specify the remote password to lock the module. (The module cannot communicate.)



When the command is sent to the locked module, the lock status is not changed. (The password verification is not performed.)

Message format

The following shows the message format of the request data and response data of the command.

11

■Request data

| | | | |
|---------|------------|------------------------|-----------------|
| Command | Subcommand | Remote password length | Remote password |
|---------|------------|------------------------|-----------------|

■Response data

There is no response data for this command.

Data specified by request data

■Command

| ASCII code | Binary code |
|----------------------------|-------------|
| 1 6 3 1 31H 36H 33H 31H | 31H 16H |

■Subcommand

| ASCII code | Binary code |
|----------------------------|-------------|
| 0 0 0 0 30H 30H 30H 30H | 00H 00H |

■Remote password length

Specify the number of characters of remote password. ([Page 164 Remote password length](#))

■Remote password

Specify the set remote password. ([Page 164 Remote password](#))

Communication example (for MELSEC-Q series)

Lock the MELSEC-Q/L series module in which the following remote password has been set.

- Remote password: '1234' (4 characters)

■Data communication using ASCII code

(Request data)

| Subcommand | Remote password length | Remote password |
|------------|------------------------|--|
| 1 6 3 1 | 0 0 0 0 | 0 0 0 4 1 2 3 4 31H, 36H, 33H, 31H 30H, 30H, 30H 30H, 30H, 30H, 34H 31H, 32H, 33H, 34H |

■Data communication using binary code

(Request data)

| Subcommand | Remote password length | Remote password |
|------------|------------------------|---|
| 31H, 16H | 00H, 00H | 04H, 00H 1 2 3 4 31H, 32H, 33H, 34H |

Communication example (for the files of MELSEC iQ-R series)

Lock the MELSEC iQ-R series module in which the following password has been set.

- Remote password: 'abcdefghijklmnoprstuvwxyz' (26 characters)

■Data communication using ASCII code

(Request data)

| Subcommand | Remote password length | Remote password |
|--|------------------------|---|
| 1 6 3 1 | 0 0 0 0 | 0 0 1 A 31H, 36H, 33H, 31H 30H, 30H, 30H, 30H 30H, 30H, 31H, 41H |
| ----- | | |
| ----- a b c d e f g h i j k l m n o p q r s t u v w x y z 61H, 62H, 63H, 64H, 65H, 66H, 67H, 68H, 69H, 6AH, 6BH, 6CH, 6DH, 6EH, 6FH, 70H, 71H, 72H, 73H, 74H, 75H, 76H, 77H, 78H, 79H, 7AH | | |

■Data communication using binary code

(Request data)

| Subcommand | Remote password length | Remote password |
|--|------------------------|-------------------|
| 31H, 16H | 00H, 00H | 1AH, 00H ----- |
| ----- a b c d e f g h i j k l m n o p q r s t u v w x y z 61H, 62H, 63H, 64H, 65H, 66H, 67H, 68H, 69H, 6AH, 6BH, 6CH, 6DH, 6EH, 6FH, 70H, 71H, 72H, 73H, 74H, 75H, 76H, 77H, 78H, 79H, 7AH | | |

11.4 Loopback Test

This chapter explains the commands for testing the connection and data communication between an external device and connected station.

Restriction

The command can only be used for C24 (including multidrop connection station) and E71 connected to the external device. It cannot be used via network.

Loopback test (command: 0619)

Perform the test to check whether the communication between the external device and connected station is normal. By performing the loopback test, the connection with external devices and operation of data communication can be checked.

Message format

The following shows the message format of the request data and response data of the command.

■Request data

| | | | |
|---------|------------|-------------------------|---------------|
| Command | Subcommand | Number of loopback data | Loopback data |
|---------|------------|-------------------------|---------------|

■Response data

The same data as "Number of loopback data" and "Loopback data" specified for request message is stored. (☞ Page 165 Loopback data)

| | |
|-------------------------|---------------|
| Number of loopback data | Loopback data |
|-------------------------|---------------|

Data specified by request data

■Command

| ASCII code | Binary code |
|----------------------------|-------------|
| 0 6 1 9 30H 36H 31H 39H | 19H 06H |

■Subcommand

| ASCII code | Binary code |
|----------------------------|-------------|
| 0 0 0 0 30H 30H 30H 30H | 00H 00H |

■Number of loopback data, loopback data

Specify the data to be transmitted by loopback test. (☞ Page 165 Loopback data)

The data can be specified within the range (numerals 0 to 9 and characters A to F) of 1 to 960 bytes.

Communication example

Perform the loopback test with the following loopback data.

- Loopback data: "ABCDE" (5 characters)

■Data communication using ASCII code

(Request data)

| Subcommand | Number of loopback data | Loopback data |
|----------------------------|----------------------------|----------------------------|
| 0 6 1 9 30H 36H 31H 39H | 0 0 0 0 30H 30H 30H 30H | 0 0 0 5 30H 30H 30H 35H |

(Response data)

| Number of loopback data | Loopback data |
|----------------------------|----------------------------------|
| 0 0 0 5 30H 30H 30H 35H | A B C D E 41H 42H 43H 44H 45H |

■Data communication using binary code

(Request data)

| Subcommand | Number of loopback data | Loopback data |
|----------------------|-------------------------------------|--------------------------------------|
| 19H, 06H 00H, 00H | 05H, 00H 41H, 42H, 43H, 44H, 45H | A B C D E 41H, 42H, 43H, 44H, 45H |

(Response data)

| Number of loopback data | Loopback data |
|-------------------------------------|--------------------------------------|
| 05H, 00H 41H, 42H, 43H, 44H, 45H | A B C D E 41H, 42H, 43H, 44H, 45H |

11.5 Clear Error Information

This section explains the command to initialize LED display and error information of buffer memory, and recover the supported device.

For details of the related LEDs, input/output signals, and buffer memory, refer to the manual of the access target module.

Turn indicator LED OFF, initialize error code (command: 1617)

Turn OFF the indicator LED of the serial communication module, and initialize the communication error information and error codes.

Restriction

The command can only be used for C24 (including multidrop connection station) connected to the external device. It cannot be used via network.

Message format

The following shows the message format of the request data and response data of the command.

■Request data

| | | | |
|---------|------------|---------------------------------------|---------------------------------------|
| Command | Subcommand | Communication error information (CH1) | Communication error information (CH2) |
|---------|------------|---------------------------------------|---------------------------------------|

■Response data

There is no response data for this command.

Data specified by request data

■Command

| ASCII code | Binary code |
|----------------------------|-------------|
| 1 6 1 7 31H 36H 31H 37H | 17H 16H |

■Subcommand

| Target | | ASCII code | Binary code | | | | | | | | | | |
|--------------------|--|--|-------------|---|---|-----|-----|-----|-----|--|--|-----|-----|
| MELSEC-Q/L series | CH1 side | <table border="1" style="margin-left: auto; margin-right: auto;"> <tr><td>0</td><td>0</td><td>0</td><td>5</td></tr> <tr><td>30H</td><td>30H</td><td>30H</td><td>35H</td></tr> </table> | 0 | 0 | 0 | 5 | 30H | 30H | 30H | 35H | <table border="1" style="margin-left: auto; margin-right: auto;"> <tr><td>05H</td><td>00H</td></tr> </table> | 05H | 00H |
| 0 | 0 | 0 | 5 | | | | | | | | | | |
| 30H | 30H | 30H | 35H | | | | | | | | | | |
| 05H | 00H | | | | | | | | | | | | |
| CH2 side | <table border="1" style="margin-left: auto; margin-right: auto;"> <tr><td>0</td><td>0</td><td>0</td><td>A</td></tr> <tr><td>30H</td><td>30H</td><td>30H</td><td>41H</td></tr> </table> | 0 | 0 | 0 | A | 30H | 30H | 30H | 41H | <table border="1" style="margin-left: auto; margin-right: auto;"> <tr><td>0AH</td><td>00H</td></tr> </table> | 0AH | 00H | |
| 0 | 0 | 0 | A | | | | | | | | | | |
| 30H | 30H | 30H | 41H | | | | | | | | | | |
| 0AH | 00H | | | | | | | | | | | | |
| CH1 side, CH2 side | <table border="1" style="margin-left: auto; margin-right: auto;"> <tr><td>0</td><td>0</td><td>0</td><td>F</td></tr> <tr><td>30H</td><td>30H</td><td>30H</td><td>46H</td></tr> </table> | 0 | 0 | 0 | F | 30H | 30H | 30H | 46H | <table border="1" style="margin-left: auto; margin-right: auto;"> <tr><td>0FH</td><td>00H</td></tr> </table> | 0FH | 00H | |
| 0 | 0 | 0 | F | | | | | | | | | | |
| 30H | 30H | 30H | 46H | | | | | | | | | | |
| 0FH | 00H | | | | | | | | | | | | |
| MELSEC iQ-R series | CH1 side, CH2 side | <table border="1" style="margin-left: auto; margin-right: auto;"> <tr><td>0</td><td>0</td><td>0</td><td>1</td></tr> <tr><td>30H</td><td>30H</td><td>30H</td><td>31H</td></tr> </table> | 0 | 0 | 0 | 1 | 30H | 30H | 30H | 31H | <table border="1" style="margin-left: auto; margin-right: auto;"> <tr><td>01H</td><td>00H</td></tr> </table> | 01H | 00H |
| 0 | 0 | 0 | 1 | | | | | | | | | | |
| 30H | 30H | 30H | 31H | | | | | | | | | | |
| 01H | 00H | | | | | | | | | | | | |



For C24 of MELSEC-Q/L series, 0 to 3 bits of subcommands are equivalent to the following functions of C24.

The settings in the table above is recommended, even though the initialization can be performed with the values (0001H to 000FH) which combined ON/OFF arbitrarily.

- Bit 0: CH1 Error initialization request (YE) ON
- Bit 1: CH2 Error initialization request (YF) ON
- Bit 2: LED for CH1 OFF, communication error information initialization request (buffer memory address: 0H) ON
- Bit 3: LED for CH2 OFF, communication error information initialization request (buffer memory address: 1H) ON

■Communication error information

Specify the items in "LED lighting status, communication error status" to be initialized. (Page 166 Communication error information)

For MELSEC iQ-R series C24, specify '0'.

Communication example

Perform the following operations for CH1 interface of QJ71C24N-R2.

- ERR LED: OFF
- Input signal XE "Error occurrence": OFF
- Buffer memory 513 (201H) "LED lighting status, communication error status": Initialized (all items are OFF)
- Error code of buffer memory: Initialized (clear)

■Data communication using ASCII code

(Request data)

| Subcommand | Communication error information (CH1) | Communication error information (CH2) |
|------------|--|---|
| 1 6 1 7 | 0 0 0 5 30H, 30H, 30H, 35H | 0 0 F F 30H, 30H, 46H, 46H 30H, 30H, 30H, 30H |

■Data communication using binary code

(Request data)

| Subcommand | Communication error information (CH1) | Communication error information (CH2) |
|------------|---------------------------------------|---------------------------------------|
| 17H, 16H | 05H, 00H | FFH, 00H |

Turn COM.ERR. LED OFF (command: 1617)

Turn the COM.ERR.LED of Ethernet interface module OFF.



The commands can be used for E71 connected to an external device and cannot be used via network.

11

Message format

The following shows the message format of the request data and response data of the command.

■Request data

| | |
|---------|------------|
| Command | Subcommand |
|---------|------------|

■Response data

There is no response data for this command.

Data specified by request data

■Command

| ASCII code | Binary code |
|----------------------------|-------------|
| 1 6 1 7 31H 36H 31H 37H | 17H 16H |

■Subcommand

| ASCII code | Binary code |
|----------------------------|-------------|
| 0 0 0 0 30H 30H 30H 30H | 00H 00H |

Communication example

Turn COM.ERR.LED OFF.

■Data communication using ASCII code

(Request data)

Subcommand

| | |
|----------------------------|----------------------------|
| 1 6 1 7 31H 36H 33H 31H | 0 0 0 0 30H 30H 30H 30H |
|----------------------------|----------------------------|

■Data communication using binary code

(Request data)

Subcommand

| | |
|---------|---------|
| | |
| 17H 16H | 00H 00H |

12 FILE CONTROL

This chapter explains the commands that operate files in the supported devices the CPU module.

Use this function in the following situations:

- To check the parameters and programs stored in the CPU module
- To change the parameters and programs in the CPU module according to the control content

For file name, extension, and storage location of files that can be handled by MC protocol, refer to the manual of the module to be accessed.

12.1 Execution Procedure

The following shows the file control procedures.

Procedure to read information from all files in directory (folder)

1. Read the file information from the head of the file.

Specify '1' for "Head file No." and '36' (upper limit) for "Number of requested file", and execute the 'read directory/file information' command. The file information of "Number of file information" is stored in the response data.

☞ Page 205 Read directory/file information (command: 1810)

2. Check if there is a file from which the file information is not read.

When "Number of file information" of the response data is in the status as shown below, it indicates that file information of all the files have been read. Complete the processing.

- MELSEC-Q/L series (when using subcommand '0000'): "Number of file information" < "Number of requested file"
- MELSEC iQ-R series (when using subcommand '0040'): "Number of file information" = -1 (FFFFH)

3. Read the file information from a file from which the file information is not read.

Specify "Number of requested file" = 36 (upper limit) to the request data, and execute the command.

For "Head file No.", specify one of the following value.

- MELSEC-Q/L series (when using subcommand '0000'): "Head file No." = previous "Head file No." + "Number of file information"
- MELSEC iQ-R series (when using subcommand '0040'): "Head file No." = previous "Last file No." + 1

☞ Page 205 Read directory/file information (command: 1810)

4. Repeat the procedure of Step 2 and later.



The correct information cannot be obtained if the file operation is performed from other devices while reading information of all the files in the directory. Do not perform file operation from other devices while reading file information.

Procedure to read files

1. Check for file existence.

Any of the following commands can be used.

- Page 205 Read directory/file information (command: 1810)
- Page 211 Search directory/file information (command: 1811)

2. Read the files.

For read command, use open and close command to prohibit access from other devices.

Execute the commands in the following order.

- Page 230 Open file (command: 1827)
- Page 234 Read file (command: 1828)
- Page 238 Close file (command: 182A)

Procedure to overwrite files

1. Check for file existence.

Any of the following commands can be used.

- Page 205 Read directory/file information (command: 1810)
- Page 211 Search directory/file information (command: 1811)

2. Write data to the file.

For writing data, use open and close command to prohibit access from other devices.

Execute the commands in the following order.

- Page 230 Open file (command: 1827)
- Page 236 Write to file (command: 1829)
- Page 238 Close file (command: 182A)



In the following cases, create a new file and write data to it after deleting the target file.

- When the target file is sequence program file (*.PRG) or FB file (*.PFB) of MELSEC iQ-R series.
 - When changing the file size of MELSEC-Q/L series is required.
- Page 187 Procedure to delete files
- Page 186 Procedure to create new file and write data

Procedure to create new file and write data

The procedure varies depending on types of file. Refer to procedure according to file types.

| File type | File extension | | Reference |
|---------------------------|-------------------|--------------------|--|
| | MELSEC-Q/L series | MELSEC iQ-R series | |
| Header statement file | DAT | — | Page 186 When creation of temporary file is required |
| Sequence program file | QPG | PRG | |
| Device comment file | QCD | DCM | |
| Device initial value file | QDI | DID | |
| FB file | — | PFB | |
| File other than above | | | Page 186 When creation of temporary file is not required |



Before creating a file, secure the free space of the target memory. It can be checked and secured by an Engineering tool.

When creation of temporary file is required

1. Create a new temporary file.

Register the file name and reserve the required capacity for the file.

The extension of the temporary file must be other than DAT, PRG, QPG, PFB, QCD, DCM, QDI, DID.

- ☞ Page 214 Create new file (command: 1820)

2. Write data to the file.

For writing data, use open and close command to prohibit access from other devices.

Execute the commands in the following order.

- ☞ Page 230 Open file (command: 1827)
- ☞ Page 236 Write to file (command: 1829)
- ☞ Page 238 Close file (command: 182A)

3. Create a file with the target file extension using the copy function.

After copying a file, delete the temporary file of the copy source as necessary.

- ☞ Page 220 Copy file (command: 1824)
- ☞ Page 217 Delete file (command: 1822)

When creation of temporary file is not required

1. Check for file existence.

Any of the following commands can be used.

- ☞ Page 205 Read directory/file information (command: 1810)
- ☞ Page 211 Search directory/file information (command: 1811)

2. Create a new file.

Register the file name to reserve the required capacity for the file.

- ☞ Page 214 Create new file (command: 1820)

3. Write data to the file.

For writing data, use open and close command to prohibit access from other devices.

Execute the commands in the following order.

- ☞ Page 230 Open file (command: 1827)
- ☞ Page 236 Write to file (command: 1829)
- ☞ Page 238 Close file (command: 182A)

Procedure to delete files

1. Check for file existence.

Any of the following commands can be used.

-  Page 205 Read directory/file information (command: 1810)
-  Page 211 Search directory/file information (command: 1811)

2. Delete the file.

-  Page 217 Delete file (command: 1822)

Procedure to copy files



Before copying file, secure the free area of the target memory. It can be checked and secured with Engineering tool.

1. Check for file existence.

Any of the following commands can be used.

-  Page 205 Read directory/file information (command: 1810)
-  Page 211 Search directory/file information (command: 1811)

2. Copy the file.

After copying a file, delete the file of the copy source as necessary.

-  Page 220 Copy file (command: 1824)
-  Page 217 Delete file (command: 1822)

Procedure to modify file creation date and time

Modify the date of file creation by the 'modify file creation date and time' (command: 1826).

-  Page 227 Modify file creation date and time (command: 1826)

12.2 Considerations

The following shows the considerations for file control.

Files such as read programs and parameters

If the files such as program files and parameters which affect the system are read from the CPU module, keep the files for backup. Do not edit the data in the file on an external device. If the programs or parameters are required to be changed, use an Engineering tool.

Access for '\$MELPRJ\$' folder of RCPU

The \$MELPRJ\$ folder of RCPU is the folder that controls the data written from an Engineering. Do not access the \$MELPRJ\$ folder other than the purpose of data backup or restoration.

When performing data backup or restoration in the '\$MELPRJ\$' folder, read/write all the files in the '\$MELPRJ\$' folder. If only a part of '\$MELPRJ\$' folder is changed, it may not operate properly.

If the file is protected

When executing the following command, disable the protection (system protect of CPU module, the protection switch of SD memory card) of the access target CPU. If the command is executed with the access target protected, the command will be terminated abnormally.

| Function | Reference |
|--------------------------------------|---|
| Create new file (Register file name) | Page 214 Create new file (command: 1820) |
| Delete file | Page 217 Delete file (command: 1822) |
| Copy file | Page 220 Copy file (command: 1824) |
| Modify file attribute | Page 224 Modify file attribute (command: 1825) |
| Modify file creation date and time | Page 227 Modify file creation date and time (command: 1826) |
| Write to file | Page 236 Write to file (command: 1829) |

Files that cannot be modified while CPU module is in RUN

The executing file cannot be modified while the CPU module is in RUN.

When data write or delete command is executed to a file being executed, place CPU module in the STOP status. If the command is executed during RUN, the command completes abnormally.

 Page 457 Applicable Commands for Online Program Change

12.3 Data to be specified in commands

This section explains the contents and specification methods for data items which are set in each command related to file control.

Password

Specify the password of the file to be accessed.

The specification of a password differs depending on the module. Specify a password which corresponds to the module of access target.

For details on the password, refer to the manual of access target module.

| Access target | Password to be set | Sub command | Data specified by Message |
|---------------------------|---------------------------------------|-------------|--|
| MELSEC-Q series module | Password (4 characters) | 0000 | Password character string (fixed to 4 characters) |
| | File password 32 (4 characters) | | |
| MELSEC-L series module | File password 32 (4 to 32 characters) | 0004 | Password character string (fixed to 32 characters) ^{*1} |
| MELSEC iQ-R series module | File password (6 to 32 characters) | 0040 | Number of password characters, password character string (variable length) |

*1 If the password is less than 32 characters, append a space (code: 20H).

Password for MELSEC-Q series module (4 characters)

Use the subcommand '0000'.

When a password is set

Send password character string in 4-digit ASCII code.

Specify the password in ASCII code during data communication in binary code as well.

Ex.

When the password is 'ABCD'

ASCII code, binary code

| | | | |
|-----|-----|-----|-----|
| A | B | C | D |
| 41H | 42H | 43H | 44H |

Restriction

When setting the password with the file password 32 function of High-speed universal model QCPU, set it with four characters.

When a password is not set

Specify 20H for 4 bytes.

ASCII code, binary code

| | | | |
|-----|-----|-----|-----|
| 20H | 20H | 20H | 20H |
|-----|-----|-----|-----|

File password 32 of MELSEC-L series (4 to 32 characters)

Use the subcommand '0004'.

Restriction

Subcommand '0000' can be used only when the password is not set to the file.

When subcommand 0000 is used to MELSEC-L series module, specify 20H for 4 bytes.

■ When a password is set

Send password character string in 32-digit ASCII code.

Specify the password in ASCII code during data communication in binary code as well.

If the password is less than 32 characters, a space (20H) is added for the shortage of the characters.

Ex.

When the password is 'ABCDEF'

ASCII code, binary code

| A | B | C | D | E | F |
|-------------|-----|-----|-----|-----|-----|
| 41H | 42H | 43H | 44H | 45H | 46H |
| 20H | 20H | 20H | 20H | 20H | 20H |
| (32 digits) | | | | | |

■ When a password is not set

Specify 20H for 32 bytes.

ASCII code, binary code

File password for MELSEC iQ-R series (6 to 32 characters)

Use the subcommand '0040'.



If multiple wrong passwords are attempted continuously, the password will be locked out and cannot unlock the password for a while.

12

■When a password is set

Specify the number of characters and the character string of password.

Specify 'Password character string' with variable length. The length of data is specified in 'Number of characters'. (☞ Page 194 Number of characters)

| ASCII code | Binary code | | | | | | | | |
|--|-------------------|--|------------|-------------------|---|--|--|-----------|-------------------|
| Number of characters Password character string <table border="1"><tr><td> </td><td> </td></tr><tr><td>(4 digits)</td><td>(Variable length)</td></tr></table> | | | (4 digits) | (Variable length) | Number of characters Password character string <table border="1"><tr><td> </td><td> </td></tr><tr><td>(2 bytes)</td><td>(Variable length)</td></tr></table> | | | (2 bytes) | (Variable length) |
| | | | | | | | | | |
| (4 digits) | (Variable length) | | | | | | | | |
| | | | | | | | | | |
| (2 bytes) | (Variable length) | | | | | | | | |

Specify the password character string in ASCII code during data communication in binary code as well.

Ex.

Password is "ABCDEFGHIJKLMNPQRSTUVWXYZ" (26 characters)

| ASCII code | Binary code | | | | | | | | |
|---|--|------------------------------|--------------------------|--|--|--|------------------------------|------------|--|
| Number of characters Password character string <table border="1"><tr><td>0 0 1 A</td><td>A B C D ... Z</td></tr><tr><td>30H 30H 31H 41H</td><td>41H 42H 43H 44H ... 5AH</td></tr></table> | 0 0 1 A | A B C D ... Z | 30H 30H 31H 41H | 41H 42H 43H 44H ... 5AH | Number of characters Password character string <table border="1"><tr><td> </td><td>A B C D ... Z</td></tr><tr><td>1AH 00H</td><td>41H 42H 43H 44H ... 5AH</td></tr></table> | | A B C D ... Z | 1AH 00H | 41H 42H 43H 44H ... 5AH |
| 0 0 1 A | A B C D ... Z | | | | | | | | |
| 30H 30H 31H 41H | 41H 42H 43H 44H ... 5AH | | | | | | | | |
| | A B C D ... Z | | | | | | | | |
| 1AH 00H | 41H 42H 43H 44H ... 5AH | | | | | | | | |

■When a password is not set

Specify with the number of character '0'.

| ASCII code | Binary code |
|--|-------------|
| 0 0 0 0 30H 30H 30H 30H | 00H 00H |

Drive No.

This is a data to specify the drive of CPU module of which files are to be controlled.

| Specified value | Target drive | |
|-----------------|--|--------------------------------------|
| | RCPU | LCPU/QCPU/QnACPU |
| 0000H | — | Program memory |
| 0001H | Device/label memory (file storage area) The same operation when 0003H is specified. | SRAM card |
| 0002H | SD memory card | Flash card, ATA card, SD memory card |
| 0003H | Device/label memory (file storage area) | Standard RAM |
| 0004H | Data memory | Standard ROM |



The program memory of RCPUs cannot be accessed. When reading/writing program files, use the data memory. (☞ Page 188 Access for '\$MELPRJ\$' folder of RCPUs)

Setting method

■Data communication using ASCII code

Convert the numerical value that indicates access target drive to 4-digit ASCII code (hexadecimal) and send it from upper digits.

■Data communication using binary code

Send 2-byte numerical values that indicate access target drive from the lower byte (L: bits 0 to 7).

Ex.

Drive No. is '0003H'

| ASCII code | Binary code |
|----------------------------|-------------|
| 0 0 0 3 30H 30H 30H 33H | 03H 00H |

File No.

This is a number for module to control files.

File No. can be obtained by following command.

☞ Page 211 Search directory/file information (command: 1811)

File No. of MELSEC-Q/L series module

Use the subcommand '0000'.

A file No. can be specified within the range of 1 to 256 (1H to 100H).

■Data communication using ASCII code

Convert the file No. to 4-digit ASCII code (hexadecimal) and send it from upper digits.

■Data communication using binary code

Send 2-byte^{*1} numerical values that indicate file No. from lower byte (L: bits 0 to 7).

*1 For C24, the additional code may be added. (☞ Page 33 Additional code (10H))

Ex.

1FH

| ASCII code | Binary code |
|----------------------------|-------------|
| 0 0 1 F 30H 30H 31H 46H | 1FH 00H |

File No. of MELSEC iQ-R series module

Use the subcommand '0040'.

■Data communication using ASCII code

Convert the file No. to ASCII code 8 digits (hexadecimal), and transmit it from the upper digits.

■Data communication using binary code

Send 4-byte^{*1} numerical values that indicate file No. from the lower byte (L: bits 0 to 7).

*1 For C24, the additional code may be added. (☞ Page 33 Additional code (10H))

Ex.

For 1FH

| ASCII code | Binary code |
|--|-----------------|
| 0 0 0 0 0 0 1 F 30H 30H 30H 30H 30H 30H 31H 46H | 1FH 00H 00H 00H |

Number of files

Specify the number of files to be accessed. The number of registered files or the number of accessed files are returned.

Setting method

■Data communication using ASCII code

Convert the numerical value to 4-digit ASCII code (hexadecimal) and send it from upper digits.

■Data communication using binary code

Send 2-byte numerical values^{*1} from the lower byte (L: bits 0 to 7).

*1 For C24, the additional code may be added. (☞ Page 33 Additional code (10H))

Ex.

Number of file is 3

| ASCII code | Binary code |
|-------------------------------|-------------|
| 0 0 0 3 30H, 30H, 30H, 33H | 03H, 00H |

Number of characters

This indicates the number of characters of variable length character string to be specified.

Setting method

■Data communication using ASCII code

Convert the numerical value to 4-digit ASCII code (hexadecimal) and send it from upper digits.

■Data communication using binary code

Send 2-byte numerical values^{*1} from the lower byte (L: bits 0 to 7).

*1 For C24, the additional code may be added. (☞ Page 33 Additional code (10H))

Ex.

Number of characters is 86 characters (56H)

| ASCII code | Binary code |
|-------------------------------|-------------|
| 0 0 5 6 30H, 30H, 35H, 36H | 56H, 00H |

Directory specification

Specify the absolute path to a file to be accessed.

Specify 'Path name' with variable length. The length of data is specified in 'Number of characters'. (☞ Page 194 Number of characters)

| ASCII code | Binary code |
|--|---|
| Number of characters Path name (4 digits) (Variable length) | Number of characters Path name (2 bytes) (Variable length) |

Specify '0' for the root directory (root folder)

| ASCII code | Binary code |
|-------------------------------|-------------|
| 0 0 0 0 30H, 30H, 30H, 30H | 00H, 00H |

Path name

Specify the absolute path by UTF-16 from the root folder.

"Drive name:\\" is not required in front of the path. Use '\' (005CH) for the delimiter between the folder names.

■Data communication using ASCII code

Convert the numerical value of UTF-16, which indicates a path character string, to ASCII code (hexadecimal) and send it from upper digits.

■Data communication using binary code

Send the numerical value of UTF-16 indicating the path character string from lower bytes (L: bits 0 to 7).

Ex.

Folder root folder 'A'

(UTF-16: 'A'= 0041)

| ASCII code | Binary code |
|----------------------------------|-------------|
| 0 0 4 1 30H , 30H , 34H , 31H | 41H , 00H |

File name specification

Specify the file name to be accessed.

A file name differs in specification by the module. Specify a corresponding file name for the module of access target. For details of usable file name, refer to the manual of module for access target.

| Access target | Usable file name | Sub command | Data specified by Message |
|---------------------------|--|--------------|---|
| MELSEC-Q/L series module | File name (up to 8 characters) and extension (3 characters) Specify with ASCII code character string. | 0000 0004 | <ul style="list-style-type: none">• File name (8 characters fixed) and extension (3 characters fixed)• Number of characters (up to 12 characters) and file name character string (including the extension) |
| MELSEC iQ-R series module | Path name, file name (up to 60 characters) and extension Specify with Unicode character string. | 0040 | <ul style="list-style-type: none">• Number of characters (up to 252 characters) and file name character string (including the path and extension) |

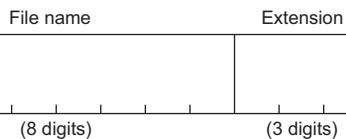


Files other than the one described as usable files in the manuals of modules are for system. Do not change the file name.

File name of MELSEC-Q/L series module (File name and extension)

When using the subcommand 0000 for the read directory/file information (command: 1810), the file name (8 characters fixed) and the extension (3 characters fixed) are stored in the response data. A period is not inserted between the file name and the extension.

ASCII code, binary code



Specify the file name and the extension in ASCII code during data communication in binary code as well.

Ex.

The file name and the extension are "ABCDEFGH.QPG".

ASCII code, binary code

| | | | | | | | | | | |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| A | B | C | D | E | F | G | H | Q | P | G |
| 41H | 42H | 43H | 44H | 45H | 46H | 47H | 48H | 51H | 50H | 47H |

File name of MELSEC-Q/L series module (Number of characters and file name)

For the data other than response data of read directory/file information (command: 1810), specify with number of characters and file name character string.

Use the subcommand '0000' or '0004'.

The file name and the extension are specified as a variable length character string with a period inserted between them.

Specify the length of character string with "number of characters". ( Page 194 Number of characters)

| ASCII code | Binary code |
|---|---|
| Number of characters File name + "." + Extension  (4 digits) (Variable length) | Number of characters File name + "." + Extension  (2 bytes) (Variable length) |

12

Specify "File name" + "Period (2EH)" + "Extension" in ASCII code during data communication in binary code as well.

Ex.

File name and the extension are "ABC.QPG". (7 characters including a period)

| ASCII code | Binary code |
|----------------------------|--|
| 0 0 0 7 30H 30H 30H 37H | A B C . Q P G 41H 42H 43H 2EH 51H 50H 47H 07H 00H 41H 42H 43H 2EH 51H 50H 47H |

File name of MELSEC iQ-R series module (Number of characters and file name)

Use the subcommand '0040'.

Specify the file name including absolute path from the root folder with variable length.

The file name and the extension are specified by inserting a period between them. Specify the length of character string with "number of characters". (☞ Page 194 Number of characters)

| ASCII code | Binary code | | | | |
|---|-------------|--|--|--|--|
| Number of characters File name + "." + Extension <table border="1"><tr><td></td><td></td></tr></table> (4 digits) (Variable length) | | | Number of characters File name + "." + Extension <table border="1"><tr><td></td><td></td></tr></table> (2 bytes) (Variable length) | | |
| | | | | | |
| | | | | | |

Specify "Path name" + "Period (002EH)" + "Extension" by Unicode(UTF-16).

"Drive name:" is not required in front of the path. Use '\' (005CH) for the delimiter between the folder names.

For the characters that cannot be used for a file name and naming rules, refer to the manual of the access target module.

■Data communication using ASCII code

Convert the numerical value of UTF-16, which indicates a file name, to ASCII code (hexadecimal) and send it from the upper digits.

■Data communication using binary code

Send the numerical value of UTF-16 that indicates file name from the lower byte (L: bits 0 to 7).

Ex.

When the file name and the extension are 'LINE\LINE.CSV' (13 characters)

| ASCII code | Binary code |
|----------------------------|--------------------------|
| 0 0 0 D 30H 30H 30H 44H | LINE/LINE.CSV 0Dh 00H |

The value of the file name 'LINE\LINE.CSV' is as follows.

| Item | Value of code corresponds to character | | | | | | | | | | | | | |
|-------------|--|------|------|------|------|------|------|------|------|------|------|------|------|--|
| File name | L | I | N | E | \ | L | I | N | E | . | C | S | V | |
| UTF-16 | 004C | 0049 | 004E | 0045 | 005C | 004C | 0049 | 004E | 0045 | 002E | 0043 | 0053 | 0056 | |
| ASCII code | 3030 | 3030 | 3030 | 3030 | 3030 | 3030 | 3030 | 3030 | 3030 | 3030 | 3030 | 3030 | 3030 | |
| | 3443 | 3439 | 3445 | 3435 | 3543 | 3443 | 3439 | 3445 | 3435 | 3245 | 3433 | 3533 | 3536 | |
| Binary code | 4C00 | 4900 | 4E00 | 4500 | 5C00 | 4C00 | 4900 | 4E00 | 4500 | 2E00 | 4300 | 5300 | 5600 | |

Attribute

This indicates whether the data can be written to a file or directory.

File attribute

| Read-only attribute | Archive attribute | ASCII code | Binary code | | | | | | | | | | |
|---------------------|--|--|-------------|---|---|-----|-----|-----|-----|--|--|-----|-----|
| Read-only | OFF | <table border="1" style="margin-left: auto; margin-right: auto;"> <tr><td>0</td><td>0</td><td>0</td><td>1</td></tr> <tr><td>30H</td><td>30H</td><td>30H</td><td>31H</td></tr> </table> | 0 | 0 | 0 | 1 | 30H | 30H | 30H | 31H | <table border="1" style="margin-left: auto; margin-right: auto;"> <tr><td>01H</td><td>00H</td></tr> </table> | 01H | 00H |
| 0 | 0 | 0 | 1 | | | | | | | | | | |
| 30H | 30H | 30H | 31H | | | | | | | | | | |
| 01H | 00H | | | | | | | | | | | | |
| ON | <table border="1" style="margin-left: auto; margin-right: auto;"> <tr><td>0</td><td>0</td><td>2</td><td>1</td></tr> <tr><td>30H</td><td>30H</td><td>32H</td><td>31H</td></tr> </table> | 0 | 0 | 2 | 1 | 30H | 30H | 32H | 31H | <table border="1" style="margin-left: auto; margin-right: auto;"> <tr><td>21H</td><td>00H</td></tr> </table> | 21H | 00H | |
| 0 | 0 | 2 | 1 | | | | | | | | | | |
| 30H | 30H | 32H | 31H | | | | | | | | | | |
| 21H | 00H | | | | | | | | | | | | |
| Writable, readable | OFF | <table border="1" style="margin-left: auto; margin-right: auto;"> <tr><td>0</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>30H</td><td>30H</td><td>30H</td><td>30H</td></tr> </table> | 0 | 0 | 0 | 0 | 30H | 30H | 30H | 30H | <table border="1" style="margin-left: auto; margin-right: auto;"> <tr><td>00H</td><td>00H</td></tr> </table> | 00H | 00H |
| 0 | 0 | 0 | 0 | | | | | | | | | | |
| 30H | 30H | 30H | 30H | | | | | | | | | | |
| 00H | 00H | | | | | | | | | | | | |
| ON | <table border="1" style="margin-left: auto; margin-right: auto;"> <tr><td>0</td><td>0</td><td>2</td><td>0</td></tr> <tr><td>30H</td><td>30H</td><td>32H</td><td>30H</td></tr> </table> | 0 | 0 | 2 | 0 | 30H | 30H | 32H | 30H | <table border="1" style="margin-left: auto; margin-right: auto;"> <tr><td>20H</td><td>00H</td></tr> </table> | 20H | 00H | |
| 0 | 0 | 2 | 0 | | | | | | | | | | |
| 30H | 30H | 32H | 30H | | | | | | | | | | |
| 20H | 00H | | | | | | | | | | | | |

Do not access the file in which the value other than above is stored in the attribute since the files are reserved for system use.

Attribute of directory (folder)

| Read-only attribute | Archive attribute | ASCII code | Binary code | | | | | | | | | | |
|---------------------|--|--|-------------|---|---|-----|-----|-----|-----|--|--|-----|-----|
| Read-only | OFF | <table border="1" style="margin-left: auto; margin-right: auto;"> <tr><td>0</td><td>0</td><td>1</td><td>1</td></tr> <tr><td>30H</td><td>30H</td><td>31H</td><td>31H</td></tr> </table> | 0 | 0 | 1 | 1 | 30H | 30H | 31H | 31H | <table border="1" style="margin-left: auto; margin-right: auto;"> <tr><td>11H</td><td>00H</td></tr> </table> | 11H | 00H |
| 0 | 0 | 1 | 1 | | | | | | | | | | |
| 30H | 30H | 31H | 31H | | | | | | | | | | |
| 11H | 00H | | | | | | | | | | | | |
| ON | <table border="1" style="margin-left: auto; margin-right: auto;"> <tr><td>0</td><td>0</td><td>3</td><td>1</td></tr> <tr><td>30H</td><td>30H</td><td>33H</td><td>31H</td></tr> </table> | 0 | 0 | 3 | 1 | 30H | 30H | 33H | 31H | <table border="1" style="margin-left: auto; margin-right: auto;"> <tr><td>31H</td><td>00H</td></tr> </table> | 31H | 00H | |
| 0 | 0 | 3 | 1 | | | | | | | | | | |
| 30H | 30H | 33H | 31H | | | | | | | | | | |
| 31H | 00H | | | | | | | | | | | | |
| Writable, readable | OFF | <table border="1" style="margin-left: auto; margin-right: auto;"> <tr><td>0</td><td>0</td><td>1</td><td>0</td></tr> <tr><td>30H</td><td>30H</td><td>31H</td><td>30H</td></tr> </table> | 0 | 0 | 1 | 0 | 30H | 30H | 31H | 30H | <table border="1" style="margin-left: auto; margin-right: auto;"> <tr><td>10H</td><td>00H</td></tr> </table> | 10H | 00H |
| 0 | 0 | 1 | 0 | | | | | | | | | | |
| 30H | 30H | 31H | 30H | | | | | | | | | | |
| 10H | 00H | | | | | | | | | | | | |
| ON | <table border="1" style="margin-left: auto; margin-right: auto;"> <tr><td>0</td><td>0</td><td>3</td><td>0</td></tr> <tr><td>30H</td><td>30H</td><td>33H</td><td>30H</td></tr> </table> | 0 | 0 | 3 | 0 | 30H | 30H | 33H | 30H | <table border="1" style="margin-left: auto; margin-right: auto;"> <tr><td>30H</td><td>00H</td></tr> </table> | 30H | 00H | |
| 0 | 0 | 3 | 0 | | | | | | | | | | |
| 30H | 30H | 33H | 30H | | | | | | | | | | |
| 30H | 00H | | | | | | | | | | | | |

Do not access the directory (folder) in which the value other than above is stored in the attribute since it is reserved for system use.

Creation date and time (last edit date and time)

This is a date and time when the current file contents are registered.

Setting method

Represent the date (year, month, day) and time (hour, minute, second) with 16-bit value, respectively.

■Data communication using ASCII code

Convert the respective numerical value to 4-digit ASCII code (hexadecimal) and send from the upper digits (time, year).

■Data communication using binary code

Send the respective 2-byte numerical value from the lower byte (L: bits 0 to 7).

Date (year, month, day)

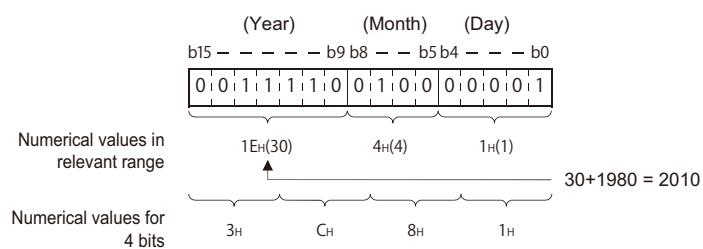
Represent the year, month, and day with 16-bit value.

- Year: The binary value is represented with bits 9 to 15 by setting the year 1980 to '0'. *1
- Month: The binary value is represented with bits 5 to 8.
- Day: The binary value is represented with bits 0 to 4.

*1 This indicates incremental number of year by regarding 1980 as '0'.

Ex.

For April 1, 2010



| ASCII code | Binary code |
|---|-------------|
| 3 C 8 1 33H 43H 38H 31H | 81H 3CH |

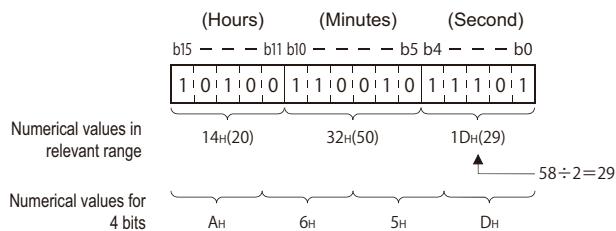
Time (hour, minute, second)

Represent the hour, minute, second with 16-bit value.

- Hour: The binary value is represented with bits 11 to 15.
- Minute: The binary value is represented with bits 5 to 10.
- Second: The binary value divided by 2 is represented with bits 0 to 4.

Ex.

When 20:50:58



| ASCII code | Binary code |
|----------------------------|-------------|
| A 6 5 D 41H 36H 35H 44H | 5DH A6H |

File size

This indicates the file capacity in byte units.

Setting method

■Data communication using ASCII code

Convert the numerical value to 8-digit ASCII code (hexadecimal), and send it from the upper digits.

■Data communication using binary code

Send 4-byte numerical values^{*1} from the lower byte (L: bits 0 to 7).

*1 For C24, the additional code may be added. (☞ Page 33 Additional code (10H))

Ex.

The file size is 7168 bytes

| ASCII code | Binary code |
|--|-----------------|
| 0 0 0 0 1 C 0 0 30H 30H 30H 30H 31H 43H 30H 30H | 00H 1CH 00H 00H |

File pointer No.

This is a number for CPU module to manage files.

The file pointer No. can be acquired with the following command.

☞ Page 230 Open file (command: 1827)

Setting method

■Data communication using ASCII code

Send 4-byte^{*1} ASCII code data.

■Data communication using binary code

Send 2-byte^{*1} numerical value.

*1 For C24, the additional code may be added. (☞ Page 33 Additional code (10H))

Ex.

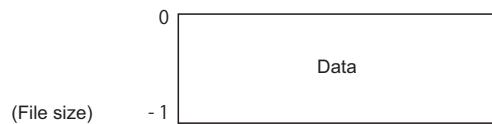
For AH

| ASCII code | Binary code |
|-------------------------------|-------------|
| 0 0 0 A 30H, 30H, 30H, 3AH | 0AH, 00H |

Offset address

Specify the address (one address/one byte) from the head (offset address: 0H) of each file with an even number.

Offset address



Setting method

■Data communication using ASCII code

Convert the numerical value to 8-digit ASCII code (hexadecimal), and send it from the upper digits.

■Data communication using binary code

Send 4-byte numerical values^{*1} from the lower byte (L: bits 0 to 7).

*1 For C24, the additional code may be added. (☞ Page 33 Additional code (10H))

Ex.

Offset address is 780H(1920)

| ASCII code | Binary code |
|--|--------------------|
| 0 0 0 0 0 7 8 0 30H, 30H, 30H, 30H, 37H, 38H, 30H | 80H, 07H, 00H, 00H |

Number of bytes

Specify the number of bytes of data to be read or written as one address/one byte.

Setting method

■Data communication using ASCII code

Convert the numerical value to 4-digit ASCII code (hexadecimal) and send it from upper digits.

■Data communication using binary code

Send 2-byte numerical values^{*1} from the lower byte (L: bits 0 to 7).

*1 For C24, the additional code may be added. (☞ Page 33 Additional code (10H))

Ex.

Number of bytes is 780H(1920)

12

| ASCII code | Binary code |
|----------------------------|-------------|
| 0 7 8 0 30H 37H 38H 30H | 80H 07H |

Read data, write data

This is a content of file to be read/written. The data for 1 address is handled as 1-byte.

Read data

The data which have been read is stored.

The "Read data" is variable length. The length of data is specified with "Number of bytes read".

Write data

The data to be written is stored.

The "Write data" is variable length. The length of data is specified with "Number of bytes written".

The order of data must be the same as the read data.

■Data communication using ASCII code

Convert the 1-byte data (1 address) to 2-digit ASCII code (hexadecimal) and send it from upper digits.

■Data communication using binary code

Send 1 address as 1 byte.

Open mode

Specify whether the specified file is open for reading or for writing with the 'open file' (command: 1827).

| Item | ASCII code | Binary code | | | | | | | | | | |
|-------------|--|-------------|-----|---|---|-----|-----|-----|-----|---|-----|-----|
| For reading | <table border="1"><tr><td>0</td><td>0</td><td>0</td><td>0</td></tr><tr><td>30H</td><td>30H</td><td>30H</td><td>30H</td></tr></table> | 0 | 0 | 0 | 0 | 30H | 30H | 30H | 30H | <table border="1"><tr><td>00H</td><td>00H</td></tr></table> | 00H | 00H |
| 0 | 0 | 0 | 0 | | | | | | | | | |
| 30H | 30H | 30H | 30H | | | | | | | | | |
| 00H | 00H | | | | | | | | | | | |
| For writing | <table border="1"><tr><td>0</td><td>1</td><td>0</td><td>0</td></tr><tr><td>30H</td><td>31H</td><td>30H</td><td>30H</td></tr></table> | 0 | 1 | 0 | 0 | 30H | 31H | 30H | 30H | <table border="1"><tr><td>00H</td><td>01H</td></tr></table> | 00H | 01H |
| 0 | 1 | 0 | 0 | | | | | | | | | |
| 30H | 31H | 30H | 30H | | | | | | | | | |
| 00H | 01H | | | | | | | | | | | |

Close type

Specify a target to be closed with the 'close file' (command: 182A).

| File to unlock | ASCII code | Binary code | | | | | | | | | | |
|--------------------------------------|--|-------------|-----|---|---|-----|-----|-----|-----|---|-----|-----|
| A file specified by the file pointer | <table border="1"><tr><td>0</td><td>0</td><td>0</td><td>0</td></tr><tr><td>30H</td><td>30H</td><td>30H</td><td>30H</td></tr></table> | 0 | 0 | 0 | 0 | 30H | 30H | 30H | 30H | <table border="1"><tr><td>00H</td><td>00H</td></tr></table> | 00H | 00H |
| 0 | 0 | 0 | 0 | | | | | | | | | |
| 30H | 30H | 30H | 30H | | | | | | | | | |
| 00H | 00H | | | | | | | | | | | |
| All files | <table border="1"><tr><td>0</td><td>0</td><td>0</td><td>1</td></tr><tr><td>30H</td><td>30H</td><td>30H</td><td>31H</td></tr></table> | 0 | 0 | 0 | 1 | 30H | 30H | 30H | 31H | <table border="1"><tr><td>01H</td><td>00H</td></tr></table> | 01H | 00H |
| 0 | 0 | 0 | 1 | | | | | | | | | |
| 30H | 30H | 30H | 31H | | | | | | | | | |
| 01H | 00H | | | | | | | | | | | |
| | <table border="1"><tr><td>0</td><td>0</td><td>0</td><td>2</td></tr><tr><td>30H</td><td>30H</td><td>30H</td><td>32H</td></tr></table> | 0 | 0 | 0 | 2 | 30H | 30H | 30H | 32H | <table border="1"><tr><td>02H</td><td>00H</td></tr></table> | 02H | 00H |
| 0 | 0 | 0 | 2 | | | | | | | | | |
| 30H | 30H | 30H | 32H | | | | | | | | | |
| 02H | 00H | | | | | | | | | | | |

When all files are specified for unlock, the external device that executes the close (command: 182A) closes all files being open regardless of the hierarchy of drive No. or directory (folder). (There is no difference between 0001H and 0002H.) The files which were opened by the other devices cannot be closed.

Executing the command to files locked by the other external device results in abnormal completion of the command.



Files are closed by restarting a module (such as resetting CPU module).

12.4 File Check

Read directory (folder), file name, file creation date and time, and file No. in the specified drive.

The presence of files to be accessed and the file No. to access can be checked.

Read directory/file information (command: 1810)

For the specified storage destination file, read the file name, file creation date and time (last edit date and time) etc.

Read the file information from the head file specified with the file No. to the specified number of files.

Message format

The following shows the message format of the request data and response data of the command.

12

■Request data

| Command | Subcommand | Fixed values | Drive No. | Head file No. | Number of requested files | Directory specification |
|---------|------------|--------------|-----------|---------------|---------------------------|-------------------------|
|---------|------------|--------------|-----------|---------------|---------------------------|-------------------------|

■Response data

| | | | | | | | | |
|--|---------------|------|-------------------------|-----------|---------------------------------------|------|-------------------------|-----------|
| Number of file information (n points) | Last file No. | File | Last edit date and time | File size | ... | File | Last edit date and time | File size |
| Directory/file information(1st point) | | | | | Directory/file information(nth point) | | | |

Data specified by request data

■Command

| ASCII code | Binary code (for C24) ^{*1} | Binary code (for E71) |
|----------------------------|-------------------------------------|-----------------------|
| 1 8 1 0 31H 38H 31H 30H | DLE 10H 10H 18H | 10H 18H |

*1 For C24, the additional code may be added. ( Page 33 Additional code (10H))

■Sub command

| Item | ASCII code | Binary code |
|------------------------|----------------------------|-------------|
| For MELSEC-Q/L series | 0 0 0 0 30H 30H 30H 30H | 00H 00H |
| For MELSEC iQ-R series | 0 0 4 0 30H 30H 34H 30H | 40H 00H |

■Fixed value

Specify 30H in 4 bytes.

ASCII code, binary code

| |
|-----------------------|
| 30H , 30H , 30H , 30H |
|-----------------------|

■Drive No.

Specify the access target drive. (☞ Page 192 Drive No.)

■Head file No.

Specify the file No. of the file from which the file information is to be read. (☞ Page 193 File No.)

File No. can be obtained by following command.

☞ Page 211 Search directory/file information (command: 1811)

■Number of requested file

Specify the number of files from which file information is read in the range of 1 to 36. (☞ Page 194 Number of files)

■Directory specification

Depending on the access target, specify any of the following:

- MELSEC iQ-R series (when using subcommand '0040'): Specify the target folder with an absolute path. (☞ Page 194 Directory specification)
- MELSEC-Q/L series (when using subcommand '0000'): Specify '0'.

| ASCII code | Binary code |
|----------------------------------|-------------|
| 0 0 0 0 30H , 30H , 30H , 30H | 00H , 00H |



When checking all files in the specified directory (folder), refer to the following section.

☞ Page 184 Procedure to read information from all files in directory (folder)

Data stored in response data

■Number of file information

The number of file information which has been read is stored. (☞ Page 194 Number of files)

In the following situation, the number of file information will be fewer than the number of requested file.

- When using subcommand '0040', the data of "Number of requested file" cannot be stored at one communication because the file name length is too long.
- When the number of files exist following the "Head file No." is less than the "Number of requested file".

When no files exist after the specified head file No., the following value is stored.

- MELSEC-Q/L series (when using subcommand '0000'): "Number of file information" = '0'
- MELSEC iQ-R series (when using subcommand '0040'): "Number of file information" = -1 (FFFFH)

■Last file No.

The file No. of the last file from which file information was read is stored. (☞ Page 193 File No.)

For MELSEC-Q/L series (subcommand '0000'), last file No. is not stored.



File No. is also assigned for the system-reserved data that cannot acquire the directory/file information, therefore, the "Last file No." may differ from the file No. of the last file for the read file information.

Acquire the file No. using the following command.

☞ Page 211 Search directory/file information (command: 1811)

Use "Last file No." in the following process.

☞ Page 184 Procedure to read information from all files in directory (folder)

■Directory/file information

Information for number of file information is stored.

When folders exist in the specified storage destination, the information for the folder is also read.

| File name specification | Attribute | Spare data ASCII: 18 bytes Binary: 9 bytes | Last edit time | Last edit date | Spare data ASCII: 4 bytes Binary: 2 bytes | File size |
|----------------------------|-----------|--|----------------|----------------|---|-----------|
| File | | Last edit date and time | | | | |
| File information (1 point) | | | | | | |

The following items are stored for each file.

- File name specification: The file name is stored. The format differs depending on the subcommand. (Path name is not included in the file name even when using the subcommand 0040.)

| Sub command | Reference |
|---|--|
| For MELSEC-Q/L series (Subcommand: 0000) | Page 196 File name of MELSEC-Q/L series module (File name and extension) |
| For MELSEC iQ-R series (Subcommand:0040) | Page 198 File name of MELSEC iQ-R series module (Number of characters and file name) |

- Attribute: The file attributes are stored. (☞ Page 199 Attribute)
- Last edit time, Last edit date: The last edit date and time are stored. (☞ Page 200 Creation date and time (last edit date and time))
- Spare data: The system data which are 18 or 4 bytes during data communication in ASCII code and 9 or 2 bytes^{*1} during data communication in binary code are stored.
- File size: Capacity of file is stored in byte unit. (☞ Page 201 File size)

*1 For C24, the additional code may be added. (☞ Page 33 Additional code (10H))



For a folder, the following values will be stored.

- Attribute: Attribute of the directory (folder) is stored. (☞ Page 199 Attribute of directory (folder))
- Last edit time, Last edit date: The folder creation date and time are stored.
- File size: '0' is stored.

Communication example (files for MELSEC-Q/L series)

Read directory/file information under the following conditions.

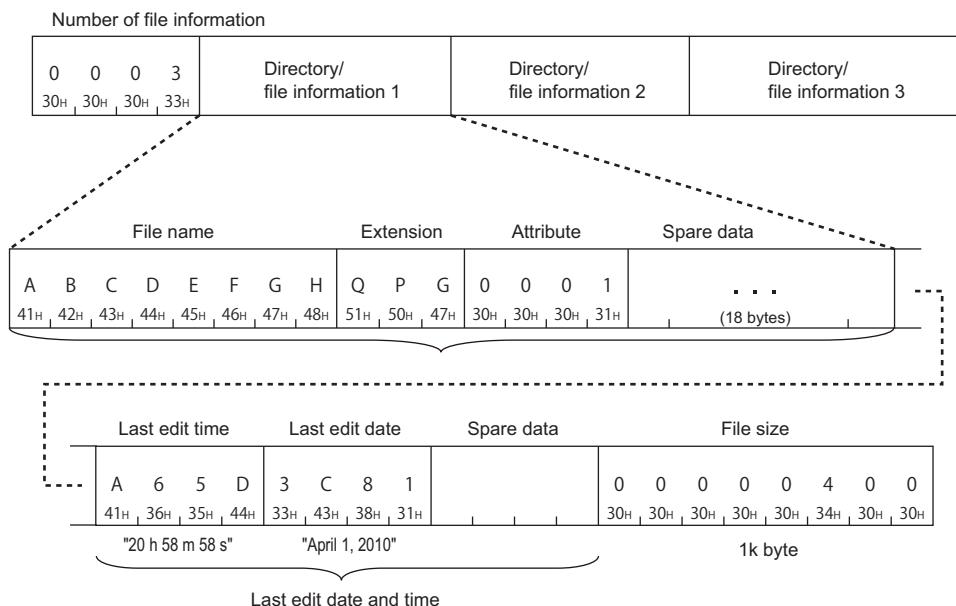
- Drive No.: 0
- Head file No.: 1
- Number of requested files: 3

■Data communication using ASCII code

(Request data)

| Subcommand | (Fixed value) | Drive No. | Head file No. | Number of file requests | Directory specification (Fixed values) |
|----------------------------------|----------------------------------|----------------------------------|----------------------------------|----------------------------------|--|
| 1 8 1 0 31H , 38H , 31H , 30H | 0 0 0 0 30H , 30H , 30H , 30H | 0 0 0 0 30H , 30H , 30H , 30H | 0 0 0 0 30H , 30H , 30H , 30H | 0 0 0 1 30H , 30H , 30H , 31H | 0 0 0 3 30H , 30H , 30H , 33H |

(Response data)

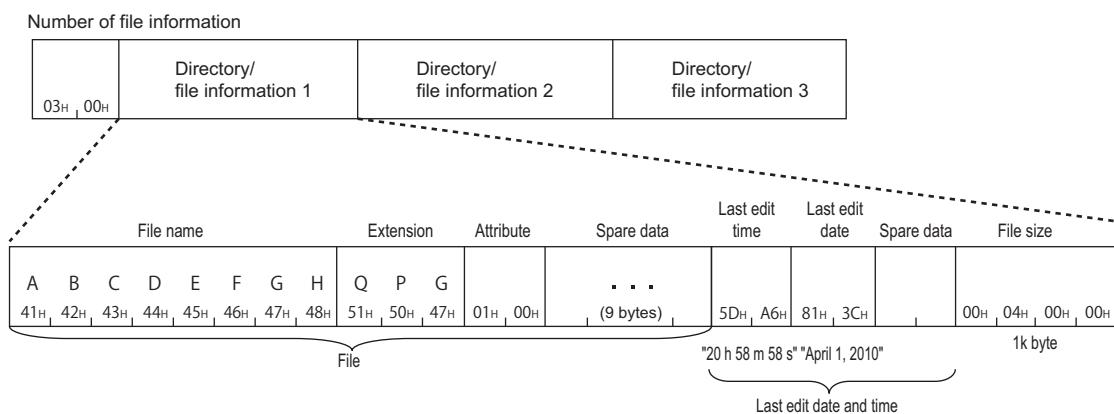


■Data communication using binary code

(Request data)

| Subcommand | (Fixed value) | Drive No. | Head file No. | Number of file requests | Directory specification (Fixed values) |
|------------------------|------------------------------------|------------------------|---------------|-------------------------|--|
| 10H , 18H 00H , 00H | 30H , 30H , 30H , 30H 00H , 00H | 01H , 00H 03H , 00H | 00H , 00H | | |

(Response data)



Communication example (files for MELSEC iQ-R series)

Read directory/file information under the following conditions.

- Drive No.: 4
- Head file No.: 1
- Number of requested files: 3

The path names of directory are shown in the following table.

(1) SUBDIR (6 characters)

| Item | Value of code corresponds to character | | | | | |
|-------------|--|----------|----------|----------|----------|----------|
| Path name | S | U | B | D | I | R |
| UTF-16 | 0053 | 0055 | 0042 | 0044 | 0049 | 0052 |
| ASCII code | 30303533 | 30303535 | 30303432 | 30303434 | 30303439 | 30303532 |
| Binary code | 5300 | 5500 | 4200 | 4400 | 4900 | 5200 |

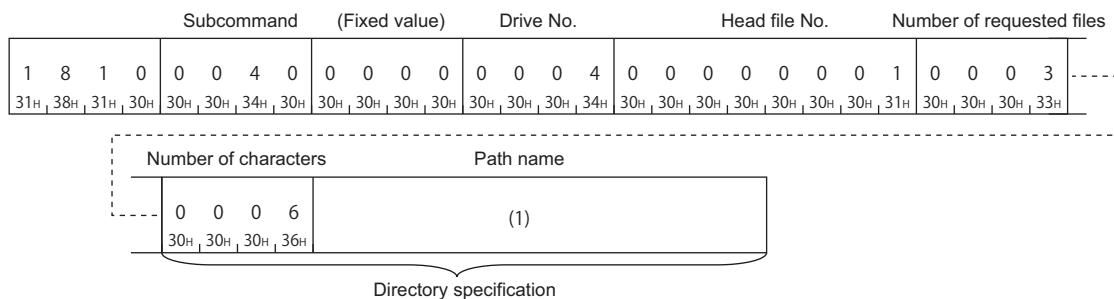
The following shows the file name of a directory/file information 1 to be read.

(2) LINE.CSV (8 characters)

| Item | Value of code corresponds to character | | | | | | | |
|-------------|--|----------|----------|----------|----------|----------|----------|----------|
| File name | L | I | N | E | . | C | S | V |
| UTF-16 | 004C | 0049 | 004E | 0045 | 002E | 0043 | 0053 | 0056 |
| ASCII code | 30303443 | 30303439 | 30303445 | 30303435 | 30303245 | 30303433 | 30303533 | 30303536 |
| Binary code | 4C00 | 4900 | 4E00 | 4500 | 2E00 | 4300 | 5300 | 5600 |

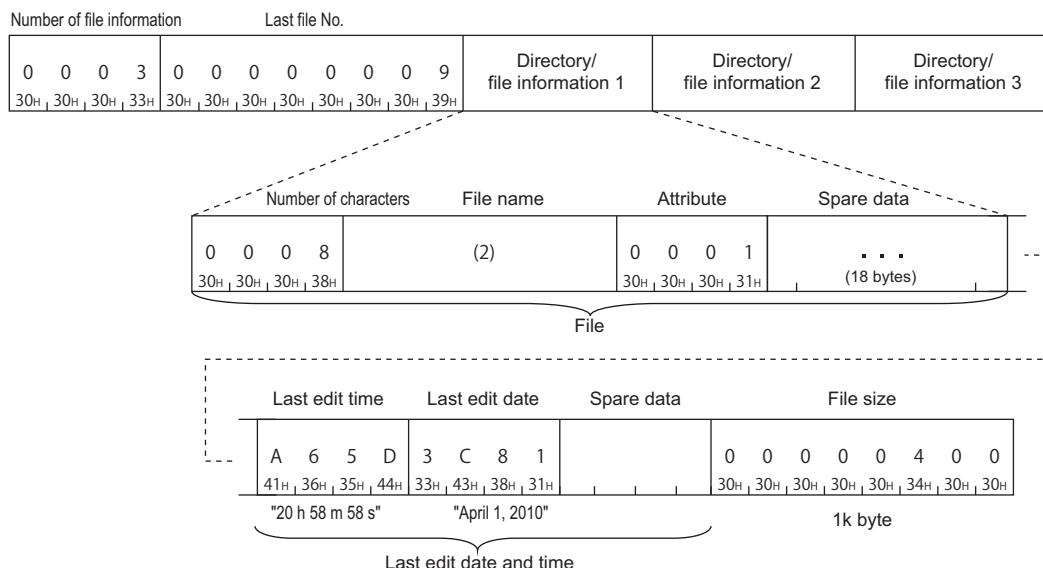
■Data communication using ASCII code

(Request data)



In the figure (1), set the value of "ASCII code" in the table of "Value of code corresponds to character" in "Path name".

(Response data)



In the figure (2), set the value of "ASCII code" in the table of "Value of code corresponds to character" in "File name".

■Data communication using binary code

(Request data)

| Subcommand | (Fixed value) | Drive No. | Head file No. | Number of file requests | Number of characters | Path name |
|------------|---------------|-----------|---------------|-------------------------|----------------------|-----------|
| 10H | 18H | 40H | 00H | 30H | 30H | 30H |

(1)

Directory specification

In the figure (1), set the value of "Binary code" in the table of "Value of code corresponds to character" in "Path name".

(Response data)

| Number of file information | Last file No. | Directory/file information 1 | Directory/file information 2 | Directory/file information 3 |
|----------------------------|--------------------|------------------------------|------------------------------|------------------------------|
| 03H, 00H | 09H, 00H, 00H, 00H | | | |

| Number of characters | File name | Attribute | Spare data | Last edit time | Last edit date | Spare data | File size |
|----------------------|-----------|-----------|------------------|----------------|----------------|------------|--------------------|
| 08H, 00H | (2) | 01H, 00H | ... (9 bytes) | 5DH, A6H | 81H, 3CH | | 00H, 04H, 00H, 00H |

"20 h 58 m 58 s" "April 1, 2010"

Last edit date and time

In the figure (2), set the value of "Binary code" in the table of "Value of code corresponds to character" in "File name".

Search directory/file information (command: 1811)

Read the file No. of the specified file.

The file No. indicates the registration number of a file when the file is written to the module.

Message format

The following shows the message format of the request data and response data of the command.

■Request data

| Command | Subcommand | Fixed values 1 | Drive No. | Fixed values 2 | Number of file name characters | File name |
|---------|------------|----------------|-----------|----------------|--------------------------------|-----------|
|---------|------------|----------------|-----------|----------------|--------------------------------|-----------|

■Response data

File No. is stored. (☞ Page 193 File No.)

If the file with the specified name does not exist, the command completes abnormally.

12

Data specified by request data

■Command

| ASCII code | Binary code |
|----------------------------|-------------|
| 1 8 1 1 31H 38H 31H 31H | 11H 18H |

■Sub command

| Item | ASCII code | Binary code |
|------------------------|----------------------------|-------------|
| For MELSEC-Q/L series | 0 0 0 0 30H 30H 30H 30H | 00H 00H |
| For MELSEC iQ-R series | 0 0 4 0 30H 30H 34H 30H | 40H 00H |

■Fixed value 1

Specify the following fixed value.

- MELSEC iQ-R series (when using subcommand '0040'): Specify '0'.

| ASCII code | Binary code |
|----------------------------|-----------------|
| 0 0 0 0 30H 30H 30H 30H | 00H 00H 00H 00H |

- MELSEC-Q/L series (when using subcommand '0000'): Specify 20H in 4 bytes.

| ASCII code, binary code |
|-------------------------|
| 20H 20H 20H 20H |

■Drive No.

Specify the access target drive. (☞ Page 192 Drive No.)

■Fixed value 2

Specify '0'.

| ASCII code | Binary code |
|----------------------------|-------------|
| 0 0 0 0 30H 30H 30H 30H | 00H 00H |

■Number of file name characters, file name

Specify the file name of which file No. is to be read. (Refer to Page 196 File name specification)

Communication example (files for MELSEC-Q series)

Read the file No. in the following conditions.

- Password: 4 spaces (code: 20H)
- Drive No.: 0
- File name: ABC.QPG (file No.6)

■Data communication using ASCII code

(Request data)

| Subcommand | (Fixed value 1) | Drive No. | (Fixed value 2) | Number of file name characters | File name |
|----------------------------|----------------------------|------------------------------------|----------------------------|--------------------------------|--|
| 1 8 1 1 31H 38H 31H 31H | 0 0 0 0 30H 30H 30H 30H | 20H 20H 20H 20H 30H 30H 30H 30H | 0 0 0 0 30H 30H 30H 30H | 0 0 0 0 30H 30H 30H 30H | A B C . Q P G 41H 42H 43H 2EH 51H 50H 47H |

(Response data)

| File No. |
|----------------------------|
| 0 0 0 6 30H 30H 30H 36H |

■Data communication using binary code

(Request data)

| Subcommand | (Fixed value 1) | Drive No. | (Fixed value 2) | 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 (files for MELSEC iQ-R series)

Read the file No. in the following conditions.

- Drive No.: 4
- File No.: 6

The file name is as follows:

(1) LINE\LINE.CSV (13 characters)

| Item | Value of code corresponds to character | | | | | | | | | | | | | | |
|-------------|--|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--|--|
| File name | L | I | N | E | \ | L | I | N | E | . | C | S | V | | |
| UTF-16 | 004C | 0049 | 004E | 0045 | 005C | 004C | 0049 | 004E | 0045 | 002E | 0043 | 0053 | 0056 | | |
| ASCII code | 3030 3443 | 3030 3439 | 3030 3445 | 3030 3435 | 3030 3543 | 3030 3443 | 3030 3439 | 3030 3445 | 3030 3435 | 3030 3245 | 3030 3433 | 3030 3533 | 3030 3536 | | |
| Binary code | 4C00 | 4900 | 4E00 | 4500 | 5C00 | 4C00 | 4900 | 4E00 | 4500 | 2E00 | 4300 | 5300 | 5600 | | |

■Data communication using ASCII code

(Request data)

| Subcommand | (Fixed value 1) | Drive No. | (Fixed value 2) | 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 D 30H 30H 30H 44H |

In the figure (1), set the value of "ASCII code" in the table of "Value of code corresponds to character" in "File name".

(Response data)

| File No. |
|--|
| 0 0 0 0 0 0 0 6 30H 30H 30H 30H 30H 39H 36H |

■Data communication using binary code

(Request data)

| Subcommand | (Fixed value 1) | Drive No. | (Fixed value 2) | Number of file name characters | File name |
|------------------------------------|------------------------------------|-----------|-----------------|--------------------------------|-----------|
| 11H 18H 40H 00H 00H 00H 00H 00H | 00H 00H 04H 00H 00H 00H 0DH 00H | | | | (1) |

In the figure (1), set the value of "Binary code" in the table of "Value of code corresponds to character" in "File name".

(Response data)

| File No. |
|-----------------|
| 06H 00H 00H 00H |

12.5 File Creation and Deletion

Create a new file or delete a file.

Create new file (command: 1820)

Specify the size to create a new file.

A folder cannot be created with this command. Create a folder with an Engineering tool.



The time on a module is registered as the last edit date and time in a file created using this function.

Message format

The following shows the message format of the request data and response data of the command.

■Request data

| Command | Subcommand | Password | Drive No. | File size | Number of file name characters | File name |
|---------|------------|----------|-----------|-----------|--------------------------------|-----------|
|---------|------------|----------|-----------|-----------|--------------------------------|-----------|

■Response data

There is no response data for this command.

Data specified by request data

■Command

| ASCII code | Binary code |
|----------------------------|-------------|
| 1 8 2 0 31H 38H 32H 30H | 20H 18H |

■Sub command

| Item | ASCII code | Binary code |
|------------------------|----------------------------|-------------|
| For MELSEC-Q/L series | 0 0 0 0 30H 30H 30H 30H | 00H 00H |
| For MELSEC iQ-R series | 0 0 4 0 30H 30H 34H 30H | 40H 00H |

■Password

Specify the following fixed value.

- MELSEC iQ-R series (when using subcommand '0040'): Specify '0'.

| ASCII code | Binary code |
|----------------------------|-----------------|
| 0 0 0 0 30H 30H 30H 30H | 00H 00H 00H 00H |

- MELSEC-Q/L series (when using subcommand '0000'): Specify 20H in 4 bytes.

| ASCII code, binary code |
|-------------------------|
| 20H 20H 20H 20H |

■Drive No.

Specify the access target drive. (☞ Page 192 Drive No.)

■File size

Specify the file capacity in byte unit. (☞ Page 201 File size)

■Number of file name characters, file name

Specify the file name to be created. (☞ Page 196 File name specification)

Communication example (files for MELSEC-Q series)

Create a new file in the following conditions:

- Password: 4 spaces (code: 20H)
- Drive No.: 0
- File name: ABC.CSV
- File size: 1K byte

■Data communication using ASCII code

(Request data)

| Subcommand | Password | Drive No. | File size |
|----------------------------|----------------------------|----------------------------|--|
| 1 8 2 0 31H 38H 32H 30H | 0 0 0 0 30H 30H 30H 30H | 0 0 0 0 20H 20H 20H 20H | 0 0 0 0 4 0 0 0 0 30H 30H 30H 30H 34H 30H 30H 30H |

| Number of file name characters | | File name |
|--------------------------------|--|-----------|
| 0 0 0 7 30H 30H 30H 37H | A B C . C S V 41H 42H 43H 2EH 43H 53H 56H | |

■Data communication using binary code

(Request data)

| Subcommand | Password | Drive No. | File size | Number of file name characters | File name |
|--------------------|------------------------------------|----------------------------|--|--------------------------------|-----------|
| 20H 18H 00H 00H | 20H 20H 20H 20H 00H 00H 00H 00H | 00H 40H 00H 00H 07H 00H | A B C . C S V 41H 42H 43H 2EH 43H 53H 56H | | |

Communication example (files for MELSEC iQ-R series)

Create a new file in the following conditions:

- Drive No.: 4
- File size: 7168 byte
- File name: LINE.CSV (8 characters)

The value of the file name is as follows:

(1) LINE.CSV (8 characters)

| Item | Value of code corresponds to character | | | | | | | |
|-------------|--|----------|----------|----------|----------|----------|----------|----------|
| File name | L | I | N | E | . | C | S | V |
| UTF-16 | 004C | 0049 | 004E | 0045 | 002E | 0043 | 0053 | 0056 |
| ASCII code | 30303443 | 30303439 | 30303445 | 30303435 | 30303245 | 30303433 | 30303533 | 30303536 |
| Binary code | 4C00 | 4900 | 4E00 | 4500 | 2E00 | 4300 | 5300 | 5600 |

■Data communication using ASCII code

(Request data)

| Subcommand | (Fixed value) | Drive No. | File size | Number of file name characters | File name |
|-------------------------------|-------------------------------|-------------------------------|-------------------------------|---|-----------------------------------|
| 1 8 2 0 31H, 38H, 32H, 30H | 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 1 C 0 0 30H, 30H, 30H, 30H, 31H, 43H, 30H, 30H | 0 0 0 8 30H, 30H, 30H, 38H (1) |

In the figure (1), set the value of "ASCII code" in the table of "Value of code corresponds to character" in "File name".

■Data communication using binary code

(Request data)

| Subcommand | Drive No. | File size | Number of file name characters | File name |
|----------------------|--------------------------------|--------------------------------|--------------------------------|-----------|
| 20H, 18H 40H, 00H | 00H, 00H, 00H, 00H 04H, 00H | 00H, 1CH, 00H, 00H 08H, 00H | | (1) |

In the figure (1), set the value of "Binary code" in the table of "Value of code corresponds to character" in "File name".

Delete file (command: 1822)

Delete a file.



If files are deleted while a programmable controller system is in operation, the system may be stopped. Determine the timing of file deletion for the entire programmable controller system.



- The files on which the 'open file' command is being executed cannot be deleted. Close the file before deleting it. ([Page 230 Open file \(command: 1827\)](#), [Page 238 Close file \(command: 182A\)](#))
- The file being executed cannot be deleted when CPU module is state of RUN. Delete the file after placing CPU module to the STOP state. ([Page 457 Applicable Commands for Online Program Change](#))

Message format

The following shows the message format of the request data and response data of the command.

■Request data

| Command | Subcommand | Password | Drive No. | Number of file name characters | File name |
|---------|------------|----------|-----------|--------------------------------|-----------|
|---------|------------|----------|-----------|--------------------------------|-----------|

■Response data

There is no response data for this command.

Data specified by request data

■Command

| ASCII code | Binary code | | | | | | | | | | |
|---|-------------|-----|-----|---|-----|-----|-----|-----|--|-----|-----|
| <table border="1"> <tr> <td>1</td> <td>8</td> <td>2</td> <td>2</td> </tr> <tr> <td>31H</td> <td>38H</td> <td>32H</td> <td>32H</td> </tr> </table> | 1 | 8 | 2 | 2 | 31H | 38H | 32H | 32H | <table border="1"> <tr> <td>22H</td> <td>18H</td> </tr> </table> | 22H | 18H |
| 1 | 8 | 2 | 2 | | | | | | | | |
| 31H | 38H | 32H | 32H | | | | | | | | |
| 22H | 18H | | | | | | | | | | |

■Sub command

| Item | ASCII code | Binary code | | | | | | | | | | |
|------------------------|---|-------------|-----|---|---|-----|-----|-----|-----|--|-----|-----|
| For MELSEC-Q series | <table border="1"> <tr> <td>0</td> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>30H</td> <td>30H</td> <td>30H</td> <td>30H</td> </tr> </table> | 0 | 0 | 0 | 0 | 30H | 30H | 30H | 30H | <table border="1"> <tr> <td>00H</td> <td>00H</td> </tr> </table> | 00H | 00H |
| 0 | 0 | 0 | 0 | | | | | | | | | |
| 30H | 30H | 30H | 30H | | | | | | | | | |
| 00H | 00H | | | | | | | | | | | |
| For MELSEC-L series | <table border="1"> <tr> <td>0</td> <td>0</td> <td>0</td> <td>4</td> </tr> <tr> <td>30H</td> <td>30H</td> <td>30H</td> <td>34H</td> </tr> </table> | 0 | 0 | 0 | 4 | 30H | 30H | 30H | 34H | <table border="1"> <tr> <td>04H</td> <td>00H</td> </tr> </table> | 04H | 00H |
| 0 | 0 | 0 | 4 | | | | | | | | | |
| 30H | 30H | 30H | 34H | | | | | | | | | |
| 04H | 00H | | | | | | | | | | | |
| For MELSEC iQ-R series | <table border="1"> <tr> <td>0</td> <td>0</td> <td>4</td> <td>0</td> </tr> <tr> <td>30H</td> <td>30H</td> <td>34H</td> <td>30H</td> </tr> </table> | 0 | 0 | 4 | 0 | 30H | 30H | 34H | 30H | <table border="1"> <tr> <td>40H</td> <td>00H</td> </tr> </table> | 40H | 00H |
| 0 | 0 | 4 | 0 | | | | | | | | | |
| 30H | 30H | 34H | 30H | | | | | | | | | |
| 40H | 00H | | | | | | | | | | | |

■Password

Specify the password of the file. ([Page 189 Password](#))

■Drive No.

Specify the access target drive. ([Page 192 Drive No.](#))

■Number of file name characters, file name

Specify the file name to be deleted. (参照 Page 196 File name specification)

Communication example (files for MELSEC-Q series)

Delete the file under the following conditions:

- Password: 1234
- Drive No.: 0
- File name: ABC.QPG

■Data communication using ASCII code

(Request data)

| Subcommand | Password | Drive No. | Number of file name characters | File name |
|-------------------------------|-------------------------------|-------------------------------|--------------------------------|--|
| 1 8 2 2 31H, 38H, 32H, 32H | 0 0 0 0 30H, 30H, 30H, 30H | 1 2 3 4 31H, 32H, 33H, 34H | 0 0 0 0 30H, 30H, 30H, 30H | 0 0 0 7 30H, 30H, 30H, 37H A B C . Q P G 41H, 42H, 43H, 2EH, 51H, 50H, 47H |

■Data communication using binary code

(Request data)

| Subcommand | Number of file name characters | Drive No. | File name |
|----------------------|--------------------------------|----------------------|--|
| 22H, 18H 00H, 00H | 1 2 3 4 31H, 32H, 33H, 34H | 00H, 00H 07H, 00H | A B C . Q P G 41H, 42H, 43H, 2EH, 51H, 50H, 47H |

Communication example (files for MELSEC-L series)

Delete the file under the following conditions:

- Password: AbCd1234□...□ (24 spaces, code: 20H)
- Drive No.: 0
- File name:MAIN.QPG

■Data communication using ASCII code

(Request data)

| Subcommand | Password (fixed to 32 characters) | Drive No. |
|-------------------------------|--|-------------------------------|
| 1 8 2 2 31H, 38H, 32H, 32H | 0 0 0 4 30H, 30H, 30H, 34H A b C d 1 2 3 4 41H, 62H, 43H, 64H, 31H, 32H, 33H, 34H, 20H, ..., 20H | 0 0 0 0 30H, 30H, 30H, 30H |

| Number of file name characters | File name |
|--------------------------------|---|
| 0 0 0 8 30H, 30H, 30H, 38H | M A I N . Q P G 4DH, 41H, 49H, 4EH, 2EH, 51H, 50H, 47H |

■Data communication using binary code

(Request data)

| Subcommand | Number of file name characters | Drive No. | File name |
|----------------------|--|----------------------|---|
| 22H, 18H 04H, 00H | A b C d 1 2 3 4 41H, 62H, 43H, 64H, 31H, 32H, 33H, 34H, 20H, ..., 20H | 00H, 00H 08H, 00H | M A I N . Q P G 4DH, 41H, 49H, 4EH, 2EH, 51H, 50H, 47H |

Communication example (files for MELSEC iQ-R series)

Delete the file under the following conditions:

- Password: A to Z (26 characters)
- Drive No.: 4

The file name is as follows:

(1) LINE.CSV (8 characters)

| Item | Value of code corresponds to character | | | | | | | | |
|-------------|--|----------|----------|----------|----------|----------|----------|----------|--|
| File name | L | I | N | E | . | C | S | V | |
| UTF-16 | 004C | 0049 | 004E | 0045 | 002E | 0043 | 0053 | 0056 | |
| ASCII code | 30303443 | 30303439 | 30303445 | 30303435 | 30303245 | 30303433 | 30303533 | 30303536 | |
| Binary code | 4C00 | 4900 | 4E00 | 4500 | 2E00 | 4300 | 5300 | 5600 | |

■Data communication using ASCII code

(Request data)

| Subcommand | Number of password characters | Password |
|-------------------------------|-------------------------------|--|
| 1 8 2 2 31H, 38H, 32H, 32H | 0 0 4 0 30H, 30H, 34H, 30H | 0 0 1 A 30H, 30H, 31H, 41H A B C D ⋯ Z 41H, 42H, 43H, 44H, 54H |
| ----- | ----- | ----- |

| Drive No. | Number of file name characters | File name |
|-----------|--------------------------------|-------------------------------|
| ----- | 0 0 0 4 30H, 30H, 30H, 34H | 0 0 0 8 30H, 30H, 30H, 38H |
| ----- | ----- | (1) |

In the figure (1), set the value of "ASCII code" in the table of "Value of code corresponds to character" in "File name".

■Data communication using binary code

(Request data)

| Subcommand | Number of password characters | Password | Drive No. | Number of file name characters | File name |
|----------------------|-------------------------------|--|----------------------|--------------------------------|-----------|
| 22H, 18H 40H, 00H | 1AH, 00H | A B C D ⋯ Z 41H, 42H, 43H, 44H, 5AH | 04H, 00H 0DH, 00H | 0 0 0 8 30H, 30H, 30H, 38H | (1) |

In the figure (1), set the value of "Binary code" in the table of "Value of code corresponds to character" in "File name".

Copy file (command: 1824)

Copy a file.

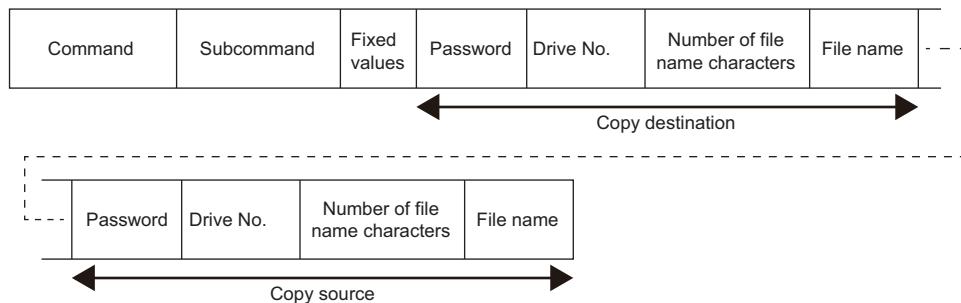


When this command is executed to files of parameter and a program being executed, place CPU module in the STOP status. ([Page 457 Commands that cannot be executed during RUN](#))

Message format

The following shows the message format of the request data and response data of the command.

■Request data



■Response data

There is no response data for this command.

Data specified by request data

■Command

| ASCII code | Binary code |
|----------------------------|-------------|
| 1 8 2 4 31H 38H 32H 34H | 24H 18H |

■Sub command

| Item | ASCII code | Binary code |
|------------------------|----------------------------|-------------|
| For MELSEC-Q series | 0 0 0 0 30H 30H 30H 30H | 00H 00H |
| For MELSEC-L series | 0 0 0 4 30H 30H 30H 34H | 04H 00H |
| For MELSEC iQ-R series | 0 0 4 0 30H 30H 34H 30H | 40H 00H |

■Fixed value

Specify '0'.

| ASCII code | Binary code |
|--|--------------------------------------|
| 30H 30H 30H 30H ... 30H (16 characters) | 00H 00H 00H 00H ... 00H (8 bytes) |

■Password

Specify the password of the access target file. (☞ Page 189 Password)

■Drive No.

Specify the access target drive. (☞ Page 192 Drive No.)



Program memory (drive No.0) of RCPU cannot be specified.

■Number of file name characters, file name

Specify the file name to be copied. (☞ Page 196 File name specification)

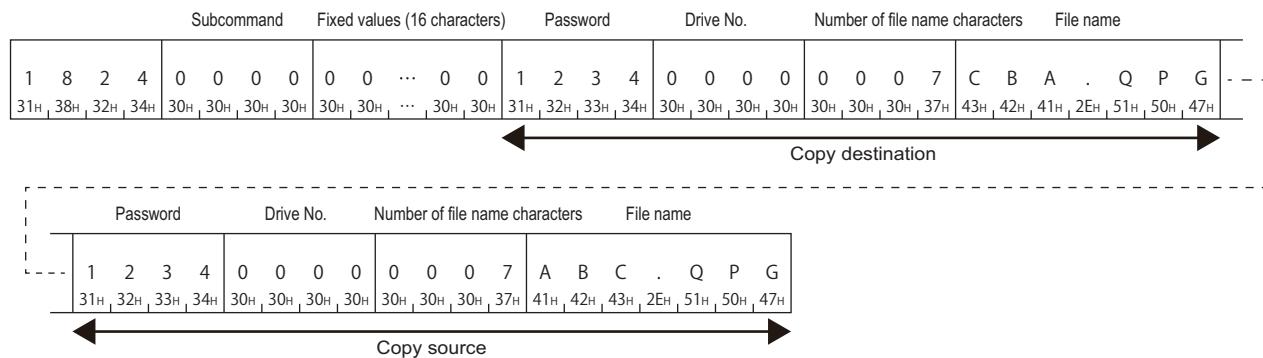
Communication example (files for MELSEC-Q series)

Copy the file under the following conditions:

- Copy source/destination password: 1234
- Drive No. of copy source, drive No. of copy destination: 0
- Copy source file name: ABC.QPG
- Copy source file name: CBA.QPG

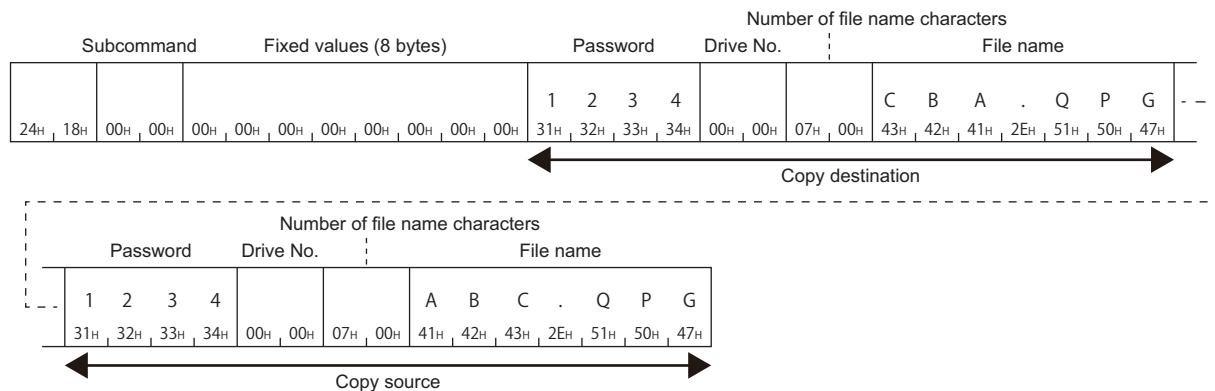
■Data communication using ASCII code

(Request data)



■Data communication using binary code

(Request data)



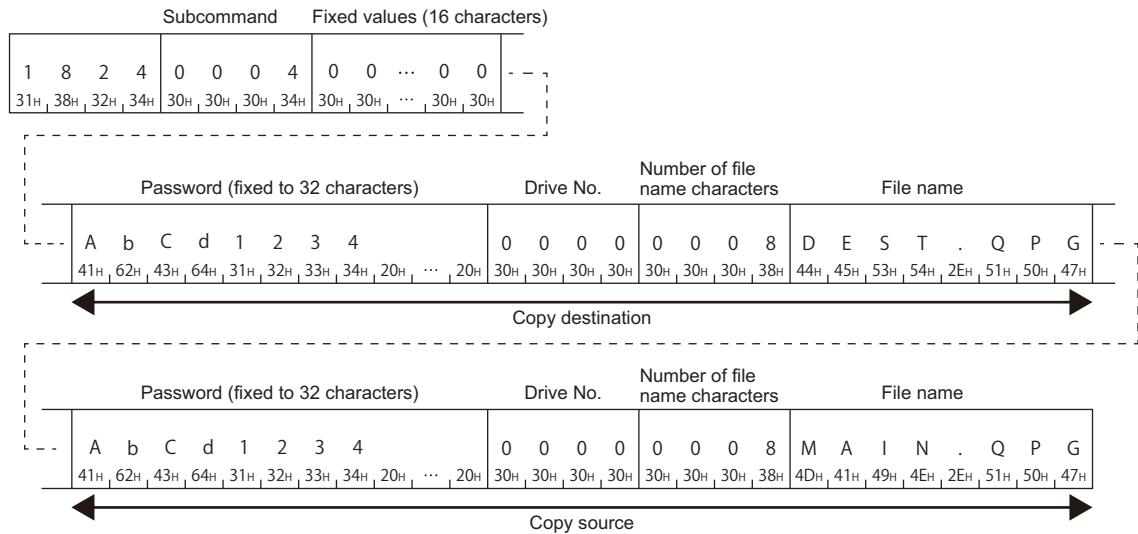
Communication example (files for MELSEC-L series)

Copy the file under the following conditions:

- Password copy source/destination AbCd1234□...□ (24 spaces, code: 20H)
- Drive No. of copy source, drive No. of copy destination: 0
- Copy source file name: MAIN.QPG
- Copy source file name: DEST.QPG

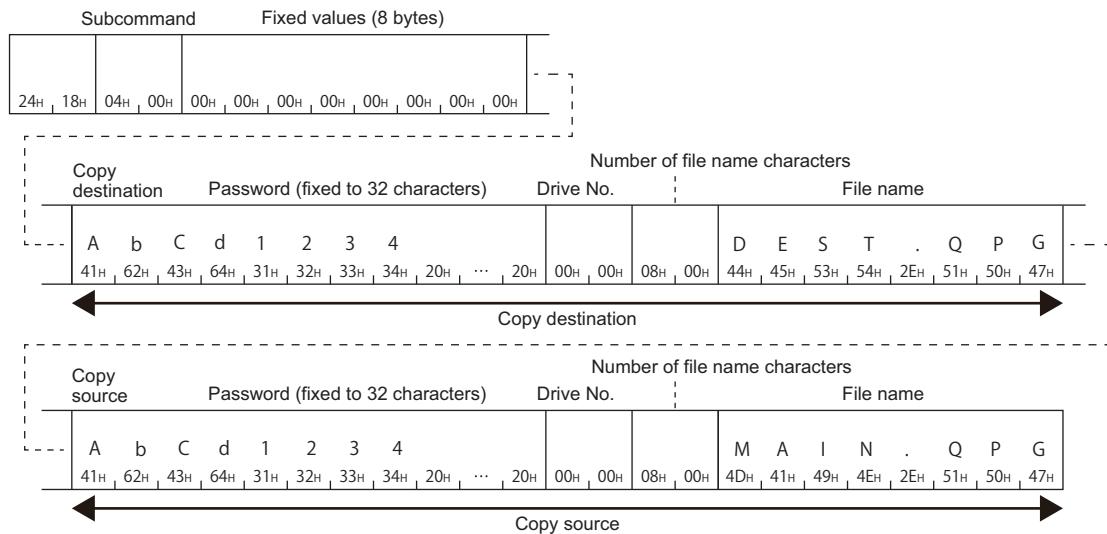
■Data communication using ASCII code

(Request data)



■Data communication using binary code

(Request data)



Communication example (files for MELSEC iQ-R series)

Copy the file under the following conditions:

- Drive No. of copy source: 2
- Drive No. of copy destination: 4

The file name of the copy source and copy destination is as follows.

(1) LINE.CSV (8 characters)

| Item | Value of code corresponds to character | | | | | | | |
|-------------|--|----------|----------|----------|----------|----------|----------|----------|
| File name | L | I | N | E | . | C | S | V |
| UTF-16 | 004C | 0049 | 004E | 0045 | 002E | 0043 | 0053 | 0056 |
| ASCII code | 30303443 | 30303439 | 30303445 | 30303435 | 30303245 | 30303433 | 30303533 | 30303536 |
| Binary code | 4C00 | 4900 | 4E00 | 4500 | 2E00 | 4300 | 5300 | 5600 |

■Data communication using ASCII code

(Request data)

| Subcommand | | Fixed values (16 characters) | | | | | | | | | | | | | |
|------------|---|------------------------------|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 8 | 2 | 4 | 0 | 0 | 0 | 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Number of copy destination password characters | Copy destination password | Copy destination drive No. | Number of copy destination file name characters | Copy destination file name |
|--|---------------------------|----------------------------|---|----------------------------|
| 0 0 1 A | A B C D ⋯ Z | 0 0 0 4 | 0 0 0 8 | (1) |

| Number of copy source password characters | Copy source password | Copy source drive No. | Number of copy source file name characters | Copy source file name |
|---|----------------------|-----------------------|--|-----------------------|
| 0 0 1 A | A B C D ⋯ Z | 0 0 0 2 | 0 0 0 8 | (1) |

In the figure (1), set the value of "ASCII code" in the table of "Value of code corresponds to character" in "File name".

■Data communication using binary code

(Request data)

| Subcommand | | Fixed values (8 bytes) | | | | | | | | Number of copy destination password characters | Copy destination password | Copy destination drive No. | Number of copy destination file name characters | Copy destination file name | |
|------------|-----|------------------------|-----|-----|-----|-----|-----|-----|-----|--|---------------------------|----------------------------|---|----------------------------|-----|
| 24H | 18H | 04H | 00H | 1AH | 00H | 41H | 42H | 43H | 44H |

| Number of copy source password characters | Copy source password | Copy source drive No. | Number of copy source file name characters | Copy source file name |
|---|----------------------|-----------------------|--|-----------------------|
| 1AH | 00H | 41H | 42H | (1) |

In the figure (1), set the value of "Binary code" in the table of "Value of code corresponds to character" in "File name".

12.6 File Modification

Read or write data from/to the specified file. Lock the file with the open/close file command during reading or writing so that the file contents will not be changed from other devices. The file is not required to be locked with the open file command when commands other than read/write file command are executed.

When modifying the file attributes and the last edit date and time, executing the open command is not required.

Modify file attribute (command: 1825)

Change the file attributes (read only/writable).



When this command is executed to files of parameter and a program being executed, place CPU module in the STOP status. (☞ Page 457 Commands that cannot be executed during RUN)

Message format

The following shows the message format of the request data and response data of the command.

■Request data

| Command | Subcommand | Password | Drive No. | Attribute | Number of file name characters | File name |
|---------|------------|----------|-----------|-----------|--------------------------------|-----------|
|---------|------------|----------|-----------|-----------|--------------------------------|-----------|

■Response data

There is no response data for this command.

Data specified by request data

■Command

| ASCII code | Binary code |
|----------------------------|-------------|
| 1 8 2 5 31H 38H 32H 35H | 25H 18H |

■Sub command

| Item | ASCII code | Binary code |
|------------------------|----------------------------|-------------|
| For MELSEC-Q series | 0 0 0 0 30H 30H 30H 30H | 00H 00H |
| For MELSEC-L series | 0 0 0 4 30H 30H 30H 34H | 04H 00H |
| For MELSEC iQ-R series | 0 0 4 0 30H 30H 34H 30H | 40H 00H |

■Password

Specify the password of the access target file. (☞ Page 189 Password)

■Drive No.

Specify the access target drive. (☞ Page 192 Drive No.)

■Attribute

Specify the file attribute. (☞ Page 199 Attribute)

- Read only: 01H
- Readable, writable: 20H

Do not specify the value other than above since the values are reserved for system use.

■Number of file name characters, file name

Specify the file name to modify attribute. (☞ Page 196 File name specification)

Communication example (files for MELSEC-Q series)

Change the file attribute in the following conditions.

- Password: 1234
- Drive No.: 0
- File name: ABC.QPG
- Attribute: Read only: 01H

■Data communication using ASCII code

(Request data)

| Subcommand | Password | Drive No. | Attribute | Number of file name characters | File name |
|----------------------------------|----------------------------------|----------------------------------|----------------------------------|----------------------------------|---|
| 1 8 2 5 31H , 38H , 32H , 35H | 0 0 0 0 30H , 30H , 30H , 30H | 1 2 3 4 31H , 32H , 33H , 34H | 0 0 0 0 30H , 30H , 30H , 30H | 0 0 0 1 30H , 30H , 30H , 31H | 0 0 0 7 30H , 30H , 30H , 37H A B C . Q P G 41H , 42H , 43H , 2EH , 51H , 50H , 47H |

■Data communication using binary code

(Request data)

| Subcommand | Password | Drive No. | Attribute | Number of file name characters | File name |
|------------------------|----------------------------------|------------------------|------------------------|--|---------------|
| 25H , 18H 00H , 00H | 1 2 3 4 31H , 32H , 33H , 34H | 00H , 00H 00H , 00H | 01H , 00H 01H , 00H | 07H , 00H 41H , 42H , 43H , 2EH , 51H , 50H , 47H | A B C . Q P G |

Communication example (files for MELSEC-L series)

Change the file attribute in the following conditions.

- Password: AbCd1234□...□(24 spaces, code: 20H)
- Drive No.: 0
- File name: MAIN.QPG
- Attribute: Read only: 01H

■Data communication using ASCII code

(Request data)

| Subcommand | Password (fixed to 32 characters) | Drive No. | Attribute |
|--|--|----------------------------------|----------------------------------|
| 1 8 2 5 31H , 38H , 32H , 35H | 0 0 0 4 A b C d 1 2 3 4 30H , 30H , 30H , 34H 41H , 62H , 43H , 64H , 31H , 32H , 33H , 34H , 20H , ... , 20H | 0 0 0 0 30H , 30H , 30H , 30H | 0 0 0 1 30H , 30H , 30H , 31H |
| Number of file name characters | | | File name |
| 0 0 0 8 M A I N . Q P G 30H , 30H , 30H , 38H 4Dh , 41H , 49H , 4EH , 2EH , 51H , 50H , 47H | | | |

■Data communication using binary code

(Request data)

| Subcommand | Password (fixed to 32 characters) | Drive No. | Attribute | Number of file name characters | File name |
|------------------------|--|------------------------|------------------------|--|-----------------|
| 25H , 18H 04H , 00H | A b C d 1 2 3 4 41H , 62H , 43H , 64H , 31H , 32H , 33H , 34H , 20H , ... , 20H | 00H , 00H 00H , 00H | 01H , 00H 01H , 00H | 08H , 00H 4DH , 41H , 49H , 4EH , 2EH , 51H , 50H , 47H | M A I N . Q P G |

Communication example (files for MELSEC iQ-R series)

Change the file attribute in the following conditions.

- Password: A to Z (26 characters)
- Drive No.: 4
- Attribute: Read only: 01H

The file name is as follows:

(1) LINE.CSV (8 characters)

| Item | Value of code corresponds to character | | | | | | | |
|-------------|--|----------|----------|----------|----------|----------|----------|----------|
| File name | L | I | N | E | . | C | S | V |
| UTF-16 | 004C | 0049 | 004E | 0045 | 002E | 0043 | 0053 | 0056 |
| ASCII code | 30303443 | 30303439 | 30303445 | 30303435 | 30303245 | 30303433 | 30303533 | 30303536 |
| Binary code | 4C00 | 4900 | 4E00 | 4500 | 2E00 | 4300 | 5300 | 5600 |

■Data communication using ASCII code

(Request data)

| Subcommand | | Number of password characters | | Password | | | | | | | | | | | | | | | |
|------------|-----|-------------------------------|-----|--------------------------------|-----|-----------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|--|
| 1 | 8 | 2 | 5 | 0 | 0 | 4 | 0 | 0 | 0 | 1 | A | A | B | C | D | ... | Z | | |
| <hr/> | | | | | | | | | | | | | | | | | | | |
| Drive No. | | Attribute | | Number of file name characters | | File name | | | | | | | | | | | | | |
| --- | --- | 0 | 0 | 0 | 4 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 8 | | | | | (1) | |
| 31H | 38H | 32H | 35H | 30H | 30H | 34H | 30H | 30H | 31H | 41H | 41H | 42H | 43H | 44H | 45H | 46H | 47H | 5AH | |

■Data communication using binary code

(Request data)

| Subcommand | | Number of password characters | | Password | | Drive No. | | Attribute | | Number of file name characters | | File name | | | | | | |
|------------|-----|-------------------------------|-----|----------|-----|-----------|-----|-----------|-----|--------------------------------|-----|-----------|-----|-----|-----|-----|-----|--|
| 25H | 18H | 40H | 00H | 1AH | 00H | 41H | 42H | 43H | 44H | 5AH | 04H | 00H | 01H | 00H | 08H | 00H | (1) | |

Modify file creation date and time (command: 1826)

Modify the file creation date and time.

Restriction

When this command is executed to files of parameter and a program being executed, place CPU module in the STOP status. (☞ Page 457 Commands that cannot be executed during RUN)

Message format

The following shows the message format of the request data and response data of the command.

■Request data

12

| | | | | | | | |
|---------|------------|--------------|-----------|----------------|----------------|--------------------------------|-----------|
| Command | Subcommand | Fixed values | Drive No. | Date to change | Time to change | Number of file name characters | File name |
|---------|------------|--------------|-----------|----------------|----------------|--------------------------------|-----------|

■Response data

There is no response data for this command.

Data specified by request data

■Command

| ASCII code | Binary code | | | | | | | | | | |
|--|-------------|-----|-----|---|-----|-----|-----|-----|---|-----|-----|
| <table border="1"><tr><td>1</td><td>8</td><td>2</td><td>6</td></tr><tr><td>31H</td><td>38H</td><td>32H</td><td>36H</td></tr></table> | 1 | 8 | 2 | 6 | 31H | 38H | 32H | 36H | <table border="1"><tr><td>26H</td><td>18H</td></tr></table> | 26H | 18H |
| 1 | 8 | 2 | 6 | | | | | | | | |
| 31H | 38H | 32H | 36H | | | | | | | | |
| 26H | 18H | | | | | | | | | | |

■Sub command

| Item | ASCII code | Binary code | | | | | | | | | | |
|------------------------|--|-------------|-----|---|---|-----|-----|-----|-----|---|-----|-----|
| For MELSEC-Q/L series | <table border="1"><tr><td>0</td><td>0</td><td>0</td><td>0</td></tr><tr><td>30H</td><td>30H</td><td>30H</td><td>30H</td></tr></table> | 0 | 0 | 0 | 0 | 30H | 30H | 30H | 30H | <table border="1"><tr><td>00H</td><td>00H</td></tr></table> | 00H | 00H |
| 0 | 0 | 0 | 0 | | | | | | | | | |
| 30H | 30H | 30H | 30H | | | | | | | | | |
| 00H | 00H | | | | | | | | | | | |
| For MELSEC iQ-R series | <table border="1"><tr><td>0</td><td>0</td><td>4</td><td>0</td></tr><tr><td>30H</td><td>30H</td><td>34H</td><td>30H</td></tr></table> | 0 | 0 | 4 | 0 | 30H | 30H | 34H | 30H | <table border="1"><tr><td>40H</td><td>00H</td></tr></table> | 40H | 00H |
| 0 | 0 | 4 | 0 | | | | | | | | | |
| 30H | 30H | 34H | 30H | | | | | | | | | |
| 40H | 00H | | | | | | | | | | | |

■Fixed value

Specify '0'.

| ASCII code | Binary code | | | | | | | | | | | | |
|--|-------------|-----|-----|---|-----|-----|-----|-----|---|-----|-----|-----|-----|
| <table border="1"><tr><td>0</td><td>0</td><td>0</td><td>0</td></tr><tr><td>30H</td><td>30H</td><td>30H</td><td>30H</td></tr></table> | 0 | 0 | 0 | 0 | 30H | 30H | 30H | 30H | <table border="1"><tr><td>00H</td><td>00H</td><td>00H</td><td>00H</td></tr></table> | 00H | 00H | 00H | 00H |
| 0 | 0 | 0 | 0 | | | | | | | | | | |
| 30H | 30H | 30H | 30H | | | | | | | | | | |
| 00H | 00H | 00H | 00H | | | | | | | | | | |

■Drive No.

Specify the access target drive. (☞ Page 192 Drive No.)

■Date to change, time to change

The file of the last edit date and time is modified with the specified date and time. (☞ Page 200 Creation date and time (last edit date and time))

■Number of file name characters, file name

Specify the file name to modify attribute. (☞ Page 196 File name specification)

Communication example (files for MELSEC-Q series)

Modify the date and time of file creation under the following conditions.

- Drive No.: 0
- Date to change: 2010/04/01
- Time to change: 20:50:58
- File name: ABC.QPG

■Data communication using ASCII code

(Request data)

| Subcommand | (Fixed value) | Drive No. | Date to change | Time to change | |
|----------------------------------|----------------------------------|--|----------------------------------|----------------------------------|----------------------------------|
| 1 8 2 6 31H , 38H , 32H , 36H | 0 0 0 0 30H , 30H , 30H , 30H | 0 0 0 0 30H , 30H , 30H , 30H | 0 0 0 0 30H , 30H , 30H , 30H | 3 C 8 1 33H , 43H , 38H , 31H | A 6 5 D 41H , 36H , 35H , 44H |
| <hr/> | | | | | |
| Number of file name characters | | File name | | | |
| 0 0 0 7 30H , 30H , 30H , 37H | | A B C . Q P G 41H , 42H , 43H , 2EH , 51H , 50H , 47H | | | |

■Data communication using binary code

(Request data)

| Subcommand | (Fixed value) | 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 | 00H , 00H 81H , 3CH | 5DH , A6H 07H , 00H | 07H , 00H 41H , 42H , 43H , 2EH , 51H , 50H , 47H | A B C . Q P G 41H , 42H , 43H , 2EH , 51H , 50H , 47H | |

Communication example (files for MELSEC iQ-R series)

Modify the date and time of file creation under the following conditions.

- Drive No.: 4
- Date to change: 2010/04/01
- Time to change: 20:50:58

The file name is as shown below.

(1) LINE.CSV (8 characters)

| Item | Value of code corresponds to character | | | | | | | |
|-------------|--|----------|----------|----------|----------|----------|----------|----------|
| File name | L | I | N | E | . | C | S | V |
| UTF-16 | 004C | 0049 | 004E | 0045 | 002E | 0043 | 0053 | 0056 |
| ASCII code | 30303443 | 30303439 | 30303445 | 30303435 | 30303245 | 30303433 | 30303533 | 30303536 |
| Binary code | 4C00 | 4900 | 4E00 | 4500 | 2E00 | 4300 | 5300 | 5600 |

■Data communication using ASCII code

(Request data)

| Subcommand | (Fixed value) | Drive No. | Date to change | Time to change |
|-----------------------------------|-------------------------------|-------------------------------|-------------------------------|-------------------------------|
| 1 8 2 6 31H, 38H, 32H, 36H | 0 0 4 0 30H, 30H, 34H, 30H | 0 0 0 0 30H, 30H, 30H, 30H | 0 0 0 4 30H, 30H, 30H, 34H | 3 C 8 1 33H, 43H, 38H, 31H |
| --- | | | | |
| Number of file name characters | | File name | | |
| 0 0 0 8 30H, 30H, 30H, 38H | | (1) | | |

■Data communication using binary code

(Request data)

| Subcommand | (Fixed value) | 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) |

Open file (command: 1827)

Open a file and lock the file so that the file contents are not modified from other devices.



The file is unlocked by any of the following.

- Close file (command: 182A)
- Restarting a module. (Resetting the CPU module, etc.)

Message format

The following shows the message format of the request data and response data of the command.

■Request data

| | | | | | | |
|---------|------------|----------|-----------|-----------|--------------------------------|-----------|
| Command | Subcommand | Password | Open mode | Drive No. | Number of file name characters | File name |
|---------|------------|----------|-----------|-----------|--------------------------------|-----------|

■Response data

The file pointer No. is stored. (☞ Page 202 File pointer No.)

Data specified by request data

■Command

| ASCII code | Binary code |
|----------------------------|-------------|
| 1 8 2 7 31H 38H 32H 37H | 27H 18H |

■Sub command

| Item | ASCII code | Binary code |
|------------------------|----------------------------|-------------|
| For MELSEC-Q series | 0 0 0 0 30H 30H 30H 30H | 00H 00H |
| For MELSEC-L series | 0 0 0 4 30H 30H 30H 34H | 04H 00H |
| For MELSEC iQ-R series | 0 0 4 0 30H 30H 34H 30H | 40H 00H |

■Password

Specify the password of the access target file. (☞ Page 189 Password)

■Open mode

Specify whether the specified file is open for reading or for writing. (☞ Page 204 Open mode)

- Open for read: 0000H
- Open for write: 0100H

■Drive No.

Specify the access target drive. (☞ Page 192 Drive No.)

■Number of file name characters, file name

Specify the file name to be open. (☞ Page 196 File name specification)

Communication example (files for MELSEC-Q series)

Open the file of QCPU in the following conditions.

- Password: 1234
- Drive No.: 0
- File name: ABC.QPG
- Open mode: Open for write

■Data communication using ASCII code

(Request data)

| Subcommand | Password | Open mode | Drive No. | Number of file name characters | File name |
|-------------------------------|-------------------------------|-------------------------------|-------------------------------|--------------------------------|---|
| 1 8 2 7 31H, 38H, 32H, 37H | 0 0 0 0 30H, 30H, 30H, 30H | 1 2 3 4 31H, 32H, 33H, 34H | 0 1 0 0 30H, 31H, 30H, 30H | 0 0 0 0 30H, 30H, 30H, 30H | 0 0 0 7 A B C . Q P G 41H, 42H, 43H, 2EH, 51H, 50H, 47H |

(Response data)

File pointer No.

| |
|-------------------------------|
| 0 0 0 0 30H, 30H, 30H, 30H |
|-------------------------------|

■Data communication using binary code

(Request data)

| Subcommand | Password | Open mode | Drive No. | Number of file name characters | File name |
|----------------------|--------------------------------|----------------------|--|--------------------------------|-----------|
| 27H, 18H 00H, 00H | 31H, 32H, 33H, 34H 00H, 01H | 00H, 00H 07H, 00H | A B C . Q P G 41H, 42H, 43H, 2EH, 51H, 50H, 47H | | |

(Response data)

File

pointer No.

| |
|----------|
| 00H, 00H |
|----------|

Communication example (files for MELSEC-L series)

Open the file of LCPU in the following condition.

- Password: AbCd1234□...□ (24 spaces, code: 20H)
- Drive No.: 0
- File name: MAIN.QPG
- Open mode: Open for write

■Data communication using ASCII code

(Request data)

| Subcommand | Password (fixed to 32 characters) | Open mode | Drive No. |
|---|-----------------------------------|-----------------|-----------|
| 1 8 2 7 | 0 0 0 4 A b C d 1 2 3 4 | 0 1 0 0 | 0 0 0 0 |
| 31H, 38H, 32H, 37H 30H, 30H, 34H 41H, 62H, 43H, 64H, 31H, 32H, 33H, 34H, 20H, ..., 20H 30H, 31H, 30H, 30H, 30H, 30H, 30H, 30H | | | |
| Number of file name characters | | File name | |
| 0 0 0 8 | | M A I N . Q P G | |
| 30H, 30H, 30H, 38H 4D _H , 41H, 49H, 4E _H , 2E _H , 51H, 50H, 47H | | | |

(Response data)

| File pointer No. |
|-------------------------------|
| 0 0 0 0 30H, 30H, 30H, 30H |

■Data communication using binary code

(Request data)

| Subcommand | Password (fixed to 32 characters) | Open mode | Drive No. | Number of file name characters | File name |
|-------------------|--|-----------|-----------|--------------------------------|--|
| 27H, 18H 04H, 00H | A b C d 1 2 3 4 41H, 62H, 43H, 64H, 31H, 32H, 33H, 34H, 20H, ..., 20H | 00H, 01H | 00H, 00H | 08H, 00H | M A I N . Q P G 4D _H , 41H, 49H, 4E _H , 2E _H , 51H, 50H, 47H |

(Response data)

| File pointer No. |
|------------------|
| 00H, 00H |

Communication example (files for MELSEC iQ-R series)

Open a file of MELSEC iQ-R series CPU module.

- Password: A to Z (26 characters)
- Drive No.: 4
- Open mode: Open for write

The file name is as follows:

(1) LINE.CSV (8 characters)

| Item | Value of code corresponds to character | | | | | | | |
|-------------|--|----------|----------|----------|----------|----------|----------|----------|
| File name | L | I | N | E | . | C | S | V |
| UTF-16 | 004C | 0049 | 004E | 0045 | 002E | 0043 | 0053 | 0056 |
| ASCII code | 30303443 | 30303439 | 30303445 | 30303435 | 30303245 | 30303433 | 30303533 | 30303536 |
| Binary code | 4C00 | 4900 | 4E00 | 4500 | 2E00 | 4300 | 5300 | 5600 |

■Data communication using ASCII code

(Request data)

| Subcommand | Number of password characters | Password |
|--|-------------------------------|--------------------------------|
| 1 8 2 7 31H, 38H, 32H, 37H | 0 0 4 0 30H, 30H, 34H, 30H | 0 0 1 A 30H, 30H, 31H, 41H |
| A B C D ... Z 41H, 42H, 43H, 44H, 5AH | | |
| Open mode | Drive No. | Number of file name characters |
| 0 1 0 0 30H, 31H, 30H, 30H | 0 0 0 4 30H, 30H, 30H, 34H | 0 0 0 8 30H, 30H, 30H, 38H |
| (1) | | |

(Response data)

| File pointer No. |
|-------------------------------|
| 0 0 0 0 30H, 30H, 30H, 30H |

■Data communication using binary code

(Request data)

| Subcommand | Number of password characters | Password | Open mode | Drive No. | Number of file name characters | File name |
|------------------------------|--|------------------------------|-----------|-----------|--------------------------------|-----------|
| 27H, 18H, 40H, 00H, 1AH, 00H | A B C D ... Z 41H, 42H, 43H, 44H, 34H | 00H, 01H, 04H, 00H, 08H, 00H | | | | (1) |

(Response data)

| File pointer No. |
|------------------|
| 00H, 00H |

Read file (command: 1828)

Read the content of a file.

Use the open/close command in order to prohibit access from other devices when this command is used.

Message format

The following shows the message format of the request data and response data of the command.

■Request data

| | | | | |
|---------|------------|------------------|----------------|----------------------|
| Command | Subcommand | File pointer No. | Offset address | Number of bytes read |
|---------|------------|------------------|----------------|----------------------|

■Response data

The data which have been read is stored.

| | |
|----------------------|-----------|
| Number of bytes read | Read data |
|----------------------|-----------|

Data specified by request data

■Command

| ASCII code | Binary code |
|----------------------------|-------------|
| 1 8 2 8 31H 38H 32H 38H | 28H 18H |

■Sub command

| ASCII code | Binary code |
|----------------------------|-------------|
| 0 0 0 0 30H 30H 30H 30H | 00H 00H |

■File pointer No.

Specify the file pointer No. obtained with the 'open file' (command: 1827). ([Page 202 File pointer No.](#))

■Offset address

Specify the start address to start reading. ([Page 202 Offset address](#))

■Number of bytes read

Specify the number of bytes to be read from file in the range of 0 to 1920. ([Page 203 Number of bytes](#))

Specify it as one address/one byte.



When the file size is 1921 bytes or more, use an offset address and read data divide into multiple steps. The file size can be checked by reading directory/file information (command: 1810).

[Page 205 Read directory/file information \(command: 1810\)](#)

Data stored in response data

■Number of bytes read

The number of data bytes read from file is stored. ([Page 203 Number of bytes](#))

■Read data

The contents of the read file are stored.

Communication example

Read the file under the following conditions:

- File pointer No.: 0
- Number of bytes read: 1K bytes

■Data communication using ASCII code

(Request data)

| Subcommand | File pointer No. | Offset address | Number of bytes read |
|-------------------------------|-------------------------------|-------------------------------|---|
| 1 8 2 8 31H, 38H, 32H, 38H | 0 0 0 0 30H, 30H, 30H, 30H | 0 0 0 0 30H, 30H, 30H, 30H | 0 0 0 0 0 0 0 0 30H, 30H, 30H, 30H, 30H, 30H, 30H, 30H |

(Response data)

| Number of bytes read | Read data |
|-------------------------------|----------------------|
| 0 4 0 0 30H, 34H, 30H, 30H | (Contents of a file) |

■Data communication using binary code

(Request data)

| Subcommand | File pointer No. | Offset address | Number of bytes read |
|------------|------------------|----------------|------------------------------|
| 28H, 18H | 00H, 00H | 00H, 00H | 00H, 00H, 00H, 00H, 00H, 04H |

(Response data)

| Number of bytes read | Read data |
|----------------------|----------------------|
| 00H, 04H | (Contents of a file) |

Write to file (command: 1829)

Write contents to a file.

Use the open/close command in order to prohibit access from other devices when this command is used.

Restriction

When this command is executed to files of parameter and a program being executed, place CPU module in the STOP status. (☞ Page 457 Commands that cannot be executed during RUN)

Message format

The following shows the message format of the request data and response data of the command.

■Request data

| Command | Subcommand | File pointer No. | Offset address | Number of bytes written | Write data |
|---------|------------|------------------|----------------|-------------------------|------------|
|---------|------------|------------------|----------------|-------------------------|------------|

■Response data

The number of data bytes written in the file is stored. (☞ Page 203 Number of bytes)

Data specified by request data

■Command

| ASCII code | Binary code |
|----------------------------|-------------|
| 1 8 2 9 31H 38H 32H 39H | 29H 18H |

■Sub command

| ASCII code | Binary code |
|----------------------------|-------------|
| 0 0 0 0 30H 30H 30H 30H | 00H 00H |

■File pointer No.

Specify the file pointer No. obtained with the 'open file' (command: 1827). (☞ Page 202 File pointer No.)

■Offset address

Specify the start address to start writing. (☞ Page 202 Offset address)

- Writing data to a file of which drive name is '00H' (program memory): Specify the address in multiples of 4 (0, 4, 8,... in decimal notation).
- Writing data to a file of which drive name is other than '00H': Specify the addresses in even numbered addresses (0, 2, 4, 6, 8,... In decimal notation).

■Number of bytes written

Specify the number of bytes to be written to a file in the range of 0 to 1920. (☞ Page 203 Number of bytes)

Specify it as one address/one byte.

Point

Write data to the file within the file size reserved by creating new file.

When the file size is 1921 bytes or more, use an offset address and write data divide into multiple steps. The file size can be checked by reading directory/file information (command: 1810).

☞ Page 205 Read directory/file information (command: 1810)

■Write data

Specify the data to be written to a file.

Communication example

Write the file under the following conditions:

- File pointer No.: 0
- Offset address: 0
- Number of bytes written: 1K byte

■Data communication using ASCII code

(Request data)

| Subcommand | File pointer No. | Offset address | Number of bytes written | Write data |
|-------------------------------|-------------------------------|-------------------------------|---|---|
| 1 8 2 9 31H, 38H, 32H, 39H | 0 0 0 0 30H, 30H, 30H, 30H | 0 0 0 0 30H, 30H, 30H, 30H | 0 0 0 0 0 0 0 0 30H, 30H, 30H, 30H, 30H, 30H, 30H, 30H | 0 4 0 0 30H, 34H, 30H, 30H (Contents of a file) |

(Response data)

Number of bytes written

| |
|-------------------------------|
| 0 4 0 0 30H, 34H, 30H, 30H |
|-------------------------------|

■Data communication using binary code

(Request data)

| Subcommand | File pointer No. | Offset address | Number of bytes written | Write data |
|----------------------|----------------------|--|-------------------------|----------------------|
| 29H, 18H 00H, 00H | 00H, 00H 00H, 00H | 00H, 00H, 00H, 00H 00H, 00H, 00H, 00H | 00H, 04H | (Contents of a file) |

(Response data)

Number of bytes written

| |
|----------|
| 00H, 04H |
|----------|

Close file (command: 182A)

Close a file and unlock the file which has been locked by the 'open file' (command: 1827).

Message format

The following shows the message format of the request data and response data of the command.

■Request data

| | | | |
|---------|------------|------------------|------------|
| Command | Subcommand | File pointer No. | Close type |
|---------|------------|------------------|------------|

■Response data

There is no response data for this command.

Data specified by request data

■Command

| ASCII code | Binary code |
|----------------------------|-------------|
| 1 8 2 A 31H 38H 32H 41H | 2AH 18H |

■Sub command

| ASCII code | Binary code |
|----------------------------|-------------|
| 0 0 0 0 30H 30H 30H 30H | 00H 00H |

■File pointer No.

Specify the file pointer No. obtained with the 'open file' (command: 1827). ([Page 202 File pointer No.](#))

■Close type

Specify the file to be closed. ([Page 204 Close type](#))

When closing a file specified by file pointer, specify '0'.

| ASCII code | Binary code |
|----------------------------|-------------|
| 0 0 0 0 30H 30H 30H 30H | 00H 00H |

The locked files which have been opened by other devices cannot be unlocked.

Executing the command to files which have been opened by the other external device results in abnormal completion of the command.



Files are closed by restarting a module (such as resetting CPU module).

Communication example

Close the file under the following conditions:

- File pointer No.: 0
- Close type: 2

■Data communication using ASCII code

(Request data)

| Subcommand | File pointer No. | Close type | |
|----------------------------|----------------------------|----------------------------|----------------------------|
| 1 8 2 A 31H 38H 32H 41H | 0 0 0 0 30H 30H 30H 30H | 0 0 0 0 30H 30H 30H 30H | 0 0 0 2 30H 30H 30H 32H |

■Data communication using binary code

(Request data)

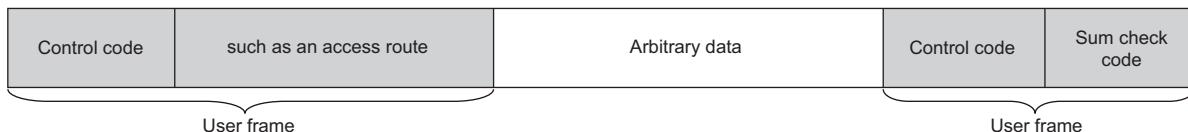
| File pointer No. | Subcommand | Close type |
|--------------------|--------------------|--------------------|
| 2AH 18H 00H 00H | 00H 00H 00H 00H | 02H 00H 02H 00H |

13 SERIAL COMMUNICATION MODULE DEDICATED COMMANDS

This chapter explains the dedicated commands for serial communication module.

13.1 User Frame

A user frame is a data name which is used to send/receive data by registering the fixed format portion in a message to be communicated between an external device and serial communication module.



Control code or sum check code in the message are registered as a default registration frame.

The data such as an access route can be registered to the user frame.

By using the user frame, the following data communication can be performed.

- Transmission of on-demand data by MC protocol
- Data communication by nonprocedural protocol

This section explains the commands for an external device to register, delete, and read user frames to C24. For transmission and reception of data using a user frame, refer to the following manual.

MELSEC-Q/L Serial Communication Module User's Manual (Application)

MELSEC iQ-R Serial Communication Module User's Manual (Application)

Restriction

The command can only be used for C24 (including multidrop connection station) connected to the external device. It cannot be used via network.

Point

This command is processed by the connected C24/E71 without waiting for the END processing by CPU module.

Data to be specified in command

This section explains the contents and specification methods for data items which are set in each command related to the user frame.

Frame No.

Specify the number of target user frame.

| Type | Setting value | Registration destination | Remarks |
|----------------------------|----------------|---|---------------------|
| Default registration frame | 1H to 3E7H | ROM for C24 operating system | Read only |
| User frame | 3E8H to 4AFH | Flash ROM of C24 | Read, write, delete |
| | 8001H to 801FH | Buffer memory of C24 (addresses 1B00H to 1FF6H) | Read, write, delete |

■Data communication in ASCII code

Convert the numerical value to 4-digit ASCII code (hexadecimal) and send it from upper digits.

■Data communication in binary code

Send 2-byte numerical values from lower byte (L: bits 0 to 7).

13

Ex.

For 3E8H

| ASCII code | Binary code |
|----------------------------|-------------|
| 0 3 E 8 30H 33H 45H 38H | E8 03H |

Number of registration data byte, number of frame byte

Specify the number of bytes of registration data.

| Function | Setting value |
|------------------------------|---------------------|
| Data registration, read data | 1H to 50H (1 to 80) |
| Deletion | 0H |

For the number of registration data byte, calculate the variable data as 2 bytes.

For the number of frame byte, calculate the variable data as 1 byte.

For the variable data, refer to the following manuals.

MELSEC-Q/L Serial Communication Module User's Manual (Application)

MELSEC iQ-R Serial Communication Module User's Manual (Application)

■Data communication in ASCII code

Convert the numerical value to 4-digit ASCII code (hexadecimal) and send it from upper digits.

■Data communication in binary code

Send 2-byte numerical values from lower byte (L: bits 0 to 7).

Ex.

When the registration data is ETX + variable data (sum check code) + CR + LF

Number of registration data byte = 1-byte (ETX) + 2-byte (variable data) + 1-byte (CR) + 1-byte (LF) = 5

Number of frame byte = 1-byte (ETX) + 1-byte (variable data) + 1-byte (CR) + 1-byte (LF) = 4

| Item | ASCII code | Binary code |
|----------------------------------|----------------------------|-------------|
| Number of registration data byte | 0 0 0 5 30H 30H 30H 35H | 05H 00H |
| Number of frame byte | 0 0 0 4 30H 30H 30H 34H | 04H 00H |

Registration data

The following shows the content of data in a user frame.

■Data communication in ASCII code

Convert the data to 2-digit (hexadecimal) ASCII code and send it from upper digit.

■Data communication in binary code

Send data from the first frame.

Ex.

When the registration data is ETX + Variable data (sum check code) + CR + LF

| ASCII code | Binary code | | | | | | | | | | | | | | | | |
|--|----------------------------------|------------------|------------------|------------------|----------------------------------|------------------|------------------|--|-----|----|----|-----|-----|-----|-----|-----|--|
| <p style="text-align: center;">Variable data (Sum check code)</p> <table border="1"><tr><td>ETX</td><td>CR</td><td>LF</td></tr><tr><td>0 3 30H 33H</td><td>F F F 1 46H 46H 46H 31H</td><td>0 D 30H 44H</td><td>0 A 30H 41H</td></tr></table> | ETX | CR | LF | 0 3 30H 33H | F F F 1 46H 46H 46H 31H | 0 D 30H 44H | 0 A 30H 41H | <p style="text-align: center;">Variable data (Sum check code)</p> <table border="1"><tr><td>ETX</td><td>CR</td><td>LF</td></tr><tr><td>03H</td><td>FFH</td><td>F0H</td></tr><tr><td>0DH</td><td>0AH</td><td></td></tr></table> | ETX | CR | LF | 03H | FFH | F0H | 0DH | 0AH | |
| ETX | CR | LF | | | | | | | | | | | | | | | |
| 0 3 30H 33H | F F F 1 46H 46H 46H 31H | 0 D 30H 44H | 0 A 30H 41H | | | | | | | | | | | | | | |
| ETX | CR | LF | | | | | | | | | | | | | | | |
| 03H | FFH | F0H | | | | | | | | | | | | | | | |
| 0DH | 0AH | | | | | | | | | | | | | | | | |

Read registered data (command: 0610)

Read the registered content of user frames.

Message format

The following shows the message format of the request data and response data of the command.

■Request data

| | | |
|---------|------------|-----------|
| Command | Subcommand | Frame No. |
|---------|------------|-----------|

■Response data

The read registration data is stored.

| | | |
|------------------------------|------------------|-------------------|
| Registration data byte count | Frame byte count | Registration data |
|------------------------------|------------------|-------------------|

13

Data specified by request data

■Command

| ASCII code | Binary code ^{*1} |
|-------------------------------|---------------------------|
| 0 6 1 0 30H, 36H, 31H, 30H | DLE 10H 10H 06H |

*1 For C24, an additional code is added. (☞ Page 33 Additional code (10H))

■Subcommand

| ASCII code | Binary code |
|-------------------------------|-------------|
| 0 0 0 0 30H, 30H, 30H, 30H | 00H, 00H |

■Frame No.

Specify the number of user frame to be read. (☞ Page 241 Frame No.)

| Type | Setting value | Registration destination |
|----------------------------|----------------|---|
| Default registration frame | 1H to 3E7H | ROM for C24 operating system |
| User frame | 3E8H to 4AFH | Flash ROM of C24 |
| | 8001H to 801FH | Buffer memory of C24 (addresses 1B00H to 1FF6H) |



When the frame No. whose user frame is not registered is specified, an error occurs and an abnormal response is returned.

Data to be stored by response data

■Number of registration data byte, number of frame byte

The data byte count of the registration data is stored. (☞ Page 241 Number of registration data byte, number of frame byte)

■Registration data

The data content of user frames to be registered is stored. (☞ Page 242 Registration data)

Example for data communication

Read the following registration data from frame No. 3E8H.

Registration data: ETX + Variable data (sum check code) + CR + LF

■Data communication in ASCII code

(Request data)

| Command | Subcommand | Frame No. |
|----------------------------|----------------------------|----------------------------|
| 0 6 1 0 30H 36H 31H 30H | 0 0 0 0 30H 30H 30H 30H | 0 3 E 8 30H 38H 45H 38H |

(Response data)

| Number of registration data byte | Number of frame byte | ETX | Variable data | CR | LF |
|----------------------------------|----------------------------|----------------|----------------------------|----------------|----------------|
| 0 0 0 5 30H 30H 30H 35H | 0 0 0 4 30H 30H 30H 34H | 0 3 30H 33H | F F F 1 46H 46H 46H 31H | 0 D 30H 44H | 0 A 30H 41H |

← Registration data →

■Data communication in binary code

(Request data)

| Subcommand | Frame No. |
|---------------------------|-----------|
| 10H, 10H, 06H 00H, 00H | E8H, 03H |

(Response data)

| Number of registration data byte | Number of frame byte | ETX | Variable data | CR | LF |
|----------------------------------|----------------------|-----|---------------|-----|-----|
| 05H, 00H | 04H, 00H | 03H | FFH, F0H | 0DH | 0AH |

← Registration data →

Register data (command: 1610)

Register user frames to C24.

Restriction

For MELSEC iQ-R series, the user frame is replaced with the contents registered using module extended parameters by powering OFF → ON or switching the CPU module STOP → RUN. For more information on user frame registration using module extended parameters, refer to the following manual.

 MELSEC iQ-R Serial Communication Module User's Manual (Application)

Point

When registering a user frame with the same frame No. again, deleted the user frame first and registered again. If it is attempted to register a user frame by specifying an already registered frame No., an error occurs and an abnormal response is returned.

13

Message format

The following shows the message format of the request data and response data of the command.

■Request data

| Command | Subcommand | Frame No. | Number of registration data byte | Number of frame byte | Registration data |
|---------|------------|-----------|----------------------------------|----------------------|-------------------|
| 1610 | | | | | |

■Response data

There is no response data for this command.

Data specified by request data

■Command

| ASCII code | Binary code *1 |
|----------------------------|-----------------|
| 1 6 1 0 31H 36H 31H 30H | DLE 10H 10H 16H |

*1 For C24, an additional code is added. ( Page 233 Additional code (10H))

■Subcommand

| ASCII code | Binary code |
|----------------------------|-------------|
| 0 0 0 0 30H 30H 30H 30H | 00H 00H |

■Frame No.

Specify the number of user frame to be registered. ( Page 241 Frame No.)

| Type | Setting value | Registration destination |
|------------|----------------|---|
| User frame | 3E8H to 4AFH | Flash ROM of C24 |
| | 8001H to 801FH | Buffer memory of C24 (addresses 1B00H to 1FF6H) |

■Number of registration data byte, number of frame byte

Specify the byte count of registration data within the range of 1 to 80. ( Page 241 Number of registration data byte, number of frame byte)

■Registration data

Store the content of user frame data to be registered. ( Page 242 Registration data)

Example for data communication

Register the following data in frame No. 3E8H.

Registration data: ETX + Variable data (sum check code) + CR + LF

■Data communication in ASCII code

(Request data)

| Command | Subcommand | Frame No. | Number of registration data byte | Number of frame byte | ETX | Variable data | CR | LF |
|----------------------------------|----------------------------------|----------------------------------|----------------------------------|----------------------------------|------------------|----------------------------------|------------------|------------------|
| 1 6 1 0 31H , 36H , 31H , 30H | 0 0 0 0 30H , 30H , 30H , 30H | 0 3 E 8 30H , 38H , 45H , 38H | 0 0 0 5 30H , 30H , 30H , 35H | 0 0 0 4 30H , 30H , 30H , 34H | 0 3 30H , 33H | F F F 1 46H , 46H , 46H , 31H | 0 D 30H , 44H | 0 A 30H , 41H |



Registration data

■Data communication in binary code

(Request data)

| Command | Subcommand | Frame No. | Number of registration data byte | Number of frame byte | ETX | Variable data | CR | LF |
|-----------------|------------|-----------|----------------------------------|----------------------|-----|---------------|-----|-----|
| 10H , 10H , 16H | 00H , 00H | E8H , 03H | 05H , 00H | 04H , 00H | 03H | FFH , F1H | 0DH | 0AH |



Registration data

Delete registered data (command: 1610)

Delete the registered user frame.

Message format

The following shows the message format of the request data and response data of the command.

■Request data

| | | | | |
|---------|------------|-----------|----------------------------------|----------------------|
| Command | Subcommand | Frame No. | Number of registration data byte | Number of frame byte |
|---------|------------|-----------|----------------------------------|----------------------|

■Response data

There is no response data for this command.

Data specified by request data

13

■Command

| ASCII code | Binary code ^{*1} |
|----------------------------|---------------------------|
| 1 6 1 0 31H 36H 31H 30H | DLE 10H 10H 16H |

*1 For C24, an additional code is added. ([Page 33 Additional code \(10H\)](#))

■Subcommand

| ASCII code | Binary code |
|----------------------------|-------------|
| 0 0 0 1 30H 30H 30H 31H | 01H 00H |

■Frame No.

Specify the number of user frame to be deleted. ([Page 241 Frame No.](#))

| Type | Setting value | Registration destination |
|------------|----------------|---|
| User frame | 3E8H to 4AFH | Flash ROM of C24 |
| | 8001H to 801FH | Buffer memory of C24 (addresses 1B00H to 1FF6H) |



When the frame No. whose user frame is not registered is specified, an error occurs and an abnormal response is returned.

■Number of registration data byte, number of frame byte

Specify '0'.

| ASCII code | Binary code |
|----------------------------|-------------|
| 0 0 0 0 30H 30H 30H 30H | 00H 00 |

Example for data communication

Delete the registration data of frame No. 3E8H.

■Data communication in ASCII code

(Request data)

| Command | Subcommand | Frame No. | Number of registration data byte | Number of frame byte |
|-------------------------------|-------------------------------|-------------------------------|----------------------------------|-------------------------------|
| 1 6 1 0 31H, 36H, 31H, 30H | 0 0 0 1 30H, 30H, 30H, 31H | 0 3 E 8 30H, 33H, 45H, 38H | 0 0 0 0 30H, 30H, 30H, 30H | 0 0 0 0 30H, 30H, 30H, 30H |

■Data communication in binary code

(Request data)

| Command | Subcommand | Frame No. | Number of registration data byte | Number of frame byte |
|---------------|------------|-----------|----------------------------------|----------------------|
| 10H, 10H, 16H | 01H, 00H | E8H, 03H | 00H, 00H | 00H, 00H |

13.2 Global Function

The global function is a function to turn ON/OFF the global signals (input signals: X1A/X1B) of serial communication module connected to external devices with the multidrop communications.

This is used as interlock signals for emergency command, simultaneous startup, and applicability of data transmission/reception to CPU modules.

Restriction

The command can only be used for C24 (including multidrop connection station) connected to the external device. It cannot be used via network.

Point

The global signal is turned OFF by powering OFF, resetting, or switching the mode of CPU module.

Compatibility with global function of Computer link module

This function (command: 1618) is compatible with GW command for dedicated protocol of Computer link module.

■ Operation of serial communication module when GW command is received

When GW command is executed from Computer link module connected with the multidrop connection, the global signal (input signal: X1A/X1B) at the side where the command was received turns ON/OFF.

- Received from CH1: X1A
- Received from CH2: X1B

■ Command: Operation of computer link module when 1618 is received

When this function (command: 1618) is used to Computer link module, the global signal (input signal: X2) of Computer link module turns ON/OFF.

Global signal ON/OFF (command: 1618)

Turn ON/OFF the global signal from an external device.

Message format

The following shows the message format of the request data and response data of the command.

■Request data

| | | |
|---------|------------|-----------------------------|
| Command | Subcommand | Global signal specification |
|---------|------------|-----------------------------|

■Response data

There is no response data for this command.

Data specified by request data

■Command

| ASCII code | Binary code |
|----------------------------|-------------|
| 1 6 1 8 31H 36H 31H 38H | 18H 16H |

■Subcommand

| Item | ASCII code | Binary code |
|------|----------------------------|-------------|
| OFF | 0 0 0 0 30H 30H 30H 30H | 00H 00H |
| ON | 0 0 0 1 30H 30H 30H 31H | 01H 00H |

■Global signal specification

Specify which global signal (X1A or X1B) is turned ON/OFF.

| Global signal to be turned ON/OFF | ASCII code | Binary code |
|--|----------------------------|-------------|
| Global signal at the side where command was received <ul style="list-style-type: none">• Received from CH1: X1A• Received from CH2: X1B | 0 0 0 0 30H 30H 30H 30H | 00H 00H |
| X1A regardless of the interface | 0 0 0 1 30H 30H 30H 31H | 01H 00H |
| X1B regardless of the interface | 0 0 0 2 30H 30H 30H 32H | 02H 00H |



The target station (all stations/only 1 specified station) of which global signal is turned ON/OFF is specified by a station No.

☞ Page 48 Station No.

Communication example (turn ON)

Turn the global signal X1A ON.

■Data communications in ASCII code (Format 1)

(Request data)

| Command | Subcommand | Global signal specification |
|----------------------------------|----------------------------------|----------------------------------|
| 1 6 1 8 31H , 36H , 31H , 38H | 0 0 0 1 30H , 30H , 30H , 31H | 0 0 0 1 30H , 30H , 30H , 31H |

■Data communications in binary code (Format 5)

(Request data)

| Command | Subcommand | Global signal specification |
|-----------|------------|-----------------------------|
| 18H , 16H | 01H , 00H | 01H , 00H |

Communication example (turn OFF)

Turn OFF the global signal X1A.

■Data communications in ASCII code (Format 1)

(Request data)

| Command | Subcommand | Global signal specification |
|----------------------------------|----------------------------------|----------------------------------|
| 1 6 1 8 31H , 36H , 31H , 38H | 0 0 0 0 30H , 30H , 30H , 30H | 0 0 0 1 30H , 30H , 30H , 31H |

■Data communications in binary code (Format 5)

(Request data)

| Command | Subcommand | Global signal specification |
|-----------|------------|-----------------------------|
| 18H , 16H | 00H , 00H | 01H , 00H |

13.3 Transmission sequence initialization function

The transmission sequence initialization function is a function to initialize the transmission sequence of data communication using format 5 of 4C frame, and to place C24 into the state where it waits to receive commands from external devices.

Restriction

The command can only be used for C24 (including multidrop connection station) connected to the external device. It cannot be used via network.

Initialize transmission sequence (command: 1615)

Initialize the transmission sequence of data communication using format 5 of 4C frame, and make C24 wait to receive commands from external devices.

Point

This function is equivalent to EOT, CL during data communication in ASCII code. Use the control codes, EOT, CL during data communication in ASCII code. (参照 Page 32 EOT(04H), CL(0CH))

Message format

The following shows the message format of the request data and response data of the command.

■Request data

| | |
|---------|------------|
| Command | Subcommand |
|---------|------------|

■Response data

There is no response data for this command.

Data specified by request data

■Command

| Binary code |
|-------------|
| 15H, 16H |

■Subcommand

| Binary code |
|-------------|
| 00H, 00H |

Example for data communication

Initialize transmission sequence.

(Request data)

| Subcommand |
|--------------------|
| 15H, 16H, 00H, 00H |

13.4 Mode Switching Function

The mode switching function is a function to switch the current communication protocol (operation mode) or transmission specification for the specified interface from external devices after C24 starts up. For more details on the mode switching function, refer to the following manuals.

- 📖 MELSEC-Q/L Serial Communication Module User's Manual (Application)
- 📖 MELSEC iQ-R Serial Communication Module User's Manual (Application)

Restriction

The command can only be used for C24 (including multidrop connection station) connected to the external device. It cannot be used via network.

Data to be specified in command

This section explains the contents and specification method for data item which is set in the command to switch mode.

Channel No.

Specify the target interface.

| Target interface | ASCII code | Binary code | | | | | |
|------------------|--|-------------|---|-----|-----|--|-----|
| CH1 | <table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td>0</td> <td>1</td> </tr> <tr> <td>30H</td> <td>31H</td> </tr> </table> | 0 | 1 | 30H | 31H | <table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td>01H</td> </tr> </table> | 01H |
| 0 | 1 | | | | | | |
| 30H | 31H | | | | | | |
| 01H | | | | | | | |
| CH2 | <table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td>0</td> <td>2</td> </tr> <tr> <td>30H</td> <td>32H</td> </tr> </table> | 0 | 2 | 30H | 32H | <table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td>02H</td> </tr> </table> | 02H |
| 0 | 2 | | | | | | |
| 30H | 32H | | | | | | |
| 02H | | | | | | | |

Switching instruction

Select which specify the contents to be switched, by data in the command or by Engineering tool.

| Bit | Item | Setting of Engineering tool | OFF (0) | ON (1) |
|----------|----------------------|--------------------------------|-------------------------------|----------------------|
| b0 | Mode No. | Communication protocol setting | Specified by Engineering tool | Specified by command |
| b1 | Transmission setting | | | |
| b2 | Communication speed | Communication speed setting | | |
| b3 to b7 | (Fixed to '0') | | | |

■Data communication in ASCII code

Convert the numerical value to 2-digit ASCII code (hexadecimal), and send it from the upper digits.

■Data communication in binary code

Send 1-byte numerical values from lower byte (L: bits 0 to 7).

Ex.

When the switching instruction is '1' (Only mode No. is specified by command)

| ASCII code | Binary code | | | | | |
|--|-------------|---|-----|-----|--|-----|
| <table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td>0</td> <td>1</td> </tr> <tr> <td>30H</td> <td>31H</td> </tr> </table> | 0 | 1 | 30H | 31H | <table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td>01H</td> </tr> </table> | 01H |
| 0 | 1 | | | | | |
| 30H | 31H | | | | | |
| 01H | | | | | | |

Mode No.

Specify the communication protocol setting after switching the C24 mode.

| Mode No. | Operation mode |
|----------|---|
| 01H | MC Protocol (Format 1) |
| 02H | MC Protocol (Format 2) |
| 03H | MC Protocol (Format 3) |
| 04H | MC Protocol (Format 4) |
| 05H | MC Protocol (Format 5) |
| 06H | Nonprocedural protocol |
| 07H | Bidirectional protocol |
| 09H | Predefined protocol |
| FFH | MELSOFT connection, GX Developer connection |

■Data communication in ASCII code

Convert the numerical value to 2-digit ASCII code (hexadecimal), and send it from the upper digits.

■Data communication in binary code

Send 1-byte numerical values from lower byte (L: bits 0 to 7).

Ex.

When mode No. is '1' (MC protocol (Format 1))

| ASCII code | Binary code | | | | | |
|--|-------------|---|-----|-----|---|-----|
| <table border="1"><tr><td>0</td><td>1</td></tr><tr><td>30H</td><td>31H</td></tr></table> | 0 | 1 | 30H | 31H | <table border="1"><tr><td>01H</td></tr></table> | 01H |
| 0 | 1 | | | | | |
| 30H | 31H | | | | | |
| 01H | | | | | | |

Transmission setting

Specify the transmission setting after switching the C24 mode.

| Bit | Item | OFF (0) | ON (1) |
|-----|-------------------|-------------|-----------|
| b0 | Operation setting | Independent | Interlink |
| b1 | Data bit | 7 | 8 |
| b2 | Parity bit | None | Yes |
| b3 | Odd/even parity | Odd | Even |
| b4 | Stop bit | 1 | 2 |
| b5 | Sum check code | None | Yes |
| b6 | Online change | Disable | Enable |
| b7 | Setting change | Disable | Enable |



For the transmission setting, refer to the following manuals.

- 📖 MELSEC iQ-R Serial Communication Module User's Manual (Application)
- 📖 MELSEC-Q/L Serial Communication Module User's Manual (Application)

13

■Data communication in ASCII code

Convert the numerical value to 4-digit ASCII code (hexadecimal) and send it from upper digits.

■Data communication in binary code

Send 1-byte numerical values from lower byte (L: bits 0 to 7).

When the transmission settings are as follows:

| Bit | Item | Setting | ON/OFF |
|-----|-------------------|-------------|--------|
| b0 | Operation setting | Independent | OFF |
| b1 | Data bit | 8 | ON |
| b2 | Parity bit | Yes | ON |
| b3 | Odd/even parity | Odd | OFF |
| b4 | Stop bit | 1 | OFF |
| b5 | Sum check code | Yes | ON |
| b6 | Online change | Enable | ON |
| b7 | Setting change | Enable | ON |

| ASCII code | Binary code |
|----------------|-------------|
| E 6 45H 36H | E6H |

Communication speed

Specify the communication speed after switching the C24 mode.

| Specified value | Communication speed |
|-----------------|---------------------|
| 0FH | 50bps |
| 00H | 300bps |
| 01H | 600bps |
| 02H | 1200bps |
| 03H | 2400bps |
| 04H | 4800bps |
| 05H | 9600bps |
| 06H | 14400bps |
| 07H | 19200bps |
| 08H | 28800bps |
| 09H | 38400bps |
| 0AH | 57600bps |
| 0BH | 115200bps |
| 0CH | 230400bps |



The communication speed that can be set differs depending on the module or channel.

For the communication speed setting, refer to the following manuals.

■ MELSEC iQ-R Serial Communication Module User's Manual (Application)

■ MELSEC-Q/L Serial Communication Module User's Manual (Application)

■ Data communication in ASCII code

Convert the numerical value to 4-digit ASCII code (hexadecimal) and send it from upper digits.

■ Data communication in binary code

Send 1-byte numerical values from lower byte (L: bits 0 to 7).

Ex.

For communication speed 05H (9600 bps)

| ASCII code | Binary code |
|----------------|-------------|
| 0 5 30H 35H | 05H |

Switch mode (command: 1612)

Switch the C24 mode from external devices.

Message format

The following shows the message format of the request data and response data of the command.

■Request data

| Command | Subcommand | Channel No. | Switching instruction | Mode No. | Transmission setting | Communication speed |
|---------|------------|-------------|-----------------------|----------|----------------------|---------------------|
| | | | | | | |

■Response data

There is no response data for this command.

Data specified by request data

13

■Command

| ASCII code | Binary code |
|----------------------------|-------------|
| 1 6 1 2 31H 36H 31H 32H | 12H 16H |

■Subcommand

| ASCII code | Binary code |
|----------------------------|-------------|
| 0 0 0 0 30H 30H 30H 30H | 00H 00H |

■Channel No.

Specify the target interface (CH1/CH2). (☞ Page 253 Channel No.)

■Switching instruction

Select which specify the contents to be switched, by data in the command or by Engineering tool. (☞ Page 253 Switching instruction)

○: Specified by command, —: Specified by Engineering tool

| Switching instruction | Mode No. | Transmission setting | Communication speed |
|-----------------------|----------|----------------------|---------------------|
| 0 | — | — | — |
| 1 | ○ | — | — |
| 2 | — | ○ | — |
| 3 | ○ | ○ | — |
| 4 | — | — | ○ |
| 5 | ○ | — | ○ |
| 6 | — | ○ | ○ |
| 7 | ○ | ○ | ○ |

■Mode No.

Specify the operation mode within the range of 1H to 9H or FFH. (☞ Page 254 Mode No.)

When the setting by command enabled (Switching instruction: 1, 3, 5, or 7), the operation mode is changed according to the specified value.

As for the setting by command is disabled (Switching instruction: 0, 2, 4, or 6), the operation mode is changed according to the Communication protocol setting set with Engineering tool

Even when the setting by command is disabled, specify the value (1 to 9H or FFH). (Do not specify '0'.)

■Transmission setting

Specify the transmission setting. ( Page 255 Transmission setting)

When the setting by command is enabled (Switching instruction: 2, 3, 6, or 7), the transmission setting is changed according to the specified value.

As for the setting by command is disabled (Switching instruction: 0, 1, 4, or 5), the transmission setting is changed according to the transmission setting set with Engineering tool

When the setting is invalid, specify '0'.

| ASCII code | Binary code |
|----------------|-------------|
| 0 0 30H 30H | 00H |

■Communication speed

Specify the communication speed. ( Page 256 Communication speed)

When the setting by command is enabled (Switching instruction: 4, 5, 6, or 7), the communication speed is changed according to the specified value.

When the setting by command is disabled (Switching instruction: 0, 1, 2, or 3), the communication speed is changed according to the communication speed setting set with Engineering tool.

When the setting is invalid, specify '0'.

| ASCII code | Binary code |
|----------------|-------------|
| 0 0 30H 30H | 00H |

Example for data communication

Perform mode switching for CH1 interface.

- Operation mode: MC protocol (Format 1) (Specified value: 01H)
- Operation setting: Following settings (Specified value: B0H)

| Item | Setting | Bit | ON/OFF |
|-------------------|-------------|-----|--------|
| Operation setting | Independent | b0 | OFF |
| Data bit | 7 | b1 | OFF |
| Parity bit | None | b2 | OFF |
| Odd/even parity | Odd | b3 | OFF |
| Stop bit | 2 | b4 | ON |
| Sum check code | Yes | b5 | ON |
| Online change | Disable | b6 | OFF |
| Setting change | Enable | b7 | ON |

- Communication speed: 9600bps (Specified value: 05H)

■Data communication in ASCII code

(Request data)

| Command | Subcommand | Channel No. | Switching instruction | Mode No. | Transmission setting | Communication speed |
|----------------------------|----------------------------|----------------|-----------------------|----------------|----------------------|---------------------|
| 1 6 1 2 31H 36H 31H 32H | 0 0 0 0 30H 30H 30H 30H | 0 1 30H 31H | 0 7 30H 37H | 0 1 30H 30H | B 0 42H 30H | 0 5 30H 35H |

■Data communication in binary code

(Request data)

| Channel No. | Switching instruction | Mode No. | Transmission setting | Communication speed |
|-------------|-----------------------|----------|----------------------|---------------------|
| 12H | 16H | 00H | 00H | 01H 07H 01H B0H 05H |

13.5 Programmable controller CPU monitoring function

Programmable controller CPU monitoring function is a function that C24 monitors CPU module with the monitoring information which was registered in advance.

For the Programmable controller CPU monitoring function, refer to the following manuals.

- MELSEC iQ-R Serial Communication Module User's Manual (Application)
- MELSEC-Q/L Serial Communication Module User's Manual (Application)

Restriction

The command can only be used for C24 (including multidrop connection station) connected to the external device. It cannot be used via network.

Data to be specified in command

Cycle time units

Specify the unit of cycle time.

| Time unit | ASCII code | Binary code | | | | | |
|-----------|--|-------------|---|-----|-----|---|-----|
| 100 ms | <table border="1"><tr><td>0</td><td>0</td></tr><tr><td>30H</td><td>30H</td></tr></table> | 0 | 0 | 30H | 30H | <table border="1"><tr><td>00H</td></tr></table> | 00H |
| 0 | 0 | | | | | | |
| 30H | 30H | | | | | | |
| 00H | | | | | | | |
| 1 second | <table border="1"><tr><td>0</td><td>1</td></tr><tr><td>30H</td><td>31H</td></tr></table> | 0 | 1 | 30H | 31H | <table border="1"><tr><td>01H</td></tr></table> | 01H |
| 0 | 1 | | | | | | |
| 30H | 31H | | | | | | |
| 01H | | | | | | | |
| 1 minute | <table border="1"><tr><td>0</td><td>2</td></tr><tr><td>30H</td><td>32H</td></tr></table> | 0 | 2 | 30H | 32H | <table border="1"><tr><td>02H</td></tr></table> | 02H |
| 0 | 2 | | | | | | |
| 30H | 32H | | | | | | |
| 02H | | | | | | | |

Cycle time

Specify the time interval (period for 1 cycle) that C24 reads the monitoring information from CPU module.

■Data communication in ASCII code

Convert the numerical value to 4-digit ASCII code (hexadecimal) and send it from upper digits.

■Data communication in binary code

Send 2-byte numerical values from lower byte (L: bits 0 to 7).

Programmable controller CPU monitoring function

Specify the send timing (fixed cycle send/condition match send) of the monitoring result.

| Programmable controller CPU monitoring function | ASCII code | Binary code | | | | | |
|---|--|-------------|---|-----|-----|---|-----|
| Fixed cycle send (Sending information in the cycle time interval.) | <table border="1"><tr><td>0</td><td>1</td></tr><tr><td>30H</td><td>31H</td></tr></table> | 0 | 1 | 30H | 31H | <table border="1"><tr><td>01H</td></tr></table> | 01H |
| 0 | 1 | | | | | | |
| 30H | 31H | | | | | | |
| 01H | | | | | | | |
| Condition match send (Sending information when the specified condition matches.) | <table border="1"><tr><td>0</td><td>2</td></tr><tr><td>30H</td><td>32H</td></tr></table> | 0 | 2 | 30H | 32H | <table border="1"><tr><td>02H</td></tr></table> | 02H |
| 0 | 2 | | | | | | |
| 30H | 32H | | | | | | |
| 02H | | | | | | | |

CPU error monitoring, CPU status information

Specify whether to perform error monitoring for the host station CPU module.

| CPU error monitoring | ASCII code | Binary code | | | | | |
|--|--|-------------|---|-----|-----|--|-----|
| Do not perform error monitoring for the host station CPU module. | <table border="1" style="display: inline-table; vertical-align: middle;"> <tr><td>0</td><td>0</td></tr> <tr><td>30H</td><td>30H</td></tr> </table> | 0 | 0 | 30H | 30H | <table border="1" style="display: inline-table; vertical-align: middle;"> <tr><td>00H</td></tr> </table> | 00H |
| 0 | 0 | | | | | | |
| 30H | 30H | | | | | | |
| 00H | | | | | | | |
| Perform error monitoring for the host station CPU module. | <table border="1" style="display: inline-table; vertical-align: middle;"> <tr><td>0</td><td>1</td></tr> <tr><td>30H</td><td>31H</td></tr> </table> | 0 | 1 | 30H | 31H | <table border="1" style="display: inline-table; vertical-align: middle;"> <tr><td>01H</td></tr> </table> | 01H |
| 0 | 1 | | | | | | |
| 30H | 31H | | | | | | |
| 01H | | | | | | | |

When the error monitoring is performed, the CPU monitoring result is stored in the response data as a CPU status information.

The following values are stored in the CPU status information.

| Specified value | CPU status |
|-----------------|--|
| 0000H | During normal operation |
| 0001H | Module warning occurring |
| 0002H | Module error/module system error occurring |

13

■Data communication in ASCII code

Convert the numerical value to 4-digit ASCII code (hexadecimal) and store it in the 'Device Data' in the following format from upper digits.

| | | | | | | | | | | |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-------------|
| 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | Device data |
| 30H | 31H | 30H | 31H | |

■Data communication in binary code

Store the 2-byte numerical values in the 'Device Data' in the following format from lower byte (L: bits 0 to 7).

| | | | | | |
|-----|-----|-----|-------------|-----|---|
| | | | Device data | | |
| 01H | 00H | 00H | 01H | 00H | L |

Monitoring condition

When 'fixed cycle send' is specified with the programmable controller CPU monitoring function, specify '0'.

| ASCII code | Binary code | | | | | | | | | | |
|--|-------------|-----|-----|---|-----|-----|-----|-----|--|-----|-----|
| <table border="1" style="display: inline-table; vertical-align: middle;"> <tr><td>0</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>30H</td><td>30H</td><td>30H</td><td>30H</td></tr> </table> | 0 | 0 | 0 | 0 | 30H | 30H | 30H | 30H | <table border="1" style="display: inline-table; vertical-align: middle;"> <tr><td>00H</td><td>00H</td></tr> </table> | 00H | 00H |
| 0 | 0 | 0 | 0 | | | | | | | | |
| 30H | 30H | 30H | 30H | | | | | | | | |
| 00H | 00H | | | | | | | | | | |

When 'condition match send' is specified with the Programmable controller CPU monitoring function, specify the value below. The timing to send the result can be selected.

- Edge trigger transmission: Send the result when the condition matches.
- Level trigger transmission: Send the result in the cycle time interval during the condition matches

| Edge trigger transmission | Level trigger transmission | Monitoring condition | Device type that can be specified |
|---------------------------|----------------------------|---|-----------------------------------|
| 0001H | 0101H | ON/OFF status of device = ON/OFF status of monitoring condition | Bit device |
| | | Device value = Monitoring condition value | Word device |
| 0002H | 0102H | ON/OFF status of device ≠ ON/OFF status of monitoring condition | Bit device |
| | | Device value ≠ Monitoring condition value | Word device |

| Edge trigger transmission | Level trigger transmission | Monitoring condition | Device type that can be specified | |
|---------------------------|----------------------------|---|-----------------------------------|-------------|
| 0003H | 0103H | Device value ≤ Monitoring condition value | Unsigned | Word device |
| 0004H | 0104H | Device value < Monitoring condition value | | |
| 0005H | 0105H | Device value ≥ Monitoring condition value | | |
| 0006H | 0106H | Device value > Monitoring condition value | | |
| 0007H | 0107H | Device value ≤ Monitoring condition value | | |
| 0008H | 0108H | Device value < Monitoring condition value | | |
| 0009H | 0109H | Device value ≥ Monitoring condition value | | |
| 000AH | 010AH | Device value > Monitoring condition value | | |

■Data communication in ASCII code

Convert the numerical value to 4-digit ASCII code (hexadecimal) and send it from upper digits.

■Data communication in binary code

Send 2-byte numerical values from lower byte (L: bits 0 to 7).

Monitoring condition value

When 'fixed cycle send' is specified with the Programmable controller CPU monitoring function, specify '0'.

| ASCII code | Binary code | | | | | | | | | | |
|---|-------------|-----|-----|---|-----|-----|-----|-----|--|-----|-----|
| <table border="1"> <tr> <td>0</td><td>0</td><td>0</td><td>0</td></tr> <tr> <td>30H</td><td>30H</td><td>30H</td><td>30H</td></tr> </table> | 0 | 0 | 0 | 0 | 30H | 30H | 30H | 30H | <table border="1"> <tr> <td>00H</td><td>00H</td></tr> </table> | 00H | 00H |
| 0 | 0 | 0 | 0 | | | | | | | | |
| 30H | 30H | 30H | 30H | | | | | | | | |
| 00H | 00H | | | | | | | | | | |

When 'condition match send' is specified with the Programmable controller CPU monitoring function, specify the value below.

| Monitoring condition value | Monitoring condition value | Device type |
|----------------------------|----------------------------|-------------|
| 0000H | OFF | Bit device |
| 0001H | ON | |
| 0000H to FFFFH | Numerical value | Word device |

■Data communication in ASCII code

Convert the numerical value to 4-digit ASCII code (hexadecimal) and send it from upper digits.

■Data communication in binary code

Send 2-byte numerical values from lower byte (L: bits 0 to 7).

Register (command: 0630)

Register the target devices to be monitored with the Programmable controller CPU monitoring function and its monitoring conditions in CPU module.

Specify the monitoring target device for multiple blocks with the consecutive word devices and bit devices treated as one block. The host station CPU module can also be specified as error monitoring target.

The monitoring is started when executing the registration command.

Message format

The following shows the message format of the request data and response data of the command.

■Request data

| Command | Subcommand | Cycle time unit | Cycle time | Programmable controller CPU monitoring function | Transmission method | Number of registered word blocks (m points) | | |
|--|----------------------|---------------------|-----------------------------------|---|---------------------|---|--|--|
| ----- | | | | | | | | |
| Number of registered bit blocks (n points) | CPU error monitoring | Block (first point) | ... | Block (mth point) | ----- | | | |
| Word device registration | | | | | | | | |
| Block (first point) | ... | Block (nth point) | CPU error monitoring registration | ----- | | | | |
| Bit device registration | | | | ----- | | | | |

■Response data

The device information of the monitoring result and programmable controller CPU status information are stored.

| | | | | | |
|-------------------------|---|---|--|----------------------|-------------------|
| Fixed value | Programmable controller CPU monitoring function | Number of registered word blocks (n points) | Number of registered bit blocks (n points) | CPU error monitoring | ----- |
| ----- | | | | | |
| Block (first point) | ... | Block (mth point) | Block (first point) | ... | Block (nth point) |
| Word device information | | | Bit device information | | |
| CPU status information | | | ----- | | |



The response data of this command will be sent as same as that of the on-demand function.

For the transmission methods and its timing of monitoring result, refer to the following manuals.

- MELSEC iQ-R Serial Communication Module User's Manual (Application)
- MELSEC-Q/L Serial Communication Module User's Manual (Application)

Data specified by request data

■Command

| ASCII code | Binary code | | | | | | | | | | |
|--|-------------|-----|-----|---|-----|-----|-----|-----|---|-----|-----|
| <table border="1" style="width: 100%; text-align: center;"> <tr> <td>0</td><td>6</td><td>3</td><td>0</td></tr> <tr> <td>30H</td><td>36H</td><td>33H</td><td>30H</td></tr> </table> | 0 | 6 | 3 | 0 | 30H | 36H | 33H | 30H | <table border="1" style="width: 100%; text-align: center;"> <tr> <td>30H</td><td>06H</td></tr> </table> | 30H | 06H |
| 0 | 6 | 3 | 0 | | | | | | | | |
| 30H | 36H | 33H | 30H | | | | | | | | |
| 30H | 06H | | | | | | | | | | |

■Subcommand

| ASCII code | Binary code | | | | | | | | | | |
|--|-------------|-----|-----|---|-----|-----|-----|-----|---|-----|-----|
| <table border="1" style="width: 100%; text-align: center;"> <tr> <td>0</td><td>0</td><td>0</td><td>0</td></tr> <tr> <td>30H</td><td>30H</td><td>30H</td><td>30H</td></tr> </table> | 0 | 0 | 0 | 0 | 30H | 30H | 30H | 30H | <table border="1" style="width: 100%; text-align: center;"> <tr> <td>00H</td><td>00H</td></tr> </table> | 00H | 00H |
| 0 | 0 | 0 | 0 | | | | | | | | |
| 30H | 30H | 30H | 30H | | | | | | | | |
| 00H | 00H | | | | | | | | | | |

■Cycle time units, cycle time

Specify the following time interval (period for 1 cycle). ([Page 260 Cycle time](#))

- Time interval that C24 reads monitoring information from CPU module
- Transmission interval of monitoring result when 'fixed cycle send' is specified with the Programmable controller CPU monitoring function

The time unit (100 ms/1 second/1 minute) can be selected. ([Page 260 Cycle time units](#))

■Programmable controller CPU monitoring function

Specify the send timing (fixed cycle send/condition match send) of the monitoring result. ([Page 260 Programmable controller CPU monitoring function](#))

■Transmission method

Fixed '0'.

| ASCII code | Binary code | | | | | |
|--|-------------|---|-----|-----|---|-----|
| <table border="1" style="width: 100%; text-align: center;"> <tr> <td>0</td><td>0</td></tr> <tr> <td>30H</td><td>30H</td></tr> </table> | 0 | 0 | 30H | 30H | <table border="1" style="width: 100%; text-align: center;"> <tr> <td>00H</td></tr> </table> | 00H |
| 0 | 0 | | | | | |
| 30H | 30H | | | | | |
| 00H | | | | | | |

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

Specify the number of blocks of word device registration and bit device registration. ([Page 69 Number of blocks](#))

■CPU error monitoring, CPU error monitoring registration

Specify whether to perform error monitoring (status monitoring) of host station CPU module together. ([Page 261 CPU error monitoring, CPU status information](#))

- When do not perform CPU error monitoring (0): CPU error monitoring registration is not required.
- When perform CPU error monitoring (1): Register the following fixed values to CPU error monitoring registration.

| ASCII code | Binary code | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|--|-------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|---|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|---|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| <table border="1" style="width: 100%; text-align: center;"> <tr> <td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>5</td><td>0</td><td>0</td><td>0</td><td>1</td></tr> <tr> <td>30H</td><td>31H</td><td>30H</td><td>30H</td><td>30H</td><td>30H</td><td>30H</td><td>30H</td><td>30H</td><td>30H</td><td>31H</td><td>30H</td><td>30H</td><td>30H</td><td>35H</td><td>30H</td><td>30H</td><td>30H</td><td>31H</td></tr> </table> | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 5 | 0 | 0 | 0 | 1 | 30H | 31H | 30H | 31H | 30H | 30H | 30H | 35H | 30H | 30H | 30H | 31H | <table border="1" style="width: 100%; text-align: center;"> <tr> <td>00H</td><td>00H</td><td>00H</td><td>01H</td><td>01H</td><td>00H</td><td>05H</td><td>00H</td><td>01H</td><td>00H</td></tr> </table> | 00H | 00H | 00H | 01H | 01H | 00H | 05H | 00H | 01H | 00H |
| 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 5 | 0 | 0 | 0 | 1 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 30H | 31H | 30H | 31H | 30H | 30H | 30H | 35H | 30H | 30H | 30H | 31H | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 00H | 00H | 00H | 01H | 01H | 00H | 05H | 00H | 01H | 00H | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

■Word device registration, bit device registration

Specify the monitoring target device for multiple blocks with the consecutive word devices and bit devices treated as one block.

Specify the following items for every block.

| | | | |
|-------------------------|-----------------------------|----------------------|----------------------------|
| Monitoring start device | Number of registered points | Monitoring condition | Monitoring condition value |
|-------------------------|-----------------------------|----------------------|----------------------------|

Block (1 point)

- Monitoring start device: Specify the device code and device number. ([Page 63 Devices](#))
- Number of registered points: Specify the device points from the start device in word units. ([Page 67 Number of device points](#))
- Monitoring condition: Specify the monitoring condition and transmission method for the device to be registered. ([Page 261 Monitoring condition](#))
- Monitoring condition value: Specify a value to judge the conditions have been matched or ON/OFF state of the bit. ([Page 262 Monitoring condition value](#))

Data to be stored by response data

■Fixed value

| ASCII code | Binary code |
|----------------------------------|-------------|
| 2 1 0 2 32H , 31H , 30H , 32H | 02H , 21H |

■Programmable controller CPU monitoring function

The send timing (fixed cycle send/condition match send) of the monitoring result is stored. (☞ Page 260 Programmable controller CPU monitoring function)

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

The number of blocks of word device information and bit device information are stored. (☞ Page 69 Number of blocks)

■Word device information, bit device information

The information of device, whose monitoring condition is satisfied, is stored for multiple blocks with the consecutive word devices and bit devices treated as one block.

The following items are stored for each block.

| | | |
|-------------------------|-----------------------------|-------------|
| Monitoring start device | Number of registered points | Device data |
|-------------------------|-----------------------------|-------------|

Block (1 point)

- Monitoring start device: Device code and device number are stored. (☞ Page 63 Devices)
- Number of registered points: Number of device data points is stored in word unit. (☞ Page 67 Number of device points)
- Device data: The device value is stored.

The order of device data differ depending on the setting of "Word/byte units designation" in Engineering tool.

For byte units, handle the word data as 2-byte data and send the numerical values from the lower byte (L: bits 0 to 7).

Ex.

2 points data of word device (1234H, 5678H)

- Word unit (0)

| ASCII code | Binary code |
|--|-----------------------|
| 1 2 3 4 5 6 7 8 31H , 32H , 33H , 34H 35H , 36H , 37H , 38H | 12H 34H 56H 78H |

- Byte unit (1)

| ASCII code | Binary code |
|--|-----------------------|
| 3 4 1 2 7 8 5 6 33H , 34H , 31H , 32H 37H , 38H , 35H , 36H | 34H 12H 78H 56H |

■CPU error monitoring, CPU status information

Specify whether to include the result of error monitoring (state monitoring) of the host station CPU module. (☞ Page 261 CPU error monitoring, CPU status information)

- When do not perform CPU error monitoring (0): CPU status information is not stored.
- When perform CPU error monitoring (1): CPU status information is stored.

Communication example (for fixed cycle send)

Perform the Programmable controller CPU monitoring registration with the following conditions.

- Cycle time units, cycle time: 30 seconds
- Programmable controller CPU monitoring function: Fixed cycle send
- Number of registered word blocks: 2 blocks
- Number of registered bit blocks: 1 block
- CPU error monitoring: Included

| Block | Monitoring start device | Registered points |
|------------------------------------|-------------------------|-------------------|
| Word device registration (1-point) | D0 | 4 points |
| Word device registration (2-point) | W100 | 8 points |
| Bit device registration (1-point) | M0 | 2 points |

The response data indicates the following data.

(1) Word device information (1-point)

| Device data | D0 | D1 | D2 | D3 |
|--------------|------------------------|----------------------|--|----------------------|
| Device value | Decimal Hexadecimal | 99 0063H | 4144 1030H | 5445 1545H |
| | Word unit Byte unit | 30303633 36333030 | 31303330 33303130 | 31353435 34353135 |
| ASCII code | Word unit Byte unit | 0063 6300 | 101030 ^{*1} 301010 ^{*1} | 1545 4515 |
| | | | | 2800 0028 |
| Binary code | | | | |

*1 The additional code is added. (☞ Page 33 Additional code (10H))

(2) Bit device information (1-point)

| Device data | M15 to M8 | M7 to M0 | M31 to M24 | M23 to M16 |
|--------------|-----------|-----------------|-----------------|-----------------|
| Device value | Bit | 0 0 0 1 0 0 0 1 | 0 0 1 1 0 0 0 1 | 0 1 0 0 1 0 0 0 |
| | Byte | 11H | 31H | 48H |
| | Word | 1131H | | 4849H |
| ASCII code | Word unit | 31313331 | | 34383439 |
| | Byte unit | 33313131 | | 34393438 |
| Binary code | Word unit | 1131 | | 4849 |
| | Byte unit | 3111 | | 4948 |

(3) CPU status information

- CPU status information: Module warning occurring (0001)

■Data communication in ASCII code

(Request data)

| Subcommand | Cycle time unit | Cycle time | Programmable controller CPU monitoring function | Transmission method | Number of registered word blocks | Number of registered bit blocks | CPU error monitoring |
|---|-----------------|-----------------------------|---|----------------------|----------------------------------|---------------------------------|----------------------|
| 0 6 3 0 | 0 0 0 0 | 0 1 | 0 0 1 E | 0 1 | 0 0 | 0 2 | 0 1 0 1 |
| 30H 36H 33H 30H | 30H 30H 30H 30H | 30H 31H | 30H 30H 31H 45H | 30H 31H | 30H 30H | 30H 32H | 30H 31H 30H 31H |
| Word device registration (first point) | | Number of registered points | | Monitoring condition | | Monitoring condition value | |
| D * | 0 0 0 0 0 0 | 0 0 0 4 | 0 0 0 0 | 0 0 0 0 | 0 0 0 0 | 0 0 0 0 | 0 0 0 0 |
| 44H 2AH | 30H 30H 30H 30H | 30H 34H | 30H 30H 30H 30H | 30H 30H 30H 30H | 30H 30H 30H 30H | 30H 30H 30H 30H | 30H 30H 30H 30H |
| D0 to D3 | | | | | | | |
| Word device registration (second point) | | Number of registered points | | Monitoring condition | | Monitoring condition value | |
| W * | 0 0 0 1 0 0 | 0 0 0 8 | 0 0 0 0 | 0 0 0 0 | 0 0 0 0 | 0 0 0 0 | 0 0 0 0 |
| 57H 2AH | 30H 30H 31H 30H | 30H 38H | 30H 30H 30H 30H | 30H 30H 30H 30H | 30H 30H 30H 30H | 30H 30H 30H 30H | 30H 30H 30H 30H |
| W100 to D107 | | | | | | | |
| Bit device registration (first point) | | Number of registered points | | Monitoring condition | | Monitoring condition value | |
| M * | 0 0 0 0 0 0 | 0 0 0 2 | 0 0 0 0 | 0 0 0 0 | 0 0 0 0 | 0 0 0 0 | 0 0 0 0 |
| 4DH 2AH | 30H 30H 30H 30H | 30H 32H | 30H 30H 30H 32H | 30H 30H 30H 30H | 30H 30H 30H 30H | 30H 30H 30H 30H | 30H 30H 30H 30H |
| M0 to M31 | | | | | | | |
| CPU error monitoring registration (Fixed value) | | | | | | | |
| 0 1 | 0 0 0 0 0 0 | 0 0 0 1 | 0 0 0 5 | 0 0 0 1 | 0 0 0 1 | 0 0 0 1 | 0 0 0 1 |
| 30H 31H | 30H 30H 30H 30H | 30H 31H | 30H 30H 35H | 30H 30H 31H | 30H 30H 35H | 30H 30H 31H | 30H 30H 31H |

(Response data)

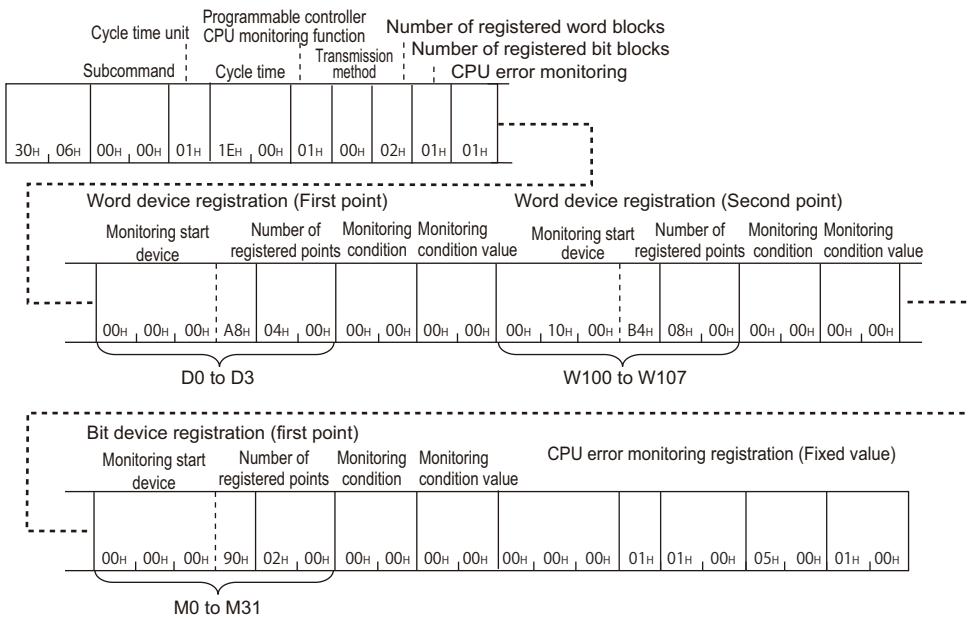
| Fixed value | Programmable controller CPU monitoring function | Number of registered word blocks | Number of registered bit blocks | CPU error monitoring |
|-------------------------|---|----------------------------------|---------------------------------|----------------------|
| 2 1 0 2 | 0 1 | 0 2 | 0 1 | 0 1 |
| 32H 31H 30H 32H | 30H 31H | 30H 32H | 30H 31H | 30H 31H |
| Word device information | | | | |
| Monitoring start device | | Number of registered points | | Device data |
| D * | 0 0 0 0 0 0 | 0 0 0 4 | (1) | ... |
| 44H 2AH | 30H 30H 30H 30H | 30H 34H | (first point) | (second point) |
| Bit device information | | | | |
| Monitoring start device | | Number of registered points | | Device data |
| M * | 0 0 0 0 0 0 | 0 0 0 2 | (2) | - |
| 4DH 2AH | 30H 30H 30H 30H | 30H 32H | | |
| CPU status information | | | | |
| Device data | | | | |
| 0 1 | 0 0 0 0 0 0 | 0 0 0 1 | 0 0 0 1 | 0 0 0 1 |
| 30H 31H | 30H 30H 30H 30H | 30H 31H | 30H 30H 31H | 30H 30H 31H |

In the figure (1) and (2), set the value of "ASCII Code" in the table of "Device data" of each response data.

The order of device data differ depending on the setting of "Word/byte units designation" in Engineering tool.

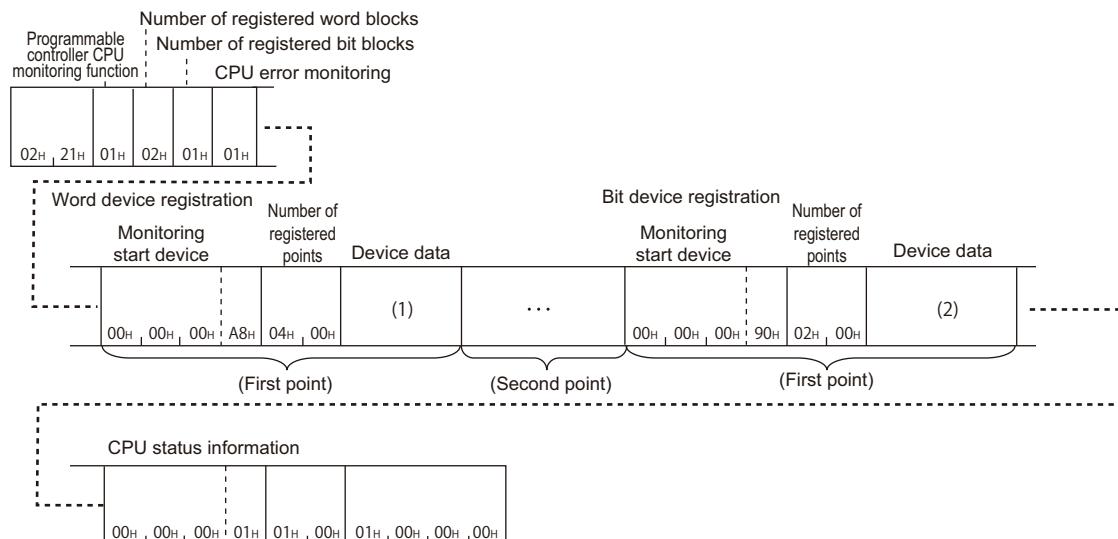
■Data communication in binary code

(Request data)



13

(Response data)



In the figure (1) and (2), set the value to "Binary code" in the tables of "Device data" of each response data. The order of device data differ depending on the setting of "Word/byte units designation" in Engineering tool.

Communication example (for condition match send)

Perform the Programmable controller CPU monitoring registration function with the following conditions.

- Cycle time units, cycle time: 30 seconds
- Programmable controller CPU monitoring function: Condition match send
- Number of registered word blocks: 2 blocks
- Number of registered bit blocks: 1 block
- CPU error monitoring designation: Included

| Block | Monitoring start device | Number of registered points | Monitoring condition | Monitoring condition value |
|------------------------------------|-------------------------|-----------------------------|--|----------------------------|
| Word device registration (1-point) | D0 | 4 points | Edge triggered transmission: Device value = Monitoring condition value | 99 |
| Word device registration (2-point) | W100 | 8 points | Edge triggered transmission: Device value ≠ Monitoring condition value | 0 |
| Bit device registration (1-point) | M0 | 2 points | Edge trigger transmission: ON/OFF status of device ≠ ON/OFF status of monitoring condition | OFF |

The response data indicates the following data.

For the condition match send, the registration data information will be sent individually. (Device information will be sent per block.)

(1) When the condition of D0 = 99 (word device registration (first point) of request data) is satisfied.

| Device data | | D0 | D1 | D2 | D3 |
|--------------|-------------|----------|----------------------|----------|----------|
| Device value | Decimal | 99 | 4144 | 5445 | 10240 |
| | Hexadecimal | 0063H | 1030H | 1545H | 2800H |
| ASCII code | Word unit | 30303633 | 31303330 | 31353435 | 32383030 |
| | Byte unit | 36333030 | 33303130 | 34353135 | 30303238 |
| Binary code | Word unit | 0063 | 101030 ^{*1} | 1545 | 2800 |
| | Byte unit | 6300 | 301010 ^{*1} | 4515 | 0028 |

*1 The additional code is added. (参照 Page 33 Additional code (10H))

(2) When the condition of M0 ≠ OFF (bit device registration (1 point) of request data) is satisfied.

| Device data | | M15 to M8 | M7 to M0 | M31 to M24 | M23 to M16 |
|--------------|-----------|-----------------|-----------------|-----------------|-----------------|
| Device value | Bit | 0 0 0 1 0 0 0 1 | 0 0 1 1 0 0 0 1 | 0 1 0 0 1 0 0 0 | 0 1 0 0 1 0 0 1 |
| | Byte | 11H | 31H | 48H | 49H |
| | Word | 1131H | | 4849H | |
| ASCII code | Word unit | 31313331 | | 34383439 | |
| | Byte unit | 33313131 | | 34393438 | |
| Binary code | Word unit | 1131 | | 4849 | |
| | Byte unit | 3111 | | 4948 | |

(3) When CPU error monitoring condition is satisfied

- CPU status information: Module warning occurring (0001)

When the error is detected at the first time, the monitoring result is sent with the equivalent method to the edge trigger transmission.

■Data communication in ASCII code

(Request data)

| Subcommand | Cycle time unit | Cycle time | Programmable controller CPU monitoring function | Transmission method | Number of registered word blocks | Number of registered bit blocks | CPU error monitoring |
|--|---------------------|-----------------------------|---|----------------------|----------------------------------|---------------------------------|----------------------|
| 0 6 3 0 | 0 0 0 0 | 0 1 | 0 0 1 E | 0 2 0 0 | 0 2 | 0 1 | 0 1 |
| 30H 36H 33H 30H | 30H 30H 30H 30H | 30H 31H | 30H 30H 31H 45H | 30H 32H 30H 30H | 30H 32H 30H 31H | 30H 31H | 30H 31H |
| Word device registration (first point) | | | | | | | |
| Monitoring start device | | Number of registered points | | Monitoring condition | | Monitoring condition value | |
| D * | 0 0 0 0 0 0 | 0 0 0 4 | 0 0 0 1 | 0 0 6 3 | | | |
| 44H 2AH 30H 30H 30H 30H | 30H 30H 30H 34H | 30H 30H 30H 31H | 30H 30H 36H 33H | | | | |
| Word device registration (second point) | | | | | | | |
| Monitoring start device | | Number of registered points | | Monitoring condition | | Monitoring condition value | |
| W * | 0 0 0 1 0 0 | 0 0 0 8 | 0 0 0 2 | 0 0 0 0 | | | |
| 57H 2AH 30H 30H 31H 30H | 30H 30H 38H | 30H 30H 32H | 30H 30H 30H | 30H 30H 30H | | | |
| Bit device registration (first point) | | | | | | | |
| Monitoring start device | | Number of registered points | | Monitoring condition | | Monitoring condition value | |
| M * | 0 0 0 0 0 0 | 0 0 0 2 | 0 0 0 2 | 0 0 0 0 | | | |
| 4DH 2AH 30H 30H 30H 30H | 30H 30H 32H | 30H 30H 32H | 30H 30H 32H | 30H 30H 30H | | | |
| CPU error monitoring registration (Fixed value) | | | | | | | |
| 0 1 | 0 0 0 0 0 0 | 0 0 0 1 | 0 0 0 5 | 0 0 0 1 | | | |
| 30H 31H | 30H 30H 30H 30H 30H | 30H 30H 31H | 30H 30H 35H | 30H 30H 30H 31H | | | |

(Response data)

When the condition (D0 = 99) of word device registration (first point) is established

| Fixed value | Programmable controller CPU monitoring function | Number of registered word blocks | Number of registered bit blocks | CPU error monitoring | Monitoring start device | Number of registered points | Device data |
|-----------------|---|----------------------------------|---------------------------------|----------------------|-------------------------|-----------------------------|-------------|
| 2 1 0 2 | 0 2 | 0 1 | 0 0 | 0 0 | D * | 0 0 0 0 0 0 | 0 0 0 4 (1) |
| 32H 31H 30H 32H | 30H 32H | 30H 31H | 30H 30H | 30H 30H | 44H 2AH | 30H 30H 30H 30H 30H 34H | |

Word device information

When the condition (M0 ≠ OFF) of bit device registration (first point) is established

| Fixed value | Programmable controller CPU monitoring function | Number of registered word blocks | Number of registered bit blocks | CPU error monitoring | Monitoring start device | Number of registered points | Device data |
|-----------------|---|----------------------------------|---------------------------------|----------------------|-------------------------|-----------------------------|-------------|
| 2 1 0 2 | 0 2 | 0 0 | 0 1 | 0 0 | M * | 0 0 0 0 0 0 | 0 0 0 2 (2) |
| 32H 31H 30H 32H | 30H 32H | 30H 30H | 30H 31H | 30H 30H | 4DH 2AH | 30H 30H 30H 30H 30H 32H | |

Bit device information

When the condition of CPU error monitoring is established(Module warning being generated)

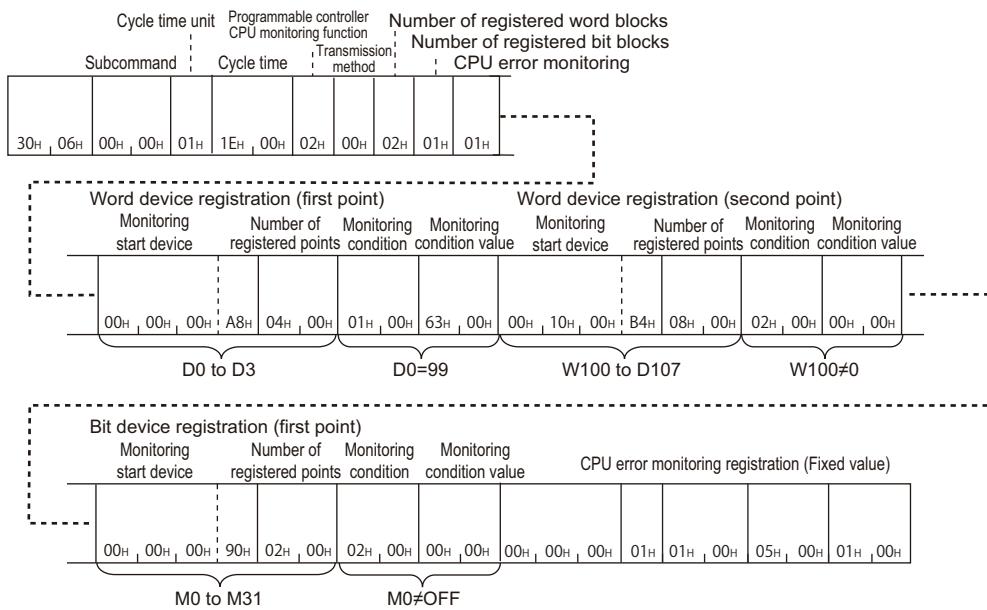
| Fixed value | Programmable controller CPU monitoring function | Number of registered word blocks | Number of registered bit blocks | CPU error monitoring | Fixed value | Device data |
|-----------------|---|----------------------------------|---------------------------------|----------------------|-------------------------|-----------------|
| 2 1 0 2 | 0 2 | 0 0 | 0 0 | 0 1 | 0 1 | 0 0 0 1 |
| 32H 31H 30H 32H | 30H 32H | 30H 30H | 30H 30H | 30H 31H | 30H 30H 30H 30H 30H 32H | 30H 30H 30H 32H |

CPU status information

In the figure (1) and (2), set the value of "ASCII Code" in the table of "Device data" of each response data.
The order of device data differ depending on the setting of "Word/byte units designation" in Engineering tool.

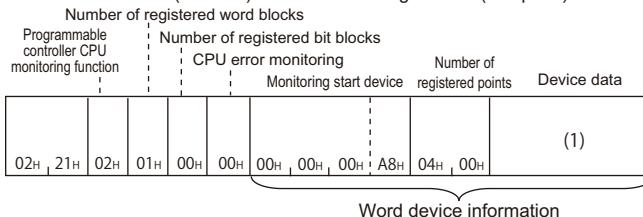
■Data communication in binary code

(Request data)

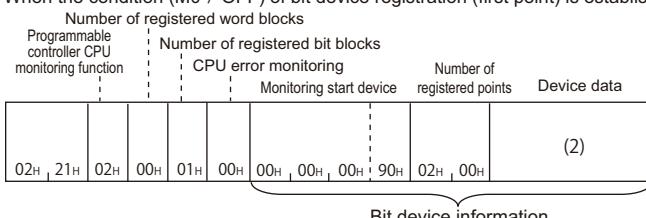


(Response data)

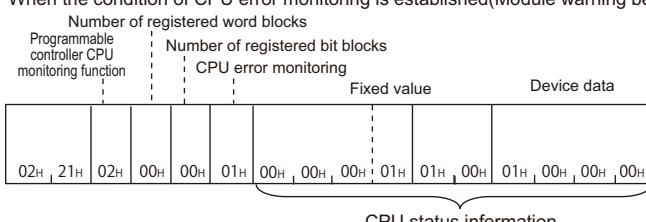
When the condition (D0 = 99) of word device registration (first point) is established



When the condition (M0 ≠ OFF) of bit device registration (first point) is established



When the condition of CPU error monitoring is established(Module warning being generated)



In the figure (1) and (2), set the value of "Binary code" in the tables of "Device data" of each response data.

The order of device data differ depending on the setting of "Word/byte units designation" in Engineering tool.

Deregister (command: 0631)

Deregister the Programmable controller CPU monitoring function.

Execute the cancel command to end monitoring.

Message format

The following shows the message format of the request data and response data of the command.

■Request data

| | |
|---------|------------|
| Command | Subcommand |
|---------|------------|

■Response data

There is no response data for this command.

13

Data specified by request data

■Command

| ASCII code | Binary code |
|----------------------------|-------------|
| 0 6 3 1 30H 36H 33H 31H | 31H 06H |

■Subcommand

| ASCII code | Binary code |
|----------------------------|-------------|
| 0 0 0 0 30H 30H 30H 30H | 00H 00H |

Example for data communication

Deregister the Programmable controller CPU monitoring function.

■Data communication in ASCII code

(Request data)

| Subcommand | |
|----------------------------|----------------------------|
| 0 6 3 1 30H 36H 33H 31H | 0 0 0 0 30H 30H 30H 30H |

■Data communication in binary code

(Request data)

| Subcommand | |
|------------|---------|
| 31H 06H | 00H 00H |

13.6 On-demand function

On-demand function is a function that transmits data to external device using MC protocol after starting up the function from CPU module.

This function can be used to sent the emergency data that is required to notify to external devices from the CPU module.

Restriction

The command can only be used for C24 connected to the external device on 1:1 basis. It cannot be used via network.

If the on-demand function is used with multidrop connection of 1:n station or m:n station, the on-demand data will be collapsed and proper data is not sent.

Settings for using the on-demand function

The following shows the setting items for the on-demand function.

Settings of serial communication module

Set the following values in the parameter of serial communication module or the buffer memory.

| Setting item | Description | Corresponding buffer memory | |
|---|---|-----------------------------|------------|
| | | CH1 | CH2 |
| Communication protocol setting | Set the format of the message format to be used. <ul style="list-style-type: none">• MC protocol (Format 1) to MC protocol (Format 4): Data is sent by 1C frame• MC protocol (Format 5): Data is send by 4C frame. | — | — |
| Word/byte units designation | Set the unit of data length (number of data). | 150 (96H) | 310 (136H) |
| Buffer memory start address designation | Set the start address of the buffer memory to be used for the on-demand function. | 160 (A0H) | 320 (140H) |
| Data length designation | Set the data length of the area to which the transmitted data is stored by the on-demand function. | 161 (A1H) | 321 (141H) |

The execution result is stored in the following:

| Execution result | Description | Corresponding buffer memory | |
|-----------------------------|---|-----------------------------|------------|
| | | CH1 | CH2 |
| On-demand execution results | <ul style="list-style-type: none">• Normal completion: 0• Abnormal completion (error code): Other than 0 | 598 (256H) | 614 (266H) |

Point

The setting content can be checked and changed by using any of the following Engineering tools.

- GX Works3 (Module parameter)
- GX Works2, GX Configurator-SC

For details on the setting items and buffer memory, refer to the following manuals.

- MELSEC iQ-R Serial Communication Module User's Manual (Application)
- MELSEC-Q/L Serial Communication Module User's Manual (Application)

Specification of interface to be sent and send data

Specify the transmission channel (CH1/CH2) and data equivalent to the response data with the arguments of the following dedicated instruction. Set the arguments in the program.

- G(P).ONDEMAND

Point

For the dedicated instruction "G(P).ONDEMAND", refer to the following manuals.

- MELSEC iQ-R Programming Manual (Instructions, Standard Functions/Function Blocks)
- MELSEC-Q/L Serial Communication Module User's Manual (Application)

Execution procedure

The procedures for the on-demand function are as shown below.

Procedure for CPU module and C24

The following shows the procedure to send data using the on-demand function from CPU module.

1. Set the items for the on-demand function.
 Page 274 Settings for using the on-demand function
2. Execute the serial communication module dedicated instruction 'G (P). ONDEMAND'.
 -  MELSEC iQ-R Programming Manual (Instructions, Standard Functions/Function Blocks)
 -  MELSEC-Q/L Serial Communication Module User's Manual (Application)

Procedure for external device

The following shows the procedure to receive the data (on-demand data) transmitted using the on-demand function from CPU module.

1. Receive the message.
2. Judge whether the received message is issued from the on-demand function by "PC No.". For the message of on-demand function, 'PC No.' will be 'FE'. ( Page 50 Network No., PC No.)
3. Process the received on-demand data.

Execution timing

The following shows the execution timing of send/receive processing while C24 is being sending/receiving other message of MC protocol, when data transmission instruction by the on-demand function is executed in the CPU module.

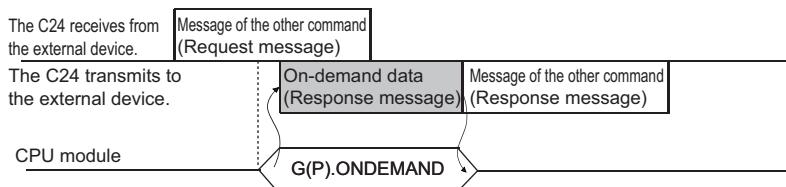
Point

When sending on-demand data or response data, the timeout check is performed by send monitoring timer (timer 2). If a timeout error occurs, change the send monitoring time (timer 2).

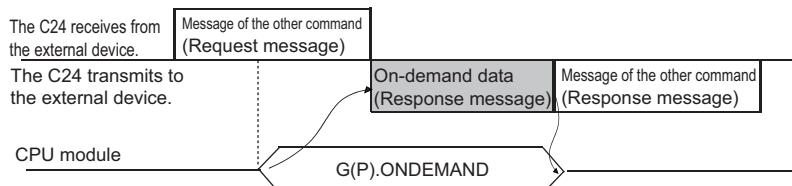
When C24 is receiving request message of other command from external device

C24 sends on-demand data before the response message to the external device is sent.

- Full-duplex communication: C24 sends on-demand data during the request message from the external device is being received.

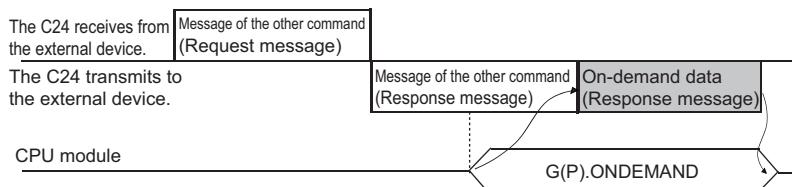


- Half-duplex communication: C24 sends on-demand data after the request message from the external device has been received.



When C24 is sending response message of other command to external device

C24 sends the on-demand data to the external device after the response message to the external device.



On-demand (command: 2101)

Send data using the on-demand function from CPU module. Any message other than send data is added automatically in the message format selected in the communication protocol setting. ('FE' is specified to the 'PC No.' for the access route.)

- Communication protocol setting is MC protocol (Format 1) to MC protocol (Format 4): Message format of 1C frame
- Communication protocol setting is MC protocol (Format 5): Message format of 4C frame

Point

When using 2C/3C/4C frame (Format 1 to 4), register the message format of each format in the user frame.

☞ Page 27 Message Formats of Each Protocol

For the user frames, refer to the following manuals.

- MELSEC iQ-R Serial Communication Module User's Manual (Application)
- MELSEC-Q/L Serial Communication Module User's Manual (Application)

Message format

The following shows the message format of the request data and response data of the command.

■Request data

There is no request data for this command.

■Response data

| | |
|---------|-------------------|
| Command | Transmission data |
|---------|-------------------|

Data stored in response data

■Command

It is added only when the message is sent with MC protocol format 5 specified in the communication protocol setting.

Binary code

| | |
|-----|-----|
| 01H | 21H |
|-----|-----|

■Transmission data

The transmission data specified in the dedicated instruction 'G(P).ONDEMAND' is stored.

The order of data differ depending on the setting of "Word/byte units designation" in Engineering tool.

For byte units, regard the word data as 2-byte data and send numerical values from lower byte (L: bits 0 to 7).

Ex.

When transmitting 2-word data (1234H, 5678H)

- Word unit (0)

| ASCII code | Binary code |
|--|-------------------|
| 1 2 3 4 5 6 7 8 31H 32H 33H 34H 35H 36H 37H 38H | 12H 34H 56H 78H |

- Byte unit (1)

| ASCII code | Binary code |
|--|-------------------|
| 3 4 1 2 7 8 5 6 33H 34H 31H 32H 37H 38H 35H 36H | 34H 12H 78H 56H |

Communication example (communication protocol setting is format 1 to 4)

Send 2-word data (1234H, 5678H) to the external device from the CPU module under the following settings.

- Communication protocol setting: MC protocol (Format 1 to 4)
- Word/byte units designation: Word unit

(Response data)

| Transmission data | | | | | | | |
|-------------------|-----|-----|-----|-----|-----|-----|-----|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| 31H | 32H | 33H | 34H | 35H | 36H | 37H | 48H |

Communication example (communication protocol is format 5)

Send 2-word data (1234H, 5678H) to the external device from the CPU module under the following settings.

- Communication protocol setting: MC Protocol (Format 5)
- Word/byte units designation: Word unit

(Response data)

| Command | Transmission data |
|----------|--------------------|
| 01H, 21H | 12H, 34H, 56H, 78H |

PART 4**COMPATIBILITY WITH
QnA SERIES**

This part explains the specifications when using MELSEC-QnA series devices.

[14 MELSEC-QnA SERIES SUPPORTED SPECIFICATIONS](#)

[15 QnACPU DEDICATED COMMANDS](#)

14 MELSEC-QnA SERIES SUPPORTED SPECIFICATIONS

This chapter explains the specifications of the messages of MC protocol and access ranges when using the MELSEC-QnA series devices as follows:

- When accessing system including MELSEC-QnA series modules
- When utilizing software for data communication created for MELSEC-QnA series programmable controller.

14.1 Frames and Commands that can be used

When accessing MELSEC-QnA series modules, all frames of MC protocol can be used.

However, the commands that can be used have some restrictions. (☞ Page 465 Accessible Modules for Each Command)

QnA-compatible frame

The following frames have compatibility with the message protocol and the message format for MELSEC-QnA series.

| Frame | Compatible message format | Access range |
|----------|---|--|
| 4C frame | Dedicated protocols for MELSEC-QnA series serial communication modules (QnA extension frame) | Page 43 Accessible range of 4C frame |
| 3C frame | Dedicated protocols for MELSEC-QnA series serial communication modules (QnA frame) | Page 44 Accessible range of 3C frame |
| 2C frame | Dedicated protocols for MELSEC-QnA series serial communication modules (QnA simplified frame) | Page 44 Accessible range of 2C frame |
| 3E frame | Message formats for MELSEC-QnA series Ethernet interface modules | Page 46 Accessible range of 4E frame, 3E frame |

14.2 Accessible Modules

Accessible to the following MELSEC-QnA series modules within the access range.

Accessible modules to other stations

The following modules of other station can be accessed.

| Type | Model name |
|-------------------------|---|
| CPU module | Q2ACPU, Q2ACPU-S1, Q2ASCPU, Q2ASCPU-S1, Q2ASHCPU, Q2ASHCPU-S1, Q3ACPU, Q4ACPU |
| Redundant system | Q4ARCPU |
| MELSECNET/10 remote I/O | AJ72QLP25(G), AJ72QBR15, A1SJ72QLP25, A1SJ72QBR15 |
| Special function module | Refer to the following section. ☞ Page 153 Accessible modules |

Modules that can be relayed between networks

The following shows the modules that can be relayed between networks when accessing a MELSEC-QnA series module.

■MELSEC-QnA series modules

| Network | Model name |
|------------------------|--|
| MELSECNET/10 | QJ71LP21, QJ71LP21-25, QJ71LP21S-25, QJ71LP21G, QJ71BR11 (MELSECNET/10 mode) AJ71QLP21 (S/G), AJ71QBR11, A1SJ71QLP21 (S), A1SJ71QBR11 |
| Ethernet ^{*1} | AJ71QE71N3-T, AJ71QE71N-B5, AJ71QE71N-B2, AJ71QE71N-T, AJ71QE71N-B5T, A1SJ71QE71N3-T, A1SJ71QE71N-B5, A1SJ71QE71N-B2, A1SJ71QE71N-T, A1SJ71QE71N-B5T, AJ71QE71, AJ71QE71-B5, A1SJ71QE71-B2, A1SJ71QE71B5 |

*1 QnA series Ethernet interface modules can be relayed between networks when using the module with the function version B or later.

Check the function version with the production date printed on the rating plate on the side of the module.

(Manufactured date: Year (lower two digits), Month (two digits), function version (one digit))

■ Modules other than MELSEC-QnA series

| Network | Module type | Model |
|-------------------------------|--|---|
| CC-Link IE Field Network | CC-Link IE Field Network master/local module | LJ71GF11-T2, QJ71GF11-T2, QS0J71GF11-T2 |
| CC-Link IE Controller Network | CC-Link IE Controller Network module | QJ71GP21-SX, QJ71GP21S-SX |
| MELSECNET/H | MELSECNET/H module | QJ71LP21, QJ71LP21-25, QJ71LP21S-25, QJ71LP21G, QJ71BR11, QJ71NT11B |
| Ethernet | Ethernet interface module | LJ71E71-100, QJ71E71-100, QJ71E71-B5, QJ71E71-B2, QJ71E71 |

☞ Page 19 Modules that can be relayed between networks

14.3 Considerations

The following shows the considerations when using MELSEC-QnA series devices.

Considerations when connecting E71

■ Setting range of monitoring timer

When accessing QnACPU for the first time, the wait time for CPU monitoring timer is required before receiving a response message because QnACPU identifies the CPU type. Be sure to set a value within the setting range shown below.

| Access target | Monitoring timer |
|----------------------------------|--------------------------|
| Connected station (host station) | 1H to 28H (0.25s to 10s) |
| Other station | 2H to F0H (0.5s to 60s) |

15 QnACPU DEDICATED COMMANDS

This chapter shows the QnACPU dedicated commands.

15.1 QnACPU Dedicated Commands List

The following shows the list of QnACPU dedicated commands.

Restriction

There are some commands that cannot be executed while the CPU module is in RUN. Refer to the following section.

☞ Page 457 Applicable Commands for Online Program Change

Point

For the number of points processed per communication, refer to the following section.

☞ Page 459 Number of Processing per One Communication

Drive memory defragmentation

| Function | Command | Description | Subcommand |
|--------------------------|---------|--|------------|
| Read memory usage status | 0205 | Reads the cluster usage status of the driver. | 0000 |
| Defragment memory | 1207 | Increases the consecutive free area by defragmenting the drive memory. (Defragmentation of file storage position) | 0000 |

File control

| Function | Command | Description | Subcommand | |
|--------------------------------------|----------------------------------|---|---|--------------------------------|
| Read file information table | Without header statement 0201 | Reads a file list (file name, last edit date and time, file size). | 0000 | — |
| | With header statement 0202 | Reads a file list (header statement attached to a file, file name, last edit date and time, file size). | 0000 | — |
| Read file presence (Search file) | | 0203 | Reads existence of specified file, file No., and file size. | 0000 |
| Read file No. usage status | | 0204 | Reads file No. usage status. | 0000 |
| Read file | | 0206 | Reads content of file. | 0000 |
| Lock file | 0808 | Locks the file in order to prevent the file contents from being edited from other devices while accessing the specified file. | 0001 | Register |
| | | Unlocks the file. | 0000 | Deregister |
| Create new file (Register file name) | 1202 | Secures the file area for the specified file name. | 0000 | — |
| Write to file | 1203 | Writes the specified data (n byte) to a file. | 0000 | Arbitrary data |
| | | Writes the specified data (1 word) for a file for n bytes. | 0001 | Identical data (FILL) |
| Modify file information | 1204 | Modifies the date and time of the last edit of the file. | 0000 | Modify last edit date and time |
| | | Modifies the file name and file size. | 0001 | File name, size |
| | | Modifies the file name, file size, and last edit date and time. | 0002 | Batch modify |
| Delete file | 1205 | Deletes a file. | 0000 | — |
| Copy file | 1206 | Copies a file. | 0000 | — |

15.2 Defragmentation of Drive Memory

An external device performs the following functions for other station QnACPU drive which stores parameters and program files that the sequence programs are written.

Read drive memory usage status

Check the usage status of the drive memory (cluster usage status) of the specified drive.

Defragment drive memory

When the valid data written in the drive memory is not consecutive, increase the consecutive free area in cluster unit by defragmentation.



A cluster is the minimum unit when files are stored to drive memory (such as memory card) and the memory containing data is managed by FAT^{*1}.

The size of one cluster of each QnACPU drive is as follows:

- Internal memory : 4096 bytes
- Other memory : 512 bytes

For example, if a data less than 512 bytes is written to a memory card, one cluster of drive memory is used to write the data.

If 513 to 1024 bytes of data are written, two clusters of drive memory is used to write the data.

15

*1 An abbreviation for 'File Allocation Table'. A table used by the operating system to manage the location of files on the drive memory.

■Images of drive memory defragmentation

□: Cluster in not used (empty), ■: Cluster is being used (valid data is written in it)

- Before drive memory defragmentation

| Cluster | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|-------------------|-----------------|-----|-----|-----|-----|-----|----|----|----|----|----|----|----|----|----|-----------------|
| First 16 clusters | ■ ^{*1} | ■ | ■ | □ | □ | □ | □ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ ^{*2} |
| . | ■ | ■ | ■ | ■ | ■ | □ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | □ | □ |
| . | ■ | ■ | ■ | ■ | ■ | □ | □ | □ | □ | □ | □ | □ | □ | ■ | ■ | ■ |
| . | □ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | □ | □ | ■ | ■ | ■ | ■ | ■ | ■ |
| . | □ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | □ | □ | ■ | ■ | ■ | ■ | ■ | ■ |
| . | □ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | □ | □ | ■ | ■ | ■ | ■ | ■ | ■ |
| . | □ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | □ | □ | ■ | ■ | ■ | ■ | ■ | ■ |
| Last 16 clusters | □ ^{*3} | □ | □ | □ | □ | □ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | □ | ■ | □ |

↓

- After drive memory defragmentation

| Cluster | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|-------------------|-----------------|-----|-----|-----|-----|-----|----|----|----|----|----|----|----|----|----|-----------------|
| First 16 clusters | ■ ^{*1} | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ ^{*2} |
| . | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ |
| . | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ |
| . | □ | □ | □ | □ | □ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ |
| . | □ | □ | □ | □ | □ | □ | □ | □ | □ | □ | □ | □ | □ | □ | □ | □ |
| . | □ | □ | □ | □ | □ | □ | □ | □ | □ | □ | □ | □ | □ | □ | □ | □ |
| Last 16 clusters | □ ^{*3} | □ | □ | □ | □ | □ | □ | □ | □ | □ | □ | □ | □ | □ | □ | □ |

*1 16th cluster from the head cluster

*2 First cluster

*3 Last cluster

Data to be specified in commands

The following explains the data to be specified in commands when an external device performs defragmentation to the drive memory of QnACPU.

Keyword

A data to allow/prohibit the access to the drive, which is the character string (up to 6 characters) that are registered to the specified drive.

When a keyword has already registered, specify the same keyword.

■Data communication in ASCII code

Use the keyword registered in the specified drive and send as is.

■Data communication in binary code

Convert the keyword registered in the specified drive to 3-byte binary code, and send it from lower byte (L: bits 0 to 7).

Ex.

When the registered keyword is '012345'

| Values converted to binary code | Transmission order |
|---------------------------------|--------------------|
| 01H, 23H, 45H | 45H, 23H, 01H |

Ex.

When the keyword registered is '012300'

| Values converted to binary code | Transmission order |
|---------------------------------|--------------------|
| 01H, 23H, 00H | 00H, 23H, 01H |

■The content of data when a keyword is not registered in a specified drive

- Data communication in ASCII code: '000000'
- Data communication in binary code: '00H', '00H', '00H'

Setting flag

A data that indicates whether the keyword registered to the specified drive is specified to the keyword above.

■Data communication in ASCII code

Convert the numerical value to 2-digit ASCII code (hexadecimal), and send it from the upper digits.

■Data communication in binary code

Send 1-byte numerical value.

■Setting flag to be specified

| Setting value | Content to be specified |
|---------------|---|
| 00H | Keyword is invalid (specify a dummy) |
| 01H | Keyword is enabled (specify the keyword registered in the specified drive). |

Drive name

A data to perform reading usage status of drive memory, or specify the QnACPU drive to perform defragmentation.

■Data communication in ASCII code

Convert the numerical value that indicates access target drive to 4-digit ASCII code (hexadecimal) and send it from upper digits.

■Data communication in binary code

Send 2-byte numerical values that indicate access target drive from the lower byte (L: bits 0 to 7).

■Specified content of drive name (Other content cannot be specified.)

| Setting value | Target drive |
|---------------|---|
| 0000H | Internal memory (built-in RAM) |
| 0001H | RAM area of memory card A |
| 0002H | ROM area of memory card A |
| 0003H | RAM area of memory card B |
| 0004H | ROM area of memory card B |
| 000FH | Drive that the parameter files currently used (specified with the QnACPU dip switch) is stored. |

Cluster No.

A data to specify the first cluster No. within the range of drive memory whose usage status is to be read. Specify the data in multiples of 16 (00H, 10H, 20H... in hexadecimal notation).

■Data communication in ASCII code

Convert the cluster No. which is higher than '00H' to 4-digit (hexadecimal) ASCII code and send it from upper digits.

■Data communication in binary code

Send 2-byte numerical values that indicate cluster No. which is higher than '00H' from lower byte (L: bits 0 to 7).

■Drive memory defragmentation

When performing defragmentation of drive memory, specification of the cluster No. is not required.

Number of clusters to be read

A data to specify the number of clusters within the range of drive memory whose usage status is to be read. Specify the data in multiples of 16 (10H, 20H... in hexadecimal notation).

■Data communication in ASCII code

Convert the number of clusters '10H' to '100H' (16 to 256) to 4-digit ASCII code (hexadecimal) and send them from upper digits.

■Data communication in binary code

Send 2-byte numerical value that indicates the number of clusters from '10H' to '100H' (16 to 256) from lower byte (L: bits 0 to 7).

■Drive memory defragmentation

When performing defragmentation of drive memory, specification of the number of clusters to be read is not required.



Specify the number of clusters to be read within the range of the usable memory size after formatting the drive to be read from.

- Number of clusters = Usable memory size/Number of bytes in one cluster (4096 or 512)

For the number of bytes in one cluster, refer to the following section.

☞ Page 283 Defragment drive memory

Free cluster table

A data indicating the status of cluster usage. It is a response data to 'read drive memory usage status' (command: 0205).

■Data communication in ASCII code

The numerical value that indicates usage status is converted to n-digit (hexadecimal) ASCII code and sent to the external device. (16 clusters/4 digits)

■Data communication in binary code

An m-byte numerical value that indicates usage status is sent to the external device. (16 clusters/2 bytes)

■Content of free cluster table

The usage status of each cluster is indicated by one cluster per bit.

Ex.

0: Empty, 1: Used

| Cluster | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 | Values for 16 bits |
|-------------------|-----------------|-----|-----|-----|-----|-----|----|----|----|----|----|----|----|----|----|-----------------|--------------------|
| First 16 clusters | 0 ^{*1} | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 ^{*2} | 3CFFH |
| . | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 003FH | |
| . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | |
| . | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0000H | |
| Last 16 clusters | 0 ^{*3} | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0000H | |

*1 16th cluster from the head cluster

*2 First cluster

*3 Last cluster

When the usage status is as shown above, the content of free cluster table that is returned to the external device is as follows:

- When 32 clusters are returned while communicating in ASCII code
'3CFF003F' is returned and it is sent from '3'.
- When 32 clusters are returned while communicating in binary code
'FFH', '3CH', '3FH', and '00H' are returned and these are sent from 'FFF'.

■Drive memory defragmentation

When performing defragmentation of drive memory, the free cluster table is not returned.

Read drive memory usage status (command: 0205)

Read the usage status of the drive memory.



When creating a new file (registering new file), a consecutive free area for the file size to be created is required.

To calculate the capacity (size) of the consecutive free area of the specified drive, check the number of consecutive free clusters (number of consecutive OFF bits) by reading the usage status of that drive memory.

Capacity of consecutive free area (size) = number of consecutive free cluster × 4096 or 512 (byte)

When the consecutive area is insufficient, perform the following memory defragmentation.

Page 289 Defragment drive memory (command: 1207)

Message format

The following shows the message format of the request data and response data of the command.

■Request data

ASCII

| | | | | | | |
|----------------------------|----------------|-------------|------------------|----------------|-----------------|-------------------------------|
| 0 2 0 5 30H,32H,30H,35H | Subcommand | Keyword | Setting flag | Drive name | Cluster No. | Number of clusters to be read |
|----------------------------|----------------|-------------|------------------|----------------|-----------------|-------------------------------|

Binary

| | | | | | | |
|---------|----------------|------------------|-------------|----------------|-----------------|-------------------------------|
| 05H,02H | Subcommand | Setting flag | Keyword | Drive name | Cluster No. | Number of clusters to be read |
|---------|----------------|------------------|-------------|----------------|-----------------|-------------------------------|

■Response data

The free cluster table is stored. The order of data differs depending on the ASCII code or binary code.

Page 286 Free cluster table

Data specified by request data

■Command

| ASCII code | Binary code |
|----------------------------|-------------|
| 0 2 0 5 30H,32H,30H,35H | 05H,02H |

■Subcommand

| ASCII code | Binary code |
|----------------------------|-------------|
| 0 0 0 0 30H,30H,30H,30H | 00H,00H |

■Keyword

Set whether to allow/prohibit access to a drive. (Page 292 Keyword)

■Setting flag

This indicates whether the keyword registered in the specified drive matches the keyword of request data. (Page 284 Setting flag)

■Drive name

Read the usage status of the drive memory, and specify the QnACPU drive to be defragmented. (Page 285 Drive name)

■Cluster No.

Specify the first cluster No. within the range of drive memory whose usage status is to be read. (Page 285 Cluster No.)

■Number of clusters to be read

Specify the number of clusters within the range of drive memory whose usage status is to be read. ([Page 285 Number of clusters to be read](#))

Communication example

Read the usage status of the drive memory of the RAM area in a memory card A (drive name: 01H) for 32 clusters.

■Data communication in ASCII code

(Request data)

| Subcommand | Keyword | Setting flag | Drive name | Cluster No. | Number of clusters to be read | |
|-------------------------------|-------------------------------|---|-----------------|-------------------------------|-------------------------------|-------------------------------|
| 0 2 0 5 30H, 32H, 30H, 35H | 0 0 0 0 30H, 30H, 30H, 30H | 0 1 2 3 4 5 30H, 31H, 32H, 33H, 34H, 35H | 0 1 30H, 31H | 0 0 0 1 30H, 30H, 30H, 31H | 0 0 0 0 30H, 30H, 30H, 30H | 0 0 2 0 30H, 30H, 32H, 30H |

(Response data)

| | |
|-------------------------------|-------------------------------|
| Free cluster table | |
| Cluster No.0 to 15 | Cluster No.16 to 31 |
| 3 C F F 33H, 43H, 46H, 46H | 0 0 3 F 30H, 30H, 33H, 46H |

■Data communication in binary code

(Request data)

| Subcommand | Setting flag | Keyword | Drive name | Cluster No. | Number of clusters to be read |
|----------------------|----------------------|-----------------|----------------------|----------------------|-------------------------------|
| 05H, 02H 00H, 00H | 01H 45H, 23H, 01H | 01H 01H, 00H | 00H, 00H 00H, 00H | 20H, 00H 20H, 00H | |

(Response data)

| | |
|----------------------|---------------------|
| Free cluster table | |
| Cluster No.0 to 15 | Cluster No.16 to 31 |
| FFH, 3CH FFH, 00H | 3FH, 00H |

Defragment drive memory (command: 1207)

Perform the defragmentation of drive memory.

Perform the defragmentation of drive memory in the following situation.

- When the specified QnACPU is STOP.
- The usage status of drive memory (check with command 0205) is not consecutive and the files cannot be stored.

Message format

The following shows the message format of the request data and response data of the command.

■Request data

ASCII

| | | | | | | | |
|-----|-----|-----|-----|------------|---------|--------------|------------|
| 1 | 2 | 0 | 7 | Subcommand | Keyword | Setting flag | Drive name |
| 31H | 32H | 30H | 37H | | | | |

Binary

| | | | | |
|-----|------------|--------------|---------|------------|
| | Subcommand | Setting flag | Keyword | Drive name |
| 07H | 12H | | | |

■Response data

There is no response data for this command.

15



In the following cases, an error occurs and an abnormal response will be returned.

- When QnACPU is protected by system protection.
- When the keyword registered in the specified drive is not specified.
- When the drive memory is faulty (defective cluster, etc.).
- When the subdirectory is created by using an IC memory card read/writer.
- When the file is not stored in one consecutive area by using an IC memory card read/writer.

Data specified by request data

■Command

| ASCII code | Binary code |
|----------------------------------|-------------|
| 1 2 0 7 31H , 32H , 30H , 37H | 07H , 12H |

■Subcommand

| ASCII code | Binary code |
|----------------------------------|-------------|
| 0 0 0 0 30H , 30H , 30H , 30H | 00H , 00H |

■Keyword

Set whether to allow/prohibit access to a drive. (☞ Page 292 Keyword)

■Setting flag

This indicates whether the keyword registered in the specified drive matches the keyword of request data. (☞ Page 284 Setting flag)

■Drive name

Read the usage status of the drive memory, and specify the QnACPU drive to be defragmented. (☞ Page 285 Drive name)

Communication example

Perform defragmentation for the drive memory of the RAM area in a memory card A (drive name: 01H).

■Data communication in ASCII code

(Request data)

| Subcommand | Keyword | Setting flag | Drive name |
|-------------------------------|-------------------------------|---|---|
| 1 2 0 7 31H, 32H, 30H, 37H | 0 0 0 0 30H, 30H, 30H, 30H | 0 1 2 3 4 0 30H, 31H, 32H, 33H, 34H, 30H | 0 1 0 0 0 1 30H, 31H, 30H, 30H, 30H, 31H |

■Data communication in binary code

(Request data)

| Setting flag | Subcommand | Keyword | Drive name |
|--------------|------------|----------|------------------------------|
| | 07H, 12H | 00H, 00H | 01H, 40H, 23H, 01H, 01H, 00H |

15.3 File Control

Read, register, and delete files in QnACPU modules.

Use this function in the following situations:

- Read parameters and sequence programs from an external device
- Write the files read from CPU module to an external device according to the commands.

For the file name, extension, and location of a file stored in CPU module, refer to the user's manual (function explanation, program fundamentals) of the CPU module used.

Considerations for file control

The following are the considerations when controlling files of QnACPU.

Editing files

The files read from CPU module are used for storage on an external device.

The external device cannot edit the content of a file read from CPU module.

Data read/write

When the data of the entire file size cannot be read/written in a single communication, use several communications to read/write the data.

The file size can be verified using the following functions.

| Function | Reference |
|--------------------------------------|---|
| Read file information table function | Page 302 Read file information table without header statement (command: 0201) Page 305 Read file information table with header statement (command: 0202) |
| Read file presence function | Page 308 Read file presence (Search file) (command: 0203) |

System protection

If the CPU module is protected by system protection when using the following functions, an error occurs and an abnormal completion message is returned.

| Function | Reference |
|---|---|
| Create new file (Register file name) | Page 317 Create new file (Register file name) (command: 1202) |
| Write to file | Page 319 Write to file (command: 1203) |
| Delete file | Page 329 Delete file (command: 1205) |
| Copy file | Page 331 Copy file (command: 1206) |
| Modify file information (date created, attribute) | Page 323 Modify file information (command: 1204) |

Registration of keyword

When registering a keyword in the file, keep the registered keyword.

When accessing the following files, specifying the registered keyword when opening or reading/writing to that file is required.

- Parameter file
- Program files

File attribute

The file attribute is valid only when the following functions are used. For other commands, the attribute is handled as a dummy.

| Function | Reference |
|---|---|
| Read file information table function | Page 302 Read file information table without header statement (command: 0201) Page 305 Read file information table with header statement (command: 0202) |
| Create new file (Register file name) | Page 317 Create new file (Register file name) (command: 1202) |
| Modify file information (date created, attribute) | Page 323 Modify file information (command: 1204) |

Data to be specified in commands

This section explains the data to be specified in command when an external device controls files in the CPU module.

Keyword

Allow/prohibit the access to the specified drive.

When a keyword has already been registered, specify the same keyword.

For the content of the data, refer to the following section.

☞ Page 284 Keyword

Setting flag

A data that indicates whether the keyword registered to the specified drive is specified to the keyword above.

For the content of the data, refer to the following section.

☞ Page 284 Setting flag

Drive name, Drive No.

A data that specifies the drive of the CPU module whose files are to be controlled.

For the method on how to specify the drive name and its corresponding drive, refer to the following section.

☞ Page 285 Drive name

File No.

A value to specify the registered number when a file specified by file name and extension is registered (written) to CPU module, or the registration number when registering to CPU module.

■Data communication in ASCII code

Convert the file No. to 4-digit ASCII code (hexadecimal) and send it from upper digits.

■Data communication in binary code

Send the 2-byte^{*1} value that indicates file No. from the lower byte (L: bits 0 to 7).

*1 For C24, the additional code may be added. (☞ Page 33 Additional code (10H))

Ex.

1FH

| ASCII code | Binary code |
|----------------------------|-------------|
| 0 0 1 F 30H 30H 31H 46H | 1FH 00H |

■Specified content of file No.

| Value to be specified | Description | Content to be specified |
|-----------------------|------------------|--|
| 01H to 100H | File No. | Specify when the file No. is known. |
| FFFFH | File No. unknown | Specify when C24/E71 searches for file No. (The request to read/write data from/to C24/E71 will be delayed for more than one sequence scan time.) |



The file No. of the registered file can be checked using the following functions.

☞ Page 302 Read file information table without header statement (command: 0201)

☞ Page 305 Read file information table with header statement (command: 0202)

☞ Page 308 Read file presence (Search file) (command: 0203)

The unused file No. at registering new file can be checked using the following function.

☞ Page 310 Read file No. usage status (command: 0204)

Number of requested file, total number of registered files, number of file information

A data that indicates the number of request files, number of files registered in the specified drive, and number of files that return file information when reading file information.

■Data communication in ASCII code

Convert the numerical value described in the reference section relevant to the corresponding function to 4-digit ASCII code (hexadecimal) and send it from upper digits.

■Data communication in binary code

Send the 2-byte numerical value described in the reference section relevant to the corresponding function from each low byte (L: bits 0 to 7).

File name, extension, and attribute

■For file name, extension, and storage location that can be specified

For file name, extension, and storage location of files that can be specified in CPU modules, refer to the user's manual (function explanation, program fundamentals) of the CPU module used.



Files not described in the user's manual (function explanation, program fundamentals) of the CPU module used may also be shown for file control. Do not access them since they are for system use.

■When creating files or changing the file names

When creating a new file or changing the file name, specify the file name (up to 8 characters) and extension (3 characters) in accordance with the file naming rule of Engineering tool.

■File name, extension

For QnACPU, specify the file name by a file name + extension.

| Item | Specification method |
|-----------|--|
| File name | Specify the file name within 8 characters. If the file name is less than 8 characters, add space (code: 20H) after the file name and fill the characters to 8 characters. |
| Extension | Specify the extension within 3 characters. |

Ex.

When the file name is 'ABC.QPG'

File name: ABC □□□□□QPG (□ indicates a space)

■Attribute

Add an attribute a file.

Specify the attribute with 1 byte.

Specify '20H' as an attribute value (disk files that can be read/written) at a new file creation and when a dummy is specified.

When the attribute of the created file is not changed, the attribute of readable/writable disk file will be attached.



The attributes for the existing files can be checked using the following functions.

☞ Page 302 Read file information table without header statement (command: 0201)

☞ Page 305 Read file information table with header statement (command: 0202)

The attribute of the created files can be changed between '01H' (Read Only file) ⇔ '20H' (readable/writable disk file).

☞ Page 323 Modify file information (command: 1204)

Last edit time and last edit date

A data that indicates the date and time when the current file content was registered.

■Data communication in ASCII code

Convert the numerical value to a 4-digit (hexadecimal) ASCII code and send it from upper digits (time, year). When dummy is specified, send '0000H'.

■Data communication in binary code

Send 2-byte numerical value from lower bytes (L: bits 0 to 7). When dummy is specified, send '0000H'.

■Date (year, month, day)

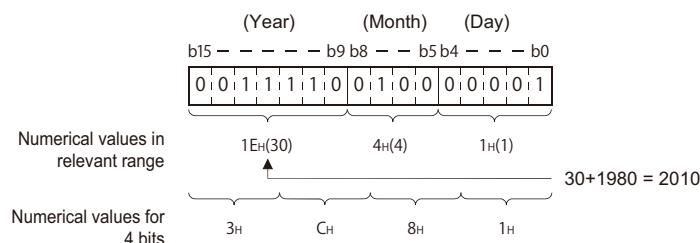
Specify year, month , and day with 16-bit value.

- Year: The binary value is represented with bits 9 to 15 by setting the year 1980 to '0'. *1
- Month: Specify the binary value in bit 5 to 8.
- Day: Specify the binary value in bit 0 to 4.

*1 This indicates incremental number of year by regarding 1980 as '0'.

Ex.

April 1st, 2010



| ASCII code | Binary code |
|--|---------------------------------|
| 3 C 8 1 33 _H 43 _H 38 _H 31 _H | 81 _H 3C _H |

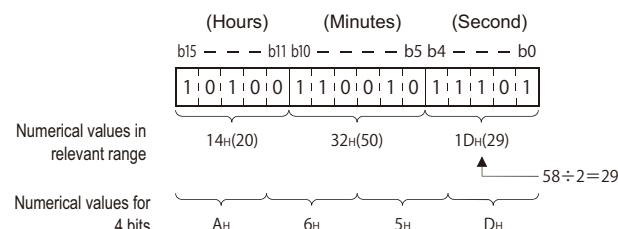
■Time (hour, minute, second)

Represent hour, minute, and second with 16-bit value.

- Hour: Represent binary value with bit 11 to 15.
- Minute: Represent binary value with bit 5 to 10.
- Second: Represent binary value ÷2 with bit 0 to 4.

Ex.

When 20:50:58



| ASCII code | Binary code |
|--|---------------------------------|
| A 6 5 D 41 _H 36 _H 35 _H 44 _H | 5D _H A6 _H |

File size

A data that indicates the current file size in number of bytes.

■Data communication in ASCII code

Convert the numerical value to 8-digit (hexadecimal) ASCII code and send it from the upper digits.

■Data communication in binary code

Send 4-byte^{*1} numerical from the lower byte (L: bits 0 to 7).

*1 For C24, the additional code may be added. (☞ Page 33 Additional code (10H))

Ex.

When the file size is 7168 bytes

| ASCII code | Binary code |
|--|-----------------------|
| 0 0 0 0 1 C 0 0 30H , 30H , 30H , 30H , 31H , 43H , 30H , 30H | 00H , 1CH , 00H , 00H |

Header statement

A header statement given to the file specified by Engineering tool supporting QnACPU (up to 32 characters).

■Data communication in ASCII code

Send data from the first character.

If the header statement is less than 32 characters, append a space (code: 20H).

15

Ex.

The header statement at the registration is '1 line-PC5'

The header statement becomes '1 line-PC5□□...', and is sequentially sent from '1'. (□: indicating a space.)

■Data communication in binary code

Use the character code of header statement as binary value, and send if from the first character.

When the header statement is less than 32 characters, 20H is added for the shortage.

Ex.

The header statement at registration is '1 line-PC5'

The header statement becomes 31H, 20H, 6CH, 69H, 6EH, 65H, 2DH, 50H, 43H, 35H, 20H, 20H..., and is sequentially sent from 31H.

Offset address

A data to specify the start address of the file range for which data is read and written.

Specify the address (one address/one byte) from the head (offset address: 0H) of each file with an even number.

■Data communication in ASCII code

Convert the address described in the reference section relevant to the corresponding function to 8-digit (hexadecimal) ASCII code and send it from the upper digits.

■Data communication in binary code

Send 4-byte numerical values described in the reference section relevant to the corresponding function from lower byte (L: bits 0 to 7).

■Offset addresses that can be specified

Check the file size (number of bytes) with the following read file information table functions, and calculate the offset address (0H to nH) by the size.

☞ Page 302 Read file information table without header statement (command: 0201)

☞ Page 305 Read file information table with header statement (command: 0202)

Number of bytes read, number of bytes written

A data to specify the number of bytes in the file range from/to which data is read/written. Specify it as one address per one byte.

■Data communication in ASCII code

Convert the numerical value described in the reference section of the relevant function to 4-digit ASCII code (hexadecimal) and send it from upper digits.

■Data communication in binary code

Send 2-byte numerical value described in the reference section of the relevant function from the lower byte (L: bits 0 to 7).

Read data, write data (for batch read and batch write functions)

A data read from or to be written to the file in the CPU module.

■Data communication in ASCII code

Convert one byte (one address) to 2-digit ASCII code (hexadecimal) and send it for the specified number of bytes from the upper digits.

■Data communication in binary code

Send the specified number of bytes as one byte for one address.

■Data order

- When reading data: Store the data to the external device without changing the order of data read from the CPU module.
- When writing data: Specify the data without changing the order of data read from CPU module.

Write data (identical data write function)

A data for the identical data write function used to wrote the identical data to the existing QnACPU files.

■Data communication in ASCII code

Convert the numerical value for one word to 4-digit ASCII code (hexadecimal) and send it from the upper digits.

■Data communication in binary code

Send the numerical value for one word from the lower byte (L: bits 0 to 7).

Capacity

A data to reserve the file area on the specified disk when registering a new file. Specify the data in the number of bytes.

■Data communication in ASCII code

Convert the numerical value which represents the area for specified file to be reserved in 2 words to 8-digit ASCII code (hexadecimal) and send it from the upper digits.

■Data communication in binary code

Send the numerical value which represents the area for the specified file to be reserved in 2 words from the lower byte (L: bits 0 to 7).



A new file with the same contents as the existing file can be registered from an external device.

The size of the relevant existing file must be checked using the read file information table function.

☞ Page 302 Read file information table without header statement (command: 0201)

☞ Page 305 Read file information table with header statement (command: 0202)

☞ Page 308 Read file presence (Search file) (command: 0203)

Fixed value

Specify '0'.

| ASCII code | Binary code |
|----------------------------|-------------|
| 0 0 0 0 30H 30H 30H 30H | 00H 00H |

Modification pattern (for changing file name and file size)

A data to specify which information to be modified when information (file name, size, date and time created) of the existing file is modified.

■Data communication in ASCII code

Convert the numerical value to 4-digit ASCII code (hexadecimal) and sent it from the upper digits.

■Data communication in binary code

Send 2-byte numerical value from the low byte (L: bits 0 to 7).

■Specified value and contents of modification pattern

| Bit | Description | Value to be specified |
|-----------|---|---|
| b0 | File name and extension specification existence | 0: Do not modify the file name and extension, 1: Modify the file name and extension |
| b1 | File attribute modification specification existence | 0: Do not modify the attribute, 1: Modify the attribute |
| b2 | Last edit time modification specification existence | 0: Do not modify the last edit time, 1: Modify the last edit time |
| b3 | Last edit date modification specification existence | 0: Do not modify the last edit date, 1: Modify the last edit date |
| b4 | Modification mode for date and time specification | When modifying date or time, specify which data to use. 0: Use date and time in a message, 1: Use date and time of QnACPU (specify dummy date and time in the message) |
| b5 | File size modification specification existence | 0: Do not modify the file size, 1: Modify the file size |
| b6 to b15 | (fixed to '0') | |

15

■Modifying the file creation date and time (command: 1204, subcommand: 0000)

04H, 08H, 0CH, 14H, 18H, or 1CH can be specified with 16 bits.

| Bit string | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---------------------|-----|-----|-----|-----|-----|-----|----|----|----|----|----|-----|-----|-----|----|----|
| Specification range | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1/0 | 1/0 | 1/0 | 0 | 0 |

■Modifying the file name and file size (command: 1204, subcommand: 0001)

01H, 02H, 03H, 20H, 21H, 22H, or 23H can be specified with 16 bits.

| Bit string | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---------------------|-----|-----|-----|-----|-----|-----|----|----|----|----|-----|----|----|----|-----|-----|
| Specification range | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1/0 | 0 | 0 | 0 | 1/0 | 1/0 |

■Modifying the file information in batch (command: 1204; subcommand: 0002)

The range from 01H to 3FH can be specified with 16 bits.

| Bit string | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---------------------|-----|-----|-----|-----|-----|-----|----|----|----|----|-----|-----|-----|-----|-----|-----|
| Specification range | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1/0 | 1/0 | 1/0 | 1/0 | 1/0 | 1/0 |

File No. usage

A data that shows the usage status of the file No. for 256 files to be returned to external devices when the usage status of file No. is read.

■Data communication in ASCII code

Convert the numerical value to 64-digit ASCII code (hexadecimal) and send it to the external device side. (File No. for 8 files: 2 digits)

■Data communication in binary code

Send the following 32-byte value which indicate the usage status to the external device side from the lower bytes (L: bits 0 to 7). (File No. for 8 files: 1 byte)

■Content of file No. usage

The usage status of each file No. is indicated as one file number per bit.

Ex.

File No. usage

0: Not used, 1: Used

| Bit string | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 | Values for 8 bits |
|------------|-----------------|----|----|----|----|----|----|-----------------|-------------------|
| 1st byte | 1 ^{*1} | 1 | 0 | 1 | 0 | 0 | 1 | 1 ^{*2} | D3H |
| 2nd byte | 1 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | B9H |
| 3rd byte | 0 | 1 | 1 | 1 | 1 | 0 | 1 | 0 | 7AH |
| 4th byte | 1 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 96H |
| 5th byte | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 71H |
| . | . | . | . | . | . | . | . | . | . |
| 31st byte | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | FFH |
| 32nd byte | 0 ^{*3} | 0 | 0 | 0 | 0 | 0 | 0 | 0 | FFH |

*1 Usage of file No. 8

*2 Usage of file No. 1

*3 Usage of file No. 256

The contents of file No. usage to be returned to external devices under the conditions above are shown below.

- Data communication in ASCII code: 'D3B97A' ... 'FFFF' are returned, and are sequentially send beginning from the 'D'.
- Data communication in binary code: 'D3H', 'B9H', '7AH' ... 'FFH', 'FFH' are returned, and sequentially send beginning from 'D3H'.

File lock mode

A data to specify whether or not a locked file is forcefully unlocked in order to allow access to the specified file from other devices.

■Data communication in ASCII code

Convert the numerical value to 4-digit ASCII code (hexadecimal) and sent it from the upper digits.

■Data communication in binary code

Send 2-byte numerical value from the low byte (L: bits 0 to 7).

■Specified value and contents of the file lock mode (Other values cannot be specified.)

| Specified value | Content to be specified |
|-----------------|------------------------------------|
| 0000H | Unlock the locked file normally. |
| 0002H | Unlock the locked file forcefully. |

■Unlock method for locked file

There are two methods for locking files:

| Method | Description |
|------------------|---|
| Normal execution | The locked file cannot be unlocked when other device has registered the file lock. When the file lock deregistration is requested, an error occurs and an abnormal response is returned. |
| Forced execution | The locked file is forcefully unlocked even if other device has registered the file lock. Use the forced execution function when a locked file cannot be unlocked because of problems in the device that registered the file lock. |

Copy mode

A data to specify whether or not the last edit data and time of the source file to the target file at the completion of copying.
The QnACPU management time at a new file creation will remain when the date and time are not copied.

■Data communication in ASCII code

Convert the numerical value to 4-digit ASCII code (hexadecimal) and sent it from the upper digits.

■Data communication in binary code

Send 2-byte numerical value from the low byte (L: bits 0 to 7).

■Specified value and contents of the copy mode

| Setting value | Content to be specified |
|---------------|--|
| 0000H | Do not copy the last edit date and time of the source file at completion of copying. |
| 0001H | Copy the last edit date and time of the source file at completion of copying. |

File control execution procedure for the QnACPU

The following shows the procedures for executing file control on QnACPU.

Procedure when reading contents of file

1. Check for file existence.

Check with any of the following commands.

- ☞ Page 302 Read file information table without header statement (command: 0201)
- ☞ Page 305 Read file information table with header statement (command: 0202)
- ☞ Page 308 Read file presence (Search file) (command: 0203)

2. Prohibit access from other devices. (Prohibit file content modification.)

- ☞ Page 315 Register and deregister file locks (command: 0808)

3. Read all content of the file.

- ☞ Page 312 Read file (command: 0206)

4. Allow access from other devices. (Allow file content modification.)

- ☞ Page 315 Register and deregister file locks (command: 0808)



Store the following file information of the file (for storage) read to an external device.

- File No.
- File name and file attribute
- File size

Procedure when overwriting data to existing file (changing file information)

1. Check for file existence.

Check with any of the following commands.

- ☞ Page 302 Read file information table without header statement (command: 0201)
- ☞ Page 305 Read file information table with header statement (command: 0202)
- ☞ Page 308 Read file presence (Search file) (command: 0203)

2. Prohibit access from other devices. (Prohibit modification of file.)

- ☞ Page 315 Register and deregister file locks (command: 0808)

3. Before changing the file size, change the QnACPU status to STOP. *1

4. Write data when overwriting or copying data, or changing file size.

- ☞ Page 319 Write to file (command: 1203)

5. Modify the file creation date and time.

- ☞ Page 323 Modify file information (command: 1204)

6. Allow access from other devices. (Allow file content modification.)

- ☞ Page 315 Register and deregister file locks (command: 0808)

*1 The file size can be changed using the file information modification function (☞ Page 323 Modify file information (command: 1204)) only when the file size is made smaller. When the file size must be made larger, write data in the same procedure as create a new file. (☞ Page 301 Procedure for creating a new file and write data to it (creating a file by copying))

Procedure for creating a new file and write data to it (creating a file by copying)

1. Check for file existence.

Check with any of the following commands.

- [Page 302 Read file information table without header statement \(command: 0201\)](#)
- [Page 305 Read file information table with header statement \(command: 0202\)](#)
- [Page 308 Read file presence \(Search file\) \(command: 0203\)](#)

2. Check unused file No. only when copying a file.

- [Page 310 Read file No. usage status \(command: 0204\)](#)

3. Check the capacity of consecutive free area.

- [Page 287 Read drive memory usage status \(command: 0205\)](#)

4. Delete an unnecessary file when free area is insufficient. Or, increase consecutive free area.*1

- [Page 301 Procedure for deleting files](#)
- [Page 289 Defragment drive memory \(command: 1207\)](#)

5. Register a file name and reserve the capacity.

- [Page 317 Create new file \(Register file name\) \(command: 1202\)](#)

6. Prohibit access from other devices. (Prohibit file content modification.)

- [Page 315 Register and deregister file locks \(command: 0808\)](#)

7. Write data to the file.

Write data with any of the following commands.

- [Page 319 Write to file \(command: 1203\)](#)
- [Page 331 Copy file \(command: 1206\)](#)

8. Modify the file creation date and time. (Can be skipped.)

- [Page 323 Modify file information \(command: 1204\)](#)

9. Allow access from other devices. (Allow file content modification.)

- [Page 315 Register and deregister file locks \(command: 0808\)](#)

10. Check the file No. of the created file only when creating a new file.

- [Page 308 Read file presence \(Search file\) \(command: 0203\)](#)

*1 Use the remote STOP function ([Page 169 Remote STOP \(command: 1002\)](#)) to place the QnACPU into the STOP status before defragmenting the memory. After the completion of the processing of this section, the remote RUN function ([Page 167 Remote RUN \(command: 1001\)](#)) can be used to set the QnACPU to the RUN status.

Procedure for deleting files

1. Check for file existence.

Check with any of the following commands.

- [Page 302 Read file information table without header statement \(command: 0201\)](#)
- [Page 305 Read file information table with header statement \(command: 0202\)](#)
- [Page 308 Read file presence \(Search file\) \(command: 0203\)](#)

2. Delete a file.*1

- [Page 329 Delete file \(command: 1205\)](#)

*1 Determine the file deletion timing for the entire system, including the QnACPU and related devices.

Read file information table without header statement (command: 0201)

The following examples show the control procedure for reading file information table without header statement within the range of the specified file No.

Message format

The following shows the message format of the request data and response data of the command.

■Request data

ASCII

| | | | | | | |
|----------------------------|------------|---------|--------------|------------|----------|---------------------------|
| 0 2 0 1 30H 32H 30H 31H | Subcommand | Keyword | Setting flag | Drive name | File No. | Number of requested files |
|----------------------------|------------|---------|--------------|------------|----------|---------------------------|

Binary

| | | | | | | | |
|-----|-----|------------|--------------|---------|------------|----------|---------------------------|
| 01H | 02H | Subcommand | Setting flag | Keyword | Drive name | File No. | Number of requested files |
|-----|-----|------------|--------------|---------|------------|----------|---------------------------|

■Response data

| | | | | | | | | | | |
|----------------------------------|----------------------------|----------|-----------|-----------|-----------|----------------|----------------|-----------|-----|------------------------------|
| Total number of registered files | Number of file information | File No. | File name | Extension | Attribute | Last edit time | Last edit date | File size | ... | File information (nth point) |
| File information (first point) | | | | | | | | | | |

Data specified by request data

■Command

| ASCII code | Binary code |
|----------------------------|-------------|
| 0 2 0 1 30H 32H 30H 31H | 01H 02H |

■Subcommand

| ASCII code | Binary code |
|----------------------------|-------------|
| 0 0 0 0 30H 30H 30H 30H | 00H 00H |

■Keyword

Allow/prohibit the access to the file of the target memory. (☞ Page 292 Keyword)

■Setting flag

Data that indicates whether the keyword is registered in the specified drive. (☞ Page 292 Setting flag)

■Drive name

A data to specify the drive of CPU module which controls files. (☞ Page 292 Drive name, Drive No.)

■File No.

A data to specify the registration number of files. (☞ Page 292 File No.)

Specify within the following range.

1 ≤ File No. ≤ 256

■Number of requested file

Number of request files when reading file information. (☞ Page 293 Number of requested file, total number of registered files, number of file information)

Specify within the following range.

1 ≤ Number of requested file ≤ 36

Data stored by response data

■Total number of registered files

Number of files registered in the specified drive. (☞ Page 293 Number of requested file, total number of registered files, number of file information)

The value is returned within the following range.

1 ≤ Total number of registered files ≤ 256

■Number of file information

Number of files that return file information when reading the file information(☞ Page 293 Number of requested file, total number of registered files, number of file information)

The value is returned within the following range.

0 ≤ Number of file information ≤ Number of requested file

(0: No registered files after the specified head file No.)

■File No.

This indicates the registration number of a file. (☞ Page 292 File No.)

15

■File name

Character string data of a file name. (☞ Page 293 File name, extension, and attribute)

■Extension

A character string data of extension of a file. (☞ Page 293 File name, extension, and attribute)

■Attribute

A data that indicates the file attribute (applicability of data read/write). (☞ Page 293 File name, extension, and attribute)

■Last edit time and last edit date

A data that indicates the date and time when the current data was registered. (☞ Page 294 Last edit time and last edit date)

■File size

A data that indicates the capacity of the current file in number of bytes. (☞ Page 295 File size)



- The total number of registered files is the current total number of files registered in the specified drive.
- When all the files are not registered within the specified file number range, the number of file information will be the number of files registered in the specified range (number of file information to be returned).

Communication example

Read the file information table without header statement in following conditions.

- Drive name: 1 (RAM area of memory card A)
- File No.: 10
- Number of requested file: 2

■Data communication in ASCII code

(Request data)

| Subcommand | Keyword | Setting flag | Drive name | File No. | Number of requested files |
|----------------------------|----------------------------|--|--|----------------------------|---------------------------|
| 0 2 0 1 30H 32H 30H 31H | 0 0 0 0 30H 30H 30H 30H | 0 1 2 0 0 0 30H 31H 32H 30H 30H 30H | 0 1 0 0 0 1 30H 31H 30H 30H 31H 31H | 0 0 0 A 30H 30H 30H 41H | 0 0 0 2 30H 30H 32H |

(Response data)

| Total number of registered files | Number of file information | File information 1 (Information of file No. 10) | | File information 2 (Information of file No. 11) | |
|---|--|---|----------------------------|---|--|
| 0 0 1 F 30H 30H 31H 46H | 0 0 0 2 30H 30H 30H 02H | | | | |
| <hr/> | | | | | |
| File No. | File name | Extension | Attribute | Time of last update | Date of last update |
| 0 0 0 A 30H 30H 30H 41H | A B C D 1 2 41H 42H 43H 44H 31H 32H 20H 20H | Q P G 51H 50H 47H | A 6 5 D 41H 36H 35H 44H | 2 0 2 8 32H 30H 32H 38H | 0 0 0 1 C 0 0 30H 30H 30H 31H 43H 30H 30H |
| (20 hours 50 minutes 58 seconds) (January 8, 1996) | | | | | |
| File No. | File name | Extension | Attribute | Time of last update | Date of last update |
| 0 0 0 B 30H 30H 30H 42H | E F G H 5 45H 46H 47H 48H 35H 20H 20H 20H | Q P G 51H 50H 47H | 5 B C 0 35H 42H 43H 30H | 2 1 8 2 32H 31H 38H 32H | 0 0 0 1 3 C 8 30H 30H 30H 31H 33H 43H 38H |
| (11 hours 30 minutes 00 seconds) (December 2, 1996) | | | | | |

■Data communication in binary code

(Request data)

| Setting flag | File No. | Subcommand | | Keyword | Drive name | Number of requested files |
|--|----------|------------|--|---------|------------|---------------------------|
| 01H 02H 00H 00H 01H 00H 20H 01H 01H 00H 0A0H 00H 02H 00H | | | | | | |

(Response data)

| Total number of registered files | Number of file information | File information 1 (Information of file No. 10) | | File information 2 (Information of file No. 11) | |
|---|--|---|----------------------|---|-----------------|
| 1FH 00H 02H 00H 0A0H 00H | 0 0 1 F 41H 42H 43H 44H 31H 32H 20H 20H | File No. | File name | Extension | Attribute |
| <hr/> | | | | | |
| | | A B C D 1 2 41H 42H 43H 44H 31H 32H 20H 20H | Q P G 51H 50H 47H | 20H 5DH A6H 28H 20H | 00H 1CH 00H 00H |
| (20 hours 50 minutes 58 seconds) (January 8, 1996) | | | | | |
| File No. | File name | Extension | Attribute | Last edit time | File size |
| 0B0H 00H 45H 46H 47H 48H 35H 20H 20H 20H | E F G H 5 45H 46H 47H 48H 35H 20H 20H | Q P G 51H 50H 47H | 20H C0H 5BH 82H 21H | C8H 13H 00H 00H | |
| (11 hours 30 minutes 00 seconds) (December 2, 1996) | | | | | |

Read file information table with header statement (command: 0202)

The following examples show the control procedure for reading file information table with header statement within the range of the specified file No.

Message format

The following shows the message format of the request data and response data of the command.

■Request data

ASCII

| | | | | | | |
|----------------------------|----------------|-------------|------------------|----------------|--------------|---------------------------|
| 0 2 0 2 30H 32H 30H 32H | Subcommand | Keyword | Setting flag | Drive name | File No. | Number of requested files |
|----------------------------|----------------|-------------|------------------|----------------|--------------|---------------------------|

Binary

| | | | | | | |
|------------|----------------|------------------|-------------|----------------|--------------|---------------------------|
| 02H 02H | Subcommand | Setting flag | Keyword | Drive name | File No. | Number of requested files |
|------------|----------------|------------------|-------------|----------------|--------------|---------------------------|

■Response data

| | | | | | | | | | | | |
|----------------------------------|----------------------------|----------|-----------|-----------|-----------|----------------|----------------|-----------|------------------|-----|------------------------------|
| Total number of registered files | Number of file information | File No. | File name | Extension | Attribute | Last edit time | Last edit date | File size | Header statement | ... | File information (nth point) |
| File information (first point) | | | | | | | | | | | |

Data specified by request data

■Command

| ASCII code | Binary code |
|----------------------------|-------------|
| 0 2 0 2 30H 32H 30H 32H | 02H 02H |

■Subcommand

| ASCII code | Binary code |
|----------------------------|-------------|
| 0 0 0 0 30H 30H 30H 30H | 00H 00H |

■Keyword

Allow/prohibit the access to the file of the target memory. (☞ Page 292 Keyword)

■Setting flag

Data that indicates whether the keyword is registered in the specified drive. (☞ Page 292 Setting flag)

■Drive name

A data to specify the drive of CPU module which controls files. (☞ Page 292 Drive name, Drive No.)

■File No.

A data to specify the registration number of files. (☞ Page 292 File No.)

Specify within the following range.

1 ≤ File No. ≤ 256

■Number of requested file

Number of request files when reading file information. (☞ Page 293 Number of requested file, total number of registered files, number of file information)

Specify within the following range.

1 ≤ Number of requested file ≤ 16

Data stored by response data

■Total number of registered files

Number of files registered in the specified drive. (☞ Page 293 Number of requested file, total number of registered files, number of file information)

The value is returned within the following range.

1 ≤ Total number of registered files ≤ 256

■Number of file information

Number of files that return file information when reading the file information(☞ Page 293 Number of requested file, total number of registered files, number of file information)

The value is returned within the following range.

0 ≤ Number of file information ≤ Number of requested file

(0: No registered files after the specified head file No.)

■File No.

This indicates the registration number of a file. (☞ Page 292 File No.)

■File name

Character string data of a file name. (☞ Page 293 File name, extension, and attribute)

■Extension

A character string data of extension of a file. (☞ Page 293 File name, extension, and attribute)

■Attribute

A data that indicates the file attribute (applicability of data read/write). (☞ Page 293 File name, extension, and attribute)

■Last edit time and last edit date

A data that indicates the date and time when the current data is registered. (☞ Page 294 Last edit time and last edit date)

■File size

A data that indicates the capacity of the current file in number of bytes. (☞ Page 295 File size)

■Header statement

The header statement attached to the file specified with Engineering tool. (☞ Page 295 Header statement)



- The total number of registered files is the current total number of files registered in the designated drive.
- When all the files are not registered within the specified file number range, the number of file information will be the number of files registered in the specified range (number of file information to be returned).

Communication example

Read the file information table with header statement in following conditions.

- Drive name: 1 (RAM area of memory card A)
- File No.: 10
- Number of requested file: 2

■Data communication in ASCII code

(Request data)

| Subcommand | Keyword | Setting flag | Drive name | File No. | Number of requested files |
|-------------------------------|-------------------------------|---|---|-------------------------------|---------------------------|
| 0 2 0 2 30H, 32H, 30H, 32H | 0 0 0 0 30H, 30H, 30H, 30H | 0 1 2 0 0 0 30H, 31H, 32H, 30H, 30H, 30H | 0 1 0 0 0 1 30H, 31H, 30H, 30H, 30H, 31H | 0 0 0 A 30H, 30H, 30H, 41H | 0 0 0 2 30H, 30H, 32H |

(Response data)

Total number of registered files Number of file information

| | | |
|-------------------------------|-------------------------------|-------|
| 0 0 1 F 30H, 30H, 31H, 46H | 0 0 0 2 30H, 30H, 30H, 02H | - - - |
|-------------------------------|-------------------------------|-------|

File information 1 (Information of file No. 10)

| File No. | File name | Extension | Attribute | Last edit time | Last edit date | File size |
|-------------------------------|---|------------------------|------------------------------------|-------------------------------|--|-----------------------------|
| 0 0 0 A 30H, 30H, 30H, 41H | A B C D 1 2 41H, 42H, 43H, 44H, 31H, 32H, 20H, 20H | Q P G 51H, 50H, 47H | A 6 5 D 20H, 41H, 36H, 35H, 44H | 2 0 2 8 32H, 30H, 32H, 38H | 0 0 0 1 30H, 30H, 30H, 31H, 43H, 30H, 30H | C 0 0 30H, 30H, 30H, 30H |

(20 hours 50 minutes 58 seconds) (January 8, 1996)

Header statement (32 characters) File information 2 (Information of file No. 11)

| | |
|-------------------------------|--|
| A B ... 41H, 42H, 20H, 20H | (The order of data items is the same as that of file information 1.) |
|-------------------------------|--|

15

■Data communication in binary code

(Request data)

| Setting flag | File No. | Number of requested files |
|---|----------|---------------------------|
| Subcommand | Keyword | Drive name |
| 02H, 02H, 00H, 01H, 00H, 20H, 01H, 01H, 00H, 0AH, 00H, 02H, 00H | | |

(Response data)

Total number of registered files Number of file information

File information 1 (Information of file No. 10)

| File No. | File name | Extension | Attribute | Last edit time | Last edit date | File size |
|--|---|------------------------|---|--|----------------|-----------|
| 1FH, 00H, 02H, 00H, 0AH, 00H, 41H, 42H, 43H, 44H, 31H, 32H, 20H, 20H | A B C D 1 2 41H, 42H, 43H, 44H, 31H, 32H, 20H, 20H | Q P G 51H, 50H, 47H | 20H, 5DH, A6H, 28H, 20H, 00H, 1CH, 00H, 00H | (20 hours 50 minutes 58 seconds) (January 8, 1996) | | |

Header statement (32 characters) File information 2 (Information of file No. 11)

| | |
|-------------------------------|--|
| A B ... 41H, 42H, 20H, 20H | (The order of data items is the same as that of file information 1.) |
|-------------------------------|--|

Read file presence (Search file) (command: 0203)

The following examples show the control procedure for searching for the specified file, and reading its file No. and file size.

Message format

■Request data

ASCII

| | | | | | | | |
|----------------------------|------------|---------|--------------|------------|-----------|-----------|-----------|
| 0 2 0 3 30H 32H 30H 33H | Subcommand | Keyword | Setting flag | Drive name | File name | Extension | Attribute |
|----------------------------|------------|---------|--------------|------------|-----------|-----------|-----------|

Binary

| | | | | | | | |
|---------|------------|--------------|---------|------------|-----------|-----------|-----------|
| 03H 02H | Subcommand | Setting flag | Keyword | Drive name | File name | Extension | Attribute |
|---------|------------|--------------|---------|------------|-----------|-----------|-----------|

■Response data

| | |
|----------|-----------|
| File No. | File size |
|----------|-----------|



If the specified file does not exist, an error occurs and the end code at the occurrence of error is returned.

Data specified by request data

■Command

| ASCII code | Binary code |
|----------------------------|-------------|
| 0 2 0 3 30H 32H 30H 33H | 03H 02H |

■Subcommand

| ASCII code | Binary code |
|----------------------------|-------------|
| 0 0 0 0 30H 30H 30H 30H | 00H 00H |

■Keyword

Allow/prohibit the access to the file of the target memory. (☞ Page 292 Keyword)

■Setting flag

Data that indicates whether the keyword is registered in the specified drive. (☞ Page 292 Setting flag)

■Drive name

A data to specify the drive of CPU module which controls files. (☞ Page 292 Drive name, Drive No.)

■File name

Character string data of a file name. (☞ Page 293 File name, extension, and attribute)

■Extension

A character string data of extension of a file. (☞ Page 293 File name, extension, and attribute)

■Attribute

A data to specify the file attribute (applicability of data read/write). (☞ Page 293 File name, extension, and attribute)

Handle the file attribute to be read as dummy data.

Data stored by response data

■File No.

This indicates the registration number of a file. (☞ Page 292 File No.)

■File size

This value indicates the capacity of the current file in number of bytes. (☞ Page 295 File size)

Communication example

Read the existence of a specified file under the following condition.

- Drive name: 1 (RAM area of memory card A)
- File name: ABC12.QPG

■Data communication in ASCII code

(Request data)

| Subcommand | Keyword | Setting flag | Drive name | File name | Extension | Attribute |
|-------------------------------|-------------------------------|---|-----------------|-------------------------------|---|----------------------------|
| 0 2 0 3 30H, 32H, 30H, 33H | 0 0 0 0 30H, 30H, 30H, 30H | 0 1 2 3 0 0 30H, 31H, 32H, 33H, 30H, 30H | 0 1 30H, 31H | 0 0 0 1 30H, 30H, 30H, 31H | A B C 1 2 □ □ □ 41H, 42H, 43H, 31H, 32H, 20H, 20H, 20H | Q P G 51H, 50H, 47H 20H |

(Response data)

| File No. | File size |
|-------------------------------|--|
| 0 0 0 1 30H, 30H, 30H, 31H | 0 0 0 0 1 C 0 0 30H, 30H, 30H, 31H, 43H, 30H, 30H |

15

■Data communication in binary code

(Request data)

| Setting flag | Subcommand | Keyword | Drive name | File name | Extension | Attribute |
|--------------|--------------------------------|----------------------|--|-----------|----------------------|-----------|
| | 03H, 02H 30H, 30H, 30H, 30H | 01H 00H, 23H, 01H | 01H, 00H 41H, 42H, 43H, 31H, 32H, 20H, 20H, 20H | | 51H, 50H, 47H 20H | |

(Response data)

| File No. | File size |
|--------------------------------|-----------|
| 01H, 00H 00H, 1CH, 00H, 00H | |

Read file No. usage status (command: 0204)

The following examples show the control procedure for reading the usage status of a file No.

Message format

■Request data

ASCII

| | | | | | | | |
|-----|-----|-----|-----|------------|---------|--------------|------------|
| 0 | 2 | 0 | 4 | Subcommand | Keyword | Setting flag | Drive name |
| 30H | 32H | 30H | 34H | | | | |

Binary

| | | | | | |
|-----|-----|------------|--------------|---------|------------|
| | | Subcommand | Setting flag | Keyword | Drive name |
| 04H | 02H | | | | |

■Response data

The file No. of usage status is stored. (☞ Page 298 File No. usage)



When a drive memory that cannot store maximum of 256 files is specified, the file No. of which file cannot be stored (shortage) will be in use (the corresponding bit will be '1').

Data specified by request data

■Command

| ASCII code | Binary code |
|----------------------------|-------------|
| 0 2 0 4 30H 32H 30H 34H | 04H 02H |

■Subcommand

| ASCII code | Binary code |
|----------------------------|-------------|
| 0 0 0 0 30H 30H 30H 30H | 00H 00H |

■Keyword

Allow/prohibit the access to the file of the target memory. (☞ Page 292 Keyword)

■Setting flag

Data that indicates whether the keyword is registered in the specified drive. (☞ Page 292 Setting flag)

■Drive name

A data to specify the drive of CPU module which controls files. (☞ Page 292 Drive name, Drive No.)

Communication example

Read the file No. usage in the following conditions.

- Drive name: 1 (RAM area of memory card A)

■Data communication in ASCII code

(Request data)

| Subcommand | Keyword | Setting flag | Drive name |
|-------------------------------|-------------------------------|---|--|
| 0 2 0 4 30H, 32H, 30H, 34H | 0 0 0 0 30H, 30H, 30H, 30H | 0 1 2 3 0 0 30H, 31H, 32H, 33H, 30H, 30H | 0 1 0 0 0 1 30H, 31H, 30H, 30H, 31H |

(Response data)

| File No. usage status (64 characters) | | | | | | | |
|---------------------------------------|--|-----------------|-----------------|--|---------------|--|--|
| D 3 44H, 33H | B 9 7 A 9 6 42H, 39H, 37H, 41H, 39H, 36H, ... | F F 46H, 46H | F F 46H, 46H | | | | |
| No.1 to 8 | | No.9 to No.248 | | | No.249 to 256 | | |

■Data communication in binary code

(Request data)

| Setting flag | Subcommand | Keyword | Drive name |
|--------------|------------|----------|------------------------------|
| | 04H, 02H | 00H, 00H | 01H, 00H, 23H, 01H, 01H, 00H |

(Response data)

| File No. usage status (32 bytes) | | | | | | | |
|----------------------------------|-----|----------------|-----|-----|---------------|-----|--|
| D3H | B9H | 7AH | 96H | ... | FFH | FFH | |
| No.1 to 8 | | No.9 to No.248 | | | No.249 to 256 | | |

Read file (command: 0206)

The following examples show the control procedure for reading the data written in the specified file.

Point

The maximum number of data bytes to be read per one time is fixed. Read all the data written to the specified file with adjusting the offset address and number of bytes read, and dividing them for several times.

Additionally, save the read data to the external device.

Check the file size with the following function.

☞ Page 302 Read file information table without header statement (command: 0201)

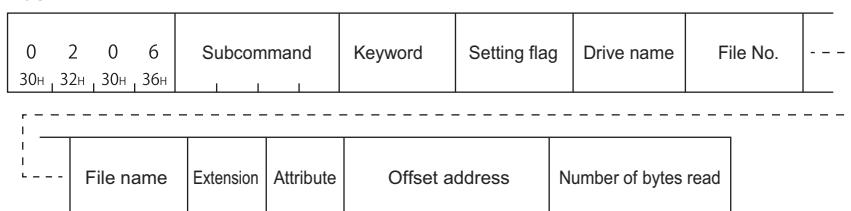
☞ Page 305 Read file information table with header statement (command: 0202)

☞ Page 308 Read file presence (Search file) (command: 0203)

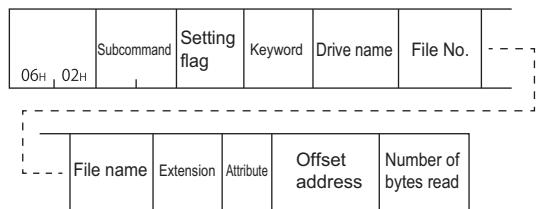
Message format

■Request data

ASCII



Binary



■Response data

The read data are stored.

☞ Page 296 Read data, write data (for batch read and batch write functions)

Data specified by request data

■Command

| ASCII code | Binary code |
|----------------------------------|-------------|
| 0 2 0 6 30H , 32H , 30H , 36H | 06H , 02H |

■Subcommand

| ASCII code | Binary code |
|----------------------------------|-------------|
| 0 0 0 0 30H , 30H , 30H , 30H | 00H , 00H |

■Keyword

Allow/prohibit the access to the file of the target memory. (☞ Page 292 Keyword)

■Setting flag

Data that indicates whether the keyword is registered in the specified drive. (☞ Page 292 Setting flag)

■Drive name

A data to specify the drive of CPU module which controls files. (☞ Page 292 Drive name, Drive No.)

■File No.

A data to specify the registration number of files. (☞ Page 292 File No.)

Specify within the following range.

1 ≤ File No. ≤ 256

■File name

Character string data of a file name. (☞ Page 293 File name, extension, and attribute)

■Extension

A character string data of extension of a file. (☞ Page 293 File name, extension, and attribute)

■Attribute

A data to specify the file attribute (applicability of data read/write). (☞ Page 293 File name, extension, and attribute)

Handle the file attribute to be read as dummy data.

■Offset address

A data to specify the start address of the range from which data is read to a file. (☞ Page 295 Offset address)

Specify within the following range.

0 ≤ Address (specify with even address) ≤ (File size - 1)

■Number of bytes read

A data to specify the number of bytes of the range from which data is read to a file.

Specify within the following range.

0 ≤ Number of bytes ≤ 960

Communication example

Read the data written in the specified file under the following condition.

- Drive name: 1 (RAM area of memory card A)
- File No.: 1
- File name: ABC12.QPG
- Offset address: 100H
- Number of bytes read: 50 bytes

■Data communication in ASCII code

(Request data)

| Subcommand | Keyword | Setting flag | Drive name | File No. | |
|---|--|---|-------------------------------|-------------------------------|-------------------------------|
| 0 2 0 6 30H, 32H, 30H, 36H | 0 0 0 0 30H, 30H, 30H, 30H | 0 1 2 3 0 0 30H, 31H, 32H, 33H, 30H, 30H | 0 1 30H, 31H | 0 0 0 1 30H, 30H, 30H, 31H | 0 0 0 1 30H, 30H, 30H, 31H |
| File name | | Extension | Attribute | Offset address | Number of bytes read |
| A B C 1 2 <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> 41H, 42H, 43H, 31H, 32H, 20H, 20H | Q P G <input type="checkbox"/> 51H, 50H, 47H, 20H | 0 0 0 0 1 0 0 30H, 30H, 30H, 30H, 31H, 30H | 0 0 3 2 30H, 30H, 33H, 32H | | |

(Response data)

| Data read (For 50 bytes) | | | | | | | | | |
|--------------------------|--|-----------------|--|--|--|--|--|--|--|
| 1 3 31H, 33H | 5 F 0 C 0 2 ⋯ 0 1 0 B 35H, 46H, 30H, 43H, 30H, 32H, ⋯, 30H, 31H, 30H, 42H, 31H, 45H | 1 E 31H, 45H | | | | | | | |

Address 100H Address 101H to 130H Address 131H

■Data communication in binary code

(Request data)

| Setting flag | File No. | | | |
|----------------|-----------|--------------------|----------------------|--|
| Subcommand | Keyword | Drive name | File name | |
| 06H, 02H | 00H, 00H | 01H | 00H, 23H, 01H | 01H, 00H, 41H, 42H, 43H, 31H, 32H, 20H, 20H, 20H |
| Offset address | | | | |
| Extension | Attribute | | Number of bytes read | |
| 51H, 50H, 47H | 20H | 00H, 01H, 00H, 00H | 32H, 00H | |

(Response data)

| Data read (For 50 bytes) | | | | |
|--|----------------------|--------------|--|--|
| 13H, 5FH, 0CH, 02H, ⋯, 01H, 0BH, 1EH Address 100H | Address 101H to 130H | Address 131H | | |

Register and deregister file locks (command: 0808)

The following examples show the control procedure for registering and deregistering a file lock.



Restarting QnACPU (such as resetting CPU module) with the file lock registered results in the file lock deregistered state.

Message format

■Request data

| Command | Subcommand | File lock mode | Drive name | File No. | File name | Extension | Attribute |
|---------|------------|----------------|------------|----------|-----------|-----------|-----------|
| | | | | | | | |

■Response data

There is no response data for this command.

Data specified by request data

■Command

| ASCII code | Binary code |
|-------------------------------|-------------|
| 0 8 0 8 30H, 38H, 30H, 38H | 08H, 08H |

15

■Subcommand

- Register

| ASCII code | Binary code |
|-------------------------------|-------------|
| 0 0 0 1 30H, 30H, 30H, 31H | 01H, 00H |

- Deregister

| ASCII code | Binary code |
|-------------------------------|-------------|
| 0 0 0 0 30H, 30H, 30H, 30H | 00H, 00H |

■File lock mode

A data to specify whether a locked file is to be forcefully unlocked in order to allow other devices to access the specified file.

(☞ Page 298 File lock mode)

■Drive name

A data to specify the drive of CPU module which controls files. (☞ Page 292 Drive name, Drive No.)

■File No.

A data to specify the registration number of files. (☞ Page 292 File No.)

■File name

Character string data of a file name. (☞ Page 293 File name, extension, and attribute)

■Extension

A character string data of extension of a file. (☞ Page 293 File name, extension, and attribute)

■Attribute

A data to specify the file attribute (applicability of data read/write). (☞ Page 293 File name, extension, and attribute)

The attribute assigned by a new file creation, etc. is valid as the attribute of the file that will be locked and unlocked. When registering and deregistering file lock, handle the file attribute as dummy data.

Communication example (Register)

Read the existence of a specified file under the following condition.

- File lock mode: 0 (usual practice)
- Drive name: 1 (RAM area of memory card A)
- File No.: 1
- File name: ABC12.QPG

■Data communication in ASCII code

(Request data)

| Subcommand | File lock mode | Drive name | File No. |
|-----------------------------------|-------------------------------|-------------------------------|---|
| 0 8 0 8 30H, 38H, 30H, 38H | 0 0 0 1 30H, 30H, 30H, 31H | 0 0 0 0 30H, 30H, 30H, 30H | 0 0 0 1 30H, 30H, 30H, 31H 30H, 30H, 30H, 31H |
| File name | Extension | Attribute | |
| A B C 1 2 □ □ □ | Q P G 51H, 50H, 47H | □ 20H | |
| 41H, 42H, 43H, 31H, 32H, 20H, 20H | | | |

■Data communication in binary code

(Request data)

| File lock mode | File No. | | | |
|----------------------|----------------------|---|----------------------|-----------|
| Subcommand | Drive name | File name | Extension | Attribute |
| 08H, 08H 01H, 00H | 00H, 00H 01H, 00H | 01H, 00H 41H, 42H, 43H, 31H, 32H, 20H, 20H | 51H, 50H, 47H 20H | |

Create new file (Register file name) (command: 1202)

The following examples show the control procedure for registering a new file and reserving a file area on the specified disk. Write data to the file created using this function with the following function.

(☞ Page 319 Write to file (command: 1203))

The content of file in which data is not written cannot be read.

Point

- A new file can be created on the specified drive memory within the size of consecutive unused clusters.
Creating a new file considering the rest of the capacity for future use is recommended.
- The QnACPU management time is registered as the last edit date and time to a new file created using this function.

Message format

■ Request data

ASCII

| | | | | | | | | | | |
|-----|-----|-----|-----|------------|---------|--------------|------------|-------------|-------------|-----|
| 1 | 2 | 0 | 2 | Subcommand | Keyword | Setting flag | Drive name | Fixed value | Fixed value | --- |
| 31H | 32H | 30H | 32H | | | | | | | |

| | | | | |
|-----------|-----------|-----------|------|-------------|
| File name | Extension | Attribute | Size | Fixed value |
|-----------|-----------|-----------|------|-------------|

15

Binary

| | | | | | | | | |
|-----|-----|------------|--------------|---------|------------|-------------|-------------|-----|
| 02 | 12 | Subcommand | Setting flag | Keyword | Drive name | Fixed value | Fixed value | --- |
| 02H | 12H | | | | | | | |

| | | | | |
|-----------|-----------|-----------|------|-------------|
| File name | Extension | Attribute | Size | Fixed value |
|-----------|-----------|-----------|------|-------------|

■ Response data

File No. is stored. (☞ Page 292 File No.)

Data specified by request data

■ Command

| ASCII code | Binary code |
|----------------------------|-------------|
| 1 2 0 2 31H 32H 30H 32H | 02H 12H |

■ Subcommand

| ASCII code | Binary code |
|----------------------------|-------------|
| 0 0 0 0 30H 30H 30H 30H | 00H 00H |

■ Keyword

Allow/prohibit the access to the file of the target memory. (☞ Page 292 Keyword)

■ Setting flag

Data that indicates whether the keyword is registered in the specified drive. (☞ Page 292 Setting flag)

■Drive name

A data to specify the drive of CPU module which controls files. (☞ Page 292 Drive name, Drive No.)

■Fixed value

'0000' (at the time of an ASCII code) or 0000H (at the time of a binary code) is transmitted. (☞ Page 297 Fixed value)

■File name

Character string data of a file name. (☞ Page 293 File name, extension, and attribute)

■Extension

A character string data of extension of a file. (☞ Page 293 File name, extension, and attribute)

■Attribute

A data to specify the file attribute (applicability of data read/write). (☞ Page 293 File name, extension, and attribute)

Specify '20H' (readable/writable disk file) for the attribute of a new file.

■Capacity

A data to reserve the file area on the specified disk when registering a new file. It is specified with the number of bytes. (☞ Page 296 Capacity)

Communication example

Read the data written in the specified file under the following condition.

- Drive name: 1 (RAM area of memory card A)
- File name: ABC12.QPG
- Capacity: 172 bytes

■Data communication in ASCII code

(Request data)

| Subcommand | Keyword | Setting flag | Drive name | Fixed value |
|-------------------------------|-------------------------------|---|------------------------------------|---|
| 1 2 0 2 31H, 32H, 30H, 32H | 0 0 0 0 30H, 30H, 30H, 30H | 0 1 2 3 0 0 30H, 31H, 32H, 33H, 30H, 30H | 0 0 0 1 30H, 31H, 30H, 30H, 31H | 0 0 0 0 0 0 30H, 30H, 30H, 30H, 30H, 30H |

| Fixed value | File name | Extension | Attribute | Size | Fixed value |
|-------------------------------|--|-----------|---|-----------------|---|
| 0 0 0 0 30H, 30H, 30H, 30H | A B C 1 2 □ □ □ Q P G 41H, 42H, 43H, 31H, 32H, 20H, 20H, 20H, 51H, 50H, 47H | □ 20H | 0 0 0 0 0 0 30H, 30H, 30H, 30H, 30H, 30H | A C 41H, 43H | 0 0 0 0 0 0 30H, 30H, 30H, 30H, 30H, 30H |

(Response data)

| |
|-------------------------------|
| 0 0 0 A 30H, 30H, 30H, 41H |
|-------------------------------|

■Data communication in binary code

(Request data)

| Command | Subcommand | Setting flag | Keyword | Drive name | Fixed value | Fixed value | File name | Extension |
|----------------------|----------------------|---------------------------|----------------------|---|----------------------|----------------------|-----------|-----------|
| 02H, 12H 00H, 00H | 00H, 00H 01H, 00H | 01H, 23H, 01H 01H, 00H | 00H, 00H 00H, 00H | 41H, 42H, 43H, 31H, 32H, 20H, 20H, 20H 51H, 50H, 47H | 00H, 00H 00H, 00H | 00H, 00H 00H, 00H | | |

| Attribute | Size | Fixed value |
|-----------------|---------------------------|----------------------|
| 20H A0H, 00H | ACH, 00H, 00H 00H, 00H | 00H, 00H 00H, 00H |

(Response data)

| |
|----------|
| A0H, 00H |
|----------|

Write to file (command: 1203)

The following example shows the control procedure for writing the data of the file that has been read from the QnACPU and stored in an external device to the specified file ('batch write'), and the control procedure for writing n-bytes of arbitrary one-word data to the specified file ('identical data write (FILL)').

Point

The maximum number of data bytes to be read per one time is static.

- For batch writing, all the data read from QnACPU and stored to an external device must be written to the specified file with adjusting the offset address and number of bytes written and divide them for several times.
- For writing identical data, adjust the offset address and number of bytes written, and divide them and write arbitrary 1 word data within the range of specified file for several times. Note that, if the data size to be written remains 1 byte, the numerical value of upper bytes of 1 word data (bit 8 to 15) is not written since the data write is performed in byte units.

Message format

■ Request data

ASCII

| | | | | | | | | | | | | |
|----------------|----------|----------|-------------------------|------------|---------|--------------|------------|----------|-----------|-----------|-----------|-----|
| 1 31H | 2 32H | 0 30H | 3 33H | Subcommand | Keyword | Setting flag | Drive name | File No. | File name | Extension | Attribute | - - |
| ----- | | | | | | | | | | | | |
| Offset address | | | Number of bytes written | | | Write data | | | ----- | | | |

15

Binary

| | | | | | | | | | | |
|----------------|-----|------------|-------------------------|---------|------------|------------|-----------|-----------|-----------|-----|
| 03H | 12H | Subcommand | Setting flag | Keyword | Drive name | File No. | File name | Extension | Attribute | - - |
| ----- | | | | | | | | | | |
| Offset address | | | Number of bytes written | | | Write data | | | ----- | |

■ Response data

There is no response data for this command.

Point

An error occurs If one of the following files is specified while the QnACPU is in RUN, and error end code will be returned.

- Parameter file
- Currently executing file in the built-in RAM (drive name: 00H)

Data specified by request data

■Command

| ASCII code | Binary code | | | | | | | | | | |
|--|-------------|-----|-----|---|-----|-----|-----|-----|---|-----|-----|
| <table border="1"><tr><td>1</td><td>2</td><td>0</td><td>3</td></tr><tr><td>31H</td><td>32H</td><td>30H</td><td>33H</td></tr></table> | 1 | 2 | 0 | 3 | 31H | 32H | 30H | 33H | <table border="1"><tr><td>03H</td><td>12H</td></tr></table> | 03H | 12H |
| 1 | 2 | 0 | 3 | | | | | | | | |
| 31H | 32H | 30H | 33H | | | | | | | | |
| 03H | 12H | | | | | | | | | | |

■Subcommand

| Type | ASCII code | Binary code | | | | | | | | | | |
|-----------------------------------|--|-------------|-----|---|---|-----|-----|-----|-----|---|-----|-----|
| Batch write | <table border="1"><tr><td>0</td><td>0</td><td>0</td><td>0</td></tr><tr><td>30H</td><td>30H</td><td>30H</td><td>30H</td></tr></table> | 0 | 0 | 0 | 0 | 30H | 30H | 30H | 30H | <table border="1"><tr><td>00H</td><td>00H</td></tr></table> | 00H | 00H |
| 0 | 0 | 0 | 0 | | | | | | | | | |
| 30H | 30H | 30H | 30H | | | | | | | | | |
| 00H | 00H | | | | | | | | | | | |
| Writing the identical data (FILL) | <table border="1"><tr><td>0</td><td>0</td><td>0</td><td>1</td></tr><tr><td>30H</td><td>30H</td><td>30H</td><td>31H</td></tr></table> | 0 | 0 | 0 | 1 | 30H | 30H | 30H | 31H | <table border="1"><tr><td>01H</td><td>00H</td></tr></table> | 01H | 00H |
| 0 | 0 | 0 | 1 | | | | | | | | | |
| 30H | 30H | 30H | 31H | | | | | | | | | |
| 01H | 00H | | | | | | | | | | | |

■Keyword

Allow/prohibit the access to the file of the target memory. (☞ Page 292 Keyword)

■Setting flag

Data that indicates whether the keyword is registered in the specified drive. (☞ Page 292 Setting flag)

■Drive name

A data to specify the drive of CPU module which controls files. (☞ Page 292 Drive name, Drive No.)

■File No.

A data to specify the registration number of files. (☞ Page 292 File No.)

■File name

Character string data of a file name. (☞ Page 293 File name, extension, and attribute)

■Extension

A character string data of extension of a file. (☞ Page 293 File name, extension, and attribute)

■Attribute

A data to specify the file attribute (applicability of data read/write). (☞ Page 293 File name, extension, and attribute)

The attribute assigned by a new file creation, etc. is valid as the attribute of the file to which data is written. When writing data, handle the attribute as dummy data.

■Offset address

A data which specifies the start address of the range in which data is written to a file. (☞ Page 295 Offset address)

Specify within the following range.

$0 \leq \text{Address} \leq (\text{File size} - 1)$

- Writing data to file of which drive name is '00H' (built-in RAM): Specify with multiples of 4 (0, 4, 8, ... in decimal notation).
- Writing to the file of which drive name is other than '00H': Specify with even number (for decimal data, 0, 2, 4, 6, 8, ...).

■Number of bytes written

Data to specify the number of bytes of data to be written. (☞ Page 296 Number of bytes read, number of bytes written)

Specify within the following range.

$0 \leq \text{Number of bytes} \leq 960$

■Write data

Data to be written to a file. (☞ Page 296 Read data, write data (for batch read and batch write functions))

Communication example (Batch write)

Batch write is executed under the following condition.

- Drive name: 1 (RAM area of memory card A)
- File No.: 1
- File name: ABC12.QPG
- Offset address: 7CH
- Number of bytes written: 416 bytes

■Data communication in ASCII code

(Request data)

| Subcommand | Keyword | Setting flag | Drive name | File No. |
|----------------------------------|---------------------------------------|--|---|----------------------------|
| 1 2 0 3 31H 32H 30H 33H | 0 0 0 0 30H 30H 30H 30H | 0 1 2 3 0 0 30H 31H 32H 33H 30H 30H | 0 1 30H 31H | 0 0 0 1 30H 30H 30H 31H |
| A B C 1 2 41H 42H 43H 31H 32H | Q P G 51H 50H 47H | □ 20H | 0 0 0 0 0 0 7 30H 30H 30H 30H 30H 37H 43H | C 30H 31H 41H 30H |
| Address 7CH 1 9 31H 39H | Address 7DH to 21AH 9 9 39H 39H | Address 21BH 1 2 31H 32H | Number of bytes written ... 0 2 5 A D C 30H 32H 35H 41H 44H 43H | |
| ← Write data (For 416 bytes) → | | | | |

15

■Data communication in binary code

(Request data)

| Setting flag | | | |
|--------------------------------|----------------------|----------------------|--|
| Subcommand | Keyword | Drive name | File name |
| 03H, 12H 00H, 00H | 01H 00H, 23H, 01H | 01H, 00H 01H, 00H | 01H, 00H 41H, 42H, 43H, 31H, 32H, 20H, 20H, 20H |
| 51H, 50H, 47H 20H | 7CH, 00H, 00H, 00H | A0H, 01H 19H | 99H, 12H, ... 02H, 5AH, DC |
| ← Write data (For 416 bytes) → | | | |

Communication example (Write identical data (FILL))

The identical data are written (FILL) by FFFFH under the following condition.

- Drive name: 1 (RAM area of memory card A)
- File No.: 1
- File name: ABC12.QPG
- Offset address: 7CH
- Number of bytes written: 416 bytes

■Data communication in ASCII code

(Request data)

| Subcommand | Keyword | Setting flag | Drive name | File No. | |
|----------------------------------|----------------------------|--|----------------|--|--|
| 1 2 0 3 31H 32H 30H 33H | 0 0 0 1 30H 30H 30H 31H | 0 1 2 3 0 0 30H 31H 32H 33H 30H 30H | 0 1 30H 31H | 0 0 0 1 30H 30H 30H 31H | 0 0 0 1 30H 30H 30H 31H |
| File name | Extension | Attribute | Offset address | Number of bytes written | Write data |
| A B C 1 2 41H 42H 43H 31H 32H | □ □ □ 20H 20H 20H | Q P G 51H 50H 47H | □ 20H | 0 0 0 0 0 0 7 C 30H 30H 30H 30H 30H 37H 43H | 0 1 A 0 30H 31H 41H 30H 46H 46H 46H |

■Data communication in binary code

(Request data)

| Setting flag | File No. | | |
|-------------------------|-----------|-----------------|---|
| | | | |
| Subcommand | Keyword | Drive name | File name |
| 03H 12H | 01H 00H | 01H 00H | 23H 01H 01H 00H 41H 42H 43H 31H 32H 20H 20H 20H |
| Number of bytes written | | | |
| Extension | Attribute | Offset address | Write data |
| 51H 50H 47H | 20H | 7CH 00H 00H 00H | A0H 01H FFH FFH |

Modify file information (command: 1204)

The following examples show the procedure to modify the file information of the specified file.

Message format

■Request data

- Modify file creation date and time (subcommand: 0000)

ASCII

| | | | | | | | | | | | | |
|----------------------|-----|-----|-----|----------------|----------------|--------------|------------|----------|-----------|-----------|-----------|-------|
| 1 | 2 | 0 | 4 | Subcommand | Keyword | Setting flag | Drive name | File No. | File name | Extension | Attribute | - - - |
| 31H | 32H | 30H | 34H | | | | | | | | | |
| Modification pattern | | | | Last edit time | Last edit date | | | | | | | |

Binary

| | | | | | | | | | | |
|----------------------|-----|------------|--------------|----------------|----------------|----------|-----------|-----------|-----------|-------|
| 04H | 12H | Subcommand | Setting flag | Keyword | Drive name | File No. | File name | Extension | Attribute | - - - |
| | | | | | | | | | | |
| Modification pattern | | | | Last edit time | Last edit date | | | | | |

- Modify file name, attribute, file size (subcommand: 0001)

ASCII

| | | | | | | | | | | | |
|------------------------------------|-----|----------------------|-----|-----------------------------------|---------|-----------------------------------|------------|-----------------------------------|------------------------------------|------------------------------------|-------------|
| 1 | 2 | 0 | 4 | Subcommand | Keyword | Setting flag | Drive name | File No. | File name (Before modification) | Extension (Before modification) | - - - |
| 31H | 32H | 30H | 34H | | | | | | | | |
| Attribute (Before modification) | | Modification pattern | | File name (After modification) | | Extension (After modification) | | Attribute (After modification) | | File size | Fixed value |
| | | | | | | | | | | | |

Binary

| | | | | | | | | | | | |
|------------------------------------|-----|----------------------|--------------|-----------------------------------|------------|-----------------------------------|------------------------------------|------------------------------------|-------|-----------|-------------|
| 04H | 12H | Subcommand | Setting flag | Keyword | Drive name | File No. | File name (Before modification) | Extension (Before modification) | - - - | | |
| | | | | | | | | | | | |
| Attribute (Before modification) | | Modification pattern | | File name (After modification) | | Extension (After modification) | | Attribute (After modification) | | File size | Fixed value |



The size can be modified only while the QnACPU is in STOP. A consecutive free area of the specified size is necessary on the specified drive. The free area can be checked by reading memory usage status shown in the following section.

☞ Page 287 Read drive memory usage status (command: 0205)

- Batch modify file information (subcommand: 0002)

ASCII

| | | | | | | | | | | |
|------------------------------------|------------|----------------------|--------------|-----------------------------------|----------|------------------------------------|-----------------------------------|----------------|--|--|
| 1 2 0 4 31H 32H 30H 34H | Subcommand | Keyword | Setting flag | Drive name | File No. | File name (Before modification) | Extension (After modification) | - - - | | |
| <hr/> | | | | | | | | | | |
| Attribute (Before modification) | | Modification pattern | | File name (After modification) | | Extension (After modification) | Attribute (After modification) | Last edit time | | |
| <hr/> | | | | | | | | | | |
| File size | | Fixed value | | | | | | | | |

Binary

| | | | | | | | | | | |
|------------------------------------|------------|----------------------|---------|-----------------------------------|----------|------------------------------------|------------------------------------|----------------|--|--|
| 04H 12H | Subcommand | Setting flag | Keyword | Drive name | File No. | File name (Before modification) | Extension (Before modification) | - - - | | |
| <hr/> | | | | | | | | | | |
| Attribute (Before modification) | | Modification pattern | | File name (After modification) | | Extension (After modification) | Attribute (After modification) | Last edit time | | |
| <hr/> | | | | | | | | | | |
| File size | | Fixed value | | | | | | | | |

■Response data

There is no response data for this command.



An error occurs If one of the following files is specified while the QnACPU is in RUN, and error end code will be returned.

- Parameter file
- Currently executing file in the built-in RAM (drive name: 00H)

Data specified by request data

■Command

| ASCII code | Binary code | | | | | | | | | | |
|---|-------------|-----|-----|---|-----|-----|-----|-----|---|-----|-----|
| <table border="1"> <tr><td>1</td><td>2</td><td>0</td><td>4</td></tr> <tr><td>31H</td><td>32H</td><td>30H</td><td>34H</td></tr> </table> | 1 | 2 | 0 | 4 | 31H | 32H | 30H | 34H | <table border="1"> <tr><td>04H</td><td>12H</td></tr> </table> | 04H | 12H |
| 1 | 2 | 0 | 4 | | | | | | | | |
| 31H | 32H | 30H | 34H | | | | | | | | |
| 04H | 12H | | | | | | | | | | |

■Subcommand

| Type | ASCII code | Binary code | | | | | | | | | | |
|--|---|-------------|-----|---|---|-----|-----|-----|-----|---|-----|-----|
| Modify file creation date and time | <table border="1"> <tr><td>0</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>30H</td><td>30H</td><td>30H</td><td>30H</td></tr> </table> | 0 | 0 | 0 | 0 | 30H | 30H | 30H | 30H | <table border="1"> <tr><td>00H</td><td>00H</td></tr> </table> | 00H | 00H |
| 0 | 0 | 0 | 0 | | | | | | | | | |
| 30H | 30H | 30H | 30H | | | | | | | | | |
| 00H | 00H | | | | | | | | | | | |
| Modify file name, attribute, file size | <table border="1"> <tr><td>0</td><td>0</td><td>0</td><td>1</td></tr> <tr><td>30H</td><td>30H</td><td>30H</td><td>31H</td></tr> </table> | 0 | 0 | 0 | 1 | 30H | 30H | 30H | 31H | <table border="1"> <tr><td>01H</td><td>00H</td></tr> </table> | 01H | 00H |
| 0 | 0 | 0 | 1 | | | | | | | | | |
| 30H | 30H | 30H | 31H | | | | | | | | | |
| 01H | 00H | | | | | | | | | | | |
| Batch modify file information | <table border="1"> <tr><td>0</td><td>0</td><td>0</td><td>2</td></tr> <tr><td>30H</td><td>30H</td><td>30H</td><td>32H</td></tr> </table> | 0 | 0 | 0 | 2 | 30H | 30H | 30H | 32H | <table border="1"> <tr><td>02H</td><td>00H</td></tr> </table> | 02H | 00H |
| 0 | 0 | 0 | 2 | | | | | | | | | |
| 30H | 30H | 30H | 32H | | | | | | | | | |
| 02H | 00H | | | | | | | | | | | |

■Keyword

Allow/prohibit the access to the file of the target memory. (☞ Page 292 Keyword)

15

■Setting flag

Data that indicates whether the keyword is registered in the specified drive. (☞ Page 292 Setting flag)

■Drive name

A data to specify the drive of CPU module which controls files. (☞ Page 292 Drive name, Drive No.)

■File No.

A data to specify the registration number of files. (☞ Page 292 File No.)

Specify within the following range.

1 ≤ File No. ≤ 256

■File name

Character string data of a file name. (☞ Page 293 File name, extension, and attribute)

■Extension

A character string data of extension of a file. (☞ Page 293 File name, extension, and attribute)

■Attribute

A data to specify the file attribute (applicability of data read/write). (☞ Page 293 File name, extension, and attribute)

The attribute can only be modified between 01H (read only file) ⇔ 20H (read/write enabled file)

■Modification pattern

A data to specify which information is to modify when modifying information (file name, size, created date and time) of the existing file. (☞ Page 297 Modification pattern (for changing file name and file size))

■Last edit time and last edit date

A data that indicates the date and time when the current data was registered. (☞ Page 294 Last edit time and last edit date)

■File size

A data that indicates the capacity of the current file in number of bytes. (☞ Page 295 File size)

■Fixed value

Send '0000' (at the time of ASCII code) or 0000H (at the time of binary code). (☞ Page 297 Fixed value)

Communication example (Modify file creation time) (subcommand: 0000)

Modify the creation time of the following file.

- Drive name: 1 (RAM area of memory card A)
- File name: ABC12.QPG
- Modification pattern: 14H (specify the time of QnACPU for the last edit time.)

■Data communication in ASCII code

(Request data)

| Subcommand | Keyword | Setting flag | Drive name | File No. |
|-------------------------------|-------------------------------|---|---|-------------------------------|
| 1 2 0 4 31H, 32H, 30H, 34H | 0 0 0 0 30H, 30H, 30H, 30H | 0 1 2 0 0 0 30H, 31H, 32H, 30H, 30H, 30H | 0 1 0 0 0 1 30H, 31H, 30H, 30H, 30H, 31H | F F F F 46H, 46H, 46H, 46H |

| File name | Extension | Attribute | Modification pattern | Last edit time | Last edit date |
|-----------------|-----------|-----------|----------------------|----------------|----------------|
| A B C 1 2 □□□ | Q P G □ | 0 0 1 4 | 0 0 0 0 | 0 0 0 0 | 0 0 0 0 |

■Data communication in binary code

(Request data)

| Subcommand | Setting flag | Keyword | Drive name | File No. |
|----------------------|----------------------|---------------------------|----------------------|----------|
| 04H, 12H 00H, 00H | 01H 00H, 20H, 01H | 00H, 20H, 01H 01H, 00H | FFH, FFH FFH, FFH | - - - |

| File name | Extension | Attribute | Modification pattern | Last edit time | Last edit date |
|--|--------------------------------|------------|----------------------|----------------------|----------------------|
| 41H, 42H, 43H, 31H, 32H, 20H, 20H, 20H 41H, 42H, 43H, 31H, 32H, 20H, 20H, 20H | 51H, 50H, 47H 51H, 50H, 47H | 20H 20H | 14H, 00H 14H, 00H | 00H, 00H 00H, 00H | 00H, 00H 00H, 00H |

Communication example (Modify file name, attribute) (subcommand: 0001)

Modify the file name and attribute of the following file.

- Drive name: 1 (RAM area of memory card A)
- File No.: 1
- File name (before modification): ABC12.QPG
- File name (after modification): ABC1234.QPG
- Attribute (before modification): 20H (disk files that can be read/written)
- Attribute (after modification): 01H (read only file)

■Data communication in ASCII code

(Request data)

| Subcommand | Keyword | Setting flag | Drive name | File No. |
|---|---|---|---|---|
| 1 2 0 4 31H, 32H, 30H, 34H | 0 0 0 1 30H, 30H, 30H, 30H | 0 1 2 3 0 0 30H, 31H, 32H, 33H, 30H, 30H | 0 1 0 0 0 1 30H, 30H, 30H, 31H | 0 0 0 1 30H, 30H, 30H, 31H |
| <hr/> | | | | |
| File name (Before modification) A B C 1 2 41H, 42H, 43H, 31H, 32H, 20H, 20H | Extension (Before modification) Q P G 51H, 50H, 47H | Attribute (Before modification) 20H | Modification pattern 0 0 0 3 30H, 30H, 30H, 33H | File name (After modification) A B C 1 2 3 4 41H, 42H, 43H, 31H, 32H, 33H, 34H, 20H |
| <hr/> | | | | |
| Extension (After modification) Q P G 51H, 50H, 47H | Attribute (After modification) 01H | File size 30H, 30H, 30H, 30H, 30H, 30H | Fixed value 0 0 0 0 30H, 30H, 30H, 30H | |

■Data communication in binary code

(Request data)

| Subcommand | Keyword | Drive name | Setting flag | File No. |
|---|---|---|--|--------------------------------|
| 04H, 12H 01H, 00H | 01H 00H, 23H | 01H 00H | 01H, 00H 01H, 00H | 01H, 00H |
| <hr/> | | | | |
| File name (Before modification) A B C 1 2 41H, 42H, 43H, 31H, 32H, 20H, 20H | Extension (Before modification) Q P G 51H, 50H, 47H | Attribute (Before modification) 20H | Modification pattern 03H, 00H 41H, 42H, 43H, 31H, 32H, 33H, 34H, 20H | File name (After modification) |
| <hr/> | | | | |
| Extension (After modification) Q P G 51H, 50H, 47H | Attribute (After modification) 01H | File size 00H, 00H, 00H, 00H, 00H, 00H | Fixed value 00H, 00H, 00H, 00H | |

Communication example (Batch modify file information) (subcommand: 0002)

Modify the file name of the following files, attribute, and last edit date and time.

- Drive name: 1 (RAM area of memory card A)
- File No.: 1
- File name (before modification): ABC12.QPG
- File name (after modification): ABC1234.QPG
- Attribute (before modification): 20H (disk files that can be read/written)
- Attribute (after modification): 01H (read only file)
- Modification pattern: 1FH (Specify the date and time of QnACPU to the last edit day and last edit time)

■Data communication in ASCII code

(Request data)

| Subcommand | Keyword | Setting flag | Drive name | File No. |
|------------------------------------|-------------------------|---|-------------------------|--------------------------------|
| 1 2 0 4 | 0 0 0 2 | 0 1 2 3 0 0 | 0 1 | 0 0 0 1 0 0 0 1 |
| ----- | | | | |
| Attribute (Before modification) | | | | |
| File name (Before modification) | | Extension (Before modification) | Modification pattern | File name (After modification) |
| A B C 1 2 | Q P G | 0 0 1 F | A B C 1 2 3 4 | ----- |
| 41H 42H 43H 31H 32H 20H 20H | 51H 50H 47H 20H | 30H 31H 46H 41H 42H 43H 31H 32H 33H 34H 20H | ----- | ----- |
| ----- | | | | |
| Attribute (After modification) | | | | |
| Extension (After modification) | Last edit time | Last edit date | File size | Fixed value |
| Q P G | 0 0 0 0 | 0 0 0 0 | 0 0 0 0 0 0 0 0 | 0 0 0 0 |
| 51H 50H 47H | 01H 30H 30H 30H 30H 30H | 30H 30H 30H 30H 30H 30H | 30H 30H 30H 30H 30H 30H | 30H 30H 30H 30H |

■Data communication in binary code

(Request data)

| Setting flag | File No. |
|---|---|
| Subcommand | Keyword Drive name |
| ----- | |
| 04H 12H | 02H 00H 01H 00H 23H 01H 01H 00H 01H 00H |
| ----- | |
| Extension (Before modification) Modification pattern | |
| File name (Before modification) Attribute (Before modification) | |
| 41H 42H 43H 31H 32H 20H 20H 51H 50H 47H 20H 1FH 00H 41H 42H 43H 31H 32H 33H 34H 20H | |
| ----- | |
| Extension (After modification) Last edit time File size | |
| Attribute (After modification) Last edit date Fixed value | |
| 51H 50H 47H 01H 00H | |

Delete file (command: 1205)

The following examples show the procedure to delete the existing files.



Determine the file deletion timing for the entire system, including the QnACPU and related devices.

The file in which the file lock has been registered cannot be deleted.

When QnACPU is in RUN, the following files cannot be deleted.

- Program file (□.QPG)
- Parameter file (□.QPA)
- Boot setting file (□.QBT)

Message format

■Request data

ASCII

| | | | | | | | | |
|----------------------------|------------|---------|--------------|------------|----------|-----------|-----------|-----------|
| 1 2 0 5 31H 32H 30H 35H | Subcommand | Keyword | Setting flag | Drive name | File No. | File name | Extension | Attribute |
|----------------------------|------------|---------|--------------|------------|----------|-----------|-----------|-----------|

Binary

| | | | | | | | | | |
|-----|-----|------------|--------------|---------|------------|----------|-----------|-----------|-----------|
| 05H | 12H | Subcommand | Setting flag | Keyword | Drive name | File No. | File name | Extension | Attribute |
|-----|-----|------------|--------------|---------|------------|----------|-----------|-----------|-----------|

15

■Response data

There is no response data for this command.

Data specified by request data

■Command

| ASCII code | Binary code |
|----------------------------|-------------|
| 1 2 0 5 31H 32H 30H 35H | 05H 12H |

■Subcommand

| ASCII code | Binary code |
|----------------------------|-------------|
| 0 0 0 0 30H 30H 30H 30H | 00H 00H |

■Keyword

Allow/prohibit the access to the file of the target memory. (☞ Page 292 Keyword)

■Setting flag

Data that indicates whether the keyword is registered in the specified drive. (☞ Page 292 Setting flag)

■Drive name

A data to specify the drive of CPU module which controls files. (☞ Page 292 Drive name, Drive No.)

■File No.

A data to specify the registration number of files. (☞ Page 292 File No.)

■File name

Character string data of a file name. (☞ Page 293 File name, extension, and attribute)

■Extension

A character string data of extension of a file. ([Page 293 File name, extension, and attribute](#))

■Attribute

A data to specify the file attribute (applicability of data read/write). ([Page 293 File name, extension, and attribute](#))

When deleting a file, handle the attribute as dummy data.

Communication example

Delete files under the following conditions.

- Drive name: 1 (RAM area of memory card A)
- File No.: 1
- File name: ABC12.QPG

■Data communication in ASCII code

(Request data)

| Subcommand | Keyword | Setting flag | Drive name | File No. |
|-------------------------------|-------------------------------|---|-----------------|-------------------------------|
| 1 2 0 5 31H, 32H, 30H, 35H | 0 0 0 0 30H, 30H, 30H, 30H | 0 1 2 3 0 0 30H, 31H, 32H, 33H, 30H, 30H | 0 1 30H, 31H | 0 0 0 1 30H, 30H, 30H, 31H |

| File name | | | Extension | Attribute |
|-----------|----------|----------|---------------------------|---|
| A 41H | B 42H | C 43H | 1 2 31H, 32H, 20H, 20H | <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> Q P G 51H, 50H, 47H, 20H |

■Data communication in binary code

(Request data)

| Subcommand | Setting flag | Keyword | Drive name | File No. |
|------------|--------------|---------|---------------|--------------------|
| 05H, 12H | 00H, 00H | 01H | 00H, 23H, 01H | 01H, 00H, 01H, 00H |

| File name | | | Extension | Attribute |
|-----------------------------------|---------------|-----|-----------|-----------|
| 41H, 42H, 43H, 31H, 32H, 20H, 20H | 51H, 50H, 47H | 20H | | |

Copy file (command: 1206)

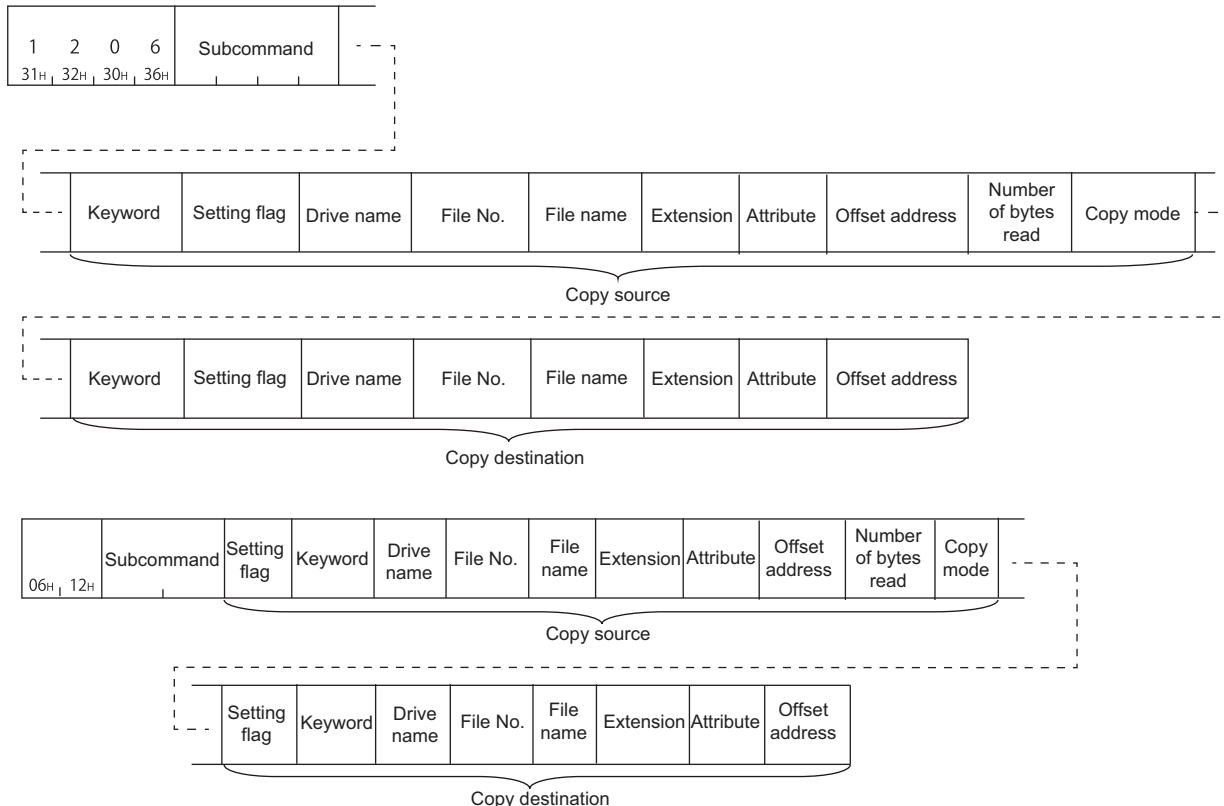
The following examples show the procedure to copy a file. (For QnACPU)

The maximum number of data bytes to be copies per one time is static. For the data which are already written, adjust the offset address and number of copy bytes, and divide them and write to the new registration file for several times.

Message format

■Request data

ASCII



15

■Response data

There is no response data for this command.



An error occurs if one of the following files is specified while the QnACPU is in RUN, and error end code will be returned.

- Parameter file
- Currently executing file in the built-in RAM (drive name: 00H)

Data specified by request data

■Command

| ASCII code | Binary code |
|----------------------------------|-------------|
| 1 2 0 6 31H , 32H , 30H , 36H | 06H , 12H |

■Subcommand

| ASCII code | Binary code |
|----------------------------------|-------------|
| 0 0 0 0 30H , 30H , 30H , 30H | 00H , 00H |

■Keyword

Allow/prohibit the access to the file of the target memory. ([Page 292 Keyword](#))

■Setting flag

This indicates whether the keyword registered in the specified drive matches the keyword of request data. ([Page 284 Setting flag](#))

■Drive name

A data to specify the drive of CPU module which controls files. ([Page 292 Drive name, Drive No.](#))

■File No.

A data to specify the registration number of files. ([Page 292 File No.](#))

■File name

Character string data of a file name. ([Page 293 File name, extension, and attribute](#))

■Extension

A character string data of extension of a file. ([Page 293 File name, extension, and attribute](#))

■Attribute

A data to specify the file attribute (applicability of data read/write). ([Page 293 File name, extension, and attribute](#))

The attribute assigned by a new file creation, etc. is valid as the attribute of the source and destination files. When copying a file, handle the attribute as dummy data.

■Offset address

A data to specify the start address of the range from which data is read to a file. ([Page 295 Offset address](#))

Specify within the following range.

Offset address: $0 \leq \text{Address} \leq (\text{File size} - 1)$

- Copy to the file whose drive name is '00H' (Built-in RAM): Specify with multiples of 4 (0, 4, 8, ... in decimal notation).
- Copy to the file whose drive name is other than '00H': Specify with even number (for decimal data, 0, 2, 4, 6, 8, ...).

■Number of bytes read (number of copy bytes)

A data to specify the range from which data is read in byte unit to file. Specify the data as one address per one byte.

([Page 296 Number of bytes read, number of bytes written](#))

Specify within the following range.

Number of copied bytes: $0 \leq \text{Number of bytes} \leq 480$

■Copy mode

A data to specify whether to copy the last edit time and data of the source file to the target file at the completion of copying.

When the time and date are not copied, the QnACPU management time at a new file creation will remain. ([Page 299 Copy mode](#))

Communication example

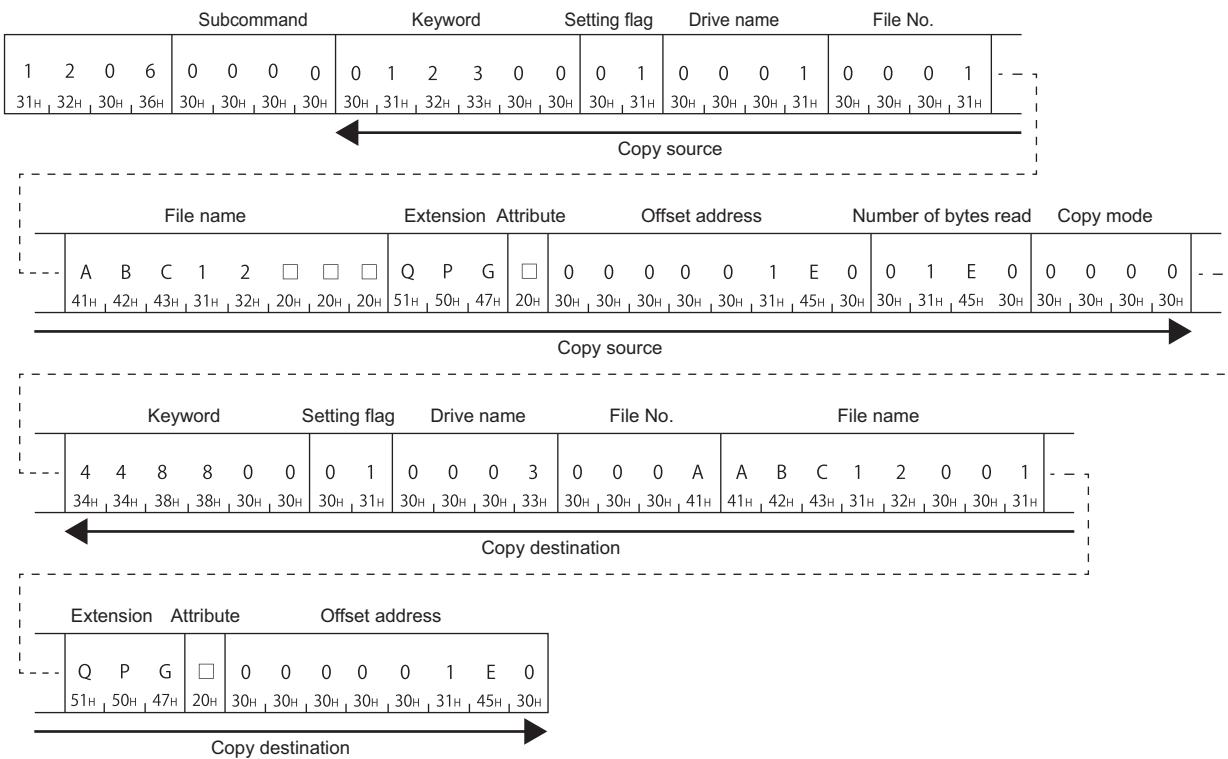
Copy the file with the following condition.

- Drive name (copy source): 1 (RAM area of memory card A)
- Drive name (copy destination): 3 (RAM area of memory card B)
- File No. (copy source): 1H (1)
- File No. (copy destination): AH (10)
- File name (copy source): ABC12.QPG
- File name (copy destination): ABC12001.QPG
- Offset address (copy source): 1E0H
- Number of bytes read (copy source): 480 bytes
- Copy mode (copy source): 0 (Do not copy the last edit date and time of the copy source.)

■Data communication in ASCII code

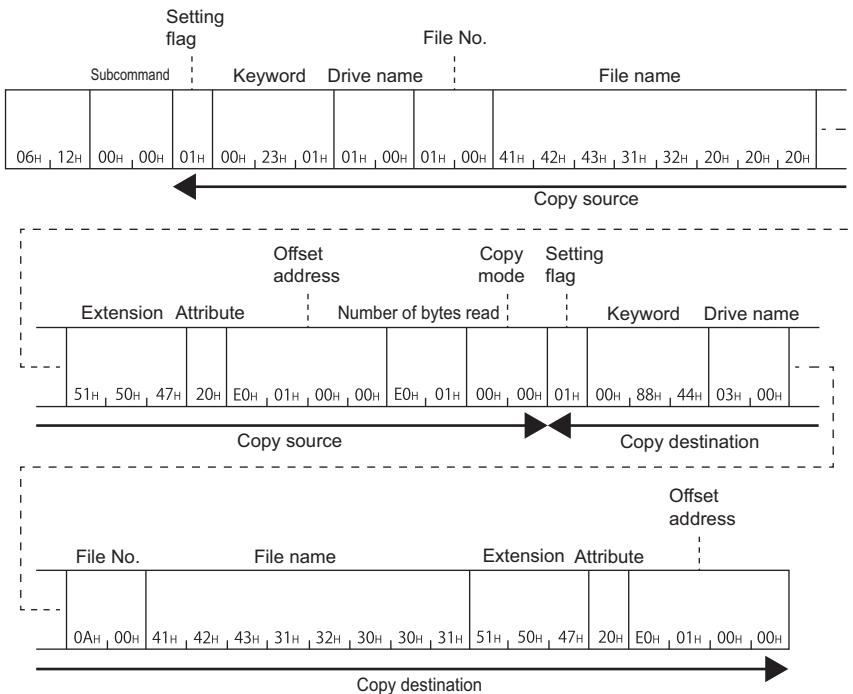
(Request data)

15



■Data communication in binary code

(Request data)



PART 5**COMPATIBILITY WITH A
SERIES**

This part explains the specifications when using MELSEC-A series devices.

[16 MELSEC-A SERIES SUPPORTED SPECIFICATIONS](#)

[17 COMMUNICATING USING 1C FRAMES](#)

[18 COMMUNICATING USING 1E FRAMES](#)

16 MELSEC-A SERIES SUPPORTED SPECIFICATIONS

This chapter explains the specifications of the messages of MC protocol and access ranges when using the MELSEC-A series devices as follows:

- When accessing system including MELSEC-A series modules
- When utilizing the software for data communication created for MELSEC-A series programmable controller.

16.1 Frames and Commands that can be Used

When accessing MELSEC-A series modules, all frames of MC protocol can be used.

However, the commands that can be used have some restrictions. (☞ Page 465 Accessible Modules for Each Command)

A compatible frame

The following frames have compatibility with the message protocol and the message format for MELSEC-A series.

| Frame | Compatible message format | Access range |
|----------|--|--------------------------------------|
| 1C frame | Dedicated protocol for MELSEC-A series computer link modules | Page 45 Accessible range of 1C frame |
| 1E frame | Message formats for MELSEC-A series Ethernet interface modules | Page 47 Accessible range of 1E frame |

16.2 Accessible modules

The following MELSEC-A series modules can be accessed within the access range.

Accessible modules to other stations

The following modules can be accessed.

| Type | Model name |
|-------------------------|---|
| CPU module | A1NCPU, A2NCPU, A2NCPU-S1, A3NCPU, A2ACPU, A2ACPU-S1, A3ACPU, A2UCPU, A2UCPU-S1, A3UCPU, A4UCPU A1SCPU, A1SJCPU(-S3), A1SHCPU, A1SJHCPU, A2SCPU, A2SHCPU, A2USCPU, A2USCPU-S1, A2USHCPU-S1 A0J2HCPU |
| | Q02CPU-A, Q02HCPU-A, Q06HCPU-A |
| | A2CCPUC24 and A2CCPUC24-PRF (when connected to an external device by multidrop connection.) |
| MELSECNET/10 remote I/O | AJ72LP25 (G), AJ72BR15 |
| Special function module | Refer to the following section. ☞ Page 379 Accessible modules |

Modules that can be relayed between networks

For the modules that can be relayed between networks when accessing MELSEC-A series module, refer to the following table.

■MELSEC-A series module

| Network | Model name |
|--------------|--|
| MELSECNET/10 | AJ71LP21 (G), AJ71BR11, A1SJ71LP21, A1SJ71BR11 |

■Module other than MELSEC-A series

☞ Page 280 Modules that can be relayed between networks

16.3 Considerations

The following shows the considerations when using MELSEC-A series devices.

Considerations when connecting C24

■When a computer link module is included in multidrop connection

Access in ASCII code (format 1 to format 4). A binary code (format 5) cannot be used. (Including the access to the connected station)

Considerations when connecting E71

■Setting range of monitoring timer

When accessing ACPU for the first time, the wait time for CPU monitoring timer is required before receiving a response message because QnACPU identifies the CPU type. Be sure to set a value within the setting range shown below.

| Access target | Monitoring timer |
|----------------------------------|----------------------------|
| Connected station (host station) | 1H to 28H (0.25 s to 10 s) |
| Other station | 2H to F0H (0.5 s to 60 s) |

17 COMMUNICATING USING 1C FRAMES

This chapter explains the functions when accessing using 1C frame and their message format.

1C frame is compatible with the communication function of the dedicated protocols supported by A series computer link modules.

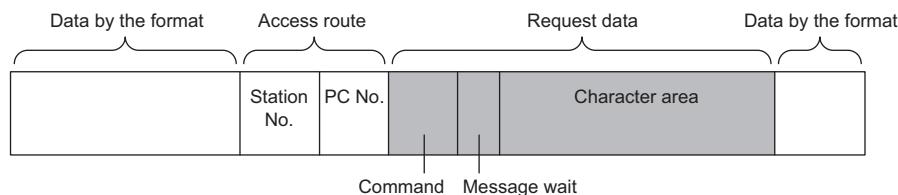
Only the commands for 1C frame explained in this chapter can be used for 1C frame.

17.1 Message Format

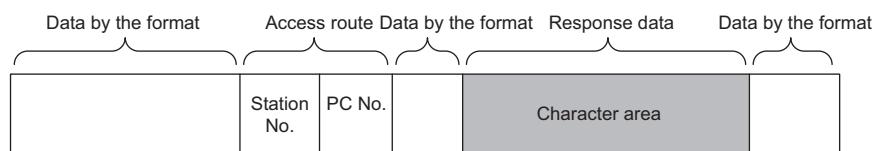
This section explains the message format when communicating data using 1C frame.

Message format

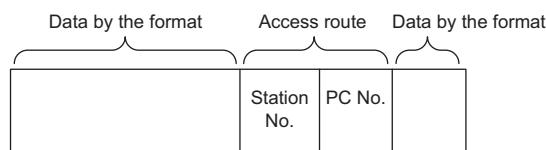
■Request message



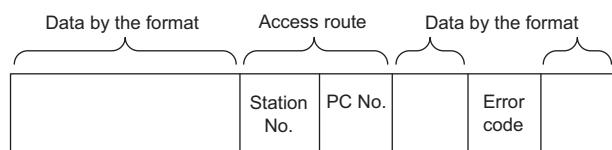
■Response message (Normal completion: Response data)



■Response message (Normal completion: No response data)



■Response message (Abnormal completion)



Setting data

Set the following items.

| Item | Description | | Reference |
|----------------|--|--|--|
| Data by format | The message formats differ depending on the set format (Format 1 to Format 4). | | Page 27 Message Formats of Each Protocol |
| Access route | Station No. | Specify the station to be connected from an external device. | Page 48 Station No. |
| | PC No. | Specify the network module station No. of an access target. | Page 50 Network No., PC No. |
| Request data | Command | Specify the function to request such as read or write. | Page 340 Command |
| | Message wait | A data to generate a delay time for response transmission. Specify the wait time within the range of 0 to 150 ms in 10 ms units. | Page 341 Message wait |
| | Character area | A data that instructs the CPU module to execute a request specified by command. The content of the character areas differs depending on the command. | Page 341 Character area |
| Response data | Character area | A data that C24 returns to a request specified by a command. The content of the character areas differs depending on the command. | |
| Error code | Error code indicates the content of occurred error. | | Page 342 Error code |

17.2 Details of Setting Data

This section explains how to specify the common data items and their content in each message.

Command

Set the command type. (☞ Page 343 Command and Function Lists for 1C Frame)

The setting values for each command are as follows.

| Function | ACPU common command | | AnA/AnUCPU common command | | Reference |
|---|---------------------|------------|---------------------------|------------|---|
| | Sym bol | ASCII code | Sym bol | ASCII code | |
| Device memory read and write | BR | 42H, 52H | JR | 4AH, 52H | Page 348 Batch read (bit units) (command: BR, JR) |
| | WR | 57H, 52H | QR | 51H, 52H | Page 350 Batch read (word units) (command: WR, QR) |
| | BW | 42H, 57H | JW | 4AH, 57H | Page 352 Batch write (bit units) (command: BW, JW) |
| | WW | 57H, 57H | QW | 51H, 57H | Page 354 Batch write (word units) (command: WW, QW) |
| | BT | 42H, 54H | JT | 4AH, 54H | Page 356 Test (random write) (bit units) (command: BT, JT) |
| | WT | 57H, 54H | QT | 51H, 54H | Page 358 Test (random write) (word units) (command: WT, QT) |
| | BM | 42H, 4DH | JM | 4AH, 4DH | Page 361 Register monitor data (bit units) (command: BM, JM) |
| | WM | 57H, 4DH | QM | 51H, 4DH | Page 362 Register monitor data (word units) (command: WM, QM) |
| | MB | 4DH, 42H | MJ | 4DH, 4AH | Page 363 Monitor (bit units) (command: MB, MJ) |
| MN | 4DH, 4EH | MQ | 4DH, 51H | | Page 364 Monitor (word units) (command: MN, MQ) |
| Read and write extended file register | ER | 45H, 52H | — | — | Page 369 Batch read (command: ER) |
| | EW | 45H, 57H | — | — | Page 370 Batch write (command: EW) |
| | — | — | NR | 4EH, 52H | Page 375 Direct read (command: NR) |
| | — | — | NW | 4EH, 57H | Page 376 Direct write (command: NW) |
| | ET | 45H, 54H | — | — | Page 371 Test (random write) (command: ET) |
| | EM | 45H, 4DH | — | — | Page 373 Register monitor data (command: EM) |
| ME | 4DH, 45H | — | — | — | Page 374 Monitor (command: ME) |
| Read/write buffer memory of special function module | TR | 54H, 52H | — | — | Page 380 Batch read (command: TR) |
| | TW | 54H, 57H | — | — | Page 382 Batch write (command: TW) |
| Loopback test | TT | 54H, 54H | — | — | Page 383 Loopback test (Command: TT) |

Setting method

Use the commands by converting to 2-digit (hexadecimal) ASCII codes.

Ex.

Device memory batch read (BR) in bit unit

| | |
|-----|-------|
| B | R |
| 42H | , 52H |

Message wait

Message wait is a data to generate a delay time for response transmission.

Some external devices may take time to become receiving status after sending a command.

Specify the minimum wait time to send the result after C24 is received a command from an external device. Specify the wait time in accordance with the specifications of the external device.

Setting method

Specify the wait time within the range of 0 to 150 ms in 10 ms units.

Convert 0H to FH (0 to 15) to 1-digit (hexadecimal) ASCII codes regarding 10 ms as 1H.

Ex.

When the message wait time is 100 ms

If the following value is set to message wait in request message, after passing 100 ms or more, transmission of a response message will be started.

| |
|-----|
| A |
| 41H |

Character area

The content of the character areas differs depending on the command.

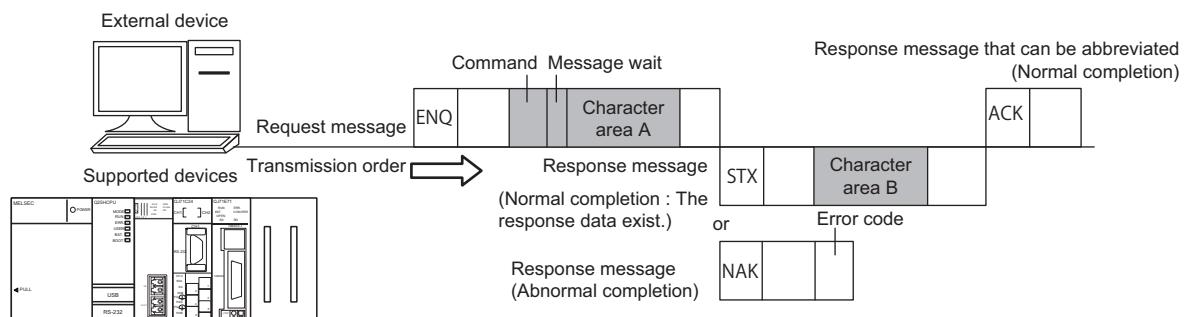
The character area of request data is equivalent to the character A area and the character C area of the dedicated protocols for A series computer link module. The character area of response data is equivalent to the character B area of a dedicated protocols.

- Character area A: A data that C24 instructs the CPU module to perform the read request specified by command.
- Character area B: A data that C24 returns to a request specified by a command.
- Character area C: A data that C24 instructs the CPU module to perform the write request specified by command.

When reading data (Response data)

The following shows the image when the response data (character B area of the dedicated protocol) is included in the response message.

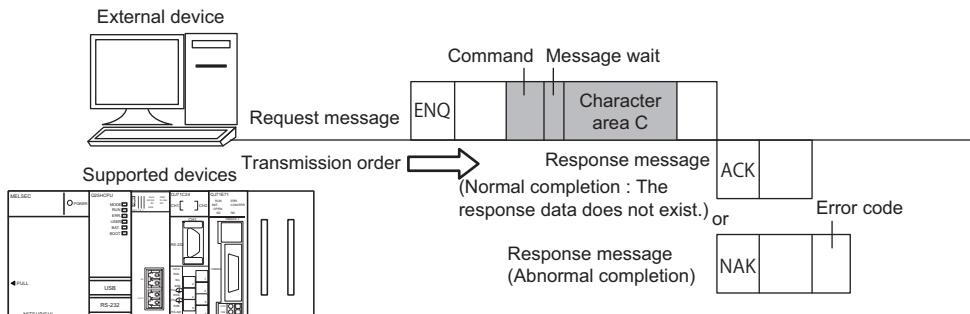
(The head of the message data in the figure is a control code of format 1. [Page 27 Message Formats of Each Protocol](#))



When writing data (No response data)

The following shows the image when the response data is not included in the response message.

(The head of the message data in the figure is a control code of format 1. [Page 27 Message Formats of Each Protocol](#))



Error code

Error code indicates the content of occurred error.

If more than one error occurs at the same time, the error code detected first is returned.

For the content of error code and its corrective action, refer to the user's manual of the module used.

MELSEC iQ-R Serial Communication Module User's Manual (Application)

Q Corresponding Serial Communication Module User's Manual (Basic)

MELSEC-L Serial Communication Module User's Manual (Basic)

Setting method

Convert the numerical value to 2-digit ASCII code (hexadecimal), and send it from the upper digits.

Ex.

For error code 05H

| | |
|-----|-----|
| 0 | 5 |
| 30H | 35H |

17.3 Command and Function Lists for 1C Frame

Use the following commands for data communication using 1C frame.

| Function | | | ACPU common command | AnA/AnUCPU common command | Description |
|-------------------------|-------------------------|------------|---------------------|---------------------------|--|
| Device memory *1 | Batch read | Bit units | BR | JR | Reads bit devices (X, Y, M, etc.) in 1-point units. |
| | | Word units | WR | QR | Reads bit devices (X, Y, M, etc.) in 16-point units. Reads word devices (D, T, C, etc.) in 1-point units. |
| | Batch write | Bit units | BW | JW | Writes bit devices (X, Y, M, etc.) in 1-point units. |
| | | Word units | WW | QW | Writes bit devices (X, Y, M, etc.) in 16-point units. Writes word devices (D, T, C, etc.) in 1-point units. |
| | Test (random write) | Bit units | BT | JT | Set/reset devices and device numbers to bit devices (X, Y, M, etc.) by specifying them randomly in 1 point unit. |
| | | Word units | WT | QT | Set/reset devices and device numbers to bit devices (X, Y, M, etc.) by specifying them randomly in 16 point units. Write devices and device numbers to word devices (D, T, C, etc.) by specifying them randomly in 1 point units. |
| | Register monitor data*2 | Bit units | BM | JM | Registers bit devices (X, Y, M, etc.) to be monitored in 1-point units |
| | | Word units | WM | QM | Registers bit devices (X, Y, M, etc.) to be monitored in 16-point units. Registers word devices (D, T, C, etc.) to be monitored in 1-point units. |
| | Monitor | Bit units | MB | MJ | Monitors the devices registered by monitor data registration. |
| | | Word units | MN | MQ | |
| Extended file register | Batch read | | ER | — | Reads extended file register (R) in 1-point units. |
| | Batch write | | EW | | Writes extended file register (R) in 1-point units. |
| | Direct read | Word units | — | NR | Read data in 1 point units by specifying the consecutive device number regardless of the block number of extended file register. |
| | Direct write | Word units | | NW | Write data in 1 point units by specifying the consecutive device number regardless of the block number of extended file register. |
| | Test (Random write) | | ET | — | Write block numbers and device numbers to extended file register (R) by specifying them randomly in 1 point units. |
| | Register monitor data*2 | | EM | | Register extended file register (R) in 1-point units. |
| | Monitor | Word units | ME | | Monitor the extended file register (R) registered by monitor data registration. |
| Special function module | Batch read | | TR | — | Reads data in the buffer memory of a special function module. |
| | Batch write | | TW | | Writes data to the buffer memory of a special function module. |
| Loopback test | | | TT | — | Send (returns) characters received from an external device back to the external device unchanged. |

*1 Use the dedicated commands for extended registers to read/write extended file registers from/to ACPU.

*2 The devices for the five types of commands (BM, JM, WM, QM, EM) for registering monitor data can be registered simultaneously in C24 for each interface.

ACPU common command, AnA/AnUCPU common command

ACPU common command is a communication function issued by MC protocol. The command is accessible for ACPU.

AnA/AnUCPU common command is a command for AnACPU and AnUCPU. The command cannot be executed for other than AnA/AnUCPU.

○: Executable, △: Executable (with restrictions), ×: Not executable

| Type | ACPU other than AnA/AnUCPU | AnA/AnUCPU | Module other than ACPU |
|---------------------------|----------------------------|------------|------------------------|
| ACPU common command | ○ | ○ | △ |
| AnA/AnUCPU common command | × | ○ | △ |

When accessing modules other than ACPU, there is a restriction for the accessible device range.

☞ Page 344 Considerations when accessing devices other than ACPU module

17.4 Device Memory Read and Write

This section explains the specification content and examples of the control procedure when reading from/writing to the device memory are as shown below.

For the message formats other than request data and response data, refer to the following sections.

☞ Page 338 Message Format, Page 340 Details of Setting Data



To read and write the extended file register, use the commands dedicated to the extended file register.

☞ Page 365 Read and Write Extended File Register

Considerations

The considerations when reading/writing device memory using the commands described in this section.

Considerations when accessing devices other than ACPU module

■ Accessible devices

Only the devices with the same names that exist in ACPU can be accessed within the device range of AnACPU.

☞ Page 346 Accessible device range

The following devices cannot be accessed from the external devices:

- Added devices
- Latch relay (L) and step relay (S)^{*1}
- File register (R) of QnACPU

^{*1} Even when the latch relay (L) or step relay (S) is specified, the internal relay (M) can be accessed.

■ Special relays and special registers

Special relays and special registers can be accessed within the following range.

- Access SM1000 to SM1255 by specifying M9000 to M9255.
- Access SD1000 to SD1255 by specifying D9000 to D9255.

■ Universal model QCPU

Use the Universal model QCPU with a serial number whose first five digits are '10102' or later.

As for the serial number whose first five digits are '10101' or earlier, access using 2C/3C/4C frame.

Data to be specified in command

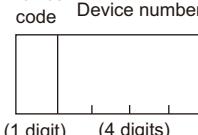
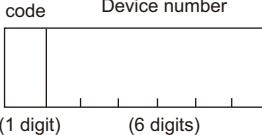
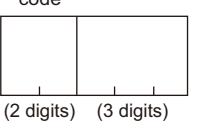
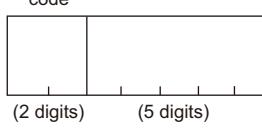
Device codes, device numbers

The settings of each device when reading/writing device memory can be performed using device code and device number as shown in the following figure.

Specify the device to be accessed by a device code and a device number.

The setting data size differ between ACPU common commands and AnA/AnUCPU commands.

The setting data size differ when the device type is timer or counter.

| Device type | ACPU common command | AnA/AnUCPU common command |
|------------------------------|---|---|
| Other than timer and counter | Device code Device number  (1 digit) (4 digits) | Device code Device number  (1 digit) (6 digits) |
| Timer, counter | Device code Device number  (2 digits) (3 digits) | Device code Device number  (2 digits) (5 digits) |

■ACPU common commands

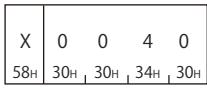
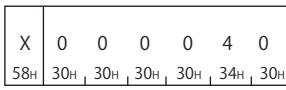
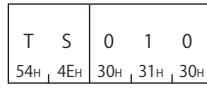
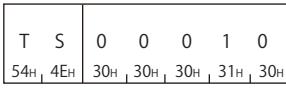
- Device code: Convert the device name to 1-digit ASCII code (2-digits for timer or counter), and send it from the upper digits.
- Device number: Convert the numerical value to 4-digit ASCII code (3-digits for timer or counter), and send it from the upper digits.

■AnA/AnUCPU common commands

- Device code: Convert the device name to 1-digit ASCII code (2-digits for timer or counter), and send it from the upper digits.
- Device number: Convert the numerical value to 6-digit ASCII code (5-digits for timer and counter), and send it from the upper digits.

Ex.

Current value of input (X) 40 and timer (T) 10

| Device | ACPU common command | AnA/AnUCPU common command |
|-------------------------------|---|---|
| Input (X) 40 |  58H 30H, 30H, 34H, 30H |  58H 30H, 30H, 30H, 30H, 34H, 30H |
| Current value of timer (T) 10 |  54H, 4EH 30H, 31H, 30H |  54H, 4EH 30H, 30H, 30H, 31H, 30H |

Accessible device range

The following table shows the devices and device number range that can be specified when accessing the device memory. Access the CPU module within the range of device number that can be used by commands and the range of the device number that can be used in the access target CPU. (☞ Page 344 Considerations when accessing devices other than ACPU module)

□: Specify 0 (30H) or space (20H)

| Device | | | | Device code | Device number | |
|------------------|---------------|--------|-------------|-------------|---------------------|---------------------------|
| Device name | | Symbol | Type | | ACPU common command | AnA/AnUCPU common command |
| Input | X | Bit | Hexadecimal | X | □□□0 to □7FF | □□□□□0 to □□1FFF |
| Output | Y | Bit | Hexadecimal | Y | □□□0 to □7FF | □□□□□0 to □□1FFF |
| Internal relay | M | Bit | Decimal | M | □□□0 to 2047 | □□□□□0 to □□8191 |
| Latch relay | L | Bit | Decimal | L | □□□0 to 2047 | □□□□□0 to □□8191 |
| Step relay | S | Bit | Decimal | S | □□□0 to 2047 | □□□□□0 to □□8191 |
| Annunciator | F | Bit | Decimal | F | □□□0 to □255 | □□□□□0 to □□2047 |
| Link relay | B | Bit | Hexadecimal | B | □□□0 to □3FF | □□□□□0 to □□1FFF |
| Timer | Current value | T | Word | Decimal | TN | □□0 to 255 |
| | Contact | | Bit | Decimal | TS | |
| | Coil | | Bit | Decimal | TC | |
| Counter | Current value | C | Word | Decimal | CN | □□0 to 255 |
| | Contact | | Bit | Decimal | CS | |
| | Coil | | Bit | Decimal | CC | |
| Data register | D | Word | Decimal | D | □□□0 to 1023 | □□□□□0 to □□8191 |
| Link register | W | Word | Hexadecimal | W | □□□0 to □3FF | □□□□□0 to □□1FFF |
| File register | R | Word | Decimal | R | □□□0 to 8191 | □□□□□0 to □□8191 |
| Special relay | M | Bit | Decimal | M | 9000 to 9255 | □□9000 to □□9255 |
| Special register | D | Word | Decimal | D | 9000 to 9255 | □□9000 to □□9255 |

Restriction

- Do not write data to the devices which cannot be written in the range of the special relays (M9000 to M9255) and special registers (D9000 to D9255). For details on the special relays and special registers, refer to manual of ACPU to be accessed.
- The range of M, L, and S devices can be specified for MELSEC-A series CPU module, however, if the range of M is specified by L or S or vice versa, they are processed equivalently.

Point

- For word unit specification, the head device number of bit device must be specified in multiple of 16.
- For special relay M9000 or later, (9000 + multiple of 16) can be specified.

Number of device points

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

Convert the numerical value to 2-digit ASCII code (hexadecimal), and send it from the upper digits.

Specify the number of device points in one command within the device points that can be processed in one communication.

(Page 459 Number of Processing per One Communication)

Specify '00' for 256 points.

Ex.

5 points, 20 points, 256 points

| Number of device points | ASCII code | | | | |
|-------------------------|--|---|---|-----|-----|
| 5 points | <table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td>0</td> <td>5</td> </tr> <tr> <td>30H</td> <td>35H</td> </tr> </table> | 0 | 5 | 30H | 35H |
| 0 | 5 | | | | |
| 30H | 35H | | | | |
| 20 points | <table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td>1</td> <td>4</td> </tr> <tr> <td>31H</td> <td>34H</td> </tr> </table> | 1 | 4 | 31H | 34H |
| 1 | 4 | | | | |
| 31H | 34H | | | | |
| 256 points | <table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td>0</td> <td>0</td> </tr> <tr> <td>30H</td> <td>30H</td> </tr> </table> | 0 | 0 | 30H | 30H |
| 0 | 0 | | | | |
| 30H | 30H | | | | |

Read data, write data

The data storage method is the same as reading/writing data with device access of 4C/3C/2C frame. (Page 70 Read data, write data)

Batch read (bit units) (command: BR, JR)

Reads bit devices (X, Y, M, etc.) in batch.

Message format

The following shows the message format of the request data and response data of the command.

■Request data

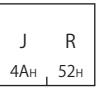
| | | | |
|---------|--------------|-------------|-------------------------|
| Command | Message wait | Head device | Number of device points |
|---------|--------------|-------------|-------------------------|

■Response data

The value of read device is stored in bit units. (☞ Page 70 Read data, write data)

Data specified by request data

■Command

| ACPU common | AnA/AnUCPU common |
|---|---|
|  |  |

■Message wait

Specify the delayed time of the response transmission. (☞ Page 341 Message wait)

■Head device

Specify the head device. (☞ Page 345 Device codes, device numbers)

■Number of device points

Convert the numerical value to 2-digit ASCII code (hexadecimal), and send it from the upper digits.

Specify the number of device points within the following range:

- $1 \leq$ Number of device points ≤ 256 (for 256 points, specify '00H')
- Head device No. + Number of device points - 1 \leq Maximum device No.

Communication example

Read data in bit units under the following conditions.

- Message wait: 100 ms
- Head device: X040
- Number of device points: 5 points

(Request data)

■When using BR (ACPU common command)

| | | | | Message wait | | Head device | | Number of device points | |
|-----|-----|-----|-----|--------------|-----|-------------|-----|-------------------------|-----|
| B | R | A | X | 0 | 0 | 4 | 0 | 0 | 5 |
| 42H | 52H | 41H | 58H | 30H | 30H | 34H | 30H | 30H | 35H |

■When using JR (AnA/AnUCPU common command)

| | | | | Message wait | | Head device | | Number of device points | |
|-----|-----|-----|-----|--------------|-----|-------------|-----|-------------------------|-----|
| J | R | A | X | 0 | 0 | 0 | 0 | 4 | 0 |
| 4AH | 52H | 41H | 58H | 30H | 30H | 30H | 30H | 34H | 30H |

(Response data)

| Data read | | | | |
|-----------|-----|-----|-----|-----|
| 0 | 1 | 1 | 0 | 1 |
| 30H | 31H | 31H | 30H | 31H |

(X40) (X41) (X42) (X43) (X44)

Batch read (word units) (command: WR, QR)

Reads bit devices (X, Y, M, etc.) in 16-point units.

Reads word devices (D, T, C, etc.) in 1-point units

Message format

The following shows the message format of the request data and response data of the command.

■Request data

| | | | |
|---------|--------------|-------------|--|
| Command | Message wait | Head device | Number of device points (Number of words) |
|---------|--------------|-------------|--|

■Response data

The value of read device is stored in word units. ([Page 70 Read data, write data](#))

Data specified by request data

■Command

| ACPU common | AnA/AnUCPU common |
|---|---|
|  |  |

■Message wait

Specify the delayed time of the response transmission. ([Page 341 Message wait](#))

■Head device

Specify the head device. ([Page 345 Device codes, device numbers](#))

■Number of device points

Convert the numerical value to 2-digit ASCII code (hexadecimal), and send it from the upper digits.

Specify the number of device points within the following range:

- For bit device: $1 \leq \text{Number of device points} \leq 32$
- For bit device: Head device No. + Number of device points $\times 16 - 1 \leq \text{Maximum device No.}$
- For word device: $1 \leq \text{Number of device points} \leq 64$
- For word device: Head device No. + Number of device points - 1 $\leq \text{Maximum device No.}$



- When specifying bit devices, set the head device No. in multiples of 16 (0, 16, ... in decimal notation).

Communication example (Reading bit device memory)

Read bit devices in 16-point units under the following conditions.

- Message wait: 0 ms
- Head device: X040
- Number of device points: 32 points (2 words)

(Request data)

■When using WR (ACPU common command)

| | | | | Message wait | | Head device | | Number of device points (Number of words) | |
|-----|-----|-----|-----|--------------|-----|-------------|-----|--|-----|
| W | R | 0 | X | 0 | 0 | 4 | 0 | 0 | 2 |
| 57H | 52H | 30H | 58H | 30H | 30H | 34H | 30H | 30H | 32H |

■When using QR (AnA/AnUCPU common command)

| | | | | Message wait | | Head device | | Number of device points (Number of words) | |
|-----|-----|-----|-----|--------------|-----|-------------|-----|--|-----|
| Q | R | 0 | X | 0 | 0 | 0 | 0 | 4 | 0 |
| 51H | 52H | 30H | 58H | 30H | 30H | 30H | 30H | 34H | 30H |
| | | | | | | | | | 32H |

(Response data)

Data read

| | | | | | | | |
|----------------------------------|----------------------------------|----------------------------------|----------------------------------|----------------------------------|----------------------------------|----------------------------------|----------------------------------|
| 0001 (Hexadecimal : 1) 31H | 0010 (Hexadecimal : 2) 32H | 0011 (Hexadecimal : 3) 33H | 0100 (Hexadecimal : 4) 34H | 1010 (Hexadecimal : A) 41H | 1011 (Hexadecimal : B) 42H | 1100 (Hexadecimal : C) 43H | 1101 (Hexadecimal : D) 44H |
| (X4F) to (X4C) | (X4B) to (X48) | (X47) to (X44) | (X43) to (X40) | (X5F) to (X5C) | (X5B) to (X58) | (X57) to (X54) | (X53) to (X50) |

Communication example (Reading word device memory)

Read word devices in 1-point units under the following conditions.

- Message wait: 0 ms
- Head device: Current value of T123
- Number of device points: 2 points (2 words)

(Request data)

■When using WR (ACPU common command)

| | | | | Message wait | | Head device | | Number of device points (Number of words) | |
|-----|-----|-----|-----|--------------|-----|-------------|-----|--|-----|
| W | R | 0 | T | N | 1 | 2 | 3 | 0 | 2 |
| 57H | 52H | 30H | 54H | 4EH | 31H | 32H | 33H | 30H | 32H |

■When using QR (AnA/AnUCPU common command)

| | | | | Message wait | | Head device | | Number of device points (Number of words) | |
|-----|-----|-----|-----|--------------|-----|-------------|-----|--|-----|
| Q | R | 0 | T | N | 0 | 0 | 1 | 2 | 3 |
| 51H | 52H | 30H | 54H | 4EH | 30H | 31H | 32H | 33H | 30H |
| | | | | | | | | | 32H |

(Response data)

Data read

| | | | | | | | |
|-----|-----|-----|-----|-----|-----|-----|-----|
| 7 | B | C | 9 | 1 | 2 | 3 | 4 |
| 37H | 42H | 43H | 39H | 31H | 32H | 33H | 34H |

(T123) (T124)

Batch write (bit units) (command: BW, JW)

Writes bit devices (X, Y, M, etc.) in batch.

Message format

The following shows the message format of the request data and response data of the command.

■Request data

ACPU common command

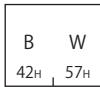
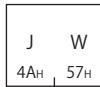
| | | | | |
|---------|--------------|-------------|-------------------------|--|
| Command | Message wait | Head device | Number of device points | Write data for the number of device points |
|---------|--------------|-------------|-------------------------|--|

■Response data

There is no response data for this command.

Data specified by request data

■Command

| ACPU common | AnA/AnUCPU common |
|---|---|
|  |  |

■Message wait

Specify the delayed time of the response transmission. ([Page 341 Message wait](#))

■Head device

Specify the head device. ([Page 345 Device codes, device numbers](#))

■Number of device points

Convert the numerical value to 2-digit ASCII code (hexadecimal), and send it from the upper digits.

Specify the number of device points within the following range:

- $1 \leq$ Number of device points ≤ 160
- Head device No. + Number of device points - 1 \leq Maximum device No.

■Write data for the number of device points

Store the data to be written in batch. ([Page 347 Read data, write data](#))

Communication example

Write data in bit units in batch under the following conditions.

- Message wait: 0 ms
- Head device: M903
- Number of device points: 5 points

(Request data)

■When using BW (ACPU common command)

| Message wait | | Number of device points (Number of words) | | | | | | | | | | | | |
|------------------------------------|-----|---|-----|-----|-----|-----|--|-----|-----|-----|-----|-----|-----|-----|
| | | Head device | | | | | Write data for the number of device points | | | | | | | |
| B | W | 0 | M | 0 | 9 | 0 | 3 | 0 | 5 | 0 | 1 | 1 | 0 | 1 |
| 42H | 57H | 30H | 4DH | 30H | 39H | 30H | 33H | 30H | 35H | 30H | 30H | 30H | 31H | 30H |
| (M903) (M904) (M905) (M906) (M907) | | | | | | | | | | | | | | |

■When using JW (AnA/AnUCPU common command)

| Message wait | | Number of device points (Number of words) | | | | | | | | | | | | | | |
|------------------------------------|-----|---|-----|-----|-----|-----|--|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| | | Head device | | | | | Write data for the number of device points | | | | | | | | | |
| J | W | 0 | M | 0 | 0 | 0 | 9 | 0 | 3 | 0 | 5 | 0 | 1 | 1 | 0 | 1 |
| 4AH | 57H | 30H | 4DH | 30H | 30H | 30H | 39H | 30H | 33H | 30H | 35H | 30H | 30H | 30H | 31H | 30H |
| (M903) (M904) (M905) (M906) (M907) | | | | | | | | | | | | | | | | |

Batch write (word units) (command: WW, QW)

Write data to bit devices (X, Y, M, etc.) in 16-point units.

Write data to word devices (D, T, C, etc.) in 1-point units.

Message format

The following shows the message format of the request data and response data of the command.

■Request data

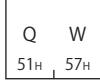
| | | | | |
|---------|--------------|-------------|-------------------------|--|
| Command | Message wait | Head device | Number of device points | Write data for the number of device points |
|---------|--------------|-------------|-------------------------|--|

■Response data

There is no response data for this command.

Data specified by request data

■Command

| ACPU common | AnA/AnUCPU common |
|---|---|
|  |  |

■Message wait

Specify the delayed time of the response transmission. ( Page 341 Message wait)

■Head device

Specify the head device. ( Page 345 Device codes, device numbers)

■Number of device points

Convert the numerical value to 2-digit ASCII code (hexadecimal), and send it from the upper digits.

Specify the number of device points within the following range:

- For bit device: $1 \leq$ Number of device points ≤ 10
- For bit device: Head device No. + Number of device points $\times 16 - 1 \leq$ Maximum device No.
- For word device: $1 \leq$ Number of device points ≤ 64
- For word device: Head device No. + Number of device points - 1 \leq Maximum device No.

■Write data for the number of device points

Store 4-digit data per one device point. ( Page 347 Read data, write data)



When specifying bit devices, set the head device No. in multiples of 16 (0, 16, ... in decimal notation).

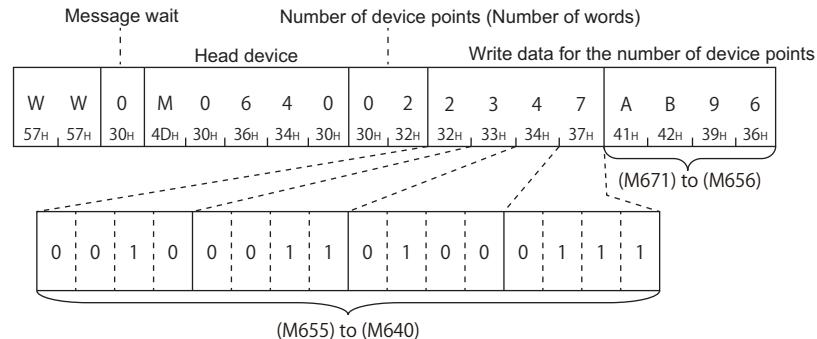
Communication example (Writing to word bit memory)

Write data to bit devices in 16-point units under the following conditions.

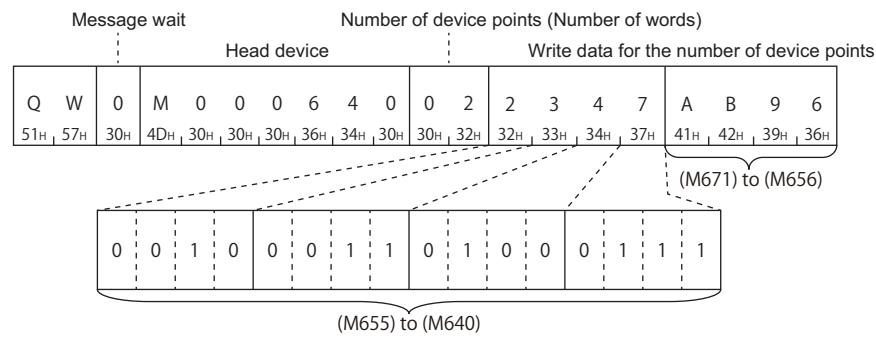
- Message wait: 0 ms
- Head device: M640
- Number of device points: 32 points (2 words)

(Request data)

■When using WW (ACPU common command)



■When using QW (AnA/AnUCPU common command)



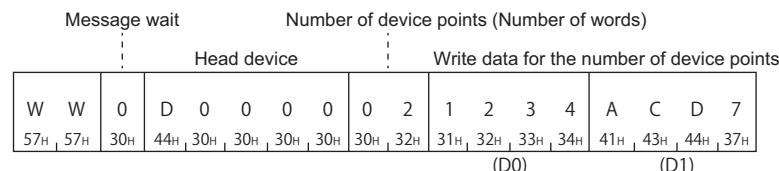
Communication example (Writing to word device memory)

Write data to word devices in 1-point units under the following conditions.

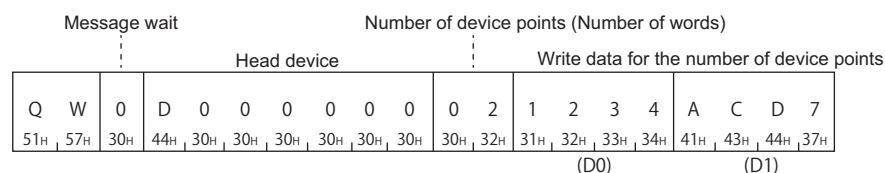
- Message wait: 0 ms
- Head device: D0
- Number of device points: 2 points (2 words)

(Request data)

■When using WW (ACPU common command)



■When using QW (AnA/AnUCPU common command)



Test (random write) (bit units) (command: BT, JT)

Set/reset devices and device numbers to bit devices (X, Y, M, etc.) by specifying them randomly in 1 point units.

Message format

The following shows the message format of the request data and response data of the command.

■Request data

| | | | | | | | |
|---------|--------------|------------------------------------|----------------------|-------------------------|-----|--------------------|-----------------------|
| Command | Message wait | Number of device points (n points) | Device (first point) | Set/reset (first point) | ... | Device (nth point) | Set/reset (nth point) |
|---------|--------------|------------------------------------|----------------------|-------------------------|-----|--------------------|-----------------------|

■Response data

There is no response data for this command.

Data specified by request data

■Command

| ACPU common | AnA/AnUCPU common |
|---|---|
|  |  |

■Message wait

Specify the delayed time of the response transmission. ( Page 341 Message wait)

■Number of device points

Convert the numerical value to 2-digit ASCII code (hexadecimal), and send it from the upper digits.

Specify the number of device points within the following range:

- $1 \leq$ Number of device points ≤ 20

■Device

Specify the device to test. ( Page 345 Device codes, device numbers)

■Set/Reset

- 0 (30H): Reset (OFF)
- 1 (31H): Set (ON)

Communication example

Perform the test in bit units under the following conditions.

- Message wait: 0 ms
- Number of device points: 3 points
- Device: Turn ON M50, turn OFF B31A, and turn ON Y02F

(Request data)

■When using BT (ACPU common command)

| Message wait | Number of device points (Number of words) | | | Set/ reset | Device | Set/ reset | Device | Set/ reset | Device | Set/ reset |
|--------------|--|-----|-----|---------------|-----------------|---------------|--------|---------------|--------|---------------|
| B | T | 0 | 0 | 3 | M | 0 | 0 | 5 | 0 | 1 |
| 42H | 54H | 30H | 30H | 33H | 4D _H | 30H | 30H | 35H | 30H | 31H |

■When using JT (AnA/AnUCPU common command)

| Message wait | Number of device points (Number of words) | | | Set/ reset | Device | Set/ reset | Device | Set/ reset | Device | Set/ reset |
|-----------------|--|-----|-----|---------------|-----------------|---------------|--------|---------------|--------|---------------|
| J | T | 0 | 0 | 3 | M | 0 | 0 | 0 | 5 | 0 |
| 4A _H | 54H | 30H | 30H | 33H | 4D _H | 30H | 30H | 30H | 35H | 30H |
| Set/ reset | | | | | | | | | | |
| Y | 0 | 0 | 0 | 0 | 0 | 2 | F | 1 | | |
| 59H | 30H | 30H | 30H | 30H | 32H | 46H | | 31H | | |

Test (random write) (word units) (command: WT, QT)

Set/reset devices and device numbers to bit devices (X, Y, M, etc.) by specifying them randomly in 16 point units.
Write devices and device numbers to word devices (D, T, C, etc.) by specifying them randomly in 1 point units.
A mixture of word devices and bit devices (16 bit units) can be specified.

Message format

The following shows the message format of the request data and response data of the command.

■Request data

| | | | | | | | |
|---------|--------------|------------------------------------|----------------------|--------------------------|-----|--------------------|------------------------|
| Command | Message wait | Number of device points (n points) | Device (first point) | Write data (first point) | ... | Device (nth point) | Write data (nth point) |
|---------|--------------|------------------------------------|----------------------|--------------------------|-----|--------------------|------------------------|

■Response data

There is no response data for this command.

Data specified by request data

■Command

| ACPU common | AnA/AnUCPU common |
|---|---|
|  |  |

■Message wait

Specify the delayed time of the response transmission. (☞ Page 341 Message wait)

■Number of device points

Convert the numerical value to 2-digit ASCII code (hexadecimal), and send it from the upper digits.

Specify the number of device points within the following range:

- $1 \leq$ Number of device points ≤ 10 (for bit device : 10 (16 points are designated as 1))

■Device

Specify the device to test. (☞ Page 345 Device codes, device numbers)

■Write data

Store 4-digit data per one device point. (☞ Page 347 Read data, write data)



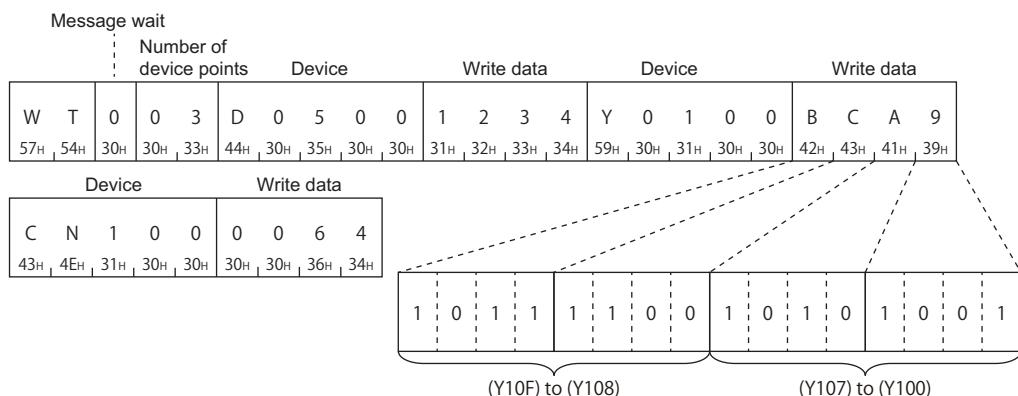
When specifying bit devices, set the head device No. in multiples of 16 (0, 16, ... in decimal notation).

Communication example

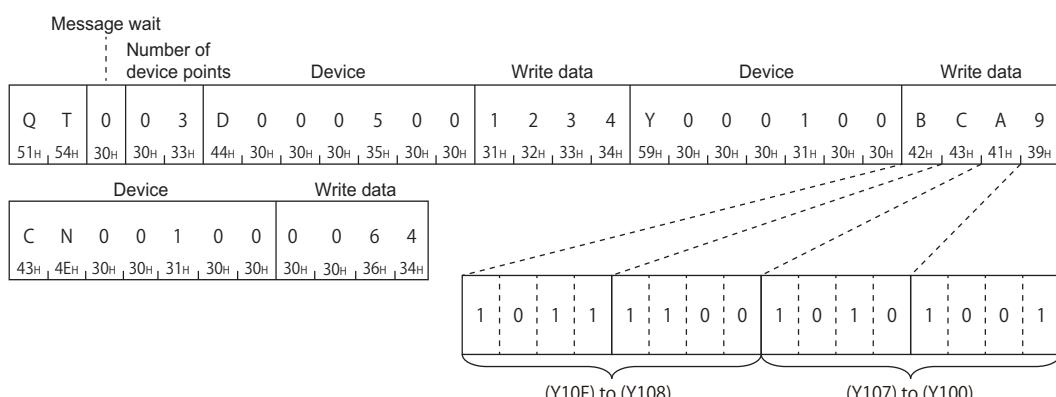
Write data with mixture specification of word devices and bit devices (16-point unit) under the following conditions.

- Message wait: 0 ms
 - Number of device points: 3 points (3 words)
 - Device: Set 1234H to D500, BCA9H from Y100 to Y10F, and 64H to current values of C100 (Request data)

■When using WT (ACPU common command)



■When using QT (AnA/AnUCPU common command)



Monitor (Command: BM, JM, WM, QM, MB, MJ, MN, MQ)

The monitor data registration function registers the devices and numbers to be monitored from an external device to C24. The monitor function reads the data of the registered devices from the CPU module and processes it in the external device. When the batch read (BR/WR/JR/QR) is performed, the read device numbers will be consecutive, however, by using this function, devices can be monitored by specifying the device numbers randomly. The following example shows the control procedure for monitoring and registering name and number of the devices to be monitored to the C24.

Monitoring procedure

- 1.** Process the monitor data registration (Edit of commands for registration and transmission of device specification.)
 - ACPU common commands: BM, WM
 - AnA/AnUCPU common commands: JM, QM
- 2.** Perform read process. (Execution of command for monitoring.)
 - ACPU common commands: MB, MN
 - AnA/AnUCPU common commands: MJ, MQ
- 3.** Process the data. (Screen display, etc.)
- 4.** If do not change the devices to be monitored, return to step 2, and repeat the process.

Point

- When monitoring data as the procedure shown above, the monitor data registration is required. If monitoring data without registering the data, a protocol error occurs.
- The content of registered monitor data are deleted when C24 is rebooted.
- The devices can be registered for each command for bit units (BM or JM), word units (WM or QM), or the extended file register (EM) in C24. (Monitoring extended file register  Page 372 Monitor (command: EM, ME))
- When registering device memory of the CPU module as a monitor data from more than one C24s on the same station, the recently registered device memory will be available since the registration data is overwritten.

Register monitor data (bit units) (command: BM, JM)

Set the bit devices (X, Y, M, etc.) to be monitored in 1-point units.

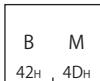
Monitor the device memory registered in bit units using the following command.

 Page 363 Monitor (bit units) (command: MB, MJ)

■Request data

| Command | Message wait | Number of device points (n points) | Device (first point) | ... | Device (nth point) |
|---------|--------------|------------------------------------|----------------------|-----|--------------------|
|---------|--------------|------------------------------------|----------------------|-----|--------------------|

- Command

| ACPU common | AnA/AnUCPU common |
|---|---|
|  |  |

- Message wait: Specify the delayed time of the response transmission. ( Page 341 Message wait)
- Number of device points: Convert the numerical value to 2-digit ASCII code (hexadecimal), and send it from the upper digits.
- Device: Specify the devices to be monitored. ( Page 345 Device codes, device numbers)



Specify the number of device points within the following range:

1 ≤ Number of device points ≤ 40

When using the BM command and accessing ACPU other than AnA/AnU, device X (input) has two processing points per point.

■Response data

There is no response data for this command.

■Communication example (Monitor data registration in bit units)

Perform monitor data registration of the bit device under the following conditions.

- Message wait: 0 ms
- Number of device points: 3 points (3 bits)
- Device: Contacts of X40, Y060, and T123.

(Request data)

- When using BM (ACPU common command)

| Message wait | | Number of device points | | Device | | Device | | Device | |
|--------------|---|-------------------------|---|--------|---|--------|---|--------|---|
| B | M | 0 | 0 | 3 | X | 0 | 0 | 4 | 0 |

42H, 4DH 30H, 33H 58H, 30H, 35H, 34H, 30H 59H, 30H, 30H, 36H, 30H 54H, 53H, 31H, 32H, 33H

- When using JM (AnA/AnUCPU common command)

| Message wait | | Number of device points | | Device | | Device | | Device | | |
|--------------|---|-------------------------|---|--------|---|--------|---|--------|---|---|
| J | M | 0 | 0 | 3 | X | 0 | 0 | 0 | 4 | 0 |

4AH, 4DH 30H, 33H 58H, 30H, 30H, 30H, 34H, 30H 59H, 30H, 30H, 30H, 36H, 30H 54H, 53H, 30H, 31H, 32H, 33H

Register monitor data (word units) (command: WM, QM)

Register the bit devices (X, Y, M, etc.) to be monitored in 16-point units.

Register the word devices (D, T, C, etc.) to be monitored in 1-point units.

A mixture of word devices and bit devices (16 bit units) can be specified.

Monitor the device memory registered in word units using the following command.

☞ Page 364 Monitor (word units) (command: MN, MQ)

■Request data

| Command | Message wait | Number of device points (n points) | Device (first point) | ... | Device (nth point) |
|---------|--------------|------------------------------------|----------------------|-----|--------------------|
|---------|--------------|------------------------------------|----------------------|-----|--------------------|

- Command

| ACPU common | AnA/AnUCPU common | | | | | | | | |
|--|-------------------|---|-----|-----|--|---|---|-----|-----|
| <table border="1"><tr><td>W</td><td>M</td></tr><tr><td>57H</td><td>4DH</td></tr></table> | W | M | 57H | 4DH | <table border="1"><tr><td>Q</td><td>M</td></tr><tr><td>51H</td><td>4DH</td></tr></table> | Q | M | 51H | 4DH |
| W | M | | | | | | | | |
| 57H | 4DH | | | | | | | | |
| Q | M | | | | | | | | |
| 51H | 4DH | | | | | | | | |

- Message wait: Specify the delayed time of the response transmission. (☞ Page 341 Message wait)
- Number of device points: Convert the numerical value to 2-digit ASCII code (hexadecimal), and send it from the upper digits.
- Device: Specify the devices to be monitored. (☞ Page 345 Device codes, device numbers)



Specify the number of device points within the following range:

1 ≤ Number of device points ≤ 20

When using the WM command and accessing ACPU other than AnA/AnU, device X (input) has two processing points per point.

■Response data

There is no response data for this command.

■Communication example (Monitor data registration in word units)

Register the monitor data in a word unit under the following conditions.

- Message wait: 0 ms
- Number of device points: 4 points (4 words)
- Device: Current value of D15, W11E, and T123, and Y060 to Y06F

(Request data)

- When using WM (ACPU common command)

| Message wait | | Number of device points | | Device | | Device | | Device | | Device | | | | | | | | | | | | | | |
|--------------|-----|-------------------------|-----|--------|-----|--------|-----|--------|-----|--------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| W | M | 0 | 0 | 4 | D | 0 | 0 | 1 | 5 | W | 0 | 1 | 1 | E | T | N | 1 | 2 | 3 | Y | 0 | 0 | 6 | 0 |
| 57H | 4DH | 30H | 30H | 34H | 44H | 30H | 30H | 31H | 35H | 57H | 30H | 31H | 31H | 45H | 54H | 4EH | 31H | 32H | 33H | 59H | 30H | 30H | 36H | 30H |

- When using QM (AnA/AnUCPU common command)

| Message wait | | Number of device points | | Device | | Device | | Device | | Device | | | | | | | | | | | | | | |
|--------------|-----|-------------------------|-----|--------|-----|--------|-----|--------|-----|--------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| Q | M | 0 | 0 | 4 | D | 0 | 0 | 0 | 1 | 5 | W | 0 | 0 | 0 | 1 | 1 | E | T | N | 0 | 0 | 1 | 2 | 3 |
| 51H | 4DH | 30H | 30H | 34H | 44H | 30H | 30H | 30H | 31H | 35H | 57H | 30H | 30H | 30H | 31H | 31H | 45H | 54H | 4EH | 30H | 30H | 31H | 32H | 33H |
| Device | | | | | | | | | | | | | | | | | | | | | | | | |
| Y | 0 | 0 | 0 | 0 | 0 | 6 | 0 | 59H | 30H | 30H | 30H | 30H | 36H | 30H | | | | | | | | | | |

Monitor (bit units) (command: MB, MJ)

Monitor the registered bit devices (X, Y, M, etc.).

Point

- The bit device memory registered with BM command is monitored using MB command.
- The bit device memory registered with JM command is monitored using MJ command.

■Request data

| | |
|---------|--------------|
| Command | Message wait |
|---------|--------------|

- Command

| ACPU common | AnA/AnUCPU common | | | | | | | | |
|---|-------------------|---|-----|-----|---|---|---|-----|-----|
| <table border="1"> <tr> <td>M</td> <td>B</td> </tr> <tr> <td>4DH</td> <td>42H</td> </tr> </table> | M | B | 4DH | 42H | <table border="1"> <tr> <td>M</td> <td>J</td> </tr> <tr> <td>4DH</td> <td>4AH</td> </tr> </table> | M | J | 4DH | 4AH |
| M | B | | | | | | | | |
| 4DH | 42H | | | | | | | | |
| M | J | | | | | | | | |
| 4DH | 4AH | | | | | | | | |

- Message wait: Specify the delayed time of the response transmission. ([Page 341 Message wait](#))

■Response data

| |
|---|
| Monitoring result (For the number of device points) |
|---|

The value of read device is stored in bit units. ([Page 70 Read data, write data](#))

■Communication example

Monitor bit devices specified with monitor data registration under the following conditions.

- Message wait: 0 ms
 - Registered devices for monitoring: 3 points (3 bits) of contacts for X040, Y060 and T123.
- (Request data)
- When using MB (ACPU common command)

| | | |
|--------------|---|---|
| Message wait | | |
| M | B | 0 |

4DH 42H 30H

- When using MJ (AnA/AnUCPU common command)

| | | |
|--------------|---|---|
| Message wait | | |
| M | J | 0 |

4DH 4AH 30H

(Response data)

Monitoring result (For the number of device points)

| | | |
|---|---|---|
| 1 | 0 | 1 |
|---|---|---|

31H 30H 31H

(X040) (Y060) (Contact of T123)

Monitor (word units) (command: MN, MQ)

Monitor the registered bit devices (X, Y, M, etc.) in 16-point units.

Monitor the registered word device (D, T, C, etc.) in 1-point unit.



- The bit device memory registered with WM command is monitored using MN command.
- The bit device memory registered with QM command is monitored using MQ command.

■Request data

| | |
|---------|--------------|
| Command | Message wait |
|---------|--------------|

- Command

| ACPU common | AnA/AnUCPU common |
|----------------|-------------------|
| M N 4DH 4EH | M Q 4DH 51H |

- Message wait: Specify the delayed time of the response transmission. ([Page 341 Message wait](#))

■Response data

| |
|---|
| Monitoring result (For the number of device points) |
|---|

The value of read device is stored in word units. ([Page 70 Read data, write data](#))

■Communication example

Monitor bit devices or word devices specified with monitor data registration under the following conditions.

- Message wait: 0 ms
- Registered devices for monitoring: Current values of D15, W11E, T123, and 4 points (4 words) of Y060 to Y06F.
(Request data)
- When using MN (ACPU common command)

| | | |
|--------------|---|---|
| Message wait | | |
| M | N | 0 |

- When using MQ (AnA/AnUCPU common command)

| | | |
|--------------|---|---|
| Message wait | | |
| M | Q | 0 |

(Response data)

Monitoring result (For the number of device points)

| | | | | | | |
|--|--|---|---|----------------|----------------|----------------|
| 1 2 3 4 (Decimal : 4660) 31H 32H 33H 34H | 0 0 5 0 (Decimal : 80) 30H 30H 35H 30H | 0 0 6 4 (Decimal : 100) 30H 30H 36H 34H | 0 0 0 0 30H | 0 1 1 1 37H | 0 1 1 0 36H | 0 1 0 0 34H |
| (D15) | (W11E) | (Current value of T123) | (Y06F) to (Y06C) (Y06B) to (Y068) (Y067) to (Y064) (Y063) to (Y060) | | | |

17.5 Read and Write Extended File Register

The extended file register is a memory area that stores required data and operation result for various data processing by using the software package for extended file register 'SWOGHP-UTLPC-FN1' or 'SW0SRX-FNUP' (hereinafter abbreviated to UTLPC-FN1 and FNUP) and AnACPU and AnUSCPU extended file register dedicated instructions. The extended file register uses free area of user memory area in CPU module as a file register.

The following example shows the control procedure to read and write extended file register.

Considerations for reading and writing extended file register

The following shows the considerations when reading/writing extended file register using the commands described in this section.

Accessible CPU modules

Only CPU modules that can handle the extended file register can be accessed.

This function cannot be used for CPU modules that cannot handle an extended file register (such as A1N).

Error detection for block numbers that do not exist

Depending on the type of memory cassette inserted in the CPU module, an error (character area error '06H') may not be detected even when a read/write operation is performed by specifying the block numbers that do not exist. In this case, the read data is incorrect. If writing data to the CPU module, the user memory of the CPU module will be collapsed.

Check the type of memory cassette and parameter settings before using these functions.

| Memory cassette model name | Block numbers that do not cause character area error (06H) | | |
|----------------------------|--|----------------|------------------------------|
| | A0J2H, A2, A3CPU | A2N, A3NCPU | A3H, AnA, AnUCPU |
| A3NMCA-12 | No.10 to No.11 | | |
| A3NMCA-18 | — | No.10 to No.28 | |
| A3NMCA-24 | — | No.13 to No.20 | No.13 to No.28 |
| A3NMCA-40 | — | | No.21 to No.28 |
| A3AMCA-96 | — | | No.21 to No.48 ^{*1} |

*1 A3AMCA-96 can be used for A3A, A3U, and A4UCPU.

For details, refer to the manual of UTLPC-FN1 or FNUP, or user's manual of the access target CPU module.

Block numbers of extended file register that can be handled by A2USCPU(S1)

The block numbers of the extended file register that can be handled by A2USCPU(S1) is as follows.

- A2USCPU: No.1 to 3
- A2USCPU-S1: No.1 to 8, No.10 to 16

Extended file register of R/L/Q/QnACPU

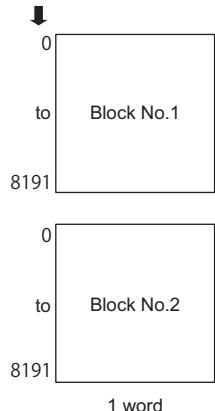
The extended file register of R/L/Q/QnACPU cannot be read/written.

Specification method for extended file register

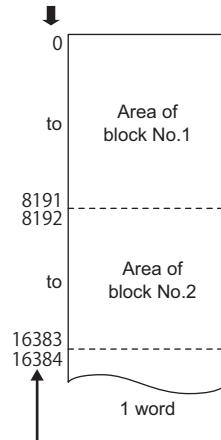
The specification method differs depending on the command.

- ACPU common command: Specify with device number and block number.
- AnA/AnUCPU common command: Specify the address from device number 0 of block number 1 as a device number.
(Access with the consecutive device number of extended file register using usable number of blocks × 8192 points.)

Device numbers specified
with the ACPU common commands



Device numbers specified with the
AnA/AnUCPU common commands



Range of block number and device number that can be specified

An extended file register has blocks numbered from '0' to 'n' (the n differs depending on the memory cassette). Block number '0' has a number of points registered with a parameter of the CPU module, while block numbers '1' to 'n' have a register of 8192 points in each block.

However, the range that can be read from or written to the CPU module will be the range specified in the parameter of 0 block. The range of block numbers and device numbers that can be specified differ depending on the type of memory cassette and the parameter setting in the CPU module. For details, refer to the operating manual of UTLP-FN1 or FNUP, or user's manuals of AnACPU and AnUCPU.

Point

- The AnA/AnUCPU common commands can be used for reading/writing data to the extended file register of block number 1 to 256. In addition, the commands can be used regardless of the existence of file register parameter.
- When accessing the file register (R) set by parameter or when accessing it by specifying block number, use ACPU common commands.

Device number (address) specification using AnA/AnUCPU common commands

By using AnA/AnU common command function, the extended file register of block number 1 to 256 can be accessed regardless of each block number by specifying the address from device number '0' of the block number '1' as a device number. (Access with the consecutive device number of extended file register using usable number of blocks × 8192 points.)

■Device number calculation method

The calculation formula of the head device number to be specified by AnA/AnUCPU common commands is as follow:

(When specifying the device number 'm' (0 to 8191) of n block (more than 1) from the head of the block.)

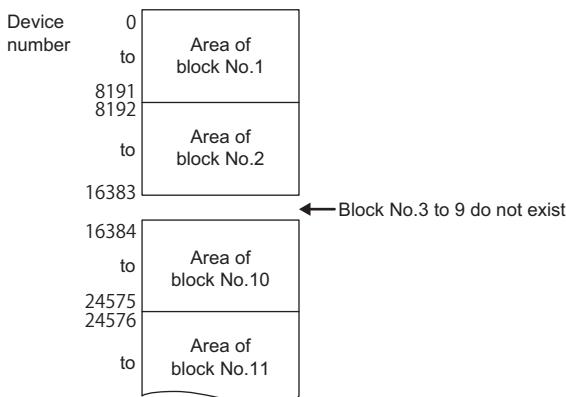
$$\text{Head device number} = (n - 1) \times 8192 + m$$

■Range of device number that can be specified

The range of device number that can be specified are as follows.

$$0 \text{ to } (\text{number of available blocks} \times 8192) - 1$$

A device number is not assigned to a block number which does not exist in the memory cassette. As shown below, the device numbers are automatically assigned by skipping block numbers that do not exist in the memory cassette.



The following table shows the range of device numbers to be specified when using AnA/AnUCPU common commands for 28 blocks per each block.

| Device number | Position of target block | Device number | Position of target block |
|------------------|--------------------------|------------------|--------------------------|
| 0 to 8191 | First block R0 to R8191 | 114688 to 122879 | 15th block R0 to R8191 |
| 8192 to 16383 | 2nd block R0 to R8191 | 122880 to 131071 | 16th block R0 to R8191 |
| 16384 to 24575 | 3rd block R0 to R8191 | 131072 to 139263 | 17th block R0 to R8191 |
| 24576 to 32767 | 4th block R0 to R8191 | 139264 to 147455 | 18th block R0 to R8191 |
| 32768 to 40959 | 5th block R0 to R8191 | 147456 to 155647 | 19th block R0 to R8191 |
| 40960 to 49151 | 6th block R0 to R8191 | 155648 to 163839 | 20th block R0 to R8191 |
| 49152 to 57343 | 7th block R0 to R8191 | 163840 to 172031 | 21st block R0 to R8191 |
| 57344 to 65535 | 8th block R0 to R8191 | 172032 to 180223 | 22nd block R0 to R8191 |
| 65536 to 73727 | 9th block R0 to R8191 | 180224 to 188415 | 23rd block R0 to R8191 |
| 73728 to 81919 | 10th block R0 to R8191 | 188416 to 196607 | 24th block R0 to R8191 |
| 81920 to 90111 | 11th block R0 to R8191 | 196608 to 204799 | 25th block R0 to R8191 |
| 90112 to 98303 | 12th block R0 to R8191 | 204800 to 212991 | 26th block R0 to R8191 |
| 98304 to 106495 | 13th block R0 to R8191 | 212992 to 221183 | 27th block R0 to R8191 |
| 106496 to 114687 | 14th block R0 to R8191 | 221184 to 229375 | 28th block R0 to R8191 |

Data to be specified in command

Device number

■ACPU common command

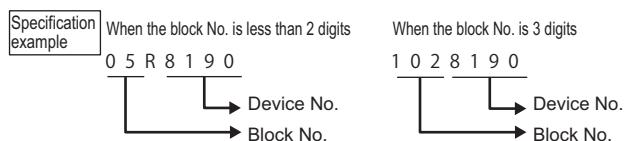
Specify the block number and device number with 7 digits.

- When the block number is less than 2 digits

'Block number (2 digits)' + 'R' + 'Device number (4 digits)'

- When the block number is 3 digits

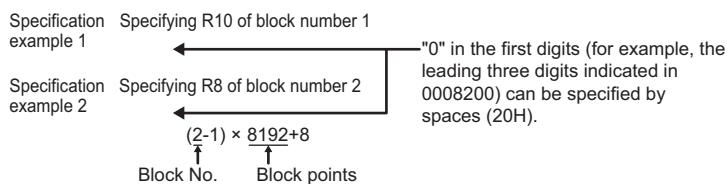
'Block number (3 digits)' + 'Device number (4 digits)'



■AnA/AnUCPU common command

Specify the address from device number 0 of block number 1 in 7 digits.

☞ Device number (address) specification using AnA/AnUCPU common commands



Batch read (command: ER)

Read extended file register (R) in 1-point units.

Message format

The following shows the message format of the request data and response data of the command.

■Request data

| | | | | |
|----------|----------|--------------|-----------------|--|
| E 45H | R 52H | Message wait | Head device No. | Number of device points (Number of words) |
|----------|----------|--------------|-----------------|--|

■Response data

The value of read device is stored in word units. (☞ Page 70 Read data, write data)

Data specified by request data

■Command

ACPU common

| | |
|----------|----------|
| E 45H | R 52H |
|----------|----------|

■Message wait

Specify the delayed time of the response transmission. (☞ Page 341 Message wait)

■Head device No.

Specify the block number and device number of head device with 7 digits. (☞ Page 368 Device number)

■Number of device points

Convert the numerical value to 2-digit ASCII code (hexadecimal), and send it from the upper digits.

Specify the number of device points within the following range:

- $1 \leq$ Number of device points ≤ 64
- Head device No. + Number of device points - 1 \leq Maximum device No.

Communication example

Read data of 2 points (R8190 and R8191) of block number 12 under the following conditions.

- Message wait: 0 ms
- Block number: 12
- Head device: R8190
- Number of device points: 2 points (2 words)

(Request data)

| | | |
|--------------|-----------------|--|
| Message wait | Head device No. | Number of device points (Number of words) |
| : | | |
| E 45H | R 52H | 0 1 2 R 8 1 9 0 0 2 30H 31H, 32H, 52H, 38H, 31H, 39H, 30H, 30H, 32H |

(Response data)

| | |
|--|---|
| Data read | |
| 1 2 3 4 (Decimal : 4660) 31H, 32H, 33H, 34H (R8190 in NO.12) | 7 A B C (Decimal : 31420) 37H, 41H, 42H, 43H (R8191 in NO.12) |

Batch write (command: EW)

Write extended file register (R) in 1-point units.

Message format

The following shows the message format of the request data and response data of the command.

■Request data

| | | | | |
|----------------|--------------|-----------------|-------------------------|--|
| E W 45H 57H | Message wait | Head device No. | Number of device points | Write data for the number of device points |
|----------------|--------------|-----------------|-------------------------|--|

■Response data

There is no response data for this command.

Data specified by request data

■Command

ACPU common

| |
|----------------|
| E W 45H 57H |
|----------------|

■Message wait

Specify the delayed time of the response transmission. (☞ Page 341 Message wait)

■Head device No.

Specify the block number and device number of head device with 7 digits. (☞ Page 368 Device number)

■Number of device points

Convert the numerical value to 2-digit ASCII code (hexadecimal), and send it from the upper digits.

Specify the number of device points within the following range:

- $1 \leq$ Number of device points ≤ 64
- : Head device No. + Number of device points - 1 \leq Maximum device No.

■Write data for the number of device points

Store the data to be written in batch.

Communication example

Write data of 3 points (R7010 to R7012) of block number 5 under the following conditions.

- Message wait: 0 ms
- Block number: 5
- Head device: R7010
- Number of device points: 3 points (3 words)

(Request data)

| Message wait | | Number of device points (Number of words) | | | | | | | | | | | | | | | | | | | | | | |
|--------------|-----|---|-----|-----|-----|-----|--|-----|-----|-----|-----|-----------------|-------------------|-------------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| | | Head device No. | | | | | Write data for the number of device points | | | | | 0 | 1 | 2 | 3 | A | B | C | 7 | 3 | 3 | 2 | 2 | |
| E | W | 0 | 0 | 5 | R | 7 | 0 | 1 | 0 | 0 | 3 | (Decimal : 291) | (Decimal : 21753) | (Decimal : 13090) | 41H | 42H | 43H | 37H | 33H | 33H | 33H | 32H | 32H | |
| 45H | 57H | 30H | 30H | 35H | 35H | 52H | 37H | 30H | 31H | 30H | 30H | 33H | 30H | 31H | 32H | 33H | 41H | 42H | 43H | 37H | 33H | 33H | 32H | 32H |

Test (random write) (command: ET)

Specify the block number and device number to the extended file register (R) in 1-point units and write them randomly.

Message format

The following shows the message format of the request data and response data of the command.

■Request data

| Data for the number of device points | | | | | |
|--------------------------------------|----------|--------------|-------------------------|------------|------------|
| E 45H | T 54H | Message wait | Number of device points | Device No. | Write data |

■Response data

There is no response data for this command.

Data specified by request data

■Command

ACPU common

| | |
|-----|-----|
| E | T |
| 45H | 54H |

■Message wait

Specify the delayed time of the response transmission. (☞ Page 341 Message wait)

■Number of device points

Convert the numerical value to 2-digit ASCII code (hexadecimal), and send it from the upper digits.

Specify the number of device points within the following range:

- $1 \leq$ Number of device points ≤ 10

■Device number

Specify the block number and device number to be test with 7 digits. (☞ Page 368 Device number)

■Write data

Store 4 characters per one device point.

Communication example

Write data of three points (3 words) randomly under the following conditions.

- Message wait: 0 ms
- Number of device points: 3 points (3 words)
- Device: Set R1234H to R1050 of block number 5, 1A1BH to R2121 of block number 7, and 506H to R3210 of block number 10.

(Request data)

| Message wait | | Number of device points | | Device | | Write data | | Device | | Write data | |
|--------------|----------|-------------------------|----------|----------|----------|------------|----------|----------|----------|------------|----------|
| E 45H | T 54H | 0 30H | 0 30H | 3 33H | 0 30H | 5 35H | R 52H | 1 31H | 0 30H | 5 35H | 0 30H |
| 1 31H | 0 30H | 2 52H | 1 33H | 3 32H | 1 31H | 4 30H | 1 32H | 2 33H | 3 34H | 1 30H | 2 37H |
| 0 30H | 7 52H | R 52H | 2 32H | 1 31H | 1 32H | 1 31H | 2 31H | 1 31H | 1 31H | A 41H | 1 31H |
| 1 31H | 0 30H | 3 52H | 2 33H | 1 32H | 0 31H | 0 30H | 5 35H | 0 30H | 6 36H | B 42H | |

Monitor (command: EM, ME)

The monitor data registration function registers the devices and numbers to be monitored from an external device to C24. The monitor function reads the data of the registered devices from the CPU module and processes it in the external device. When the batch read (ER) or direct read (NR) is performed, the read device numbers will be consecutive, however, by using this function, devices can be monitored by specifying the device numbers randomly. The following example shows the control procedure for monitoring and registering name and number of the devices to be monitored to the C24.

Monitoring procedure

1. Process the monitor data registration (Edit of EM command and transmission of device specification.)
 - Register monitor data (command: EM)
2. Perform read process. (Execution of ME command)
 - Monitor (command: ME)
3. Process the data. (Screen display, etc.)
4. If do not change the devices to be monitored, return to step 2, and repeat the process.

Point

- When monitoring data as the procedure shown above, the monitor data registration is required. If monitoring data without registering the data, a protocol error occurs.
- The content of registered monitor data are deleted when C24 is rebooted.
- Five kinds of monitor data can be registered for the extended file register (EM), device memory in bit units (BM or JM), and word units (WM or QM).
- When registering device memory of the CPU module as a monitor data from more than one external devices on the same station, the recently registered device memory will be available since the registration data is overwritten. (Monitoring device memory Page 360 Monitor (Command: BM, JM, WM, QM, MB, MJ, MN, MQ))

Register monitor data (command: EM)

Register the device number to be monitored in 1-point units.

Monitor the extended file register registered with EM command using the following command.

☞ Page 374 Monitor (command: ME)

■Request data

| | | | |
|----------------|--------------|-------------------------|---|
| E M 45H 4DH | Message wait | Number of device points | Device No. (for the number of device points) |
|----------------|--------------|-------------------------|---|

- Command

ACPU common

| |
|----------------|
| E M 45H 4DH |
|----------------|

- Message wait: Specify the delayed time of the response transmission. (☞ Page 341 Message wait)
- Number of device points: Convert the numerical value to 2-digit ASCII code (hexadecimal), and send it from the upper digits.
- Device number: Specify the block number and device number of the device to be monitored with 7 digits. (☞ Page 368 Device number)



Specify the number of device points within the following range:

1 ≤ Number of device points ≤ 20

■Response data

There is no response data for this command.

■Communication example

Perform monitor data registration of 4 points (4 words) under the following conditions.

- Message wait: 0 ms
- Number of device points: 4 points (4 words)
- Device number: Register R1234 of block number 5, R2345 of block number 6, R3055 of block number 15, and R8000 of block number 17.

(When the extended file register of block number 1 to 8 and 10 to 17 exist.)

| Message wait | | | | | | | | | | | | | | | | | | | | | | | | | |
|--------------|-----|-------------------------|-----|------------|-----|-----|-----|------------|-----|-----|-----|------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| | | Number of device points | | Device No. | | | | Device No. | | | | Device No. | | | | | | | | | | | | | |
| E | M | 0 | 0 | 4 | 0 | 5 | R | 1 | 2 | 3 | 4 | 0 | 6 | R | 2 | 3 | 4 | 5 | 1 | 5 | R | 3 | 0 | 5 | 5 |
| 45H | 4DH | 30H | 30H | 34H | 30H | 35H | 52H | 31H | 32H | 33H | 34H | 30H | 36H | 52H | 32H | 33H | 34H | 35H | 31H | 35H | 52H | 33H | 30H | 35H | 35H |
| Device No. | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | 7 | R | 8 | 0 | 0 | 0 | 0 | 31H | 37H | 52H | 38H | 30H | 30H | 30H | 30H | 30H | 30H | 30H | 31H | 35H | 52H | 33H | 30H | 35H | 35H |

Monitor (command: ME)

Monitor the registered extended file registers.

■Message format

- Request data

| | | |
|----------|----------|--------------|
| M 4DH | E 45H | Message wait |
|----------|----------|--------------|

- Response data

| |
|---|
| Monitoring result (For the number of device points) |
|---|

The value of read device is stored in word units. ([Page 70 Read data, write data](#))

■Data specified by request data

- Command

ACPU common

| | |
|----------|----------|
| M 4DH | E 45H |
|----------|----------|

- Message wait: Specify the delayed time of the response transmission. ([Page 341 Message wait](#))

■Communication example

Monitor data of 4 points (4 word) specified with monitor data registration under the following conditions.

- Message wait: 0 ms
- Registered devices for monitoring: R1234 of block number 5, R2345 of block number 6, R3055 of block number 15, and 4-point of R8000 of block number 17.

(Request data)

| | |
|--------------|----------|
| Message wait | |
| M 4DH | E 45H |

(Response data)

| Monitoring result (For the number of device points) | | | |
|---|---|--|---|
| 3 5 0 1 (Decimal : 13569) (R1234 in No.5) | 4 F 5 B (Decimal : 20315) (R2345 in No.6) | 0 1 5 0 (Decimal : 366) (R3055 in No.15) | 1 C 2 D (Decimal : 366) (R800 in No.17) |
| 33H, 35H, 30H, 31H | 34H, 46H, 35H, 42H | 30H, 31H, 35H, 30H | 31H, 43H, 32H, 44H |

Direct read (command: NR)

Read extended file register in 1-point (1 word) units by specifying the consecutive device number of extended file register.

Message format

The following shows the message format of the request data and response data of the command.

■Request data

| | | | | |
|----------|----------|--------------|-----------------|--|
| N 4EH | R 52H | Message wait | Head device No. | Number of device points (Number of words) |
|----------|----------|--------------|-----------------|--|

■Response data

The value of read device is stored in word units. (☞ Page 70 Read data, write data)

Data specified by request data

■Command

ACPU common

| | |
|----------|----------|
| N 4EH | R 52H |
|----------|----------|

■Message wait

Specify the delayed time of the response transmission. (☞ Page 341 Message wait)

■Head device No.

Specify the device number of head device with 7 digits. (☞ Page 368 Device number)

■Number of device points

Convert the numerical value to 2-digit ASCII code (hexadecimal), and send it from the upper digits.

Specify the number of device points within the following range:

- $1 \leq$ Number of device points ≤ 64
- Head device No. + Number of device points - 1 \leq Maximum device No.

Communication example

Read data of 2 points (R8190 and R8191) of block number 2 under the following conditions.

- Message wait: 0 ms
- Block number: 2
- Head device: R8190
- Number of device points: 2 points (2 words)

(Request data)

| | | Message wait | | Head device No. | | Number of device points (Number of words) | | | | | |
|----------|----------|--------------|----------|-----------------|----------|--|----------|----------|----------|----------|----------|
| N 4EH | R 52H | 0 30H | 0 30H | 0 31H | 1 31H | 6 36H | 3 33H | 8 38H | 2 32H | 0 30H | 2 32H |

(Response data)

| Data read | | | |
|-----------------------------|-----|------------------------------|-----|
| 1 2 3 4 (Decimal : 4660) | | 7 A B C (Decimal : 31420) | |
| 31H | 32H | 33H | 34H |
| 37H | 41H | 42H | 43H |
| (R8190 in No.2) | | (R8191 in No.2) | |

Direct write (command: NW)

Write extended file register in 1-point (1 word) units by specifying the consecutive device number of extended file register.

Message format

The following shows the message format of the request data and response data of the command.

■Request data

| | | | | | |
|----------|----------|--------------|-----------------|-------------------------|--|
| N 4EH | W 57H | Message wait | Head device No. | Number of device points | Write data for the number of device points |
|----------|----------|--------------|-----------------|-------------------------|--|

■Response data

There is no response data for this command.

Data specified by request data

■Command

ACPU common

| | |
|----------|----------|
| N 4EH | W 57H |
|----------|----------|

■Message wait

Specify the delayed time of the response transmission. ( Page 341 Message wait)

■Head device No.

Specify the device number of head device with 7 digits. ( Page 368 Device number)

■Number of device points

Convert the numerical value to 2-digit ASCII code (hexadecimal), and send it from the upper digits.

Specify the number of device points within the following range:

- $1 \leq$ Number of device points ≤ 64
- Head device No. + Number of device points - 1 \leq Maximum device No.

■Write data for the number of device points

Store 4-digit data per one device point.

Communication example

Write data of 3 points under the following conditions.

- Message wait: 0 ms
- Write data: Write R8190 and R8191 of block number 12 and R0 of block number 13.
(When the extended file register of block number 1 to 8 and 10 to 13 exist.)
- Number of device points: 3 points (3 words)

(Request data)

| Message wait | | Number of device points (Number of words) | | | | | | | | | | | | | | | | | | | | | | |
|--------------|----------|---|----------|----------|----------|----------|--|----------|----------|----------|----------|----------|----------|----------|-----------------------------|------------------|---------------|----------|----------|----------|----------|----------|----------|--|
| | | Head device No. | | | | | Write data for the number of device points | | | | | | | | | | | | | | | | | |
| N 4EH | W 57H | 0 30H | 0 30H | 0 39H | 9 30H | 0 31H | 1 31H | 1 30H | 0 30H | 3 33H | 0 30H | 1 31H | 2 32H | 3 33H | (Decimal : 291) A 41H | 1 42H | 2 43H | 3 41H | 7 37H | 3 33H | 3 32H | 2 32H | 2 32H | (Decimal : 21753) (Decimal : 13090) |
| | | | | | | | | | | | | | | | (R8190 in No.12) | (R8191 in No.13) | (R0 in No.13) | | | | | | | |

17.6 Read and write Buffer Memory of Special Function Module

The following examples the commands that performs data read/write to the buffer memory of MELSEC-A series special function modules.

Point

This command accesses in byte units regardless of the word/byte specification.

Data to be specified in command

This section explains the contents and specification methods for data items which are set in each command related to the access to the special function module buffer memory.

For details of the specification method of address and module number for commands (TR, TW), refer to the following manual.
 Computer Link Module (Com. link func./Print. func.) User's Manual

Start address

Specify the start address of the buffer memory to be read/written.

Convert the numerical value to 5-digit ASCII code (hexadecimal), and send it from the upper digits.

Calculation method

Calculate the start address as follows:

Start address = (Buffer memory address $\times 2$) + the arbitrary additional value of a module

For the additional values (buffer memory start address) for each module, refer to the following section.

 Page 379 Accessible modules

Ex.

When specifying buffer memory address 1H of AD61

$(1H \times 2) + 80H = 82H$

Byte length

Specify the byte length of the special function module buffer memory data to be read/written.

Convert the numerical value to 2-digit ASCII code (hexadecimal), and send it from the upper digits.

Read data, write data

The read buffer memory value is stored for reading, and the data to be written is stored for writing.

This function reads/writes data in byte unit.

 Page 156 Read data, write data

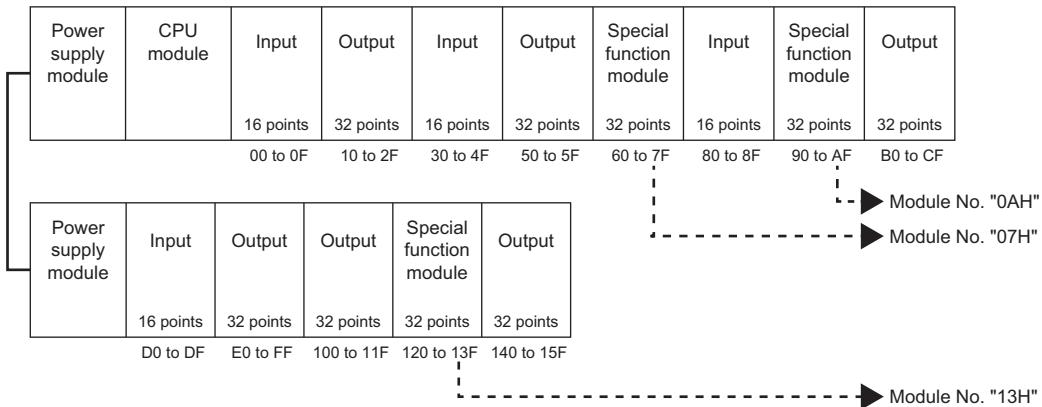
Special function module No.

Specify the last input/output signal (I/O address) of the special function module. (Specify the upper 2-digit in 3-digit representation.)

Convert the numerical value to 2-digit ASCII code (hexadecimal), and send it from the upper digits.

■Special function module No. which occupies 1 slot

Special function module No. is the upper 2 digits of the input/output signal (I/O address) of the last number represented with 3-digits on the slot where the module is mounted.



■Special function module No. which occupies 2 slots

For a special function module which occupies two slots, the number of occupied points for each slot is fixed for each module.

Special function module No. specified for the operation procedure is the upper 2 digits of the input/output signal (I/O address) of the last number represented with 3-digits of the slot assigned as a special function module on the slot.

For details of the assigned slot for each module, refer to the manual of the modules.

Ex.

A module that assign the first part of the slots as empty slots (AD72, A84AD, etc.)

| | |
|--------------|-------------------------|
| (Empty slot) | Special function module |
| 16 points | 32 points |
| 00 to 0F | 10 to 2F |

----- Module No. "02H"

Ex.

A module that assign the last part of the slots as empty slots (A61LS, etc.)

| | |
|-------------------------|--------------|
| Special function module | (Empty slot) |
| 32 points | 16 points |
| 00 to 0F | 20 to 2F |

----- Module No. "01H"

Ex.

A module in which both assignment of special function module and I/O assignment exist (A81CPU)

| | |
|-------------------------|--------------|
| Special function module | Input module |
| 64 points | 64 points |
| 00 to 3F | 40 to 7F |

----- Module No. "03H"

Accessible modules

The accessible special function modules are as follows.

| Module | Type | Additional values when calculating start address (Buffer memory start address) | Module number when module is mounted on slot 0 |
|--------------------------------------|---|---|--|
| High-speed counter module | AD61 (S1) | 80H | 01H |
| | A1SD61, A1SD62 (E/D) | 10H | 01H |
| Analog-digital converter module | A616AD | 10H | 01H |
| | A68AD(S2), A68ADN | 80H | 01H |
| | A84AD | 10H | 02H |
| | A1S64AD | 10H | 01H |
| Digital-analog converter module | A616DAI, A616DAV A62DA (S1) A68DAV/DAI A1S62DA | 10H | 01H |
| Temperature-digital converter module | A616TD A68RD3/4 A1S62RD3/4 | 10H | 01H |
| PID control module | A81CPU | 200H | 03H |
| Position detection module | A61LS | 80H | 01H |
| | A62LS (S5) | 80H | 02H |
| MELSECNET/MINI master module | AJ71PT32 (S3), AJ71T32-S3 A1SJ71PT32-S3 | 20H | 01H |
| CC-Link system master/local module | AJ61BT11 | 2000H | 01H |
| | A1SJ61BT11 | 2000H | 01H |
| Multidrop link module | AJ71C22 (S1) | 1000H | 01H |
| Computer link module | AJ71C24 (S3/S6/S8) | 1000H | 01H |
| | AJ71UC24 A1SJ71 (U) C24-R2 A1SJ71 (U) C24-PRF A1SJ71 (U) C24-R4 | 400H | 01H |
| Intelligent communication module | AD51H (S3), AD51H (S3) | 800H | 02H |
| Terminal interface module | AJ71C21 (S1) | 400H | 01H |
| B/NET interface module | AJ71B62 | 20H | 01H |
| SUMINET interface module | AJ71P41 | 400H | 01H |
| Ethernet interface module | AJ71E71 (S3) | 400H | 01H |
| | A1SJ71E71 (S3) | 4000H | 01H |
| External fault diagnosis module | AD51FD (S3) | 280H | 02H |
| Graphic controller module | AD57G (S3) | 280H | 02H |
| Vision sensor module | AS25VS, AS50VS | 100H | 02H |
| | AS50VS-GN | 80H | 02H |
| Memory card interface module | AD59 (S1) | 1800H | 01H |
| Positioning module | AD70 (D) (S2) | 80H | 01H |
| | AD71 (S1/S2/S7) | 200H | 01H |
| | AD72 A1SD71-S2/S7 | 200H | 02H |
| | AD75P1/P2/P3 (S3), AD75M1/M2/M3 A1SD75P1/P2/P3 (S3), A1SD75M1/M2/M3 | 800H | 01H |
| Positioning module for 1 axis | A1SD70 | 80H | 01H |
| Analog input/output module | A1S63ADA | 10H | 01H |
| Temperature control module | A1S64TCTT (BW)-S1 A1S64TCRT (BW)-S1 A1S64TCTRT (BW) A1S62TCTT (BW)-S2 A1S62TCRT (BW)-S2 | 20H | 01H |

Batch read (command: TR)

Read the buffer memory of a special function module.

Message format

The following shows the message format of the request data and response data of the command.

■Request data

| | | | | | |
|----------|----------|--------------|---------------|-------------|-----------------------------|
| T 54H | R 52H | Message wait | Start address | Byte length | Special function module No. |
|----------|----------|--------------|---------------|-------------|-----------------------------|

■Response data

The value read from the buffer memory is stored.

2-digit ASCII code data is stored per 1 byte of buffer memory data.

Data specified by request data

■Command

ACPU common

| | |
|----------|----------|
| T 54H | R 52H |
|----------|----------|

■Message wait

Specify the delayed time of the response transmission. (☞ Page 341 Message wait)

■Start address

Specify the start address of the buffer memory to be read in five digits. (☞ Page 380 Start address)

■Byte length

Convert the numerical value to 2-digit ASCII code (hexadecimal), and send it from the upper digits.

Specify the byte length within the following ranges:

- $1 \leq \text{Byte length} \leq 128$

■Special function module No.

Specify with 2 digits. (☞ Page 380 Special function module No.)



The content of one data may cross 2 or 3 bytes depending on the special function module. For the specifications of byte length, refer to the manual of each module.

Communication example

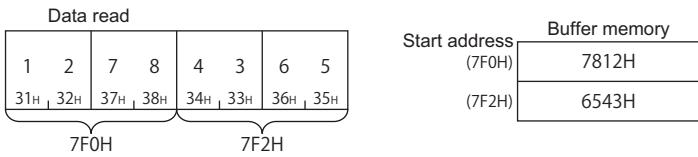
Read data of 4 bytes under the following conditions.

- Message wait: 0 ms
- Start address: 7FOH
- Byte length: 4 bytes
- Special function module No.: 13 (input/output signals are 120H to 13FH)

(Request data)

| | | Message wait | Start address | Byte length | Special function module No. |
|-----|-----|--------------|--------------------|-------------|-----------------------------|
| T | R | 0 | 0 0 7 F 0 | 0 4 | 1 3 |
| 54H | 52H | 30H | 30H, 30H, 37H, 46H | 30H | 31H, 33H |

(Response data)



Batch write (command: TW)

Write data to the buffer memory of a special function module.

Message format

The following shows the message format of the request data and response data of the command.

■Request data

| | | | | | | |
|----------|----------|--------------|---------------|-------------|-----------------------------|------------|
| T 54H | W 57H | Message wait | Start address | Byte length | Special function module No. | Write data |
|----------|----------|--------------|---------------|-------------|-----------------------------|------------|

■Response data

There is no response data for this command.

Data specified by request data

■Command

ACPU common

| | |
|----------|----------|
| T 54H | W 57H |
|----------|----------|

■Message wait

Specify the delayed time of the response transmission. (☞ Page 341 Message wait)

■Start address

Specify the start address of the buffer memory to be written in five digits. (☞ Page 380 Start address)

■Byte length

Convert the numerical value to 2-digit ASCII code (hexadecimal), and send it from the upper digits.

Specify the byte length within the following ranges:

- $1 \leq \text{Byte length} \leq 128$

■Special function module No.

Specify in two characters. (☞ Page 380 Special function module No.)

■Write data

Store the data written to buffer memory.



The content of one data may cross 2 or 3 bytes depending on the special function module. For the specifications of byte length, refer to the manual of each module.

Communication example

Write data of 4 bytes under the following conditions.

- Message wait: 0 ms
- Start address: 27FAH
- Byte length: 4 bytes
- Special function module No.: 13 (input/output signals are 120H to 13FH)

(Request data)

| Message wait | | Special function module No. | | | | | | | | | | | | | | | | | |
|--------------|----------|-----------------------------|----------|-------------|----------|------------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|
| | | Start address | | Byte length | | Write data | | | | | | | | | | | | | |
| T 54H | W 57H | 0 30H | 0 30H | 2 32H | 7 37H | F 46H | A 41H | 0 30H | 4 34H | 1 31H | 3 33H | 0 30H | 1 31H | C 43H | D 44H | A 41H | B 42H | E 45H | F 46H |
| 27FAH | | | | | | | | | | 27FCH | | | | | | | | | |

| Start address (27FAH) | Buffer memory |
|--------------------------|---------------|
| | CD01H |
| | EFABH |

17.7 Loopback Test

A loopback test checks whether the communication function between an external device and C24 operates normally.

Loopback test (Command: TT)

Return the characters received from an external device to the external device unchanged.

Message format

The following shows the message format of the request data and response data of the command.

■Request data

| | | | | |
|----------|----------|--------------|------------------|---------------|
| T 54H | T 54H | Message wait | Character length | Loopback data |
|----------|----------|--------------|------------------|---------------|

■Response data

| | |
|------------------|---------------|
| Character length | Loopback data |
|------------------|---------------|

Data specified by request data

■Command

ACPU common

| | |
|----------|----------|
| T 54H | T 54H |
|----------|----------|

■Message wait

Specify the delayed time of the response transmission. ( Page 341 Message wait)

■Character length

Convert the numerical value to 2-digit ASCII code (hexadecimal), and send it from the upper digits.

Specify the character length within the following range:

- $1 \leq \text{Character length} \leq 254$

■Loopback data

Store the loopback data for character length.

Data stored by response data

■Character length

The same data as request data is stored.

■Loopback data

The same data as request data is stored.



Specify 'FF' for PC No.

Communication example

Return 5-digit data received from an external device to the external device unchanged under the following conditions.

- Message wait: 0 ms
- Character length: 5 characters
- Loopback data: 'ABCDE'

(Request data)

| Message wait | | Character length | | Loopback data | | | | |
|--------------|-----|------------------|-----|---------------|-----|-----|-----|-----|
| T | T | 0 | 0 | 5 | A | B | C | D |
| 54H | 54H | 30H | 30H | 35H | 41H | 42H | 43H | 44H |

(Response data)

| Character length | | Loopback data | | | | |
|------------------|-----|---------------------|--|--|--|--|
| 0 | 5 | A B C D E | | | | |
| 30H | 35H | 41H 42H 43H 44H 45H | | | | |

18 COMMUNICATING USING 1E FRAMES

This chapter explains the functions when accessing using 1E frame and their message format.

1E frame is compatible with the communication function supported by MELSEC-A series Ethernet interface modules.

Only the commands for 1E frame explained in this chapter can be used for 1E frame.

18

18.1 Message Format

This section explains the message format when communicating data using 1E frame.

Message format

■Request message

| | | | | |
|--------|-----------|--------|-----------------------|--------------|
| Header | Subheader | PC No. | ACPU monitoring timer | Request data |
|--------|-----------|--------|-----------------------|--------------|

■Response message (Normal completion: Response data)

| | | | |
|--------|-----------|----------|---------------|
| Header | Subheader | End code | Response data |
|--------|-----------|----------|---------------|

■Response message (Normal completion: No response data)

| | | |
|--------|-----------|----------|
| Header | Subheader | End code |
|--------|-----------|----------|

■Response message (Abnormal completion)

| | | | |
|--------|-----------|----------|---------------|
| Header | Subheader | End code | Abnormal code |
|--------|-----------|----------|---------------|

Setting data

Set the following items.

| Item | Description | Reference |
|---------------------------|---|---|
| Header | A header of Ethernet. Normally, it is added automatically. | Page 386 Header |
| Subheader | Set the command type. | Page 386 Subheader |
| PC No. | Specify the network module station No. of an access target. | Page 387 PC No. |
| ACPU monitoring timer | Set the wait time up to the completion of reading and writing processing. | Page 388 ACPU monitoring timer |
| Request data | Set the commands that indicates request content. | Page 391 Read and Write Device Memory Page 412 Read and Write Extended File Register Page 426 Read and Write Buffer Memory of Special Function Module |
| Response data | For the response data, store the read data for the command at normal completion. Refer to "Response data" rows of each command. | |
| End code Abnormal code | The command processing result is stored. | Page 389 End code, Abnormal code |

18.2 Detail of Setting Data

This section explains how to specify the common data items and their content in each message.

Header

A header for TCP/IP and UDP/IP. A header of a request message is added on the external device side and sent. Normally, it is added automatically by an external device. A header for a response message is set automatically by E71.

Subheader

Set the command type. ( Page 390 Commands and Function List for 1E Frame)

The setting values for each command are as follows.

| Function | Subheader | | Reference |
|---|-----------------|------------------|--|
| | Request message | Response message | |
| Read and write device memory | 00H | 80H | Page 397 Batch read in bit units (command: 00) |
| | 01H | 81H | Page 399 Batch read in word units (command: 01) |
| | 02H | 82H | Page 401 Batch write in bit units (command: 02) |
| | 03H | 83H | Page 403 Batch write in word units (command: 03) |
| | 04H | 84H | Page 404 Test in bit units (random write) (command: 04) |
| | 05H | 85H | Page 406 Test in word units (random write) (command: 05) |
| | 06H | 86H | Page 409 Register monitor data(command: 06, 07) |
| | 07H | 87H | Page 409 Register monitor data(command: 06, 07) |
| | 08H | 88H | Page 410 Monitor in bit units (command: 08) |
| | 09H | 89H | Page 411 Monitor in word units (command: 09) |
| Read and write extended file register | 17H | 97H | Page 413 Batch read (command: 17) |
| | 18H | 98H | Page 415 Batch write (command: 18) |
| | 19H | 99H | Page 417 Test (random write) (command: 19) |
| | 1AH | 9AH | Page 420 Register monitor data (command: 1A) |
| | 1BH | 9BH | Page 421 Monitoring (command: 1B) |
| | 3BH | BBH | Page 422 Direct read (command: 3B) |
| | 3CH | BCH | Page 424 Direct write (command: 3C) |
| Read and write buffer memory of special function module | 0EH | 8EH | Page 428 Batch read (command: 0E) |
| | 0FH | 8FH | Page 430 Batch write (command: 0F) |

Setting method

■Data communication in ASCII code

Convert the numerical value to 2-digit ASCII code (hexadecimal), and send it from the upper digits.

■Data communication in binary code

Send 1-byte numerical value.

Ex.

Device memory batch read (word units)

Request message

| ASCII code | Binary code |
|----------------|-------------|
| 0 0 30H 30H | 00H |

Response message

| ASCII code | Binary code |
|----------------|-------------|
| 8 0 38H 30H | 80H |

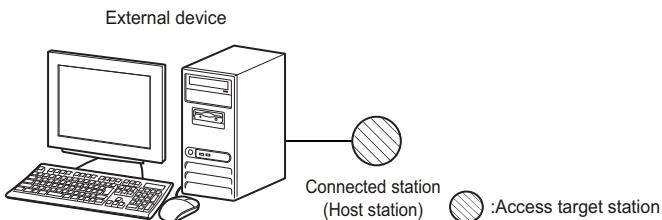
PC No.

Specify the station No. of the access target.

Accessing the connected station (host station)

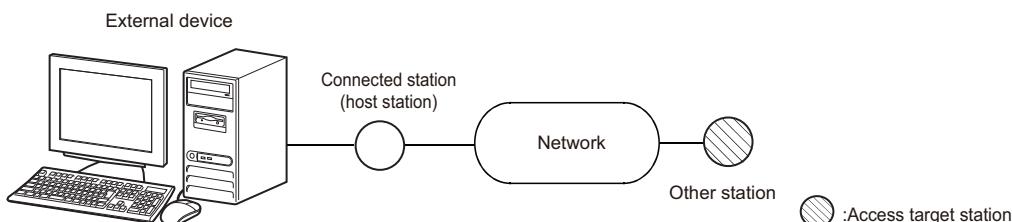
Specify 'FF'.

18



Accessing other stations via network

Specify the network module station No.01H to 40H (1 to 64) of the access target.



Setting method

■Data communication in ASCII code

Convert the numerical value to 2-digit ASCII code (hexadecimal), and send it from the upper digits.

■Data communication in binary code

Send 1-byte numerical value.

Ex.

Accessing the connected station (host station)

| ASCII code | Binary code |
|----------------|-------------|
| F F 46H 46H | FFH |

When accessing other station of network station No. '3'.

| ASCII code | Binary code |
|----------------|-------------|
| 0 3 30H 33H | 03H |



The station No. of the network module can be checked by using the following parameters of Engineering tool.

- GX Developer and GX Works2: "Network Parameter"
- GX Works3: "Module Parameter"

The network module station No. is set in decimal. However, the PC No. is set in hexadecimal.

When specifying the network of the access target is required, set "Valid Module During Other Station Access" with an Engineering tool.

ACPU monitoring timer

Set the wait time up to the completion of reading and writing processing.

- 0000H (0): Wait infinitely (Waits until a processing is completed.)
- 0001H to FFFFH (1 to 65535): Waiting time (unit: 250 ms)

To perform normal data communication, using the timer within the setting range in the table below is recommended depending on the communication destination.

| Access target | The recommended value for monitoring timer |
|----------------------------------|--|
| Connected station (host station) | 1H to 28H (0.25 s to 10 s) |
| Other station | 2H to F0H (0.5 s to 60 s) |



When accessing QnACPU or ACPU for the first time, the wait time for CPU monitoring timer is required before receiving a response message because QnACPU identifies the CPU type. Be sure to set a value within the range of recommended range.

Setting method

■Data communication in ASCII code

Convert the numerical value to 4-digit ASCII code (hexadecimal) and send it from upper digits.

■Data communication in binary code

Send 2-byte numerical values from lower byte (L: bits 0 to 7).



When specifying '10H' for the monitoring timer

| ASCII code | Binary code |
|----------------------------------|-------------|
| 0 0 1 0 30H , 30H , 31H , 30H | 10H , 00H |

End code, Abnormal code

The command processing result is stored.

End code

At normal completion, '0' is stored.

At abnormal completion, an error code of the access target is stored.

Error code indicates the content of occurred error.

If more than one error occurs at the same time, the error code detected first is returned.

For the content of error code and its corrective action, refer to the user's manual of the module used.

QCPU User's Manual (Hardware Design, Maintenance and Inspection)

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

Q Corresponding Ethernet Interface Module User's Manual (Basic)

MELSEC-L Ethernet Interface Module User's Manual (Basic)

Abnormal code

When an end code is '5BH', the details of abnormal content are displayed.

Setting method

■Data communication in ASCII code

Convert the numerical value to 2-digit ASCII code (hexadecimal), and send it from the upper digits.

■Data communication in binary code

Use a 1-byte value.

Ex.

Normal completion

| ASCII code | Binary code | | | | | |
|---|-------------|---|-----|-----|---|-----|
| <table border="1"> <tr> <td>0</td> <td>0</td> </tr> <tr> <td>30H</td> <td>30H</td> </tr> </table> | 0 | 0 | 30H | 30H | <table border="1"> <tr> <td>00H</td> </tr> </table> | 00H |
| 0 | 0 | | | | | |
| 30H | 30H | | | | | |
| 00H | | | | | | |

When Error code '10H' is returned

| ASCII code | Binary code | | | | | |
|---|-------------|---|-----|-----|---|-----|
| <table border="1"> <tr> <td>1</td> <td>0</td> </tr> <tr> <td>31H</td> <td>30H</td> </tr> </table> | 1 | 0 | 31H | 30H | <table border="1"> <tr> <td>10H</td> </tr> </table> | 10H |
| 1 | 0 | | | | | |
| 31H | 30H | | | | | |
| 10H | | | | | | |

When error code '5BH' and an abnormal code '10H' (PC No. error) are returned,

| ASCII code | Binary code | | | | | | | | | | |
|---|-------------|-----|-----|---|-----|-----|-----|-----|--|-----|-----|
| <table border="1"> <tr> <td>5</td> <td>B</td> <td>1</td> <td>0</td> </tr> <tr> <td>35H</td> <td>42H</td> <td>31H</td> <td>30H</td> </tr> </table> | 5 | B | 1 | 0 | 35H | 42H | 31H | 30H | <table border="1"> <tr> <td>5BH</td> <td>10H</td> </tr> </table> | 5BH | 10H |
| 5 | B | 1 | 0 | | | | | | | | |
| 35H | 42H | 31H | 30H | | | | | | | | |
| 5BH | 10H | | | | | | | | | | |

18.3 Commands and Function List for 1E Frame

Use the following commands for data communication using 1E frame.

| Function | | Command | Description |
|-----------------------------|-------------------------------------|---------------------|--|
| Device memory ^{*1} | Batch read | Bit units | 00H Reads bit devices (X, Y, M, etc.) in 1-point units. |
| | | Word units | 01H Reads bit devices (X, Y, M, etc.) in 16-point units. Reads word devices (D, T, C, etc.) in 1-point units. |
| | Batch write | Bit units | 02H Writes bit devices (X, Y, M, etc.) in 1-point units. |
| | | Word units | 03H Writes bit devices (X, Y, M, etc.) in 16-point units. Writes word devices (D, T, C, etc.) in 1-point units. |
| | | Test (random write) | 04H Specify the devices and device numbers of bit devices (X, Y, M, etc.) in 1-point units randomly, and set/reset them. 05H Specify the devices and device numbers of bit devices (X, Y, M, etc.) in 16-point units randomly, and set/reset them. Specify the devices and device numbers of word devices (D, T, C, etc.) in 1-point units randomly, and write them. |
| | Register monitor data ^{*2} | Bit units | 06H Registers bit devices (X, Y, M, etc.) to be monitored in 1-point units |
| | | Word units | 07H Registers bit devices (X, Y, M, etc.) to be monitored in 16-point units. Registers word devices (D, T, C, etc.) to be monitored in 1-point units. |
| | | Monitoring | 08H Monitors the devices registered by monitor data registration. 09H |
| Extended file register | Batch read | 17H | Reads extended file register (R) in 1-point units. |
| | Batch write | 18H | Writes extended file register (R) in 1-point units. |
| | Test (random write) | 19H | Specify the block numbers and device numbers in 1-point units and write them to the extended file register (R) randomly. |
| | Register monitor data ^{*2} | 1AH | Registers extended file register (R) in 1-point units. |
| | Monitoring | 1BH | Monitors extended file register (R) with monitor data registered. |
| | Direct read | 3BH | Reads extended file register in 1-point units with direct designation. |
| | Direct write | 3CH | Writes extended file register in 1-point units by direct specification. |
| Special function module | Batch read | 0EH | Reads the content in the buffer memory of a special function module. |
| | Batch write | 0FH | Writes data to the buffer memory of a special function module. |

*1 Use the dedicated commands for extended registers when performing extended file registers read/write.

*2 The devices that can be registered to E71 is for 1 command out of the three types of commands (06H, 07H, 1AH).

The specified device recently used by any of the above commands is registered to E71.

18.4 Read and Write Device Memory

This section explains the specification content and examples of the request data and the response data when reading and writing device memory.

For the message formats other than request data and response data, refer to the following sections.

☞ Page 385 Message Format, Page 385 Detail of Setting Data



To read and write the extended file register, use the commands dedicated to the extended file register.

☞ Page 412 Read and Write Extended File Register

Considerations

Considerations when reading/writing data to module other than ACPU module

■ Accessible devices

Only the devices with the same names that exist in ACPU can be accessed within the device range of AnACPU.

The following devices cannot be accessed from the external devices

- Added devices
- Latch relay (L) and step relay (S)
- File register (R)

■ Special relays and special registers

Special relays and special registers can be accessed within the following range.

- Access SM1000 to SM1255 by specifying M9000 to M9255.
- Access SD1000 to SD1255 by specifying D9000 to D9255.

■ Universal model QCPU

Use the Universal model QCPU with a serial number whose first five digits are '10102' or later.

When the module with the serial number whose first five digits are '10101' or earlier, access using 3E frame or 4E frame.

When accessing the built-in Ethernet port of the CPU module, refer to the following manual.

☞ QnUCPU User's Manual (Communication via Built-in Ethernet Port)

Data to be specified in command

Device codes and device numbers

The settings of each device when reading/writing device memory can be performed using device code and device number as shown in the following figure.

Specify the device to be accessed by a device code and a device number.

The order of data differs depending on the ASCII code or binary code.

| ASCII code | Binary code |
|--|--|
| Device code Device number (4 digits) (8 digits) | Device number Device code (4 bytes) (2 bytes) |

■ Data communication in ASCII code

- Device code: Convert the numerical value to 4-digit ASCII code (hexadecimal), and send it from upper digits.
- Device number: Convert the numerical value to 8-digit (hexadecimal) ASCII code and send it from the upper digits.

■ Data communication in binary code

- Device number: Send 4-byte numerical values from the lower byte (L: bits 0 to 7).
- Device code: Send 2-byte numerical values from lower byte (L: bits 0 to 7).

Ex.

Data register (D) 1234 (device number is decimal)

Convert a device number to hexadecimal. '1234' (decimal) → '4D2' (hexadecimal)

| ASCII code | Binary code | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|-------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|---|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|--|-----|-----|-----|-----|-----|-----|
| <table border="1"> <tr> <td>4</td><td>4</td><td>2</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>4</td><td>D</td><td>2</td></tr> <tr> <td>34H</td><td>34H</td><td>32H</td><td>30H</td><td>30H</td><td>30H</td><td>30H</td><td>30H</td><td>34H</td><td>44H</td><td>32H</td></tr> </table> | 4 | 4 | 2 | 0 | 0 | 0 | 0 | 0 | 4 | D | 2 | 34H | 34H | 32H | 30H | 30H | 30H | 30H | 30H | 34H | 44H | 32H | <table border="1"> <tr> <td>D2H</td><td>04H</td><td>00H</td><td>00H</td><td>20H</td><td>44H</td></tr> </table> | D2H | 04H | 00H | 00H | 20H | 44H |
| 4 | 4 | 2 | 0 | 0 | 0 | 0 | 0 | 4 | D | 2 | | | | | | | | | | | | | | | | | | | |
| 34H | 34H | 32H | 30H | 30H | 30H | 30H | 30H | 34H | 44H | 32H | | | | | | | | | | | | | | | | | | | |
| D2H | 04H | 00H | 00H | 20H | 44H | | | | | | | | | | | | | | | | | | | | | | | | |

For the values of each device code, refer to the following section.

□: Space

| Device name | Symbol | Type | Representation | Device code | |
|--|---------------|------|----------------|-------------|-------|
| Input | X | Bit | Hexadecimal | X□ | 5820H |
| Output | Y | Bit | Hexadecimal | Y□ | 5920H |
| Internal relay (include in latch relay and step relay) | M/L/S | Bit | Decimal | M□ | 4D20H |
| Annunciator | F | Bit | Decimal | F□ | 4620H |
| Link relay | B | Bit | Hexadecimal | B□ | 4220H |
| Timer | Current value | T | Word | TN | 544EH |
| | Contact | | Bit | TS | 5453H |
| | Coil | | Bit | TC | 5443H |
| Counter | Current value | C | Word | CN | 434EH |
| | Contact | | Bit | CS | 4353H |
| | Coil | | Bit | CC | 4343H |
| Data register | D | Word | Decimal | D□ | 4420H |
| Link register | W | Word | Hexadecimal | W□ | 5720H |
| File register | R | Word | Decimal | R□ | 5220H |

Access the devices within the range that can be used in the access target CPU.

For the accessible device range, refer to the following section.

☞ Page 393 Accessible device range

Point

- For word unit specification, the head device number of bit device must be specified in multiple of 16.
- For special relay M9000 or later, (9000 + multiple of 16) can be specified.

Accessible device range

■List of devices (ACPU other than AnU)

Specify the device number within the range of the access target module.

○: Accessible, —: No device

| Device | Device range | Device number | A1S A1SH A1SJ A1SJH A1 A1N | A2S A2SH A2 A2N A2C A2CJ A0J2H | A2-S1 A2N-S1 | A3 A3N | A2A | A2A-S1 | A3A |
|--|----------------|----------------|---|--|-----------------|-----------|-----|--------|-----|
| Input | X0 to X0FF | 0000H to 00FFH | ○ | ○ | | | ○ | | |
| | X100 to X1FF | 0100H to 01FFH | — | ○ | | | ○ | | |
| | X200 to X3FF | 0200H to 03FFH | — | — | ○ | | — | ○ | |
| | X400 to X7FF | 0400H to 07FFH | — | — | | ○ | — | | ○ |
| Output | Y0 to Y0FF | 0000H to 00FFH | ○ | ○ | | | ○ | | |
| | Y100 to Y1FF | 0100H to 01FFH | — | ○ | | | ○ | | |
| | Y200 to Y3FF | 0200H to 03FFH | — | — | ○ | | — | ○ | |
| | Y400 to Y7FF | 0400H to 07FFH | — | — | | ○ | — | | ○ |
| Internal relay (Including latch relay and step relay) | M0 to M2047 | 0000H to 07FFH | ○ | ○ | | | ○ | | |
| | M2048 to M8191 | 0800H to 1FFFH | — | — | | | ○ | | |
| | M9000 to M9255 | 2328H to 2427H | ○ | ○ | | | ○ | | |
| Annunciator | F0 to F255 | 0000H to 00FFH | ○ | ○ | | | ○ | | |
| | F256 to F2047 | 0100H to 07FFH | — | — | | | ○ | | |
| Link relay | B0 to B3FF | 0000H to 03FFH | ○ | ○ | | | ○ | | |
| | B400 to BFFF | 0400H to 0FFFH | — | — | | | ○ | | |
| Timer | Current value | T0 to T255 | 0000H to 00FFH | ○ | ○ | | ○ | | |
| | | T256 to T2047 | 0100H to 07FFH | — | — | | ○ | | |
| | Contact | T0 to T255 | 0000H to 00FFH | ○ | ○ | | ○ | | |
| | | T256 to T2047 | 0100H to 07FFH | — | — | | ○ | | |
| | Coil | T0 to T255 | 0000H to 00FFH | ○ | ○ | | ○ | | |
| | | T256 to T2047 | 0100H to 07FFH | — | — | | ○ | | |
| Counter | Current value | C0 to C255 | 0000H to 00FFH | ○ | ○ | | ○ | | |
| | | C256 to C1023 | 0100H to 03FFH | — | — | | ○ | | |
| | Contact | C0 to C255 | 0000H to 00FFH | ○ | ○ | | ○ | | |
| | | C256 to C1023 | 0100H to 03FFH | — | — | | ○ | | |
| | Coil | C0 to C255 | 0000H to 00FFH | ○ | ○ | | ○ | | |
| | | C256 to C1023 | 0100H to 03FFH | — | — | | ○ | | |
| Data register | D0 to D1023 | 0000H to 03FFH | ○ | ○ | | | ○ | | |
| | D1024 to D6143 | 0400H to 17FFF | — | — | | | ○ | | |
| | D9000 to D9255 | 2328H to 2427H | ○ | ○ | | | ○ | | |
| Link register | W0 to W3FF | 0000H to 03FFH | ○ | ○ | | | ○ | | |
| | W400 to WFFF | 0400H to 0FFFH | — | — | | | ○ | | |
| File register | R0 to R4095 | 0000H to 0FFFH | — | ○ | | | ○ | | |
| | R4096 to R8191 | 1000H to 1FFFH | — | — | | ○ | ○ | | |

Restriction

- Do not write data to the devices which cannot be written in the range of the special relays (M9000 to M9255) and special registers (D9000 to D9255). For details on the special relays and special registers, refer to manual of ACPU to be accessed.
- For L and S, perform accessing by specifying 'M' (For example, to access L100, specify M100.)

■List of devices (AnUCPU, QnACPU)

Specify the device within the range of AnACPU.

☞ Page 391 Considerations when reading/writing data to module other than ACPU module

○: Accessible, ×: Not accessible, —: No device

| Device | Device range | Device number | A2US A2U | A2US-S1 A2USH-S1 A2U-S1 | A3U A4U | Q2A Q2AS Q2ASH | Q2A-S1 Q2AS-S1 Q2ASH-S1 | Q3A Q4A Q4AR |
|--|--------------------------------------|----------------|----------------|-------------------------------|------------|----------------------|-------------------------------|--------------------|
| Input | X0 to X1FF | 0000H to 01FFH | ○ | | | ○ | | |
| | X200 to X3FF | 0200H to 03FFH | — | ○ | | — | ○ | |
| | X400 to X7FF | 0400H to 07FFH | — | | ○ | — | | ○ |
| Output | Y0 to Y1FF | 0000H to 01FFH | ○ | | | ○ | | |
| | Y200 to Y3FF | 0200H to 03FFH | — | ○ | | — | ○ | |
| | Y400 to Y7FF | 0400H to 07FFH | — | | ○ | — | | ○ |
| Internal relay (Including latch relay and step relay) | M0 to M8191 | 0000H to 1FFFH | ○ | | | ○ | | |
| | M9000 to M9255 (SM1000 to SM1255) | 2328H to 2427H | ○ | | | ○ | | |
| Annunciator | F0 to F2047 | 0000H to 07FFH | ○ | | | ○ | | |
| Link relay | B0 to BFFF | 0000H to 0FFFH | ○ | | | ○ | | |
| Timer | Current value | T0 to T2047 | 0000H to 07FFH | ○ | | ○ | | |
| | Contact | T0 to T2047 | 0000H to 07FFH | ○ | | ○ | | |
| | Coil | T0 to T2047 | 0000H to 07FFH | ○ | | ○ | | |
| Counter | Current value | C0 to C1023 | 0000H to 03FFH | ○ | | ○ | | |
| | Contact | C0 to C1023 | 0000H to 03FFH | ○ | | ○ | | |
| | Coil | C0 to C1023 | 0000H to 03FFH | ○ | | ○ | | |
| Data register | D0 to D6143 | 0000H to 17FFH | ○ | | | ○ | | |
| | D9000 to D9255 (SD1000 to SD1255) | 2328H to 2427H | ○ | | | ○ | | |
| Link register | W0 to WFFF | 0000H to 0FFFH | ○ | | | ○ | | |
| File register | R0 to R8191 | 0000H to 1FFFH | ○ | | | × | | |

Restriction

- Do not write data to the devices which cannot be written in the range of the special relays (M9000 to M9255) and special registers (D9000 to D9255). For details on the special relays and special registers, refer to the programming manual of ACPU.
- For L and S, perform accessing by specifying 'M' (For example, to access L100, specify M100.)

■Device list (QCPU, LCPU, safety CPU)

Specify the device within the range of AnACPU.

☞ Page 391 Considerations when reading/writing data to module other than ACPU module

○: Accessible, ×: Not accessible, —: No device

| Device | Device range | Device number | Basic model QCPU Safety CPU | QCPU other than left LCPU |
|----------------|--------------------------------------|----------------|--------------------------------|------------------------------|
| Input | X0 to X7FF | 0000H to 07FFH | ○ | |
| Output | Y0 to Y7FF | 0000H to 07FFH | ○ | |
| Internal relay | M0 to M8191 | 0000H to 1FFFH | ○ | |
| | M9000 to M9255 (SM1000 to SM1255) | 2328H to 2427H | — | ○ |
| Annunciator | F0 to F1023 | 0000H to 03FFH | ○ | |
| | F1024 to F2047 | 0400H to 07FFH | — | ○ |
| Link relay | B0 to B7FF | 0000H to 07FFH | ○ | |
| | B800 to BFFF | 0800H to 0FFFH | — | ○ |
| Timer | Current value | T0 to T511 | 0000H to 01FFH | ○ |
| | | T512 to T2047 | 0200H to 07FFH | — |
| | Contact | T0 to T511 | 0000H to 01FFH | ○ |
| | | T512 to T2047 | 0200H to 07FFH | — |
| | Coil | T0 to T511 | 0000H to 01FFH | ○ |
| | | T512 to T2047 | 0200H to 07FFH | — |
| Counter | Current value | C0 to C511 | 0000H to 01FFH | ○ |
| | | C512 to C1023 | 0200H to 03FFH | — |
| | Contact | C0 to C511 | 0000H to 01FFH | ○ |
| | | C512 to C1023 | 0200H to 03FFH | — |
| | Coil | C0 to C511 | 0000H to 01FFH | ○ |
| | | C512 to C1023 | 0200H to 03FFH | — |
| Data register | D0 to D6143 | 0000H to 17FFH | ○ | |
| | D9000 to D9255 (SD1000 to SD1255) | 2328H to 2427H | — | ○ |
| Link register | W0 to W7FF | 0000H to 07FFH | ○ | |
| | W800 to WFFF | 0800H to 0FFFH | — | ○ |
| File register | R0 or more | 0000H or more | × | |

Number of device points

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

Specify the number of device points in one command within the device points that can be processed in one communication.

☞ Page 459 Number of Processing per One Communication

Specify '00' for 256 points.

■Data communication in ASCII code

Convert the numerical value to 2-digit ASCII code (hexadecimal), and send it from the upper digits.

Use capitalized code for alphabetical letter.

■Data communication in binary code

Send 1-byte numerical value.

Ex.

5 points, 20 points, 256 points

| Number of device points | ASCII code | Binary code | | | | | |
|-------------------------|--|-------------|---|-----|-----|---|-----|
| 5 points | <table border="1"><tr><td>0</td><td>5</td></tr><tr><td>30H</td><td>35H</td></tr></table> | 0 | 5 | 30H | 35H | <table border="1"><tr><td>05H</td></tr></table> | 05H |
| 0 | 5 | | | | | | |
| 30H | 35H | | | | | | |
| 05H | | | | | | | |
| 20 points | <table border="1"><tr><td>1</td><td>4</td></tr><tr><td>31H</td><td>34H</td></tr></table> | 1 | 4 | 31H | 34H | <table border="1"><tr><td>14H</td></tr></table> | 14H |
| 1 | 4 | | | | | | |
| 31H | 34H | | | | | | |
| 14H | | | | | | | |
| 256 points | <table border="1"><tr><td>0</td><td>0</td></tr><tr><td>30H</td><td>30H</td></tr></table> | 0 | 0 | 30H | 30H | <table border="1"><tr><td>00H</td></tr></table> | 00H |
| 0 | 0 | | | | | | |
| 30H | 30H | | | | | | |
| 00H | | | | | | | |

Read data, write data

The read data is stored for reading, and the data to be written is stored for writing. The data order differs between bit units or word units.

Batch read in bit units (command: 00)

Read bit devices (X, Y, M, etc.) in 1-point units.

Message format

The following shows the message format of the request data and response data of the command.

Specify the command type by subheader. (☞ Page 386 Subheader)

■Request data

| | | |
|-------------|-------------------------|-------------|
| Head device | Number of device points | Fixed value |
|-------------|-------------------------|-------------|

■Response data

The data for the number of device points are stored.

(☞ Page 396 Read data, write data)



For ASCII code, when the number of device points are specified in an odd number, one byte of dummy data (30H) will be added to the response data. For example, if a data for three points are read, data for four points is returned. The last byte will be a dummy data.

Data specified by request data

■Head device

Specify the head device number of bit devices to be read. (☞ Page 391 Device codes and device numbers)

■Number of device points

Specify the number of the bit device points to be read.

Specify '00H' when the number of device points is to 256 points. (☞ Page 396 Number of device points)

■Fixed value

Fixed to '0'.

| ASCII code | Binary code | | | | | |
|---|-------------|---|-----|-----|---|-----|
| <table border="1"> <tr> <td>0</td> <td>0</td> </tr> <tr> <td>30H</td> <td>30H</td> </tr> </table> | 0 | 0 | 30H | 30H | <table border="1"> <tr> <td>00H</td> </tr> </table> | 00H |
| 0 | 0 | | | | | |
| 30H | 30H | | | | | |
| 00H | | | | | | |

Communication example

Read the bit devices in the CPU module with E71 mounted under the following conditions.

- Head device: M100
- Number of device points: 12 points

■Data communication in ASCII code

(Request data)

| Head device | Number of device points |
|--|----------------------------------|
| 4 D 2 0 0 0 0 0 0 0 6 4 34H , 44H , 32H , 30H , 30H , 30H , 30H , 30H , 30H , 36H , 34H | 0 C 0 0 30H , 43H , 30H , 30H |

(Response data)

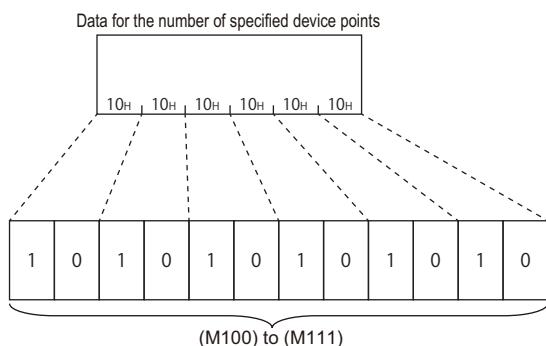
| Data for the number of specified device points | | | | | | | | | | | | |
|--|----|--|--------|--|--|--|--|--|--|--|--|--|
| 1 0 1 0 1 0 1 0 1 0 1 0 31H , 30H , 31H , 30H | to | | (M111) | | | | | | | | | |

■Data communication in binary code

(Request data)

| Head device | Number of device points |
|-----------------------------------|-------------------------|
| 64H , 00H , 00H , 00H , 20H , 4DH | 0CH , 00H |

(Response data)



Batch read in word units (command: 01)

Reads bit devices (X, Y, M, etc.) in 16-point units.

Reads word devices (D, T, C, etc.) in 1-point units

Message format

The following shows the message format of the request data and response data of the command.

■Request data

| | | |
|-------------|-------------------------|-------------|
| Head device | Number of device points | Fixed value |
|-------------|-------------------------|-------------|

■Response data

The data for the number of device points are stored.

The order of data differs depending on the ASCII code or binary code. (☞ Page 396 Read data, write data)

Data specified by request data

■Head device

Specify the head device of the device to be read. (☞ Page 391 Device codes and device numbers)

■Number of device points

Specify the device number to be read.

Specify '00H' when the number of device points is to 256 points. (☞ Page 396 Number of device points)

When specifying bit devices, set the head device No. in multiples of 16 (0, 16, ... in decimal notation).

■Fixed value

Fixed to '0'.

| ASCII code | Binary code | | | | | |
|---|-------------|---|-----|-----|---|-----|
| <table border="1"> <tr> <td>0</td> <td>0</td> </tr> <tr> <td>30H</td> <td>30H</td> </tr> </table> | 0 | 0 | 30H | 30H | <table border="1"> <tr> <td>00H</td> </tr> </table> | 00H |
| 0 | 0 | | | | | |
| 30H | 30H | | | | | |
| 00H | | | | | | |

Communication example

Read the bit devices in the CPU module with E71 mounted under the following conditions.

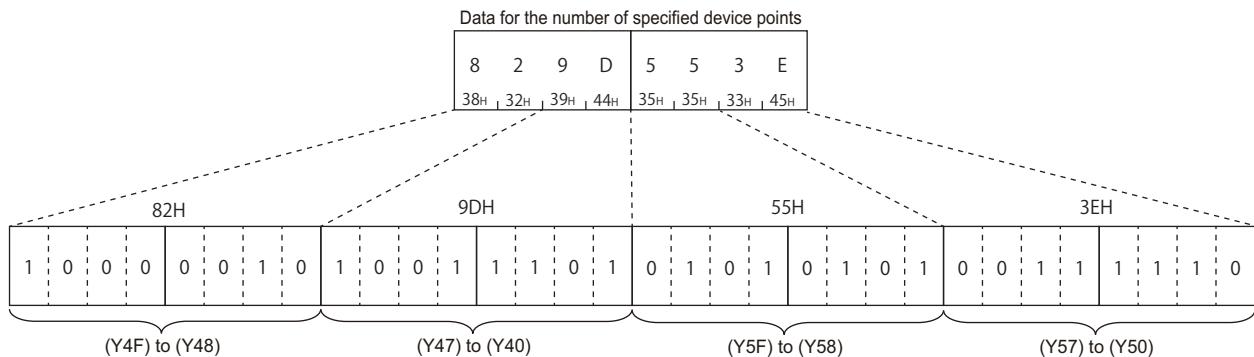
- Head device: Y40
- Number of device points: 32 points (2 bytes)

■Data communication in ASCII code

(Request data)

| Head device | Number of device points |
|---|----------------------------------|
| 5 9 2 0 0 0 0 0 0 0 0 4 0 35H , 39H , 32H , 30H , 30H , 30H , 30H , 30H , 30H , 34H , 30H 35H , 32H , 30H , 30H | 0 2 0 0 30H , 32H , 30H , 30H |

(Response data)

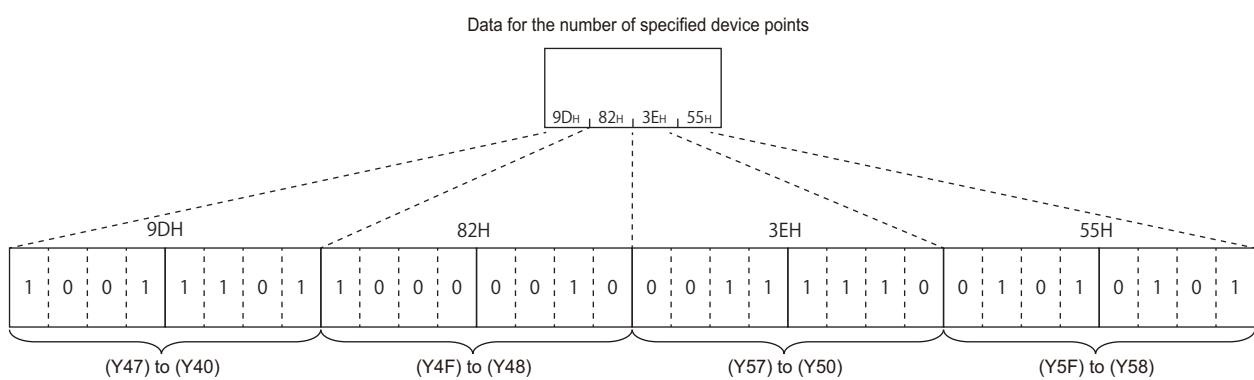


■Data communication in binary code

(Request data)

| Head device | Number of device points |
|-----------------------------------|-------------------------|
| 40H , 00H , 00H , 00H , 20H , 59H | 02H 00H |

(Response data)



Batch write in bit units (command: 02)

Write bit devices (X, Y, M, etc.) in 1-point units.

Message format

The following shows the message format of the request data and response data of the command.

■Request data

| | | | |
|-------------|-------------------------|-------------|------------|
| Head device | Number of device points | Fixed value | Write data |
|-------------|-------------------------|-------------|------------|

■Response data

There is no response data for this command.

Data specified by request data

■Head device

Specify the head device of the bit device to be written. (☞ Page 391 Device codes and device numbers)

■Fixed value

Fixed to '0'.

| ASCII code | Binary code |
|-----------------|-------------|
| 0 0 30H, 30H | 00H |

■Number of device points

Specify the bit device points to be written.

Specify '00H' when the number of device points is to 256 points. (☞ Page 396 Number of device points)

■Write data

Stored data for the number of device points to be written.

- 0 (30H): OFF
- 1 (31H): ON

Communication example

Write the bit devices in the CPU module with E71 mounted under the following conditions.

- Head device: M50
- Number of device points: 12 points

■Data communication in ASCII code

(Request data)

| Head device | Number of device points | Data for the number of specified device points |
|--|-------------------------|---|
| 4 4 2 0 0 0 0 0 0 3 2 34H 44H 32H 30H 30H 30H 30H 30H 33H 32H | 0 3 30H 43H | 0 0 30H 30H 1 1 31H 31H ... 0 1 30H 31H |

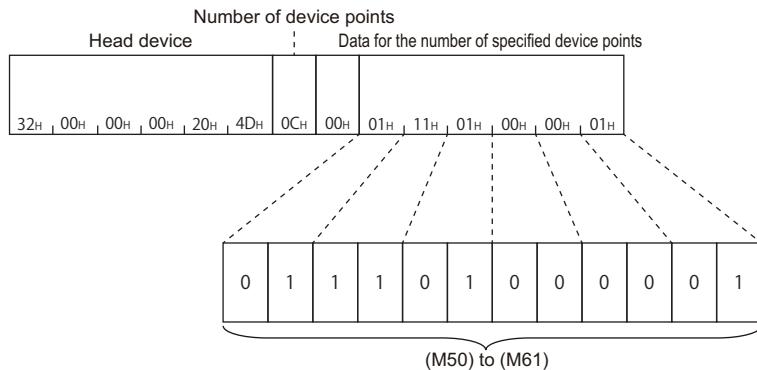
(M50)

to

(M61)

■Data communication in binary code

(Request data)



Batch write in word units (command: 03)

Write bit devices (X, Y, M, etc.) in 16-point units.

Write word devices (D, T, C, etc.) in 1-point units.

Message format

The following shows the message format of the request data and response data of the command.

■Request data

| | | | |
|-------------|-------------------------|-------------|------------|
| Head device | Number of device points | Fixed value | Write data |
|-------------|-------------------------|-------------|------------|

■Response data

There is no response data for this command.

Data specified by request data

■Head device

Specify the head device of the device to be written.

■Number of device points

Specify the number of device points to be written.

Specify '00H' when the number of device points is to 256 points.

When specifying bit devices, set the head device No. in multiples of 16 (0, 16, ... in decimal notation).

■Fixed value

Fixed to '0'.

| ASCII code | Binary code |
|-----------------|-------------|
| 0 0 30H, 30H | 00H |

■Write data

Stored data for the number of device points to be written.

The order of data differs depending on the ASCII code or binary code. (☞ Page 396 Read data, write data)

Communication example

Write the devices in the CPU module with E71 mounted under the following conditions.

- Head device: D100
- Number of device points: 3 points

■Data communication in ASCII code

(Request data)

| Head device | Number of device points | Data for the number of specified device points |
|---|-------------------------------|---|
| 4 4 2 0 0 0 0 0 0 6 4 34H, 34H, 32H, 30H, 30H, 30H, 30H, 30H, 36H, 34H | 0 3 0 0 30H, 33H, 30H, 30H | 1 2 3 4 9 8 7 6 0 1 0 9 31H, 32H, 33H, 34H, 39H, 38H, 37H, 36H, 30H, 31H, 30H, 39H (D100) (D101) (D102) |

■Data communication in binary code

(Request data)

| Head device | Number of device points | Data for the number of specified device points |
|------------------------------|---------------------------------|--|
| 64H, 00H, 00H, 00H, 20H, 44H | 03H 00H 34H 12H 76H 98H 09H 01H | (D100) (D101) (D102) |

Test in bit units (random write) (command: 04)

Specify the devices and device numbers of bit devices (X, Y, M, etc.) in 1-point units randomly, and set/reset them.

Message format

The following shows the message format of the request data and response data of the command.

■Request data

There is no request data for this command.

| | | | | | | |
|---------------------------------------|-------------|-----------------------------------|---------------------------------------|-----|---------------------------------|-------------------------------------|
| Number of device points (n points) | Fixed value | Specified device (first point) | ON/OFF specification (first point) | ... | Specified device (nth point) | ON/OFF specification (nth point) |
|---------------------------------------|-------------|-----------------------------------|---------------------------------------|-----|---------------------------------|-------------------------------------|

■Response data

There is no response data for this command.

Data specified by request data

■Number of device points

Specify the number of points of the bit device to be set/reset. (☞ Page 396 Number of device points)

■Fixed value

Fixed to '0'.

| ASCII code | Binary code | | | | | |
|--|-------------|---|-----|-----|---|-----|
| <table border="1"><tr><td>0</td><td>0</td></tr><tr><td>30H</td><td>30H</td></tr></table> | 0 | 0 | 30H | 30H | <table border="1"><tr><td>00H</td></tr></table> | 00H |
| 0 | 0 | | | | | |
| 30H | 30H | | | | | |
| 00H | | | | | | |

■Specified device

Specify the bit devices to be set/reset. (☞ Page 391 Device codes and device numbers)

■ON/OFF specification

Specify set/reset.

| Processing | ASCII code | Binary code | | | | | |
|------------|--|-------------|---|-----|-----|---|-----|
| Reset | <table border="1"><tr><td>0</td><td>0</td></tr><tr><td>30H</td><td>30H</td></tr></table> | 0 | 0 | 30H | 30H | <table border="1"><tr><td>00H</td></tr></table> | 00H |
| 0 | 0 | | | | | | |
| 30H | 30H | | | | | | |
| 00H | | | | | | | |
| Set | <table border="1"><tr><td>0</td><td>1</td></tr><tr><td>30H</td><td>31H</td></tr></table> | 0 | 1 | 30H | 31H | <table border="1"><tr><td>01H</td></tr></table> | 01H |
| 0 | 1 | | | | | | |
| 30H | 31H | | | | | | |
| 01H | | | | | | | |

Communication example

Set/reset the bit devices in CPU module with E71 mounted under the following conditions.

- Number of device points: 3 points
- Data for the number of specified device points: Specify Y94 to ON, M60 to OFF, and B26 to ON.

■Data communication in ASCII code

(Request data)

| Number of device points | | Specified device (first point) | | | | | | | | | | ON/OFF specification (first point) | |
|-------------------------|-----|---------------------------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-------------------------------------|-----|
| 0 | 3 | 0 | 0 | 5 | 9 | 2 | 0 | 0 | 0 | 0 | 0 | 9 | 4 |
| 30H | 33H | 30H | 30H | 35H | 39H | 32H | 30H | 30H | 30H | 30H | 30H | 39H | 34H |
| (Y94) | | | | | | | | | | 0 | 1 | 30H | 31H |
| (ON) | | | | | | | | | | | | | |
| | | Specified device (second point) | | | | | | | | | | ON/OFF specification (second point) | |
| | | 4 | D | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 3 | C |
| | | 34H | 44H | 32H | 30H | 33H | 43H |
| (M60) | | | | | | | | | | 0 | 0 | 30H | 30H |
| (OFF) | | | | | | | | | | | | | |
| | | Specified device (third point) | | | | | | | | | | ON/OFF specification (third point) | |
| | | 4 | 2 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 6 | 0 |
| | | 34H | 32H | 32H | 30H | 30H | 30H | 30H | 30H | 30H | 32H | 36H | 30H |
| (B26) | | | | | | | | | | 0 | 1 | 30H | 31H |
| (ON) | | | | | | | | | | | | | |

■Data communication in binary code

(Request data)

| Number of device points | Specified device (first point) | ON/OFF specification (first point) | Specified device (second point) | ON/OFF specification (second point) | Specified device (third point) | ON/OFF specification (third point) |
|-------------------------|--------------------------------|------------------------------------|---------------------------------|-------------------------------------|--------------------------------|------------------------------------|
| 03H 00H | 94H 00H | 00H 00H | 20H 00H | 59H 01H | 3CH 00H | 00H 00H |

(Y94)

(ON)

(M60)

(OFF)

(B26)

(ON)

Test in word units (random write) (command: 05)

Specify the devices and device numbers of bit devices (X, Y, M, etc.) in 16-point units randomly, and set/reset them. Specify the devices and device numbers of word devices (D, T, C, etc.) in 1-point units randomly, and write them.

Message format

The following shows the message format of the request data and response data of the command.

■Request data

| | | | | | | |
|---------------------------------------|-------------|-----------------------------------|---------------------------------------|-----|---------------------------------|-------------------------------------|
| Number of device points (n points) | Fixed value | Specified device (first point) | ON/OFF specification (first point) | ... | Specified device (nth point) | ON/OFF specification (nth point) |
|---------------------------------------|-------------|-----------------------------------|---------------------------------------|-----|---------------------------------|-------------------------------------|

■Response data

There is no response data for this command.

Data specified by request data

■Number of device points

Specify the number of device points to be set/reset. (☞ Page 396 Number of device points)

When specifying bit devices, set the head device No. in multiples of 16 (0, 16, ... in decimal notation).

■Fixed value

Fixed to '0'.

| ASCII code | Binary code | | | | | |
|---|-------------|---|-----|-----|--------------------------------------|-----|
| <table><tr><td>0</td><td>0</td></tr><tr><td>30H</td><td>30H</td></tr></table> | 0 | 0 | 30H | 30H | <table><tr><td>00H</td></tr></table> | 00H |
| 0 | 0 | | | | | |
| 30H | 30H | | | | | |
| 00H | | | | | | |

■Specified device

Specify the device to be set/reset. (☞ Page 391 Device codes and device numbers)

■ON/OFF specification

Store the data to be written.

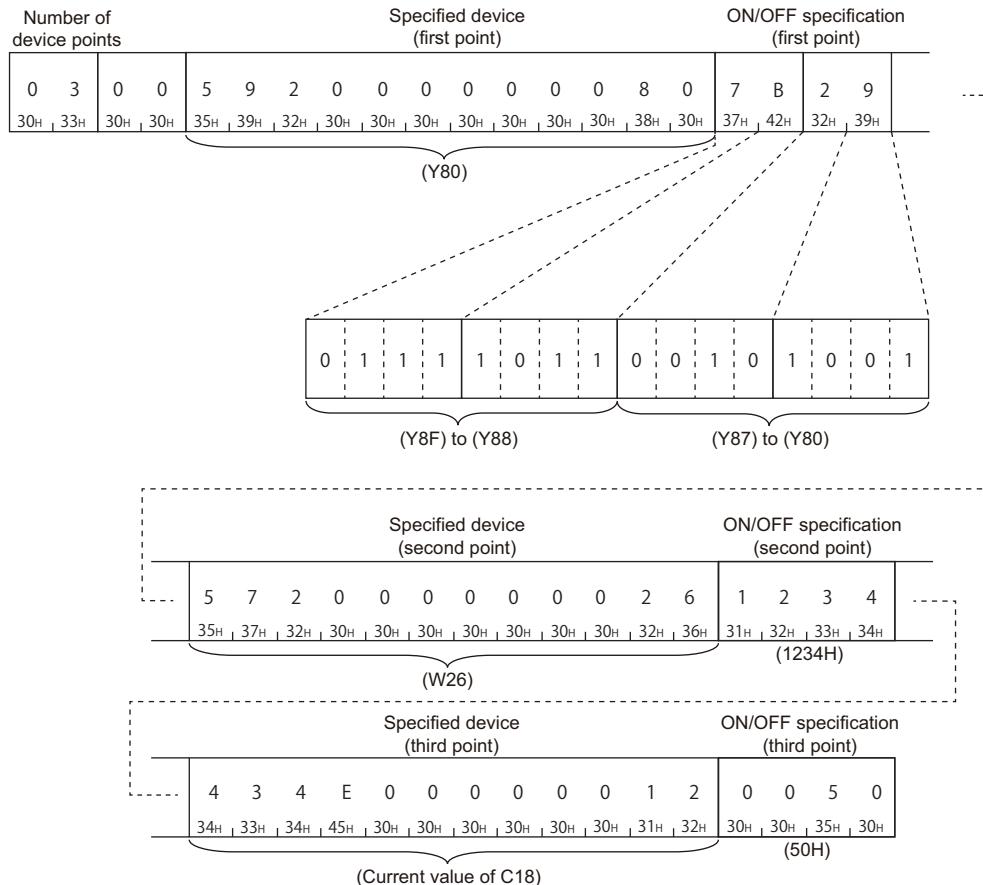
Communication example

Set/reset the devices in the CPU module with E71 mounted under the following condition.

- Number of device points: 3 points
 - Data for the number of specified device points: Specify Y80 to 8F to ON/OFF, W26 to '1234H', and the current value of C18 to '50H'.

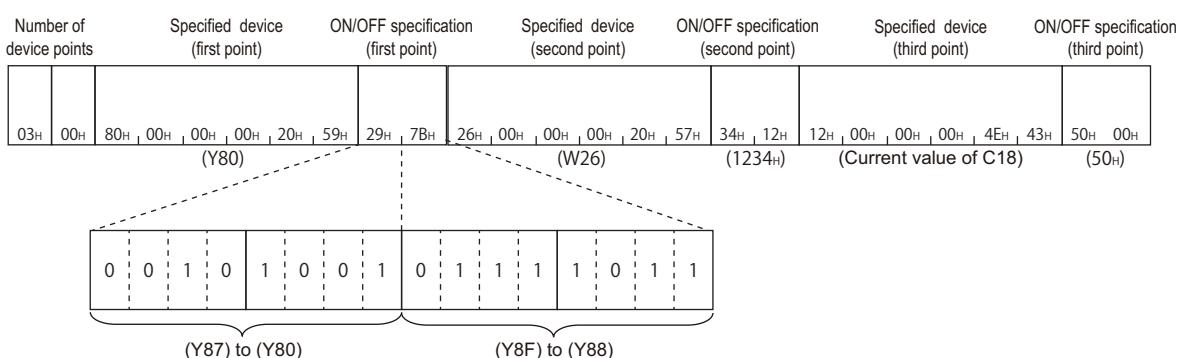
■Data communication in ASCII code

(Request data)



■ Data communication in binary code

(Request data)



Monitor device memory (command: 06, 07, 08, 09)

The ON/OFF status or contents of devices in the CPU module can be monitored from an external device by registering the devices and device numbers to be monitored to E71 in advance and executing the monitor command from the external device.

When reading device memory in batch, the read device numbers will be consecutive. However, when reading them using the monitor command, the devices can be monitored randomly by specifying the arbitrary devices and device numbers.

For monitoring the extended file register, refer to the following section.

☞ Page 419 Monitor extended file registers (command: 1A, 1B)

Procedure for monitoring

1. Register the devices to be monitored from the external device to E71 by registering monitor data.
2. Execute the read processing by a monitor command.
3. Process the data.
4. If do not change the devices to be monitored, return to step 2, and repeat the process.

Point

- When monitoring data as the procedure shown above, the monitor data registration is required. If monitoring data without registering the data, a protocol error occurs.
- The content of registered monitor data is deleted when turning the power OFF or the resetting the CPU module.
- There are three types of commands for monitor data registration; expansion file register, device memory bit unit, and device memory word unit. The recently registered one command out of three types of commands can be registered to E71.
- When registering device memory of the CPU module as a monitoring data from more than one external devices on the same station, the recently registered device memory will be available since the registration data is overwritten.

Register monitor data(command: 06, 07)

Registers bit devices (X, Y, M, etc.) to be monitored in 1-point units. (Command: 06)

Registers bit devices (X, Y, M, etc.) to be monitored in 16-point units. (Command: 07)

Registers word devices (D, T, C, etc.) to be monitored in 1-point units. (Command: 07)

■Request data

| | | | | |
|---------------------------------------|----------------|--------------------------------|-----|------------------------------|
| Number of device points (n points) | Fixed value | Device number (first point) | ... | Device number (nth point) |
|---------------------------------------|----------------|--------------------------------|-----|------------------------------|

- Number of device points: Specify the number of devices to be registered as a monitoring data. ([Page 396 Number of device points](#))
- Fixed value: '0'.

| ASCII code | Binary code | | | | | |
|---|-------------|---|-----|-----|---|-----|
| <table border="1"> <tr> <td>0</td> <td>0</td> </tr> <tr> <td>30H</td> <td>30H</td> </tr> </table> | 0 | 0 | 30H | 30H | <table border="1"> <tr> <td>00H</td> </tr> </table> | 00H |
| 0 | 0 | | | | | |
| 30H | 30H | | | | | |
| 00H | | | | | | |

- Device number: Specify the device number to be registered as a monitor data. ([Page 391 Device codes and device numbers](#))



When specifying bit devices at monitor data registration in word unit, set the device numbers in multiples of 16 (0, 16, ... in decimal notation).

■Response data

There is no response data for this command.

■Communication example

Register devices as a monitoring data in the CPU module with E71 mounted under the following conditions.

- Number of device points: 3 points
- Device number: Y46, M12, B2C

(Request data)

(ASCII code)

| Number of device points | | Device number | | | | | | | | | | | | |
|-------------------------|-----|---------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| 0 | 3 | 0 | 0 | 5 | 9 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 4 | 6 |
| <hr/> | | | | | | | | | | | | | | |
| 30H | 33H | 30H | 30H | 35H | 39H | 32H | 30H | 30H | 30H | 30H | 30H | 30H | 34H | 36H |
| (Y46) | | | | | | | | | | | | | | |
| <hr/> | | | | | | | | | | | | | | |
| Device number | | | | | | | | | | | | | | |
| 4 | D | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | C | 4 | 2 | 2 |
| 34H | 44H | 32H | 30H | 43H | 34H | 32H | 32H |
| (M12) | | | | | | | | | | | | | | |
| <hr/> | | | | | | | | | | | | | | |
| 4 | 2 | 2 | E | 0 | 0 | 0 | 0 | 0 | 0 | 0 | C | 0 | 0 | 2 |
| 34H | 32H | 32H | 45H | 30H | 43H | 30H | 30H | 32H |
| (B2C) | | | | | | | | | | | | | | |

(Binary code)

| Number of device points | Device number | Device number | Device number |
|-------------------------|---------------|---------------|---------------|
| 03H | 00H | 46H | 00H |
| | | 00H | 00H |
| | | 20H | 20H |
| | | 59H | 4DH |
| | | 0CH | 2CH |
| | | 00H | 00H |
| | | 00H | 00H |
| | | 20H | 42H |

Monitor in bit units (command: 08)

Monitor the bit devices for which monitor data is registered.

■Request data

There is no request data for this command.

■Response data

The value of the read device is stored in byte units.

The order of data differs depending on the ASCII code or binary code.



If the number of device points registered to be monitored is an odd number, dummy data 0 (30H) is added when the monitoring is executed. For example, if the number of device points registered to be monitored is three points, data for four points is returned. The last byte will be a dummy data.

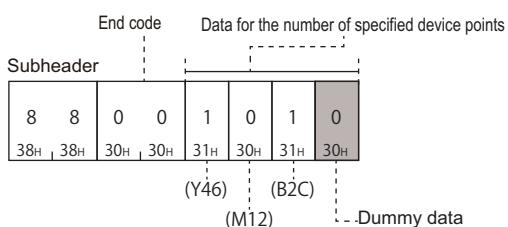
■Communication example

Monitor the devices registered with monitor data registration in the CPU module with E71 mounted under the following conditions.

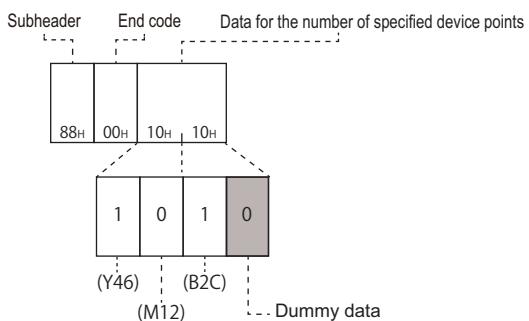
- Number of registered device points: 3 points
- Number of registered device numbers: Y46, M12, B2C

(Response data)

(ASCII code)



(Binary code)



Monitor in word units (command: 09)

Monitor word devices and bit devices (16 point units) which are registered as a monitor data.

■ Request data

There is no request data for this command.

■ Response data

The value of the read device is stored in word units.

The order of data differs depending on the ASCII code or binary code.

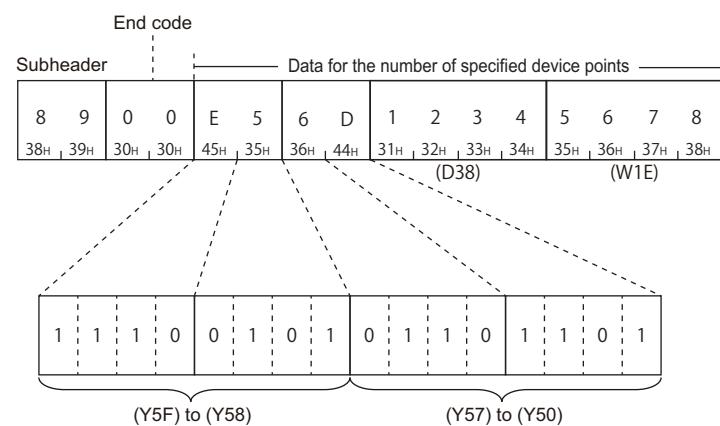
■Communication example

Monitor the devices registered with monitor data registration in the CPU module with E71 mounted under the following conditions.

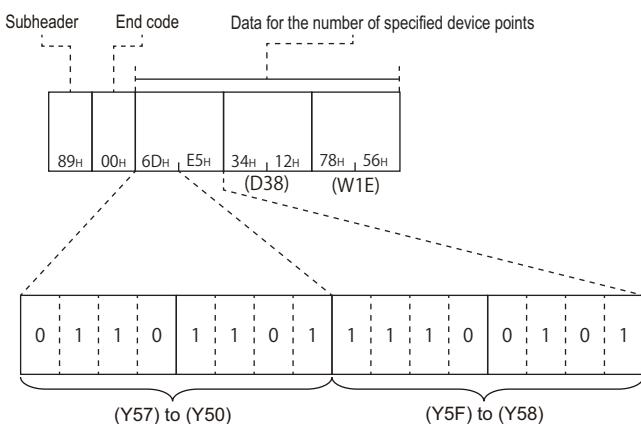
- Number of registered device numbers: Y50 to 5F, D38, W1E

(Response data)

(ASCII code)



(Binary code)



18.5 Read and Write Extended File Register

The extended file register is a memory area that stores required data and operation result for various data processing and AnACPU and AnUSCPU extended file register dedicated instructions by using the software package for extended file register 'SWOGHP-UTLPC-FN1' or 'SW0SRX-FNUP' (hereinafter abbreviated to UTLP-FN1 and FNUP). The extended file register uses free area of user memory area in CPU module as a file register.

This section explains the commands for reading and writing extended file register.

For the message formats other than request data and response data, refer to the following sections.

☞ Page 385 Message Format, Page 385 Detail of Setting Data



For the considerations when reading and writing extended file register, refer to the following section.

☞ Page 365 Considerations for reading and writing extended file register

For the specification method of the extended file register, refer to the following section.

☞ Page 366 Specification method for extended file register

Data to be specified in command

Block number

Specify the block number of the extended file register.

■Data communication in ASCII code

Convert the numerical value to 4-digit ASCII code (hexadecimal) and send it from upper digits.

■Data communication in binary code

Send 2-byte numerical value.

Device number

Specify the device number of the extended file register.

When reading or writing (command: 3B, 3C) extended file register directly, refer to the following section.

☞ Page 367 Device number (address) specification using AnA/AnUCPU common commands

■Data communication in ASCII code

Convert the numerical value to 12-digit ASCII code (hexadecimal), and send it from the upper digits.

■Data communication in binary code

Send 6-byte numerical value.

Batch read (command: 17)

Read extended file register (R) in 1-point units.

Message format

The following shows the message format of the request data and response data of the command.

■Request data

(ASCII code)

| | | | |
|-----------|---------------|-------------------------|-----------------|
| Block No. | Device number | Number of device points | 0 0 30H, 30H |
|-----------|---------------|-------------------------|-----------------|

(Binary code)

| | | | |
|---------------|-----------|-------------------------|-----|
| Device number | Block No. | Number of device points | 00H |
|---------------|-----------|-------------------------|-----|

■Response data

The data for the number of device points are stored.

The order of data differs depending on the ASCII code or binary code. (☞ Page 396 Read data, write data)

Data specified by request data

■Device number

Specify the head device of the extended file register to be read. (☞ Page 412 Device number)

■Block number

Specify the block number of the extended file register to be read. (☞ Page 412 Block number)

■Number of device points

Specify the points of extended file register to be read.

Specify '00H' when the number of device points is 256 points. (☞ Page 396 Number of device points)

Communication example

Read the extended file registers in the CPU module with E71 mounted under the following conditions.

- Block number: No.2
- Device number: R70
- Number of device points: 3 points

■Data communication in ASCII code

(Request data)

| Block No. | Device number | Number of device points |
|-----------|-------------------------|--|
| 0 0 0 2 | 5 2 2 0 0 0 0 0 0 0 4 6 | 0 3 0 0 30H , 30H , 30H , 32H 35H , 32H , 32H , 30H , 30H , 30H , 30H , 30H , 30H , 34H , 36H 30H , 33H 30H , 30H |

(Response data)

| Data for the number of specified device points | | | |
|--|-----------------------|-----------------------|--|
| 1 2 3 4 | 8 7 6 5 | 0 1 3 F | |
| 31H , 32H , 33H , 34H | 38H , 37H , 36H , 35H | 30H , 31H , 33H , 46H | |

(R70 in No.2) (R71 in No.2) (R72 in No.2)

■Data communication in binary code

(Request data)

| Device number | Block No. | Number of device points |
|-----------------------------|-----------|-------------------------|
| 46H , 00H , 00H , 20H , 52H | 02H , 00H | 03H , 00H |

(Response data)

| Data for the number of specified device points | | | |
|--|-----|-----|-----|
| 34H | 12H | 65H | 87H |
| 3F _H | 01H | | |

(R70 in No.2) (R71 in No.2) (R72 in No.2)

Batch write (command: 18)

Write to extended file register (R) in 1-point units.

Message format

The following shows the message format of the request data and response data of the command.

■Request data

(ASCII code)

| | | | | |
|-----------|---------------|-------------------------|-----------------|--|
| Block No. | Device number | Number of device points | 0 0 30H, 30H | Data for the number of specified device points |
|-----------|---------------|-------------------------|-----------------|--|

(Binary code)

| | | | | |
|---------------|-----------|-------------------------|-----|--|
| Device number | Block No. | Number of device points | 00H | Data for the number of specified device points |
|---------------|-----------|-------------------------|-----|--|

■Response data

There is no response data for this command.

Data specified by request data

■Device number

Specify the head device number of the extended file register to be written. ([Page 412 Device number](#))

■Block number

Specify the block number of the extended file register to be written. ([Page 412 Block number](#))

■Number of device points

Specify the number of the data points to be write.

Specify '00H' when the number of device points is to 256 points. ([Page 396 Number of device points](#))

■Data for the number of specified device points

Store the data to be written to the extended file registers.

Communication example

Write data to extended file registers in the CPU module with E71 mounted under the following conditions.

- Block number: No.3
- Device number: R100
- Number of device points: 3 points

■Data communication in ASCII code

(Request data)

| Block No. | Device number | Number of device points |
|---|-------------------------|-------------------------|
| 0 0 0 3 | 5 2 2 0 0 0 0 0 0 0 6 4 | 0 3 0 0 ... |
| 30H, 30H, 30H, 33H 35H, 32H, 32H, 30H, 30H, 30H, 30H, 30H, 30H, 36H, 34H 30H, 33H, 30H, 30H | | |
| 0 1 0 9 | 9 8 7 6 | 1 2 3 4 |
| 30H, 31H, 30H, 39H | 39H, 38H, 37H, 36H | 31H, 32H, 33H, 34H |
| (R100 in No.3) (R101 in No.3) (R102 in No.3) | | |

■Data communication in binary code

(Request data)

| Device number | Block No. | Number of device points | Data for the number of specified device points |
|-------------------------------|-----------|-------------------------|--|
| 64H, 00H, 00H, 00H, 20H, 52H | 03H, 00H | 03H, 00H, 09H, 01H | 76H, 98H, 34H, 12H |
| (R100 in No.3) (R101 in No.3) | | (R102 in No.3) | |

Test (random write) (command: 19)

Specify the block numbers and device numbers in 1-point units and write them randomly to extended file register (R).

Message format

The following shows the message format of the request data and response data of the command.

■Request data

(ASCII code)

| Data for the number of device points | | | | |
|--------------------------------------|------------------|-----------|---------------|------------|
| Number of device points | 0 0 30H 30H | Block No. | Device number | Write data |

(Binary code)

| Data for the number of device points | | | |
|--------------------------------------|---------------|-----------|------------|
| Number of device points | Device number | Block No. | Write data |
| 00H | | | |

■Response data

There is no response data for this command.

Data specified by request data

■Number of device points

Specify the number of points of the data to be written. (☞ Page 396 Number of device points)

■Device number

Specify the head device of the extended file register to be written. (☞ Page 412 Device number)

■Block number

Specify the block number of the extended file register to be written. (☞ Page 412 Block number)

■Write data

Store the data to be written to the extended file registers.

Communication example

Write data to extended file registers in the CPU module with E71 mounted under the following conditions.

- Data to be written: R26 in block number 2 and R19 in block number 3
- Number of device points: 2 points

■Data communication in ASCII code

(Request data)

| Number of device points | Block No. | | | | Device number | | | | Write data | | | | |
|-------------------------|-----------|---------|---------|---------------|---------------|---------|-----|------------|------------|-----|-----|-----|-----|
| 0 2 | 0 0 | 0 0 0 | 2 | 5 2 2 0 | 0 0 0 0 | 0 0 0 0 | 1 A | 1 2 3 4 | 31H | 32H | 33H | 34H | |
| (R26 in block No.2) | | | | | | | | | | | | | |
| Block No. | | | | Device number | | | | Write data | | | | | |
| 0 0 0 3 | 5 2 2 0 | 0 0 0 0 | 0 0 0 0 | 1 3 | 0 1 0 9 | 30H | 31H | 30H | 39H | 30H | 31H | 30H | 39H |
| (R19 in block No.3) | | | | | | | | | | | | | |

■Data communication in binary code

(Request data)

| Number of device points | Device number | | | | Block No. | Write data | Device number | | | | Block No. | Write data | |
|-------------------------|---------------|-----|-----|-----|-----------|------------|---------------|-----|-----|-----|-----------|------------|---------------------|
| 02H 00H | 1AH | 00H | 00H | 20H | 02H | 00H | 34H | 12H | 13H | 00H | 00H | 20H | 52H 03H 00H 09H 01H |
| (R26 in block No.2) | | | | | | | | | | | | | |
| (R19 in block No.3) | | | | | | | | | | | | | |

Monitor extended file registers (command: 1A, 1B)

The contents of extended file registers in the CPU module can be monitored from an external device by registering the block numbers and device numbers to be monitored to E71 in advance and executing the monitor command from the external device.

When reading extended file register in batch, the read device numbers will be consecutive. However, when reading them using the monitor command, the extended file registers can be read randomly by specifying the file registers of arbitrary block numbers and device numbers.

Procedure for monitoring

1. Register block numbers and device numbers of the extended file registers to be monitored to E71 by monitor data registration.
2. Execute read processing by monitoring.
3. Process the data.
4. If do not change the devices to be monitored, return to step 2, and repeat the process.

Point

- When monitoring data as the procedure shown above, the monitor data registration is required. If monitoring data without registering the data, an error occurs (END code: 57H).
- The content of registered monitor data is deleted when turning the power OFF or the resetting the CPU module.
- There are three types of commands for monitor data registration; extended file register, device memory bit unit, and device memory word unit. The recently registered one command out of three types of commands can be registered to E71.
- When registering device memory of the CPU module as a monitoring data from more than one external devices on the same station, the recently registered device memory will be available since the registration data is overwritten.

Register monitor data (command: 1A)

Register device numbers to be monitored in 1-point units.

■ Request data

(ASCII code)

| Data for the number of device points | | | |
|--------------------------------------|----------------------|-----------|---------------|
| Number of device points | 0 0 30H 30H | Block No. | Device number |

(Binary code)

| Data for the number of device points | | | |
|--------------------------------------|--|---------------|-----------|
| Number of device points | | Device number | Block No. |
| 00 | | | |

- Number of device points: Specify the number of points of the extended file registers to be registered as a monitor data. (☞ Page 396 Number of device points)
 - Device number: Specify the extended file registers to be registered as a monitor data. (☞ Page 412 Device number)
 - Block number: Specify the block number of the extended file registers to be registered as a monitor data. (☞ Page 412 Block number)

■ Response data

There is no response data for this command.

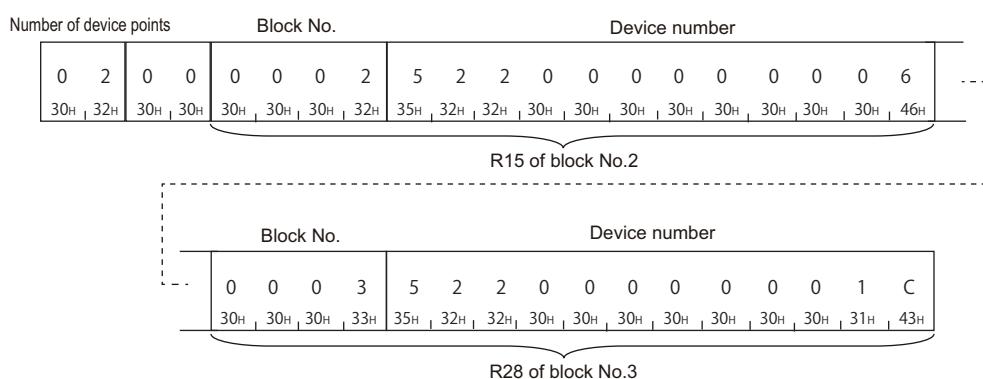
■Communication example

Register devices as a monitoring data in the CPU module with E71 mounted under the following conditions.

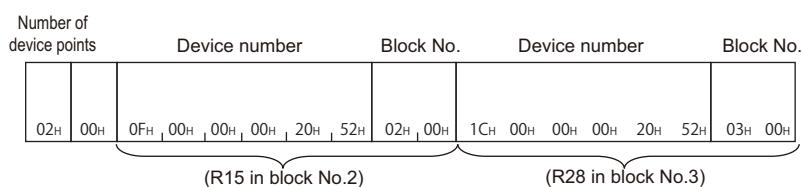
- Data to be registered: R15 of block number 2 and R28 of block number 3

(Request data)

(ASCII code)



(Binary code)



Monitoring (command: 1B)

Monitor the extended file registers registered by monitor data registration.

■Data specified by request data

There is no request data for this command.

■Response data

Store the monitoring result.

■Communication example

Monitor the following extended file registers registered by monitor data registration.

- Monitor data registration: R15 in block number 2 and R28 in block number 3

(Response data)

(ASCII code)

| Monitoring result | | | | | | | |
|-------------------|-----|-----|-----|-----|-----|-----|-----|
| E | 5 | 6 | D | 1 | 2 | 3 | 4 |
| 45H | 35H | 36H | 44H | 31H | 32H | 33H | 34H |

(R15 in block No.2) (R28 in block No.3)

(Binary code)

| Monitoring result | | | | | | | |
|-------------------|-----------------|-----------------|-----------------|--|--|--|--|
| | | | | | | | |
| 6D _H | E5 _H | 34 _H | 12 _H | | | | |

(R15 in block No.2) (R28 in block No.3)

Direct read (command: 3B)

Read extended file register in 1-point (1 word) units by specifying the consecutive device numbers of extended file register. This command is a dedicated command for AnACPU. (The command is equivalent to AnA/AnUCPU common commands of 1C protocol.)

Message format

The following shows the message format of the request data and response data of the command.

■Request data

| | | |
|---------------|-------------------------|-------------|
| Device number | Number of device points | Fixed value |
|---------------|-------------------------|-------------|

■Response data

The data for the number of device points are stored.

(☞ Page 396 Read data, write data)

Data specified by request data

■Device number

Specify the start address of the extended file register to be read. (☞ Page 412 Device number)

■Number of device points

Specify the points of extended file register to be read.

Specify '00H' when the number of device points is to 256 points. (☞ Page 396 Number of device points)

■Fixed value

Fixed to '0'.

| ASCII code | Binary code |
|-----------------|-------------|
| 0 0 30H, 30H | 00H |

Communication example

Read the extended file registers directly in the CPU module with E71 mounted under the following conditions.

- Block number: No.0
- Device number: R70
- Number of device points: 4 points

■Data communication in ASCII code

(Request data)

| Block No. | Device number | Number of device points |
|----------------------------------|--|--------------------------------|
| 0 0 0 2 30H , 30H , 30H , 32H | 5 2 2 0 35H , 32H , 32H , 30H , 30H , 30H , 30H , 30H , 30H , 34H , 36H | 0 4 6 30H , 34H , 30H , 30H |

(Response data)

| Data for the number of specified device points | | | | | | | |
|--|----------------------------------|----------------------------------|----------------------------------|---------------|---------------|---------------|---------------|
| 1 2 3 4 31H , 32H , 33H , 34H | 8 7 6 5 38H , 37H , 36H , 35H | 0 1 3 F 30H , 31H , 33H , 46H | 0 0 2 0 30H , 30H , 32H , 30H | (R70 in No.0) | (R71 in No.0) | (R72 in No.0) | (R73 in No.0) |

■Data communication in binary code

(Request data)

| Device number | Number of device points |
|-----------------------------|-------------------------|
| 46H , 00H , 00H , 20H , 52H | 04H 00H |

(Response data)

| Data for the number of specified device points | | | | | | | |
|--|--------------------------|--------------------------|--------------------------|--|--|--|--|
| 34H 12H (R70 in No.0) | 65H 87H (R71 in No.0) | 3FH 01H (R72 in No.0) | 20H 00H (R73 in No.0) | | | | |

Direct write (command: 3C)

Write extended file register in 1-point (1 word) units by specifying the consecutive device number of extended file register. This command is a dedicated command for AnACPU. (The command is equivalent to AnA/AnUCPU common commands of 1C protocol.)

Message format

The following shows the message format of the request data and response data of the command.

■Request data

| | | | |
|---------------|-------------------------|-------------|--|
| Device number | Number of device points | Fixed value | Data for the number of specified device points |
|---------------|-------------------------|-------------|--|

■Response data

There is no response data for this command.

Data specified by request data

■Device number

Specify the start address of the extended file register to be written. (☞ Page 412 Device number)

■Number of device points

Specify the number of extended file registers to be written. (☞ Page 396 Number of device points)

Specify '00H' when the number of device points is to 256 points.

■Fixed value

Fixed to '0'.

| ASCII code | Binary code |
|----------------|-------------|
| 0 0 30H 30H | 00H |

■Data for the number of specified device points

Stored data for the number of device points to be written.

Communication example

Write data to the extended file registers directly in the CPU module with E71 mounted under the following conditions.

- Block number: No.0
- Device number: R100
- Number of device points: 3 points

■Data communication in ASCII code

(Request data)

| Device number | Number of device points | Data for the number of specified device points |
|--|----------------------------------|--|
| 5 2 2 0 0 0 0 0 0 6 4 35H , 32H , 32H , 30H , 30H , 30H , 30H , 30H , 36H , 34H | 0 3 0 0 30H , 33H , 30H , 30H | 0 1 0 9 30H , 31H , 30H , 39H 9 8 7 6 39H , 38H , 37H , 36H 1 2 3 4 31H , 32H , 33H , 34H |

(R100 in No.0)

(R101 in No.0)

(R102 in No.0)

■Data communication in binary code

(Request data)

| Device number | Number of device points | Data for the number of specified device points |
|-----------------------------------|------------------------------------|---|
| 64H , 00H , 00H , 00H , 20H , 52H | 03H 00H 09H 01H 76H 98H 34H 12H | (R100 in No.0) (R102 in No.0) (R101 in No.0) |

18.6 Read and Write Buffer Memory of Special Function Module

The section explains the commands that read/write data from/to buffer memory of MELSEC-A series special function module. For the message formats other than request data and response data, refer to the following sections.

☞ Page 385 Message Format, Page 385 Detail of Setting Data

Point

This command accesses in byte units.

Data to be specified in command

This section explains the contents and specification methods for data items which are set in each command related to the access to the special function module buffer memory.

Start address

Specify the start address of the buffer memory to be read/written.

The value to specify is the same as 1C frame.

☞ Page 378 Special function module No.

■Data communication in ASCII code

Convert the numerical value to 6-digit ASCII code (hexadecimal), and send it from the upper digits.

■Data communication in binary code

Send 3-byte numerical values from lower byte (L: bits 0 to 7).

Ex.

When the head area address is 1E1H

| ASCII code | Binary code |
|--|-------------|
| 0 0 0 1 E 1 30H 30H 30H 31H 45H 31H | E1H 01H 00H |

Byte length

Specify the byte length of the buffer memory to be read or written.

Specify '00' for 256 bytes.

■Data communication in ASCII code

Convert the numerical value to 2-digit ASCII code (hexadecimal), and send it from the upper digits.

■Data communication in binary code

Send 1-byte numerical value.

Ex.

For 5 bytes, 20 bytes, 256 bytes.

| Number of device points | ASCII code | Binary code | | | | | | |
|-------------------------|--|-------------|---|-----|-----|---|----|-----|
| 5 bytes | <table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td>0</td> <td>5</td> </tr> <tr> <td>30H</td> <td>35H</td> </tr> </table> | 0 | 5 | 30H | 35H | <table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td>05</td> </tr> <tr> <td>05H</td> </tr> </table> | 05 | 05H |
| 0 | 5 | | | | | | | |
| 30H | 35H | | | | | | | |
| 05 | | | | | | | | |
| 05H | | | | | | | | |
| 20 bytes | <table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td>1</td> <td>4</td> </tr> <tr> <td>31H</td> <td>34H</td> </tr> </table> | 1 | 4 | 31H | 34H | <table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td>14</td> </tr> <tr> <td>14H</td> </tr> </table> | 14 | 14H |
| 1 | 4 | | | | | | | |
| 31H | 34H | | | | | | | |
| 14 | | | | | | | | |
| 14H | | | | | | | | |
| 256 bytes | <table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td>0</td> <td>0</td> </tr> <tr> <td>30H</td> <td>30H</td> </tr> </table> | 0 | 0 | 30H | 30H | <table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td>00</td> </tr> <tr> <td>00H</td> </tr> </table> | 00 | 00H |
| 0 | 0 | | | | | | | |
| 30H | 30H | | | | | | | |
| 00 | | | | | | | | |
| 00H | | | | | | | | |

Read data, write data

The read buffer memory value is stored for reading, and the data to be written is stored for writing.

This function reads and writes data in byte units.

☞ Page 156 Read data, write data

Special function module No.

Specify the last input/output signal (I/O address) of the special function module. (Specify the upper 2-digit of 3-digit representation.)

The value to specify is the same as 1C frame.

☞ Page 378 Special function module No.

■Data communication in ASCII code

Convert the numerical value to 2-digit ASCII code (hexadecimal), and send it from the upper digits.

■Data communication in binary code

Send 1-byte numerical value.

Accessible modules

Special function modules that can be accessed buffer memory are the same as 1C frame.

☞ Page 379 Accessible modules

Batch read (command: 0E)

Read the buffer memory of a special function module.

Message format

The following shows the message format of the request data and response data of the command.

■Request data

| | | | |
|---------------|-------------|-----------------------------|-------------|
| Start address | Byte length | Special function module No. | Fixed value |
|---------------|-------------|-----------------------------|-------------|

■Response data

The data read from buffer memory is stored. (☞ Page 427 Read data, write data)

Data specified by request data

■Start address

Specify the buffer memory start address to be read. (☞ Page 426 Start address)

■Byte length

Specify the byte length of buffer memory to be read.

Specify '00H' when the number of device points is to 256 points. (☞ Page 427 Byte length)

■Special function module No.

Specify the special function module No. of the buffer memory to be read.

■Fixed value

Fixed to '0'.

| ASCII code | Binary code |
|----------------|-------------|
| 0 0 30H 30H | 00H |

Communication example

Read the buffer memory of the special function module with E71 mounted in the same station under the following conditions.

- Special function module No.: 13H (buffer memory whose input/output signal is from 120H to 13FH)
- Start address: 7F0H
- Byte length: 4

■Data communication in ASCII code

(Request data)

| Start address | | Byte length | | Special function module No. | | | |
|---------------|-----|-------------|-----|-----------------------------|-----|-----|-----------------|
| 0 | 0 | 7 | F | 0 | 0 | 4 | 1 3 0 0 |
| 30H | 30H | 30H | 37H | 46H | 30H | 34H | 31H 33H 30H 30H |

(Response data)

| Data read | | | |
|-----------|---------|---------|---------|
| 0 9 | 1 8 | 2 0 | 3 4 |
| 30H 39H | 31H 38H | 32H 30H | 33H 34H |

(7F0H) (7F1H) (7F2H) (7F3H)

■Data communication in binary code

(Request data)

| Head address | | Byte length | | Special function module No. | | | |
|--------------|-----|-------------|-----|-----------------------------|-----|--|--|
| F0H | 07H | 00H | 04H | 13H | 00H | | |

(Response data)

| Data read | | | |
|-----------|--------|--------|--------|
| 09H | 18H | 20H | 34H |
| (7F0H) | (7F1H) | (7F2H) | (7F3H) |

Batch write (command: 0F)

Write data to the buffer memory of a special function module.

Message format

The following shows the message format of the request data and response data of the command.

■Request data

| | | | | |
|---------------|-------------|-----------------------------|-------------|---|
| Start address | Byte length | Special function module No. | Fixed value | Write data (for the length of specified bytes) |
|---------------|-------------|-----------------------------|-------------|---|

■Response data

There is no response data for this command.

Data specified by request data

■Start address

Specify the buffer memory start address to be read. (☞ Page 426 Start address)

■Byte length

Specify the byte length of buffer memory to be read.

Specify '00H' when the number of device points is to 256 points. (☞ Page 427 Byte length)

■Special function module No.

Specify the special function module No. of the buffer memory to be read.

■Fixed value

Fixed to '0'.

| ASCII code | Binary code |
|----------------|-------------|
| 0 0 30H 30H | 00H |

■Write data

Store the data to be written in a buffer memory. (☞ Page 427 Read data, write data)

Communication example

Write data to buffer the memory of the special function module with E71 mounted in the same station under the following conditions.

- Special function module No.: 13H (buffer memory whose input/output signal is from 120H to 13FH)
- Start address: 750H
- Byte length: 4

■Data communication in ASCII code

(Request data)

| Start address | Byte length | Special function module No. | Write data |
|--|----------------|--------------------------------|--|
| 0 0 0 7 5 0 30H 30H 30H 37H 35H 30H | 0 4 30H 34H | 1 3 0 0 31H 33H 30H 30H | 0 1 2 3 4 5 6 7 30H 31H 32H 33H 34H 35H 36H 37H |

(750H) (751H) (752H) (753H)

■Data communication in binary code

(Request data)

| Start address | Byte length | Special function module No. | Write data |
|-------------------------|-----------------|--------------------------------|------------|
| 50H 07H 00H 04H 13H 00H | 01H 23H 45H 67H | | |

(750H) (751H) (752H) (753H)

APPENDIX

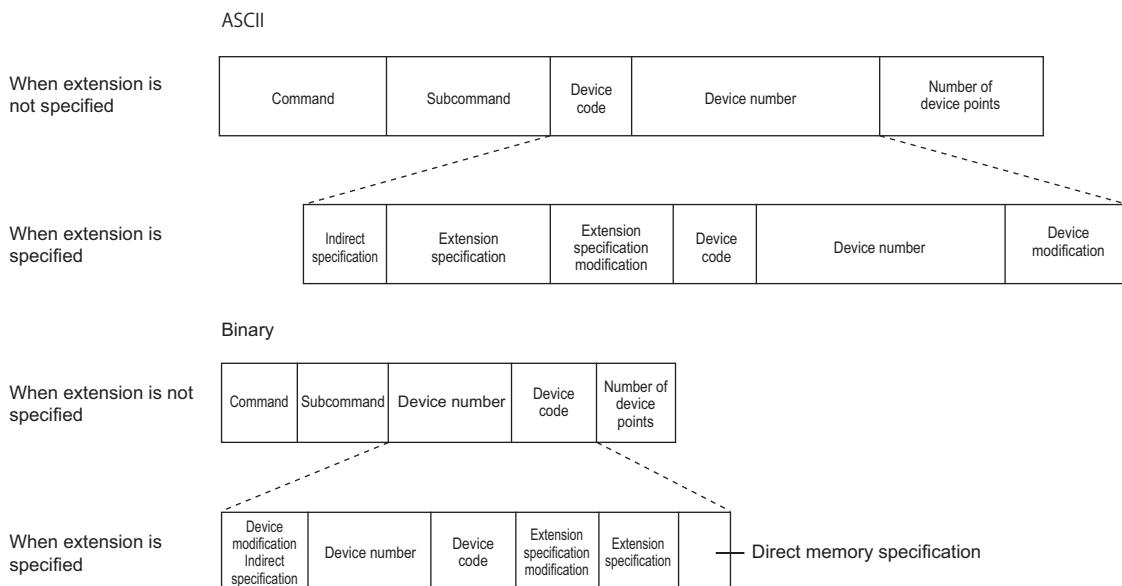
Appendix 1 Read/Write by Device Extension Specification

Accesses shown below are available by setting the subcommand in a request data to 008□ or 00C0.

- Accessing link direct devices
- Accessing module access devices
- Accessing CPU buffer memory access devices
- Accessing by indexing network No. and the start input/output number
- Accessing by indexing the device number
- Accessing by specifying the device number with the value stored in word device with indirect specification

Device specification method for device extension specification

Replace the device portion in a message format of each command with a message format for extension specification described in this section when the extension is specified.



With commands which can specify more than one device, the devices listed in the following section can be accessed by specifying '0' for "Extension specification", "Extension specification modification", "Direct memory specification".

☞ Page 66 Device code list

However, when 008□ or 00C0 is specified by "Subcommand", specify the device in the message format shown in this section. Message formats with extension not specified and message formats with extension specified cannot coexist in the same message.

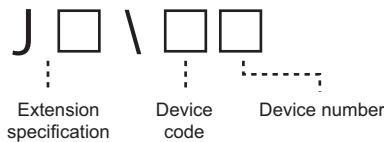
Subcommands for device extension specification

Use the following subcommands for extension specification.

| Item | ASCII code | Binary code | | | | | | | | | | |
|---|---|-------------|-----|---|-----|-----|-----|-----|---|---|-----|-----|
| For MELSEC-Q/ L series | When accessing in word units <table border="1"> <tr><td>0</td><td>0</td><td>8</td><td>0</td></tr> <tr><td>30H</td><td>30H</td><td>38H</td><td>30H</td></tr> </table> | 0 | 0 | 8 | 0 | 30H | 30H | 38H | 30H | <table border="1"> <tr><td>80H</td><td>00H</td></tr> </table> | 80H | 00H |
| 0 | 0 | 8 | 0 | | | | | | | | | |
| 30H | 30H | 38H | 30H | | | | | | | | | |
| 80H | 00H | | | | | | | | | | | |
| When accessing in bit units <table border="1"> <tr><td>0</td><td>0</td><td>8</td><td>1</td></tr> <tr><td>30H</td><td>30H</td><td>38H</td><td>31H</td></tr> </table> | 0 | 0 | 8 | 1 | 30H | 30H | 38H | 31H | <table border="1"> <tr><td>81H</td><td>00H</td></tr> </table> | 81H | 00H | |
| 0 | 0 | 8 | 1 | | | | | | | | | |
| 30H | 30H | 38H | 31H | | | | | | | | | |
| 81H | 00H | | | | | | | | | | | |
| When specifying monitor condition <table border="1"> <tr><td>0</td><td>0</td><td>C</td><td>0</td></tr> <tr><td>30H</td><td>30H</td><td>43H</td><td>30H</td></tr> </table> | 0 | 0 | C | 0 | 30H | 30H | 43H | 30H | <table border="1"> <tr><td>C0H</td><td>00H</td></tr> </table> | C0H | 00H | |
| 0 | 0 | C | 0 | | | | | | | | | |
| 30H | 30H | 43H | 30H | | | | | | | | | |
| C0H | 00H | | | | | | | | | | | |
| For MELSEC iQ-R series | When accessing in word units <table border="1"> <tr><td>0</td><td>0</td><td>8</td><td>2</td></tr> <tr><td>30H</td><td>30H</td><td>38H</td><td>32H</td></tr> </table> | 0 | 0 | 8 | 2 | 30H | 30H | 38H | 32H | <table border="1"> <tr><td>82H</td><td>00H</td></tr> </table> | 82H | 00H |
| 0 | 0 | 8 | 2 | | | | | | | | | |
| 30H | 30H | 38H | 32H | | | | | | | | | |
| 82H | 00H | | | | | | | | | | | |
| When accessing in bit units <table border="1"> <tr><td>0</td><td>0</td><td>8</td><td>3</td></tr> <tr><td>30H</td><td>30H</td><td>38H</td><td>33H</td></tr> </table> | 0 | 0 | 8 | 3 | 30H | 30H | 38H | 33H | <table border="1"> <tr><td>83H</td><td>00H</td></tr> </table> | 83H | 00H | |
| 0 | 0 | 8 | 3 | | | | | | | | | |
| 30H | 30H | 38H | 33H | | | | | | | | | |
| 83H | 00H | | | | | | | | | | | |

Accessing link direct devices

Access link devices of the network modules such as remote input (RX), remote output (RY), and link special relay (SB).



The following device access commands can be used for access.

| Function | Command | Subcommand |
|-----------------------------|---------|------------------------|
| Batch read | 0401 | 0080, 0081, 0082, 0083 |
| Batch write | 1401 | 0080, 0081, 0082, 0083 |
| Random read | 0403 | 0080, 0082, 00C0 |
| Random write | 1402 | 0080, 0081, 0082, 0083 |
| Batch read multiple blocks | 0406 | 0080, 0082 |
| Batch write multiple blocks | 1406 | 0080, 0082 |
| Register monitor data | 0801 | 0080, 0082, 00C0 |

Message format for device extension specification

■Data communication in ASCII code

| Subcommand type | ASCII code | | | | | | | | | | | | |
|-------------------------|--|----------------------------|----------------------------|-------------|---------------|----------------|----------|----------------------------|----------------------------|------------|--|------------|-------------|
| For MELSEC-Q/L series | <table border="1"> <tr> <td colspan="2">Extension specification</td> <td>Device code</td> <td>Device number</td> </tr> <tr> <td>0 0 30H 30H</td> <td>J 4AH</td> <td>0 0 0 30H 30H 30H</td> <td>0 0 0 30H 30H 30H</td> </tr> <tr> <td>(3 digits)</td> <td></td> <td>(2 digits)</td> <td>(6 digits)</td> </tr> </table> | Extension specification | | Device code | Device number | 0 0 30H 30H | J 4AH | 0 0 0 30H 30H 30H | 0 0 0 30H 30H 30H | (3 digits) | | (2 digits) | (6 digits) |
| Extension specification | | Device code | Device number | | | | | | | | | | |
| 0 0 30H 30H | J 4AH | 0 0 0 30H 30H 30H | 0 0 0 30H 30H 30H | | | | | | | | | | |
| (3 digits) | | (2 digits) | (6 digits) | | | | | | | | | | |
| For MELSEC iQ-R series | <table border="1"> <tr> <td colspan="2">Extension specification</td> <td>Device code</td> <td>Device number</td> </tr> <tr> <td>0 0 30H 30H</td> <td>J 4AH</td> <td>0 0 0 0 30H 30H 30H 30H</td> <td>0 0 0 0 30H 30H 30H 30H</td> </tr> <tr> <td>(4 digits)</td> <td></td> <td></td> <td>(10 digits)</td> </tr> </table> | Extension specification | | Device code | Device number | 0 0 30H 30H | J 4AH | 0 0 0 0 30H 30H 30H 30H | 0 0 0 0 30H 30H 30H 30H | (4 digits) | | | (10 digits) |
| Extension specification | | Device code | Device number | | | | | | | | | | |
| 0 0 30H 30H | J 4AH | 0 0 0 0 30H 30H 30H 30H | 0 0 0 0 30H 30H 30H 30H | | | | | | | | | | |
| (4 digits) | | | (10 digits) | | | | | | | | | | |

■Data communication in binary code

| Subcommand type | Binary code | | | | | | | | | | | | |
|------------------------|---|-------------------------|-------------------------|-------------------------|-------------------------|------------|------------|------------|-----------|-----------|--|----------|--|
| For MELSEC-Q/L series | <table border="1"> <tr> <td colspan="2">Device number</td> <td>Device code</td> <td>Extension specification</td> </tr> <tr> <td>00H 00H</td> <td></td> <td>00H 00H</td> <td>F9H</td> </tr> <tr> <td>(3 bytes)</td> <td></td> <td>(1 byte)</td> <td></td> </tr> </table> | Device number | | Device code | Extension specification | 00H 00H | | 00H 00H | F9H | (3 bytes) | | (1 byte) | |
| Device number | | Device code | Extension specification | | | | | | | | | | |
| 00H 00H | | 00H 00H | F9H | | | | | | | | | | |
| (3 bytes) | | (1 byte) | | | | | | | | | | | |
| For MELSEC iQ-R series | <table border="1"> <tr> <td>Device number</td> <td>Device code</td> <td>Extension specification</td> </tr> <tr> <td>00H 00H</td> <td></td> <td>00H 00H</td> <td>F9H</td> </tr> <tr> <td>(4 bytes)</td> <td>(2 bytes)</td> <td></td> <td></td> </tr> </table> | Device number | Device code | Extension specification | 00H 00H | | 00H 00H | F9H | (4 bytes) | (2 bytes) | | | |
| Device number | Device code | Extension specification | | | | | | | | | | | |
| 00H 00H | | 00H 00H | F9H | | | | | | | | | | |
| (4 bytes) | (2 bytes) | | | | | | | | | | | | |

Data to be specified

■Extension specification

Specify the access target network No. in hexadecimal.

- ASCII code: Specify J (4AH) at the head of data. Convert the numerical value to 3-digit ASCII code (hexadecimal), and send it from the upper digits.
- Binary code: Send 2-byte numerical values from the lower byte (L: bits 0 to 7).

Ex.

For network No.8

| ASCII code | Binary code |
|-------------------------------|-------------|
| J 0 0 8 4AH, 30H, 30H, 38H | 08H 00H |

Point

Using the index register of CPU module, the access target network No. can be specified indirectly.

☞ Page 440 Access to index the network No. and start input/output number

■Device code and device number

Specify the following devices. (☞ Page 63 Devices)

- Link input (X)
- Link output (Y)
- Link relay (B)
- Link special relay (SB)
- Link register (W)
- Link special register (SW)

For the values of device codes, refer to the following section.

☞ Page 66 Device code list

Point

Using the index register of CPU module, the access target device number can be specified indirectly.

☞ Page 441 Access to index the device number

Device extension specification example

Access W100 (J1\W100) of network No.1 by specifying subcommand 0080.

■Data communication in ASCII code

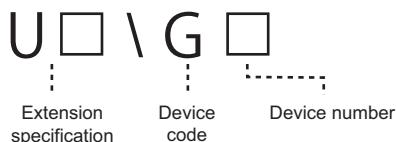
| Subcommand | Extension specification | | | Device code | Device number | | |
|-------------------------------|-------------------------|-------------------------------|------------------------|-------------|--|------------------------|--|
| 0 0 8 0 30H, 30H, 38H, 30H | 0 0 30H, 30H | J 0 0 1 4AH, 30H, 30H, 31H | 0 0 0 30H, 30H, 30H | W *57H, 2AH | 0 0 0 1 0 0 30H, 30H, 31H, 30H, 30H | 0 0 0 30H, 30H, 30H | |

■Data communication in binary code

| Subcommand | Device number | Device code | Extension specification |
|------------|---------------|----------------------|-------------------------|
| 80H, 00H | 00H, 00H | 00H, 01H, 00H B4H | 00H, 00H, 01H, 00H, F9H |

Accessing module access devices

Access buffer memory of an intelligent function module.



The following device access commands can be used for access.

| Function | Command | Subcommand |
|-----------------------------|---------|------------------|
| Batch read | 0401 | 0080, 0082 |
| Batch write | 1401 | 0080, 0082 |
| Random read | 0403 | 0080, 0082, 00C0 |
| Random write | 1402 | 0080, 0082 |
| Batch read multiple blocks | 0406 | 0080, 0082 |
| Batch write multiple blocks | 1406 | 0080, 0082 |
| Register monitor data | 0801 | 0080, 0082, 00C0 |

Message format for device extension specification

■Data communication in ASCII code

| Subcommand type | ASCII code | | | | | | | | | | | | | | | | |
|-------------------------|--|-------------------------|--------------------|-------------|---------------|-----|---|---------|---------|----------|-----|--------------------|--------------------|--|--|------------|-------------|
| For MELSEC-Q/L series | <table border="1"> <thead> <tr> <th colspan="2">Extension specification</th> <th>Device code</th> <th>Device number</th> </tr> </thead> <tbody> <tr> <td>0 0</td> <td>U</td> <td>0 0 0</td> <td>0 0 0</td> </tr> <tr> <td>30H, 30H</td> <td>55H</td> <td>30H, 30H, 30H</td> <td>30H, 30H, 30H</td> </tr> <tr> <td colspan="2"></td> <td>(2 digits)</td> <td>(6 digits)</td> </tr> </tbody> </table> | Extension specification | | Device code | Device number | 0 0 | U | 0 0 0 | 0 0 0 | 30H, 30H | 55H | 30H, 30H, 30H | 30H, 30H, 30H | | | (2 digits) | (6 digits) |
| Extension specification | | Device code | Device number | | | | | | | | | | | | | | |
| 0 0 | U | 0 0 0 | 0 0 0 | | | | | | | | | | | | | | |
| 30H, 30H | 55H | 30H, 30H, 30H | 30H, 30H, 30H | | | | | | | | | | | | | | |
| | | (2 digits) | (6 digits) | | | | | | | | | | | | | | |
| For MELSEC iQ-R series | <table border="1"> <thead> <tr> <th colspan="2">Extension specification</th> <th>Device code</th> <th>Device number</th> </tr> </thead> <tbody> <tr> <td>0 0</td> <td>U</td> <td>0 0 0 0</td> <td>0 0 0 0</td> </tr> <tr> <td>30H, 30H</td> <td>55H</td> <td>30H, 30H, 30H, 30H</td> <td>30H, 30H, 30H, 30H</td> </tr> <tr> <td colspan="2"></td> <td>(4 digits)</td> <td>(10 digits)</td> </tr> </tbody> </table> | Extension specification | | Device code | Device number | 0 0 | U | 0 0 0 0 | 0 0 0 0 | 30H, 30H | 55H | 30H, 30H, 30H, 30H | 30H, 30H, 30H, 30H | | | (4 digits) | (10 digits) |
| Extension specification | | Device code | Device number | | | | | | | | | | | | | | |
| 0 0 | U | 0 0 0 0 | 0 0 0 0 | | | | | | | | | | | | | | |
| 30H, 30H | 55H | 30H, 30H, 30H, 30H | 30H, 30H, 30H, 30H | | | | | | | | | | | | | | |
| | | (4 digits) | (10 digits) | | | | | | | | | | | | | | |

■Data communication in binary code

| Subcommand type | Binary code | | | | | | | | | |
|------------------------|--|-------------------------|-------------|-------------------------|----------|--|---------------|-----------|-----------|--|
| For MELSEC-Q/L series | <table border="1"> <thead> <tr> <th>Device number</th> <th>Device code</th> <th>Extension specification</th> </tr> </thead> <tbody> <tr> <td>00H, 00H</td> <td></td> <td>00H, 00H, F8H</td> </tr> <tr> <td>(3 bytes)</td> <td>(1 byte)</td> <td></td> </tr> </tbody> </table> | Device number | Device code | Extension specification | 00H, 00H | | 00H, 00H, F8H | (3 bytes) | (1 byte) | |
| Device number | Device code | Extension specification | | | | | | | | |
| 00H, 00H | | 00H, 00H, F8H | | | | | | | | |
| (3 bytes) | (1 byte) | | | | | | | | | |
| For MELSEC iQ-R series | <table border="1"> <thead> <tr> <th>Device number</th> <th>Device code</th> <th>Extension specification</th> </tr> </thead> <tbody> <tr> <td>00H, 00H</td> <td></td> <td>00H, 00H, F8H</td> </tr> <tr> <td>(4 bytes)</td> <td>(2 bytes)</td> <td></td> </tr> </tbody> </table> | Device number | Device code | Extension specification | 00H, 00H | | 00H, 00H, F8H | (4 bytes) | (2 bytes) | |
| Device number | Device code | Extension specification | | | | | | | | |
| 00H, 00H | | 00H, 00H, F8H | | | | | | | | |
| (4 bytes) | (2 bytes) | | | | | | | | | |

Data to be specified

■Extension specification

Specify the value obtained by dividing the start input/output number of an intelligent function module by 16 in hexadecimal.

- ASCII code: Specify U (55H) at the head of data. Convert the numerical value to 3-digit ASCII code (hexadecimal), and send it from the upper digits.
- Binary code: Send 2-byte numerical values from the lower byte (L: bits 0 to 7).

Ex.

Start input/output number 0010H of a module

| ASCII code | Binary code | | | | | | | | | | |
|---|-------------|-----|-----|---|-----|-----|-----|-----|--|-----|-----|
| <table border="1"> <tr> <td>U</td> <td>0</td> <td>0</td> <td>1</td> </tr> <tr> <td>55H</td> <td>30H</td> <td>30H</td> <td>31H</td> </tr> </table> | U | 0 | 0 | 1 | 55H | 30H | 30H | 31H | <table border="1"> <tr> <td>01H</td> <td>00H</td> </tr> </table> | 01H | 00H |
| U | 0 | 0 | 1 | | | | | | | | |
| 55H | 30H | 30H | 31H | | | | | | | | |
| 01H | 00H | | | | | | | | | | |

Point

Indirect specification of the start input/output number can also be performed by using the CPU module index register.

☞ Page 440 Access to index the network No. and start input/output number

■Device code and device number

Specify the following device. (☞ Page 63 Devices)

- Module access device (G)

For the values of device codes, refer to the following section.

☞ Page 66 Device code list

Point

Using the index register of CPU module, the access target device number can be specified indirectly.

☞ Page 441 Access to index the device number

Device extension specification example

Access the buffer memory (address: 1) of intelligent function module whose start input/output number is 0030H by specifying subcommand 0080.

■Data communication in ASCII code

| Subcommand | Extension specification | Device code | Device number |
|-------------------------------|-------------------------|-------------------------------|------------------------|
| 0 0 8 0 30H, 30H, 38H, 30H | 0 0 30H, 30H | U 0 0 3 55H, 30H, 30H, 33H | 0 0 0 30H, 30H, 30H |

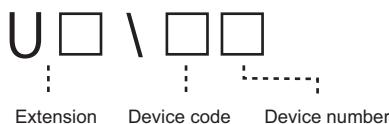
G * 0 0 0 0 0 1 0 0 0
47H, 2AH 30H, 30H, 30H, 30H, 31H 30H, 30H, 30H

■Data communication in binary code

| Subcommand | Device number | Device code | Extension specification |
|------------|---------------|---------------|------------------------------|
| 80H, 00H | 00H, 00H | 01H, 00H, 00H | ABH, 00H, 00H, 03H, 00H, F8H |

Accessing CPU buffer memory access device

Access the buffer memory of RCPU.



The following device access commands can be used for access.

| Function | Command | Subcommand |
|-----------------------------|---------|------------|
| Batch read | 0401 | 0082 |
| Batch write | 1401 | 0082 |
| Random read | 0403 | 0082 |
| Random write | 1402 | 0082 |
| Batch read multiple blocks | 0406 | 0082 |
| Batch write multiple blocks | 1406 | 0082 |
| Register monitor data | 0801 | 0082 |

Message format for device extension specification

■Data communication in ASCII code

| Subcommand type | ASCII code | | | | | | | | | | | | | | | | | | |
|-------------------------|--|-------------------------|------------|-------------|-----------------|-------------|---------------|-----|-------|---------|------------|--|---------|---------|-------------|-----------------|--|--|-----------------|
| For MELSEC iQ-R series | <table><thead><tr><th colspan="4">Extension specification</th><th>Device code</th><th>Device number</th></tr></thead><tbody><tr><td>0 0</td><td>U 3 E</td><td>0 0 0 0</td><td>(4 digits)</td><td></td><td>0 0 0 0</td></tr><tr><td>30H 30H</td><td>55H 33H 45H</td><td>30H 30H 30H 30H</td><td></td><td></td><td>30H 30H 30H 30H</td></tr></tbody></table> | Extension specification | | | | Device code | Device number | 0 0 | U 3 E | 0 0 0 0 | (4 digits) | | 0 0 0 0 | 30H 30H | 55H 33H 45H | 30H 30H 30H 30H | | | 30H 30H 30H 30H |
| Extension specification | | | | Device code | Device number | | | | | | | | | | | | | | |
| 0 0 | U 3 E | 0 0 0 0 | (4 digits) | | 0 0 0 0 | | | | | | | | | | | | | | |
| 30H 30H | 55H 33H 45H | 30H 30H 30H 30H | | | 30H 30H 30H 30H | | | | | | | | | | | | | | |

■Data communication in binary code

| Subcommand type | Binary code | | | | | | | | | | | | | | | | | | |
|------------------------|---|---------------|-------------------------|-------------|-------------------------|--|--|-----|-----|--|-----|-----|-----|-----------|-----------|--|--|--|--|
| For MELSEC iQ-R series | <table><thead><tr><th colspan="2">Device number</th><th>Device code</th><th colspan="3">Extension specification</th></tr></thead><tbody><tr><td>00H</td><td>00H</td><td></td><td>00H</td><td>00H</td><td>FAH</td></tr><tr><td>(4 bytes)</td><td>(2 bytes)</td><td></td><td></td><td></td><td></td></tr></tbody></table> | Device number | | Device code | Extension specification | | | 00H | 00H | | 00H | 00H | FAH | (4 bytes) | (2 bytes) | | | | |
| Device number | | Device code | Extension specification | | | | | | | | | | | | | | | | |
| 00H | 00H | | 00H | 00H | FAH | | | | | | | | | | | | | | |
| (4 bytes) | (2 bytes) | | | | | | | | | | | | | | | | | | |

Data to be specified

■Extension specification

Specify the start input/output number of CPU module with the upper 3 digits of 4-digit in hexadecimal.

- ASCII code: Specify U (55H) at the head of data. Convert the numerical value to 3-digit ASCII code (hexadecimal), and send it from the upper digits.
- Binary code: Send 2-byte numerical values from the lower byte (L: bits 0 to 7).

The start input/output numbers of CPU modules to be specified are as shown below.

| CPU number | Start input/output number of CPU module | Value to be specified |
|------------|---|-----------------------|
| CPU No.1 | 3E00H | 3E0H |
| CPU No.2 | 3E10H | 3E1H |
| CPU No.3 | 3E20H | 3E2H |
| CPU No.4 | 3E30H | 3E3H |

Ex.

For CPU No.4 (start input/output number: 3E30H)

| ASCII code | Binary code |
|----------------------------|-------------|
| U 3 E 3 55H 33H 45H 33H | E3H 03H |

Point

Indirect specification of the start input/output number of the CPU module can also be performed by using the CPU module index register.

☞ Page 440 Access to index the network No. and start input/output number

■Device code and device number

Specify the following devices. (☞ Page 63 Devices)

- CPU buffer memory access device (G)
- CPU buffer memory access device (HG)

For the values of device codes, refer to the following section.

☞ Page 66 Device code list

Point

Using the index register of CPU module, the access target device number can be specified indirectly.

☞ Page 441 Access to index the device number

Device extension specification example

Access the buffer memory (address: 1) of the CPU No.1 (start input/output number: 03E0H) by specifying the subcommand 0082.

■Data communication in ASCII code

| Subcommand | Extension specification | | | | Device code | Device number | | | | |
|----------------------------|-------------------------|----------------------------|----------------------------|----------------------------|--|--|--|--|--|--|
| 0 0 8 2 30H 30H 38H 32H | 0 0 30H 30H | U 3 E 0 55H 33H 45E 30H | 0 0 0 0 30H 30H 30H 30H | G * * * 47H 2AH 2AH 2AH | 0 0 0 0 0 0 0 0 1 30H 30H 30H 30H 30H 30H 30H 31H | 0 0 0 0 0 0 0 0 0 30H 30H 30H 30H 30H 30H 30H 30H | | | | |

■Data communication in binary code

| Subcommand | Device number | Device code | Extension specification |
|------------|---------------|-----------------|-----------------------------|
| 82H 00H | 00H 00H | 01H 00H 00H 00H | ABH 00H 00H 00H E0H 03H FAH |

Access for index modification

Index modification is the indirect specification using index registers.

Accesses shown below are available by setting the subcommand in a request data to 008□ or 00C0.

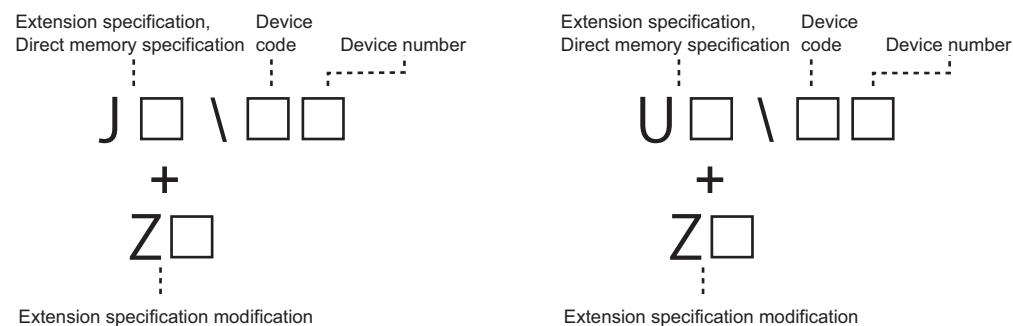
| Target data | Reference |
|--|--|
| Network No. or start input/output number | Page 440 Access to index the network No. and start input/output number |
| Device number | Page 441 Access to index the device number |

The following device access commands can be used for access.

| Function | Command | Subcommand |
|-----------------------|---------|------------------------|
| Random read | 0403 | 0080, 0082, 00C0 |
| Random write | 1402 | 0080, 0081, 0082, 0083 |
| Register monitor data | 0801 | 0080, 0082, 00C0 |

Access to index the network No. and start input/output number

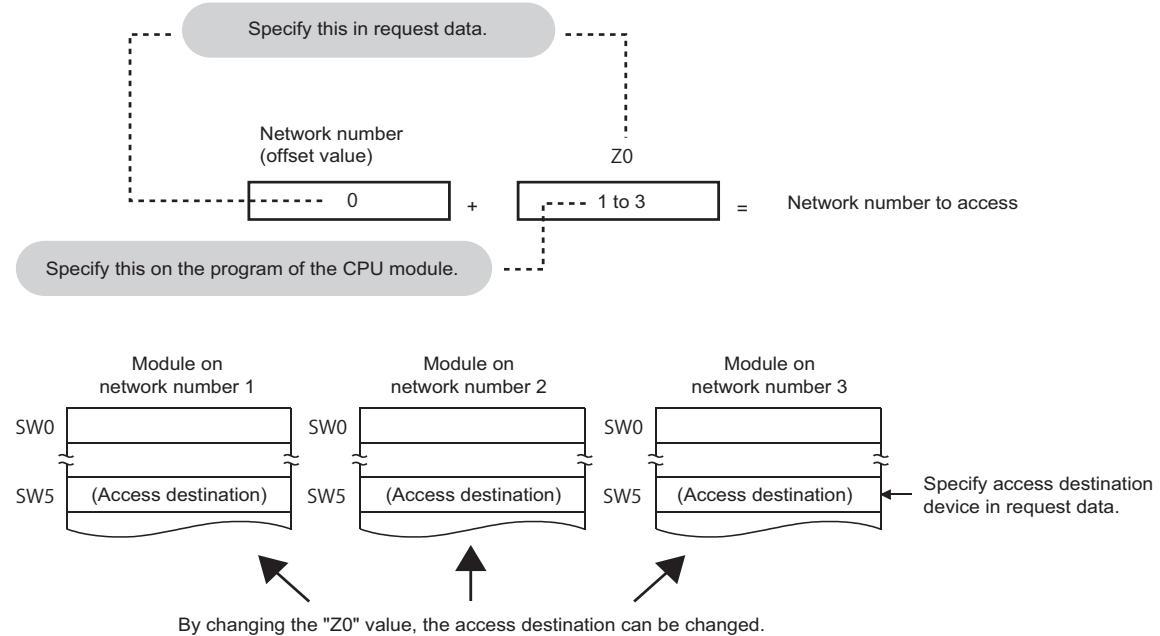
When accessing a link direct device, the access target network No. can be specified by the index register with indirect specification. Also, when accessing a module access device or a CPU buffer memory access device, the access target start input/output number can be specified by the index register with indirect specification.



The access target can be switched with one message, by changing the value of the index register in CPU module programs.

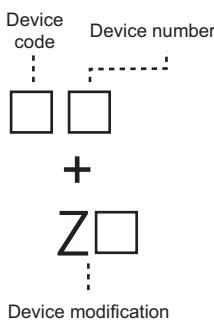
Ex.

The access target can be switched by changing the value of 'Z0', when multiple network modules are mounted onto the access target.



Access to index the device number

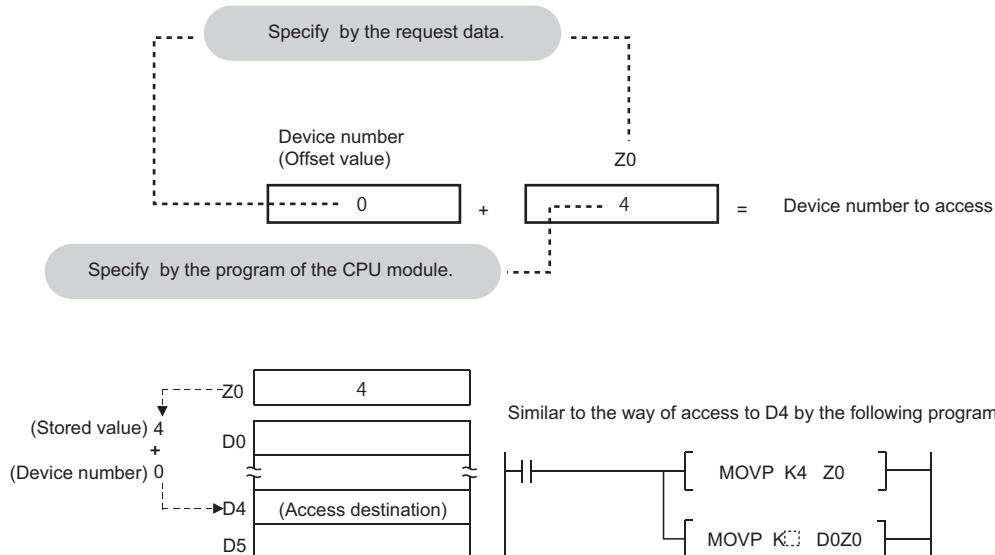
When accessing a device, indirect specification of the device number can be performed using index register.



The access target can be switched with one message, by changing the value of the index register in CPU module programs.

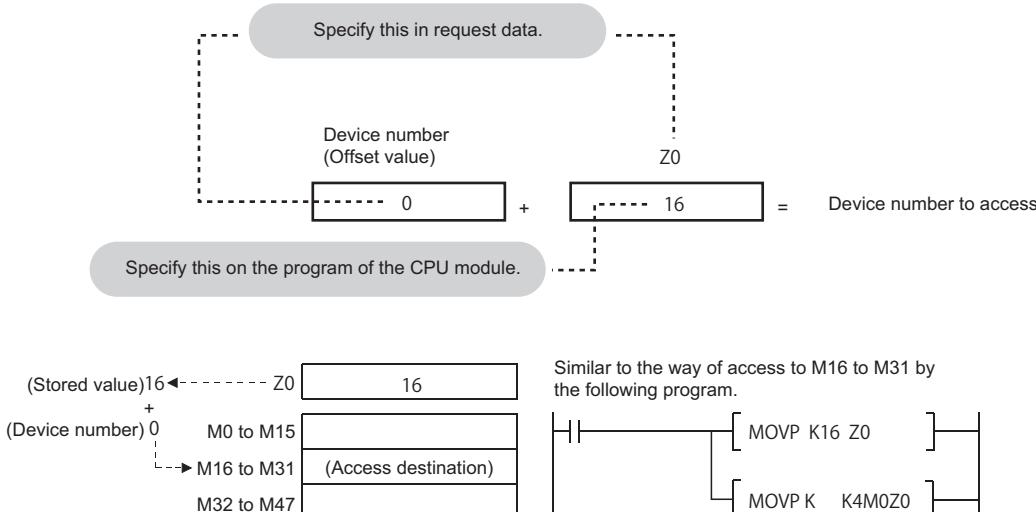
Ex.

When accessing D4 by specifying D0 and Z0



Ex.

When accessing M16 to M31 by specifying M0 and Z0 (Word units)



Message format for device extension specification

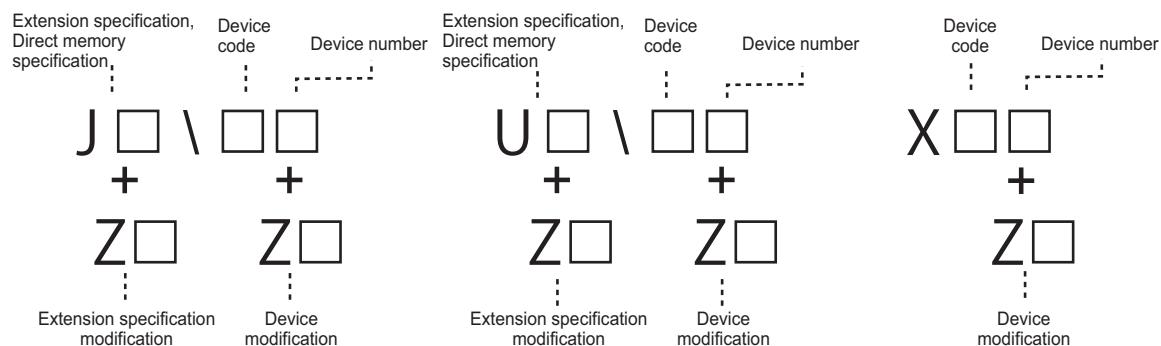
■Data communication in ASCII code

| | | | | | |
|----------------|-----------------------------|--|-----------------|-------------------|-------------------------|
| 0 0 30H 30H | Extension specification | Extension specification modification | Device code | Device number | Device modification |
|----------------|-----------------------------|--|-----------------|-------------------|-------------------------|

■Data communication in binary code

| | | | | | |
|---------|-------------------|-----------------|--|-----------------------------|---------------------------------|
| 00H 00H | Device number | Device code | Extension specification modification | Extension specification | Direct memory specification |
|---------|-------------------|-----------------|--|-----------------------------|---------------------------------|

The following shows the correspondence of each data.



Data to be specified

■Extension specification, device code, and device number

Specify the access target network No. and the offset value of the start input/output number for extension specification.

When indexing a network No. and start input/output number by "Extension specification modification", the values specified by "Extension specification" will be the offset value.

When indexing a device number by "Device modification", the values specified by "Device number" will be the offset value.

For the data to be specified for each access device, refer to the following table.

| Item | Reference |
|---------------------------------|-------------------------------|
| Link direct device | Page 435 Data to be specified |
| Module access device | Page 437 Data to be specified |
| CPU buffer memory access device | Page 438 Data to be specified |

■Direct memory specification (only when communicating in binary code)

Specify the type of access device.

| Item | Value to be specified |
|---------------------------------|-----------------------|
| Link direct device | F9H |
| Module access device | F8H |
| CPU buffer memory access device | FAH |

Specify '0' when accessing devices other than link direct device, module access device, and CPU buffer memory access device.

Binary code

Specify '0'.

| |
|---------|
| 00H 00H |
|---------|

■Extension specification modification

Specify the index register to be used when indexing the value specified by 'Extension specification'.

| Access target | Devices to be used | ASCII code | | Binary code | |
|---------------------------|---------------------|--|---|---|-------------|
| | | Device code | Device number | Device number | Fixed value |
| MELSEC iQ-R series module | Index register | Specify 'Z' and space with 2-digit ASCII code. | Specify the device number in decimal (2-digit ASCII code). (Specification range: 0 to 24) | Specify in hexadecimal. (Specification range: 00H to 18H) | 40H |
| | Long index register | Specify 'LZ' with 2-digit ASCII code. | Specify the device number in decimal (2-digit ASCII code). (Specification range: 0 to 12) | Specify in hexadecimal. (Specification range: 00H to 0CH) | 80H |
| MELSEC-Q/L series module | Index register | Specify 'Z' with 1-digit ASCII code. | Specify the device number in decimal (2-digit ASCII code). (Specification range: 0 to 15) | Specify in hexadecimal. (Specification range: 0 to F) | 40H |

Ex.

Index register (Z0)

| Subcommand type | ASCII code | Binary code |
|------------------------|--------------------------|-------------|
| For MELSEC-Q/L series | Z 0 0 5AH 30H 30H | 00H 40H |
| For MELSEC iQ-R series | Z 0 0 5AH 20H 30H 30H | 00H 40H |

Ex.

Long index register (LZ0)

| Subcommand type | ASCII code | Binary code |
|------------------------|----------------------------|-------------|
| For MELSEC iQ-R series | L Z 0 0 4CH 5AH 30H 30H | 00H 80H |

When do not perform index modification, specify '0'.

| Subcommand type | ASCII code | Binary code |
|------------------------|----------------------------|-------------|
| For MELSEC-Q/L series | 0 0 0 30H 30H 30H | 00H 00H |
| For MELSEC iQ-R series | 0 0 0 0 30H 30H 30H 30H | 00H 00H |

■Device modification

Specify the number of index register when indexing device number.

The data to be specified is the same as extension specification modification. ([Page 443 Extension specification modification](#))

Communication example (indexing network No.)

Access W100 (J1Z0\W100) of network No.1 + Z0 by specifying subcommand 0080.

■Data communication in ASCII code

| Subcommand | Extension specification | Extension specification modification | Device code | Device number |
|-------------------------------|-------------------------|--------------------------------------|------------------------|-----------------|
| 0 0 8 0 30H, 30H, 38H, 30H | 0 0 30H, 30H | J 0 0 1 4AH, 30H, 30H, 31H | Z 0 0 5AH, 30H, 30H | W * 57H, 2AH |

0 0 0 1 0 0 0 0 0 0 0 0
30H, 30H, 30H, 31H, 30H, 30H, 30H, 30H, 30H, 30H, 30H, 30H

■Data communication in binary code

| Subcom mand | Device number | Device code | Extension specification | Extension modification | Direct memory specification |
|----------------|------------------|----------------|----------------------------|---------------------------|--------------------------------|
| 80H, 00H | 00H, 00H | 00H, 01H, 00H | B4H | 00H, 40H | 01H, 00H, F9H |

Device extension specification example (When indexing the device number)

Access D100Z4 by specifying subcommand 0080.

■Data communication in ASCII code

| Subcommand | Extension specification | Extension specification modification | Device code | Device number | Device modification |
|-------------------------------|-------------------------------|--------------------------------------|-----------------|---|------------------------|
| 0 0 8 0 30H, 30H, 38H, 30H | 0 0 0 0 30H, 30H, 30H, 30H | 0 0 0 30H, 30H, 30H | D * 44H, 2AH | 0 0 0 1 0 0 30H, 30H, 30H, 31H, 30H, 30H | Z 0 4 5AH, 30H, 34H |

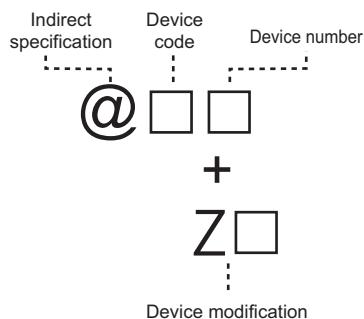
■Data communication in binary code

| Subcom mand | Device modification | Device number | Device code | Extension specification | Extension modification | Direct memory specification |
|----------------|------------------------|------------------|----------------|----------------------------|---------------------------|--------------------------------|
| 80H, 00H | 04H, 40H | 64H, 00H, 00H | A8H | 00H, 00H | 00H, 00H | 00H |

Accessing devices for indirect specification

Access the device corresponding to the address stored in the word device (2 points).

Store the address of the access target device in the device for the indirect specification, and represent the data as '@ + device for indirect specification'.



Point

For details on the indirect specification, refer to the following manual.

MELSEC iQ-R CPU Module User's Manual (Application)

The following device access commands can be used for access.

| Function | Command | Subcommand |
|-----------------------|---------|------------------|
| Random read | 0403 | 0080, 0082, 00C0 |
| Random write | 1402 | 0080, 0082 |
| Register monitor data | 0801 | 0080, 0082, 00C0 |

Message format for device extension specification

■Data communication in ASCII code

| Subcommand type | ASCII code | | | | | | | | | | | | | | | | | | | | | | |
|------------------------|---|-----|-----|-----|-----|-----|-----|-------------|---------------|---------------------|---------------|---------------------|-----|-----|-----|-----|-----|-----|-----|-----|--|--|--|
| For MELSEC-Q/L series | <table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td>0</td> <td>@</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>Device code</td> <td>Device number</td> <td>Device modification</td> </tr> <tr> <td>30H</td> <td>40H</td> <td>30H</td> <td>30H</td> <td>30H</td> <td>30H</td> <td>30H</td> <td>30H</td> <td></td> <td></td> <td></td> </tr> </table> | 0 | @ | 0 | 0 | 0 | 0 | 0 | 0 | Device code | Device number | Device modification | 30H | 40H | 30H | 30H | 30H | 30H | 30H | 30H | | | |
| 0 | @ | 0 | 0 | 0 | 0 | 0 | 0 | Device code | Device number | Device modification | | | | | | | | | | | | | |
| 30H | 40H | 30H | 30H | 30H | 30H | 30H | 30H | | | | | | | | | | | | | | | | |
| For MELSEC iQ-R series | <table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td>0</td> <td>@</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>Device code</td> <td>Device number</td> <td>Device modification</td> </tr> <tr> <td>30H</td> <td>40H</td> <td>30H</td> <td>30H</td> <td>30H</td> <td>30H</td> <td>30H</td> <td>30H</td> <td></td> <td></td> <td></td> </tr> </table> | 0 | @ | 0 | 0 | 0 | 0 | 0 | 0 | Device code | Device number | Device modification | 30H | 40H | 30H | 30H | 30H | 30H | 30H | 30H | | | |
| 0 | @ | 0 | 0 | 0 | 0 | 0 | 0 | Device code | Device number | Device modification | | | | | | | | | | | | | |
| 30H | 40H | 30H | 30H | 30H | 30H | 30H | 30H | | | | | | | | | | | | | | | | |

■Data communication in binary code

| Subcommand type | Binary code | | | | | | | | | | | | | | | | | | |
|------------------------|--|---------------|-------------|-----|-----|--|--|-----|-----|-----|-----|-----|-----|------------------------|---------------------|--|--|--|--|
| For MELSEC-Q/L series | <table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td>Device number</td> <td>Device code</td> <td> </td> <td> </td> <td> </td> <td> </td> </tr> <tr> <td>00H</td> <td>00H</td> <td>00H</td> <td>00H</td> <td>00H</td> <td>00H</td> </tr> <tr> <td>Indirect specification</td> <td>Device modification</td> <td colspan="4"></td> </tr> </table> | Device number | Device code | | | | | 00H | 00H | 00H | 00H | 00H | 00H | Indirect specification | Device modification | | | | |
| Device number | Device code | | | | | | | | | | | | | | | | | | |
| 00H | 00H | 00H | 00H | 00H | 00H | | | | | | | | | | | | | | |
| Indirect specification | Device modification | | | | | | | | | | | | | | | | | | |
| For MELSEC iQ-R series | <table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td>Device number</td> <td>Device code</td> <td> </td> <td> </td> <td> </td> <td> </td> </tr> <tr> <td>00H</td> <td>00H</td> <td>00H</td> <td>00H</td> <td>00H</td> <td>00H</td> </tr> <tr> <td>Indirect specification</td> <td>Device modification</td> <td colspan="4"></td> </tr> </table> | Device number | Device code | | | | | 00H | 00H | 00H | 00H | 00H | 00H | Indirect specification | Device modification | | | | |
| Device number | Device code | | | | | | | | | | | | | | | | | | |
| 00H | 00H | 00H | 00H | 00H | 00H | | | | | | | | | | | | | | |
| Indirect specification | Device modification | | | | | | | | | | | | | | | | | | |

Data to be specified

■Device code and device number

Specify the devices for indirect specification. (☞ Page 63 Devices)

Indirect specification can be used only for word devices.

When indexing a device number by "Device modification", the values specified by "Device number" will be the offset value.

For the values of device codes, refer to the following section.

☞ Page 66 Device code list

■Device modification

Specify the number of index register when the device number is indirectly specified with index register.

| Access target | Devices to be used | ASCII code | | Binary code | |
|---------------------------|---------------------|--|---|---|-------------|
| | | Device code | Device number | Device number | Fixed value |
| MELSEC iQ-R series module | Index register | Specify 'Z' and space with 2-digit ASCII code. | Specify the device number in decimal (2-digit ASCII code). (Specification range: 0 to 24) | Specify in hexadecimal. (Specification range: 00H to 18H) | 48H |
| | Long index register | Specify 'LZ' with 2-digit ASCII code. | Specify the device number in decimal (2-digit ASCII code). (Specification range: 0 to 12) | Specify in hexadecimal. (Specification range: 00H to 0CH) | 88H |
| MELSEC-Q/L series module | Index register | Specify 'Z' with 1-digit ASCII code. | Specify the device number in decimal (2-digit ASCII code). (Specification range: 0 to 15) | Specify in hexadecimal. (Specification range: 0 to F) | 48H |

Ex.

Index register (Z0)

| Subcommand type | ASCII code | Binary code |
|------------------------|--------------------------|-------------|
| For MELSEC-Q/L series | Z 0 0 5AH 30H 30H | 00H 48H |
| For MELSEC iQ-R series | Z 0 0 5AH 20H 30H 30H | 00H 48H |

Ex.

For long index register (LZ0)

| Subcommand type | ASCII code | Binary code |
|------------------------|----------------------------|-------------|
| For MELSEC iQ-R series | L Z 0 0 4CH 5AH 30H 30H | 00H 88H |

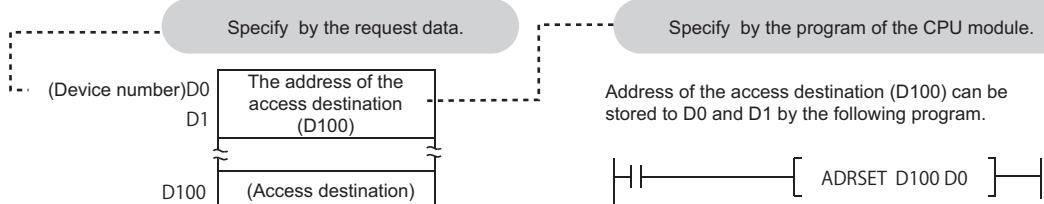
When do not perform index modification, specify the following.

| Subcommand type | ASCII code | Binary code |
|------------------------|----------------------------|-------------|
| For MELSEC-Q/L series | 0 0 0 30H 30H 30H | 00H 08H |
| For MELSEC iQ-R series | 0 0 0 0 30H 30H 30H 30H | 00H 08H |

Device extension specification example (when do not perform index modification)

Access D100 by specifying the subcommand 0080 and using devices for indirect specification '@D0'.

Store the address of D100 in D0 with a program before executing the command.



Data communication in ASCII code

| Subcommand | Indirect specification | | | | Device code | Device number | Device modification |
|------------|------------------------|---------|-------|-----|-------------|---------------|---------------------|
| 0 0 8 0 | 0 @ | 0 0 0 0 | 0 0 0 | D * | 0 0 0 0 0 0 | 0 0 0 | 0 0 0 |

Hex values: 30H, 30H, 38H, 30H | 30H, 40H | 30H, 30H, 30H, 30H | 30H, 30H, 30H | 44H, 2AH | 30H, 30H, 30H, 30H, 30H, 30H | 30H, 30H, 30H | 30H, 30H, 30H

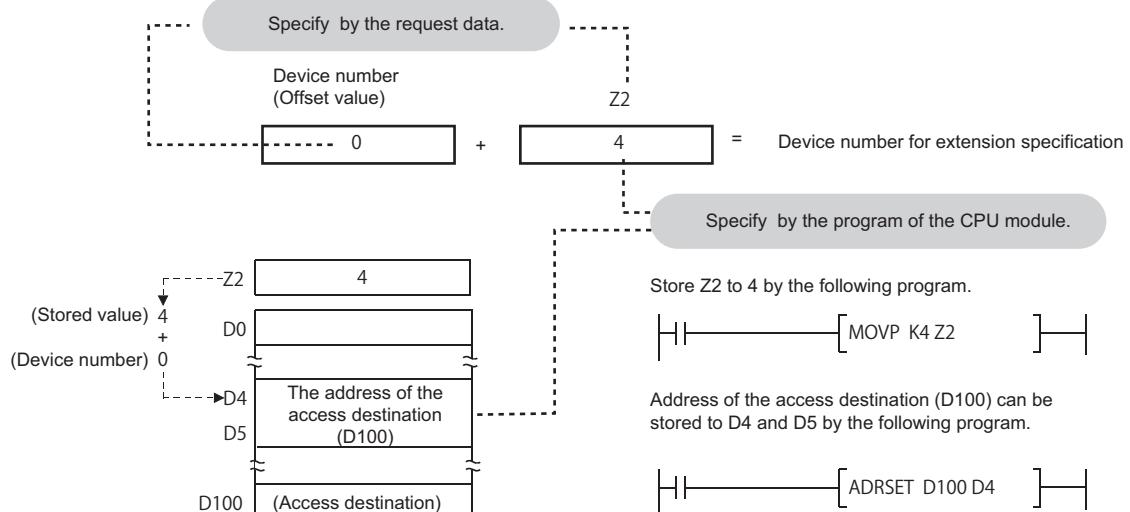
Data communication in binary code

| Subcommand | Device modification | Device | Device |
|------------|------------------------|---------------|--------|
| | Indirect specification | number | code |
| 80H, 00H | 00H, 08H | 00H, 00H, 00H | A8H |

Device extension specification example (Index modification)

Access D100 by specifying subcommand 0080 and specify the index modified device for indirect specification @D0Z2.

Before executing the command, store the D100 address in D4 and '4' in Z2 with the program.



Data communication in ASCII code

| Subcommand | Indirect specification | | | | Device code | Device number | Device modification |
|------------|------------------------|---------|-------|-----|-------------|---------------|---------------------|
| 0 0 8 0 | 0 @ | 0 0 0 0 | 0 0 0 | D * | 0 0 0 0 0 0 | Z 0 2 | 5AH, 30H, 32H |

Hex values: 30H, 30H, 38H, 30H | 30H, 40H | 30H, 30H, 30H, 30H | 30H, 30H, 30H | 44H, 2AH | 30H, 30H, 30H, 30H, 30H, 30H | 30H, 30H, 30H | 5AH, 30H, 32H

Data communication in binary code

| Subcommand | Device modification | Device | Device |
|------------|------------------------|---------------|--------|
| | Indirect specification | number | code |
| 80H, 00H | 02H, 48H | 00H, 00H, 00H | A8H |

Appendix 2 Processing Time

Time chart and communication time of C24 transmission sequence

The following figure shows a communication time chart when accessing programmable controller from an external device via C24.

When accessing CPU module with C24 mounted from an external device

The following shows the time chart when the message wait time is specified.

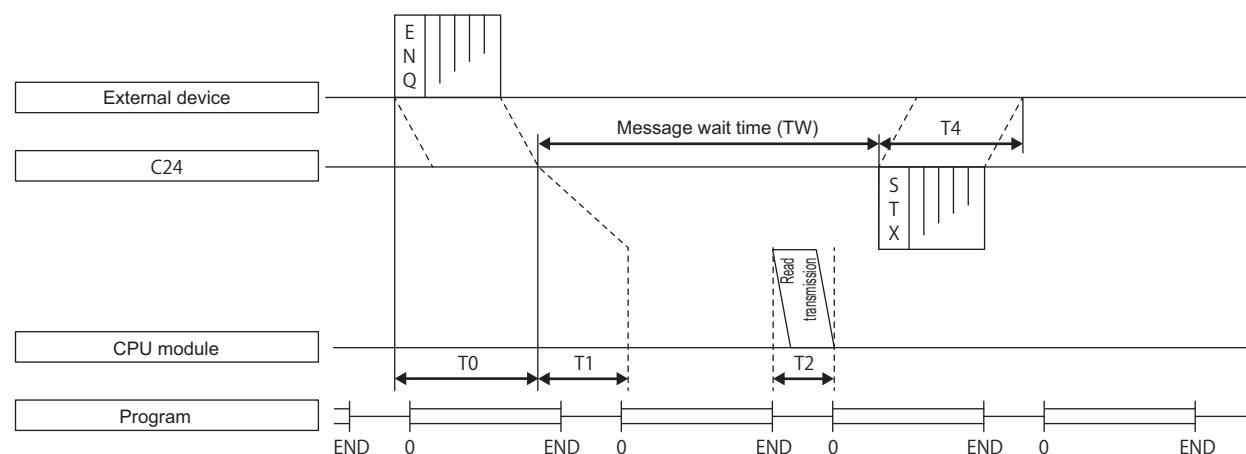
For the number of scans required for processing a read/write request, refer to the following section.

☞ Page 453 Number of scans required for processing

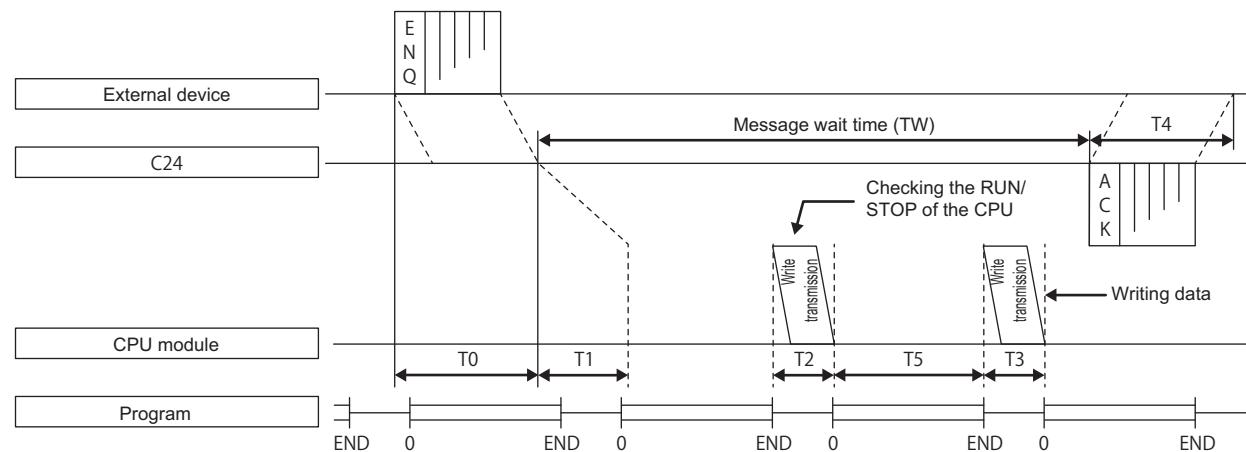
For the number of points processed per communication, refer to the following section.

☞ Page 459 Number of Processing per One Communication

■Reading data



■Writing data



- The communication between C24 and CPU module is always performed after END. Therefore, the scan time becomes longer according to the communication time (the time to interrupt to the CPU module).
- The number of scans required for processing to request read/write operation to the CPU module differs depending on the content of the request. For example, when a read operation that requires a scan is requested, it takes an extra time of 1 scan + T2.
- The number of scans required for the processing varies depending on the Service Processing Setting. It can be set in the parameter settings of Engineering tool. For details of Service Processing Setting, refer to the user's manual of the CPU module used.

Transmission time of transmission sequence

The following explains the approximate time required from when external device starts data transmission to when C24 returns the result.

The average of the above approximate time is shown below for the estimation of the processing time.

■When external device reads data from a programmable controller (unit: ms)

Communication time = Request message transmission time (T0) + [$(T1 + (1.5 \times \text{scan time}) + T2)$ or TW] + Response message transmission time (T4)
 (When the value exceeds TW)

■When external device writes data to a programmable controller

Communication time = Request message transmission time (T0) + [$(T1 + (1.5 \times \text{scan time}) + T2 + T3 + T5)$ or TW] + Response message transmission time (T4)
 (When the value exceeds TW)

$$T0, T4 = \frac{1000}{\text{Transmission rate}} \times \text{number of bits for 1 byte at transmission} \left(1 + \frac{7}{8} + 0/1 + \frac{1}{2} \right) \times \text{number of bytes}$$

T1 = C24 processing time
 Depending on the access points (MIN to MAX), the following ranges.

| | |
|---------------------------------|-------------------|
| RJ71C24(-R2/-R4) | : 12.0 to 45.0 ms |
| QJ71C24N(-R2/-R4), LJ71C24(-R2) | : 12.5 to 45.0 ms |
| QJ71C24(-R2) | : 13.0 to 70.0 ms |

T2, T3 = CPU intervention time : 0.07 to 13.1ms

T5 = Scan time (For functions that can process one scan, T3 and T5 equal 0.)

TW = The set time when message wait time is set.

Transmission time when accessing other stations

This section explains the transmission time (T1) when accessing CC-Link IE, MELSECNET/H, or MELSECNET/10 on other station.



For details on the network systems, refer to the manual of each network system.

Transmission time (T1) to access other station (CC-Link IE, PLC network)

The following formula is for calculating the transmission time of CC-Link IE, MELSECNET/H (PLC network), and MELSECNET/10 (PLC network).

$$\text{Transmission time (T1)} = \frac{\text{Transmission delay time} + \underset{*1}{\text{1 scan time of the station on which the C24/E71 is loaded or internal processing time}} \times \underset{*2}{(n+1)}}{\underset{*3}{(\text{When this value is more than the internal processing time.})}}$$

Internal processing time C24 : Approximately 50 ms, E71 : Approximately 30 ms

*1 Refer to the transmission delay time in the manual of the network system.

*2 n=6

- When performing the initial communication to the relevant station after powering ON or resetting CPU
- When performing communication to stations excluding the currently communicated 16 stations

n=1

- When performing the second communication when the number of communication stations is 16 or less
- When performing the second communication to the 16 stations communicated last

*3 When online program change is disabled in C24/E71, '1' is added only when writing data from external device.
(Set "Online change" with the "Switch setting" or "Parameter setting" of Engineering tool.)

■Transmission time calculation example

The following example shows when C24 is mounted on the station on CC-Link IE controller network, and the device memory of other station on the same network is read. (When the number of communication station is 8 stations, and the second communication is performed)

● Processing times and settings

- S_T : Scan time of transmission side 2 ms
- S_R : Scan time of reception side 1 ms
- LS : Link scan time 2 ms
- α_T : Link refresh time of transmission side 1 ms
- α_R : Link refresh time of reception side 1 ms
- Simultaneous transient request: 2
- Maximum transient request: 1

Ex.

Normal transmission delay time

$$\text{Transmission time (T1)} = \{(2 + 1 + 1 + 1) \times 2 + \underset{(72\text{ms})}{2 \times 4} + \underset{(S_T \cdot \alpha_T \cdot S_R \cdot \alpha_R)}{(2 / 1 - 1) \times 2 \times 2} + \underset{(\text{LS})}{50}\} \times 1$$

Simultaneous transient request
Maximum transient request
C24 internal processing time

Ex.

Worst-case transmission delay time

$$\text{Transmission time (T1)} = \{(2 + 1 + 1 + 1) \times 2 + \underset{(76\text{ms})}{2 \times 6} + \underset{(S_T \cdot \alpha_T \cdot S_R \cdot \alpha_R)}{(2 / 1 - 1) \times 2 \times 2} + \underset{(\text{LS})}{50}\} \times 1$$

Simultaneous transient request
Maximum transient request
C24 internal processing time

■Causes of transmission time (T1) delay

When a command that takes two scans to send is executed, the transmission time is twice the value obtained by the formula above.

When the access request is issued to the same station from multiple external devices simultaneously, refer to the precautions for data communication. (☞ Page 21 When accessing CPU module)

Point

Data transmission to the CPU module other than connected station (host station) on CC-Link IE, MELSECNET/H, or MELSECNET/10 may be delayed considerably depending on conditions. The transmission delay time can be shortened by limiting communication between an external device and the CPU module to the station with C24/E71 mounted, and communicating with programmable controllers on other stations by data link (LB, LW).

Transmission time (T_1) to access other station (remote I/O net)

The following shows the transmission time for MELSECNET/H(remote I/O net) and MELSECNET/10(remote I/O net).



MELSEC iQ-R series and MELSEC-L series are not supported.

Internal processing time C24 : Approximately 50 ms, E71 : Approximately 30 ms

- *1 Refer to the transmission delay time in the reference manual (remote I/O network) of the network system.
 - *2 n=6
 - When performing the initial communication to the relevant station after starting link.
 - When communicating with stations other than the latest 16 stations communicated with.
 - n=1
 - When performing the second communication when the number of communication stations is 16 or less
 - When performing the second communication to the 16 stations communicated last
 - *3 When online program change is disabled in C24/E71, '1' is added only when writing data from external device.
(Set the parameter in the Switch Setting for C24. As for E71, set it in Operation Settings using Engineering tool.)

■Transmission time calculation example

The following example shows when C24 is mounted on the station on MELSECNET/H (remote I/O net), and the device memory of other station on the same network is read. (When the number of communication station is 8 stations, and the second communication is performed)

● Processing times

- S_m : Scan time of remote master station sequence program 12 ms
 - α_m : Link refresh time of remote master station 5 ms
 - LS : Link scan time 9 ms

Since the processing time is $(S_m) > (LS)$, the transmission time (T_1) will be as follows:



When master station is one station

■Causes of transmission time (T1) delay

When a command that takes two scans to send is executed, the transmission time is twice the value obtained by the formula above .

When the access request is issued to the same station from multiple external devices simultaneously, refer to the precautions for data communication. ( Page 21 When accessing CPU module)



Data transmission to the CPU module other than connected station (host station) on MELSECNET/H may be delayed considerably depending on conditions. The transmission delay time can be shortened by limiting communication between an external device and the CPU module to the station with C24/E71 mounted, and communicating with programmable controllers on other stations by data link (LB, LW).

Number of scans required for processing

When connected to C24/E71

■Basic model QCPU

| Function | Command | Subcommand | Number of scans required for processing | |
|----------------------|--------------------------------------|-----------------------------------|---|---------------------------|
| | | | Online change is enabled | Online change is disabled |
| Device access | Batch read and write | Batch read in word units | 0401 | 0000 |
| | | Batch read in bit units | | 0001 |
| | Batch write in word units | | 1401 | 0000 |
| | | Batch write in bit units | | 0001 |
| | Random read and write | Random read in word units | 0403 | 0000 |
| | | Random write in word units (test) | 1402 | 0000 |
| | | Random write in bit units (test) | | 0001 |
| | Batch read and write multiple blocks | Batch read multiple blocks | 0406 | 0000 |
| | | Batch write multiple blocks | 1406 | 0000 |
| | Monitor device memory | Register monitor data | 0801 | 0000 |
| | | Monitor | 0802 | 0000 |
| Buffer memory access | Intelligent function module | Batch read | 0601 | 0000 |
| | | Batch write | 1601 | 0000 |
| Module control | Remote control | Read CPU model name | 0101 | 0000 |

■High performance model QCUP

| Function | Command | Subcommand | Number of scans required for processing | |
|----------------------|--------------------------------------|------------------------------------|---|---------------------------|
| | | | Online change is enabled | Online change is disabled |
| Device access | Batch read and write | Batch read in word units | 0401 | 0000 |
| | | Batch read in bit units | | 0001 |
| | Batch write in word units | | 1401 | 0000 |
| | | Batch write in bit units | | 0001 |
| | Random read and write | Random read in word units | 0403 | 0000 |
| | | Random write in word units (test) | 1402 | 0000 |
| | | Random write in bit units (test) | | 0001 |
| | Batch read and write multiple blocks | Batch read multiple blocks | 0406 | 0000 |
| | | Batch write multiple blocks | 1406 | 0000 |
| | Monitor device memory | Register monitor data | 0801 | 0000 |
| | | Monitor | 0802 | 0000 |
| Buffer memory access | Intelligent function module | Batch read | 0601 | 0000 |
| | | Batch write | 1601 | 0000 |
| Module control | Remote control | Read CPU model name | 0101 | 0000 |
| File control | File check | Read directory/file information | 1810 | 0000 |
| | | Search directory/file information | 1811 | 0000 |
| | File creation and deletion | Create new file | 1820 | 0000 |
| | | Delete file | 1822 | 0000 |
| | | Copy file | 1824 | 0000 |
| | File modification | Modify file attribute | 1825 | 0000 |
| | | Modify file creation date and time | 1826 | 0000 |
| | | Open file | 1827 | 0000 |
| | | Read file | 1828 | 0000 |
| | | Write to file | 1829 | 0000 |
| | | Close file | 182A | 0000 |

■Universal model QCPU, LCPU, RCPU

| Function | | | Command | Subcommand | | Number of scans required for processing | |
|----------------------|--------------------------------------|------------------------------------|---------|---------------|------|---|------------------------------|
| | | | | QCPU/ LCPU | RCPU | Online change is enabled | Online change is disabled |
| Device access | Batch read and write | Batch read in word units | 0401 | 0000 | 0002 | 1 | 1 |
| | | Batch read in bit units | | 0001 | 0003 | 1 | 1 |
| | | Batch write in word units | 1401 | 0000 | 0002 | 1 | 2 |
| | | Batch write in bit units | | 0001 | 0003 | 1 | 2 |
| | Random read and write | Random read in word units | 0403 | 0000 | 0002 | 1 | 1 |
| | | Random write in word units (test) | 1402 | 0000 | 0002 | 1 | 2 |
| | | Random write in bit units (test) | | 0001 | 0003 | 1 | 2 |
| | Batch read and write multiple blocks | Batch read multiple blocks | 0406 | 0000 | 0002 | 1 | 1 |
| | | Batch write multiple blocks | 1406 | 0000 | 0002 | 1 | 2 |
| | Monitor device memory | Register monitor data | 0801 | 0000 | 0002 | 0 | 0 |
| | | Monitor | 0802 | 0000 | | 1 | 1 |
| Buffer memory access | Intelligent function module | Batch read | 0601 | 0000 | | 1 | 1 |
| | | Batch write | 1601 | 0000 | | 1 | 2 |
| Module control | Remote control | Read CPU model name | 0101 | 0000 | | 1 | 1 |
| File control | File check | Read directory/file information | 1810 | 0000 | 0040 | 1 | 1 |
| | | Search directory/file information | 1811 | 0000 | 0040 | 1 | 1 |
| | File creation and deletion | Create new file | 1820 | 0000 | 0040 | 1 | 2 |
| | | Delete file | 1822 | 0000 | 0040 | 1 | 2 |
| | | Copy file | 1824 | 0000 | 0040 | 1 | 2 |
| | File modification | Modify file attribute | 1825 | 0000 | 0040 | 1 | 2 |
| | | Modify file creation date and time | 1826 | 0000 | 0040 | 1 | 2 |
| | | Open file | 1827 | 0000 | 0040 | 1 | 1 |
| | | Read file | 1828 | 0000 | 0040 | 1 | 1 |
| | | Write to file | 1829 | 0000 | 0040 | 1 | 2 |
| | | Close file | 182A | 0000 | 0040 | 1 | 1 |

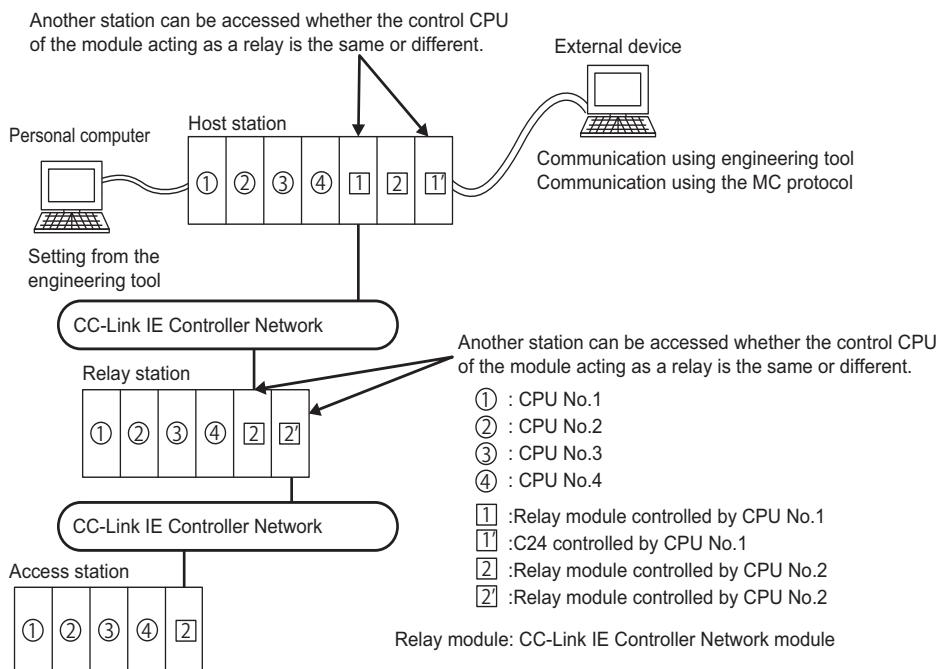
When serial communication function is used

| Function | | | Command | Subcommand | | Number of scans required for processing | |
|---------------|-----------------------|-----------------------------------|---------|---------------|------|---|------------------------------|
| | | | | QCPU/ LCPU | RCPU | Online change is enabled | Online change is disabled |
| Device access | Batch read and write | Batch read in word units | 0401 | 0000 | 0002 | 3 | 3 |
| | | Batch read in bit units | | 0001 | 0003 | 3 | 3 |
| | | Batch write in word units | 1401 | 0000 | 0002 | 3 | 3 |
| | | Batch write in bit units | | 0001 | 0003 | 3 | 3 |
| | Random read and write | Random read in word units | 0403 | 0000 | 0002 | 3 | 3 |
| | | Random write in word units (test) | 1402 | 0000 | 0002 | 3 | 3 |
| | | Random write in bit units (test) | | 0001 | 0003 | 3 | 3 |
| | Monitor device memory | Register monitor data | 0801 | 0000 | 0002 | 3 | 3 |
| | | Monitor | 0802 | 0000 | | 3 | 3 |

Appendix 3 Compatibility with Multiple CPU Systems

A

This section explains data communication using MC protocol when CPU modules are configured in the multiple CPU system.



Point

By setting routing parameters (communication route) in advance, accessing programmable controllers in the following network systems are available.

- CC-Link IE Controller Network
- CC-Link IE Field Network
- MELSECNET/H
- MELSECNET/10
- Ethernet

Up to 8 target networks (relay stations: 7 stations) can be accessed.

When RCPU is configured in the multiple CPU system

Point

When using C24/E71 in the multiple CPU system of RCPU, refer to the following manual.

BOOK MELSEC iQ-R Module Configuration Manual

Accessing multiple CPU system on connected station (host station)

■ Accessing multiple CPU system on connected station (host station) from external devices

Control CPU (set with Engineering tool) and non-control CPU of C24 can be accessed.

The usable commands vary depending on the access target RCPU (control CPU, non-control CPU).

Accessing multiple CPU system on other station

■ Accessing multiple CPU system on the other station (access station)

Control CPU and non-control CPU of the network module connected to the relay station can be accessed.

The usable command vary depending on the access target RCPU (control CPU, non-control CPU).

The modules which can be accessed to other stations via network are as follows.

- CC-Link IE, MELSECNET/H, MELSECNET/10 Network Module
- C24
- E71

When QCPU is configured in the multiple CPU system

Point

When using C24/E71 in the multiple CPU system of QCPU, refer to the following manual.

QCPU User's Manual (Multiple CPU System)

Accessing multiple CPU system on connected station (host station)

■When using C24/E71 in the multiple CPU system

Use C24/E71 of which function version is B.

■Accessing multiple CPU system on connected station (host station) from external devices

Control CPU (set with Engineering tool) and non-control CPU of C24/E71 can be accessed.

The usable commands vary depending on the access target QCPU (control CPU, non-control CPU).

Using C24/E71 of which function version is A is also possible. In this case, QCPU No.1 is the control CPU of C24/E71.

Only the control CPU of C24/E71 can be accessed when accessing QCPU on the connected station (host station) from external devices.

Accessing multiple CPU system on other station

■Use the modules of which function version is B for each station.

Use the modules of which function version is B for each station as shown below.

| Type | Module | Remarks |
|----------------------------------|--|--|
| Connected station (host station) | QCPU | — |
| | C24/E71 connected to external device | Set the same control CPU for C24/E71 and network module. |
| | Network module connected to relay station/access station | |
| Relay station | QCPU | — |
| | Two modules connecting the network between the host station side and the access station side | |
| Access station | QCPU | |
| | Network module connected to relay station | |

■Accessing multiple CPU system on the other station (access station)

Control CPU and non-control CPU of the network module connected to the relay station can be accessed.

The usable commands vary depending on the access target QCPU (control CPU, non-control CPU).

When accessing other stations from external devices, and any module of which function version is A is included in the above modules of connected station (host station), relay station, or access station, only the control CPU of network module connected to the relay station can be accessed. In addition, accessing other station via a module which is controlled by the same control CPU is available.

The modules which can be accessed other stations via network are as follows.

- CC-Link IE, MELSECNET/H ,MELSECNET/10 Network module
- C24
- E71

Appendix 4 Applicable Commands for Online Program Change

This section explains the applicability of each command according to the STOP/RUN status of CPU module and the settings by supported devices.

In this section, the commands that cannot be executed while CPU module is in the RUN status and the commands to enable/disable the online change with Engineering tool are shown. The commands not listed in the following tables can be used regardless of the CPU STOP/RUN status.

Commands that cannot be executed during RUN

The following commands cannot be executed while the CPU module is in RUN.

Set the CPU module to STOP and perform the command.

File handling

The file operation to specify the following files cannot be performed during RUN.

| Type | Command | CPU module | Target file |
|----------|--|------------|--|
| Write | <ul style="list-style-type: none"> Copy file (command: 1824) Modify file attribute (command: 1825) Modify file creation date and time (command: 1826) Write to file (command: 1829) | RCPU | Current execution file*1 |
| | <ul style="list-style-type: none"> Write to file (command: 1203) Copy file (command: 1206) Modify file information (command: 1204) Modify file name, attribute, file size (command: 1204) Batch modify file information (command: 1204) | Q/LCPU | <ul style="list-style-type: none"> Parameter file (*.QPA) Current execution file of program memory (Drive name: 00H) |
| Deleting | <ul style="list-style-type: none"> Delete file (command: 1822) | RCPU | Current execution file*1 |
| | <ul style="list-style-type: none"> Delete file (command: 1205) | Q/LCPU | <ul style="list-style-type: none"> Program file (*.QPG) Parameter file (*.QPA) Boot setting file (*.QBT) |
| | | QnACPU | |

*1 For details on the operations for the files of RCPU, refer to the following manual.

MELSEC iQ-R CPU Module User's Manual (Application)

Module control

| Type | Command | CPU module |
|---------------------------------|--|------------|
| Remote control | <ul style="list-style-type: none"> Remote latch clear (command: 1005) Remote RESET (command: 1006) | RCPU |
| | | Q/LCPU |
| Defragmentation of drive memory | Defragment drive memory (command: 1207) | QnACPU |

Commands that enable/disable online change

The following shows the commands which can be executed while the CPU module is in the RUN status only when the online change is set to be enabled.

○: Executable, ×:Not executable

| CPU state | STOP | RUN | |
|------------------------------------|----------------|--------|---------|
| Setting of online change | Enable/Disable | Enable | Disable |
| Applicability of command execution | ○ | ○ | × |

Commands for 4C /3C/2C/4E/3E frame

| Type | Command | 4C, 3C, 4E, 3E | 2C |
|---------------|------------------------------------|----------------|----|
| Device access | Batch write in word units | 1401 | 4 |
| | Batch write in bit units | | 3 |
| | Random write in word units (test) | 1402 | 7 |
| | Random write in bit units (test) | | 6 |
| | Batch write multiple blocks | 1406 | — |
| Label access | Batch write array type labels | 141A | — |
| | Random write labels | 1418 | — |
| File control | Create new file | 1820 | — |
| | Delete file | 1822 | — |
| | Copy file*1 | 1824 | — |
| | Modify file attribute | 1825 | — |
| | Modify file creation date and time | 1826 | — |
| | Write to file | 1829 | — |
| | Create new file | 1202 | — |
| | Write to file | 1203 | — |
| | Modify file information | 1204 | — |
| | Delete file | 1205 | — |

*1 For E71, data can be copied even if Online program change function is set to prohibit.

Commands for 1C/1E frame

| Type | Command | 1C | 1E |
|-----------------------------|---------------------|------------|--------|
| Device memory | Batch write | Bit units | BW, JW |
| | | Word units | WW, QW |
| | Test (random write) | Bit units | BT, JT |
| | | Word units | WT, QT |
| Extended file register | Batch write | EW | 18H |
| | Test (random write) | ET | 19H |
| | Direct write | NW | 3CH |
| Intelligent function module | Batch write | TW | 0FH |

Setting method for writing data to CPU during RUN

The setting to allowed/prohibit of Online program change function can be performed on the following screen of Engineering tool.

For details of the setting, refer to the user's manual of the supported device (C24/E71) used.

GX Works3

- C24: "Module Parameter" screen (transmission setting)
- E71: "Module Parameter" screen (Own Node Settings)

GX Works2

- C24: "Switch Settings" screen of Intelligent functional module (Transmission setting)
- E71: "Ethernet Operation Setting" screen

GX Developer

- C24: "Switch Setting for I/O and Intelligent Function Module" screen (Switch 1, b6 of switch 3)
- E71: "Ethernet Operation Setting" screen

Appendix 5 Number of Processing per One Communication

A

The following shows the number of processing per one communication with each command.

Commands for 4C/3C/4E/3E frame

Device access

The letter in the column of 'Access target' indicates the following module.

- R/Q/L: MELSEC iQ-R series, MELSEC-Q series, or MELSEC-L series module
- QnA: MELSEC-QnA series and other station module via MELSEC-QnA series network module
- A: MELSEC-QnA series module

For the restrictions of access target for each command and subcommand type, refer to the following section.

 Page 465 Accessible Modules for Each Command

| Function | | Command | Subcommand | Condition | Number of points processed per communication | | | |
|--------------------------------------|-----------------------------------|---------|--------------|--|--|----------------------------|--------------------------|--|
| | | | | | R/Q/L | QnA | A | |
| Batch read and write | Batch read in word units | 0401 | 0000 0002 | Word device points | 960 points | 480 points | 64 points | |
| | | | | Bit device points | 960 words (15360 points) | 480 words (7680 points) | 32 words (512 points) | |
| | Batch read in bit units | | 0001 0003 | C24 | 7904 points | 3952 points | 256 points | |
| | | | | E71 (ASCII code) | 3584 points | 1792 points | | |
| | | | | E71 (Binary code) | 7168 points | 3584 points | | |
| | Batch write in word units | | 0000 0002 | Word device points | 960 points | 480 points | 64 points | |
| | | | | Bit device points | 960 words (15360 points) | 480 words (7680 points) | 10 words (160 points) | |
| | | | 0001 0003 | C24 | 7904 points | 3952 points | 160 points | |
| | | | | E71 (ASCII code) | 3584 points | 1792 points | | |
| | | | | E71 (Binary code) | 7168 points | 3584 points | | |
| Random read and write | Random read in word units | 0403 | 0000 | Number of word access points + Number of double word access points | 192 points | 96 points | — | |
| | | | 0002 008□ | | 96 points | | | |
| | Random write in word units (test) | 1402 | 0000 | (Number of word access points × 12) + (Number of double word access points × 14) | 1920 points | 960 points | 10 points | |
| | | | 0002 008□ | | 960 points | | | |
| | | | 0001 | Number of bit access points | 188 points | 94 points | 20 points | |
| | | | 0003 008□ | | 94 points | | | |
| Batch read and write multiple blocks | Batch read multiple blocks | 0406 | 0000 | Number of word device blocks + Number of bit device blocks | 120 points | 60 points | — | |
| | | | 0002 008□ | | 60 points | | | |
| | Batch write multiple blocks | 1406 | 0000 | | 120 points | 60 points | — | |
| | | | 0002 008□ | | 60 points | | | |
| Monitor device memory | Register monitor data | 0801 | 0000 | Number of word access points + Number of double word access points | 192 points | 96 points | — | |
| | | | 0002 008□ | | 96 points | | | |
| | Monitor | 0802 | 0000 | Read data | (Number of registered points) | | | |

Label access

| Function | | Command | Subcommand | Number of points processed per communication |
|-----------------------|-------------------------------|---------|------------|--|
| Batch read and write | Batch read array type labels | 041A | 0000 | Depend on the label name length (total number of bytes are 1920 bytes or less) |
| | Batch write array type labels | 141A | 0000 | |
| Random read and write | Random read labels | 041C | 0000 | |
| | Random write labels | 141B | 0000 | |

Buffer memory access

| Function | | Command | Subcommand | Number of points processed per communication |
|-----------------------------|-------------|---------|------------|--|
| Buffer memory | Batch read | 0613 | 0000 | 480 words (960 bytes) |
| | Batch write | 1613 | 0000 | |
| Intelligent function module | Batch read | 0601 | 0000 | 1920 bytes (960 words) |
| | Batch write | 1601 | 0000 | |

Module control

| Function | | Command | Subcommand | Number of points processed per communication |
|-------------------------|---------------------|---------|------------|--|
| Remote control | Remote RUN | 1001 | 0000 | (1 station) |
| | Remote STOP | 1002 | 0000 | (1 station) |
| | Remote PAUSE | 1003 | 0000 | (1 station) |
| | Remote latch clear | 1005 | 0000 | (1 station) |
| | Remote RESET | 1006 | 0000 | (1 station) |
| | Read CPU model name | 0101 | 0000 | (1 station) |
| Remote password | Unlock | 1630 | 0000 | — |
| | Lock | 1631 | 0000 | — |
| Loopback test | | 0619 | 0000 | 960 bytes (applicable only with a connected station) |
| Clear error information | | 1617 | 000□ | (1 station) |

File control

| Function | | Command | Subcommand | Number of points processed per communication |
|----------------------------|------------------------------------|---------|----------------------|--|
| File check | Read directory/file information | 1810 | 0000 0040 | (for 36) |
| | Search directory/file information | 1811 | 0000 0040 | (for 1) |
| File creation and deletion | Create new file | 1820 | 0000 0040 | (for 1) |
| | Delete file | 1822 | 0000 0004 0040 | (for 1) |
| | Copy file | 1824 | 0000 0004 0040 | (for 1) |
| File modification | Modify file attribute | 1825 | 0000 0004 0040 | (for 1) |
| | Modify file creation date and time | 1826 | 0000 0040 | (for 1) |
| | Open file | 1827 | 0000 0004 0040 | (for 1) |
| | Read file | 1828 | 0000 0040 | 1920 bytes |
| | Write to file | 1829 | 0000 0040 | 1920 bytes |
| | Close file | 182A | 0000 0040 | (for 1) |

Serial communication module dedicated commands

| Function | | Command | Subcommand | Number of points processed per communication |
|--|------------------------|---------|--------------|--|
| User frame | Read registered data | 0610 | 0000 | 80 bytes |
| | Register data | 1610 | 0000 | (1 station) |
| | Delete registered data | | 0001 | (for 1 station/all stations) |
| Global | | 1618 | 0000 0001 | (for 1 station/all stations) |
| Initialize transmission sequence | | 1615 | 0000 | (1 station) |
| Switch mode | | 1612 | 0000 | (1 station) |
| Programmable controller CPU monitoring | Register | 0630 | 0000 | 960 points |
| | Deregister | 0631 | 0000 | — |
| On-demand | | 2101 | — | (1 station) |

QnA dedicated commands

| Function | | Command | Subcommand | Number of points processed per communication |
|------------------------------|--------------------------------------|---------|------------|--|
| Drive memory defragmentation | Read memory usage status | 0205 | 0000 | (256 clusters) |
| | Defragment memory | 1207 | 0000 | (1 station) |
| File control | Read file information table | 0201 | 0000 | (for 36) |
| | | 0202 | 0000 | (for 16) |
| | Read file presence (Search file) | 0203 | 0000 | (for 1) |
| | Read file No. usage status | 0204 | 0000 | (for 256) |
| | Read file | 0206 | 0000 | 960 bytes |
| | Lock file | 0808 | 0001 | (for 1) |
| | | | 0000 | |
| | Create new file (Register file name) | 1202 | 0000 | (for 1) |
| | Write to file | 1203 | 0000 | 960 bytes |
| | | | 0001 | For file size |
| | Modify file information | 1204 | 0000 | (for 1) |
| | | | 0001 | |
| | | | 0002 | |
| | Delete file | 1205 | 0000 | (for 1) |
| | Copy file | 1206 | 0000 | 480 bytes |

Commands for 2C frame

Refer to the processing points of the corresponding commands for 4C/3C/4E/3E frame

Commands for 1C frame

| Function | | | | Command | Number of points processed per communication | | |
|-------------------------|-----------------------|-------------|------------|-------------------------------------|--|--|--|
| Device memory | Batch read | Bit units | | BR JR | 256 points | | |
| | | Word units | Bit device | WR QR | 32 words (512 points) | | |
| | Batch write | Word device | | | 64 points | | |
| | | Bit units | | BW JW | 160 points | | |
| | | Word units | Bit device | WW QW | 10 words (160 points) | | |
| | | Word device | | | 64 points | | |
| Register monitor data | Test (random write) | Bit units | | BT JT | 20 points | | |
| | | Word units | Bit device | WT QT | 10 words (160 points) | | |
| | | Word device | | | 10 points | | |
| | Bit units | | BM JM | 40 points ^{*1} | | | |
| | Word units | Bit device | WM QM | 20 words (320 points) ^{*1} | | | |
| Monitor | Word device | | | 20 points | | | |
| | Bit units | | MB MJ | Number of registered points | | | |
| | Word units | | MN MQ | Number of registered points | | | |
| Extended file register | Batch read | | | ER | 64 points | | |
| | Batch write | | | EW | 64 points | | |
| | Test (random write) | | | ET | 10 points | | |
| | Register monitor data | | | EM | 20 points | | |
| | Monitor | | | ME | Number of registered points | | |
| | Direct read | Word units | | NR | 64 points | | |
| | Direct write | Word units | | NW | 64 points | | |
| Special function module | Batch read | | | TR | 128 bytes | | |
| | Batch write | | | TW | 128 bytes | | |
| Loopback test | | | | TT | 254 bytes | | |

*1 For ACPU other than AnA/AnU, device X (input) has two processing points per point. When the X is included in the specified device, set the number of device points to meet the following.

((Specified points for X × 2) + Specified points for other devices) ≤ Number of points processed per communication.

If only X is specified, the number of points processed per one communication will be one half the value shown in the table.

Commands for 1E frame

| Function | | | Command | Number of points processed per communication |
|-------------------------|-----------------------|------------|---------|--|
| Device memory | Batch read | Bit units | 00H | 256 points |
| | | Word units | 01H | 128 words (2048 points) |
| | | | | 256 points |
| | Batch write | Bit units | 02H | 256 points |
| | | Word units | 03H | 40 words (640 points) |
| | | | | 256 points |
| | Test (random write) | Bit units | 04H | 80 points |
| | | Word units | 05H | 40 words (640 points) |
| | | | | 40 points |
| | Register monitor data | Bit units | 06H | 40 points*1 |
| | | Word units | 07H | 20 words (320 points)*1 |
| | | | | 20 points |
| | Monitor | Bit units | 08H | Number of registered points |
| | | Word units | 09H | Number of registered points |
| Extended file register | Batch read | | 17H | 256 points |
| | Batch write | | 18H | 256 points |
| | Test (random write) | | 19H | 40 points |
| | Register monitor data | | 1AH | 20 points |
| | Monitor | | 1BH | — |
| | Direct read | | 3BH | 256 points |
| | Direct write | | 3CH | 256 points |
| Special function module | Batch read | | 0EH | 256 bytes (128 words) |
| | Batch write | | 0FH | 256 bytes (128 words) |

*1 For ACPU other than AnA/AnU, device X (input) has two processing points per point. When the X is included in the specified device, set the number of device points to meet the following.

((Specified points for X × 2) + Specified points for other devices) ≤ Number of points processed per communication.

If only X is specified, the number of points processed per one communication will be one half the value shown in the table.

Appendix 6 Accessible Modules for Each Command

A

The following shows the access target modules that can be specified by a request message.

For the functions and commands which can be used for supported devices, refer to the user's manual of each device.

Access target modules

The character in the column of 'Access target' indicates the following module.

- R: MELSEC-iQ-R series CPU module and mounted intelligent function module
- Q/L: MELSEC-Q series or MELSEC-L series CPU module and the mounted intelligent functional module
- QnA: MELSEC-QnA series CPU module and mounted special function module
- A: MELSEC-A series CPU module and mounted special function module
- QS: Safety CPU module and the mounted intelligent functional module
- H: Module of MELSECNET/H remote station
- 10: Module of MELSECNET/10 remote station
- Head: Head module of CC-Link IE Field Network

Commands for 4C/3C/4E/3E frame

Device access

○: Executable, △: Executable (with restrictions), ×: Not executable

| Function | Command | Subcommand | Access target | | | | | | | | | Reference | |
|-----------------------|---------|--------------|-----------------|---|-----------------|-----|---|----|------------|----|------|--|--|
| | | | CPU module | | | | | | Remote I/O | | Head | | |
| | | | R | L | Q | QnA | A | QS | H | 10 | | | |
| Batch read and write | 0401 | 0000 0080 | △ ^{*1} | ○ | ○ | ○ | ○ | ○ | ○ | × | ○ | Page 84 Batch read in word units (command: 0401) | |
| | | 0002 0082 | ○ ^{*2} | × | × | × | × | × | × | × | × | | |
| | | 0001 0081 | △ ^{*1} | ○ | ○ | ○ | ○ | ○ | ○ | × | ○ | | |
| | | 0003 0083 | ○ ^{*2} | × | × | × | × | × | × | × | × | | |
| | 1401 | 0000 0080 | △ ^{*1} | ○ | ○ | ○ | ○ | × | ○ | × | ○ | Page 89 Batch write in word units (command: 1401) | |
| | | 0002 0082 | ○ ^{*2} | × | × | × | × | × | × | × | × | | |
| | | 0001 0081 | △ ^{*1} | ○ | ○ | ○ | ○ | × | ○ | × | ○ | | |
| | | 0003 0083 | ○ ^{*2} | × | × | × | × | × | × | × | × | | |
| Random read and write | 0403 | 0000 0080 | △ ^{*1} | ○ | ○ | ○ | × | ○ | ○ | × | ○ | Page 94 Random read in word units (command: 0403) | |
| | | 0040 00C0 | × | × | △ ^{*3} | ○ | × | × | ○ | × | × | | |
| | | 0002 0082 | ○ ^{*2} | × | × | × | × | × | × | × | × | | |
| | 1402 | 0000 0080 | △ ^{*1} | ○ | ○ | ○ | ○ | × | ○ | × | ○ | Page 101 Random write in word units (test) (command: 1402) | |
| | | 0002 0082 | ○ ^{*2} | × | × | × | × | × | × | × | × | | |
| | | 0001 0081 | △ ^{*1} | ○ | ○ | ○ | ○ | × | ○ | × | ○ | Page 105 Random write in bit units (test) (command: 1402) | |
| | | 0003 0083 | ○ ^{*2} | × | × | × | × | × | × | × | × | | |

| Function | Command | Subcommand | Access target | | | | | | | | | Reference | |
|--------------------------------------|---------|--------------|-----------------|---|-----------------|-----------------|---|----|------------|----|------|--|--|
| | | | CPU module | | | | | | Remote I/O | | Head | | |
| | | | R | L | Q | QnA | A | QS | H | 10 | | | |
| Batch read and write multiple blocks | 0406 | 0000 0080 | △ ^{*1} | ○ | ○ | △ ^{*4} | × | ○ | ○ | × | ○ | Page 107 Batch read multiple blocks (command: 0406) | |
| | | 0002 0082 | ○ ^{*2} | × | × | × | × | × | × | × | × | | |
| | 1406 | 0000 0080 | △ ^{*1} | ○ | ○ | △ ^{*4} | × | × | ○ | × | ○ | Page 111 Batch write multiple blocks (command: 1406) | |
| | | 0002 0082 | ○ ^{*2} | × | × | × | × | × | × | × | × | | |
| Monitor device memory | 0801 | 0000 0080 | △ ^{*1} | ○ | ○ | ○ | × | × | ○ | × | ○ | Page 115 Register monitor data (command: 0801) | |
| | | 0040 00C0 | × | × | △ ^{*3} | × | × | × | ○ | × | × | | |
| | | 0002 0082 | ○ ^{*2} | × | × | × | × | × | × | × | × | | |
| | 0802 | 0000 | ○ | ○ | ○ | ○ | × | × | ○ | × | ○ | Page 116 Monitor (command: 0802) | |

*1 Specify the device with the specification method of MELSEC-Q/L series. The added new devices for MELSEC iQ-R and their ranges cannot be specified.

*2 The command can be used when both access target and connected station are MELSEC iQ-R series module.

*3 The command can be specified when using Basic model QCPU, High Performance model QCPU, and Process CPU.

*4 QnACPU and Q2AS(H) CPU with the version 9707B or later can be used.

Label access

The command can be used when both access target and connected station are MELSEC iQ-R series module.

☞ Page 119 LABEL ACCESS

Buffer memory access

○: Executable, △: Executable (with restrictions), ×: Not executable

| Function | Command | Subcommand | Access target | | | | | | | | | Reference | |
|-----------------------------|---------|------------|-----------------|---|---|-----|---|----|------------|----|------|-------------------------------------|--------------------------------------|
| | | | CPU module | | | | | | Remote I/O | | Head | | |
| | | | R | L | Q | QnA | A | QS | H | 10 | | | |
| Buffer memory | 0613 | 0000 | — ^{*1} | | | | | | | | | Page 149 Batch read (command: 0613) | |
| | 1613 | 0000 | — ^{*1} | | | | | | | | | | Page 151 Batch write (command: 1613) |
| Intelligent function module | 0601 | 0000 | × | ○ | ○ | ○ | × | ○ | ○ | × | ○ | Page 157 Batch read (command: 0601) | |
| | 1601 | 0000 | × | ○ | ○ | ○ | × | × | ○ | × | ○ | | |

*1 The command can only be used for C24 (including multidrop connection station) and E71 connected to the external device.

Module control

○: Executable, △: Executable (with restrictions), ×: Not executable

| Function | Command | Subcommand | Access target | | | | | | | | | Reference | |
|-------------------------|---------|------------|---------------|---|---|-----|---|----|------------|----|------|--|--|
| | | | CPU module | | | | | | Remote I/O | | Head | | |
| | | | R | L | Q | QnA | A | QS | H | 10 | | | |
| Remote control | 1001 | 0000 | ○ | ○ | ○ | ○ | × | × | × | × | ○ | Page 167 Remote RUN (command: 1001) | |
| | 1002 | 0000 | ○ | ○ | ○ | ○ | × | × | × | × | ○ | Page 169 Remote STOP (command: 1002) | |
| | 1003 | 0000 | ○ | ○ | ○ | ○ | × | × | × | × | × | Page 170 Remote PAUSE (command: 1003) | |
| | 1005 | 0000 | ○ | ○ | ○ | ○ | × | × | × | × | × | Page 171 Remote latch clear (command: 1005) | |
| | 1006 | 0000 | ○ | ○ | ○ | ○ | × | × | × | × | ○ | Page 172 Remote RESET (command: 1006) | |
| | 0101 | 0000 | ○ | ○ | ○ | × | × | ○ | × | × | ○ | Page 173 Read CPU model name (command: 0101) | |
| Remote password | 1630 | 0000 | — *1 | | | | | | | | | Page 175 Unlock (command: 1630) | |
| | 1631 | 0000 | — *2 | | | | | | | | | Page 177 Lock (command: 1631) | |
| Loopback test | 0619 | 0000 | — *1 | | | | | | | | | Page 179 Loopback test (command: 0619) | |
| Clear error information | 1617 | 0000 | — *1 | | | | | | | | | Page 181 Turn indicator LED OFF, initialize error code (command: 1617) | |

*1 The command can only be used for C24 (including multidrop connection station) and E71 connected to the external device.

*2 The command can only be used for E71 connected to an external device.

File control

○: Executable, △: Executable (with restrictions), ×: Not executable

| Function | Command | Subcommand | Access target | | | | | | | | | Reference | |
|----------------------------|---------|------------|---------------|-----|---|-----|---|----|------------|----|------|--|--|
| | | | CPU module | | | | | | Remote I/O | | Head | | |
| | | | R | L | Q | QnA | A | QS | H | 10 | | | |
| File check | 1810 | 0000 | × | ○ | ○ | × | × | ○ | × | × | ○ | Page 205 Read directory/file information (command: 1810) | |
| | | 0040 | ○*1 | × | × | × | × | × | × | × | × | | |
| | 1811 | 0000 | × | ○ | ○ | × | × | ○ | × | × | ○ | Page 211 Search directory/file information (command: 1811) | |
| | | 0040 | ○*1 | × | × | × | × | × | × | × | × | | |
| File creation and deletion | 1820 | 0000 | × | ○ | ○ | × | × | × | × | × | ○ | Page 214 Create new file (command: 1820) | |
| | | 0040 | ○*1 | × | × | × | × | × | × | × | × | | |
| | 1822 | 0000 | × | △*2 | ○ | × | × | × | × | × | △*2 | Page 217 Delete file (command: 1822) | |
| | | 0004 | × | ○*3 | × | × | × | × | × | × | ○ | | |
| | | 0040 | ○*1 | × | × | × | × | × | × | × | × | | |
| | 1824 | 0000 | × | △*2 | ○ | × | × | × | × | × | △*2 | Page 220 Copy file (command: 1824) | |
| | | 0004 | × | ○*3 | × | × | × | × | × | × | ○ | | |
| | | 0040 | ○*1 | × | × | × | × | × | × | × | × | | |

| Function | Command | Subcommand | Access target | | | | | | | | | Reference | |
|-------------|---------|------------|---------------|-----|---|-----|---|----|------------|----|------|--|--|
| | | | CPU module | | | | | | Remote I/O | | Head | | |
| | | | R | L | Q | QnA | A | QS | H | 10 | | | |
| Modify file | 1825 | 0000 | x | △*2 | ○ | x | x | x | x | x | △*2 | Page 224 Modify file attribute (command: 1825) | |
| | | 0004 | x | ○*3 | x | x | x | x | x | x | ○ | | |
| | | 0040 | ○*1 | x | x | x | x | x | x | x | x | | |
| | 1826 | 0000 | × | ○ | ○ | x | x | x | x | x | ○ | Page 227 Modify file creation date and time (command: 1826) | |
| | | 0040 | ○*1 | x | x | x | x | x | x | x | x | | |
| | 1827 | 0000 | x | △*2 | ○ | x | x | ○ | x | x | △*2 | Page 230 Open file (command: 1827) | |
| | | 0004 | x | ○*3 | x | x | x | x | x | x | ○ | | |
| | | 0040 | ○*1 | x | x | x | x | x | x | x | x | | |
| | 1828 | 0000 | ○ | ○ | ○ | x | x | ○ | x | x | ○ | Page 234 Read file (command: 1828) | |
| | 1829 | 0000 | ○ | ○ | ○ | x | x | x | x | x | ○ | Page 236 Write to file (command: 1829) | |
| | 182A | 0000 | ○ | ○ | ○ | x | x | ○ | x | x | ○ | Page 238 Close file (command: 182A) | |

*1 The command can be used when both access target and connected station are MELSEC iQ-R series module.

*2 It can be used when password is not set to the target file.

*3 The command can be used when both access target and connected station are MELSEC-L series module.

Serial communication dedicated commands

The command can only be used for C24 (including multidrop connection station) connected to the external device.

☞ Page 240 SERIAL COMMUNICATION MODULE DEDICATED COMMANDS

QnACPU dedicated commands

The QnACPU dedicated commands can only be executed when access target is QnACPU.

☞ Page 282 QnACPU DEDICATED COMMANDS

Commands for 2C frame

The command can only be used for C24 (including multidrop connection station) and its control CPU module connected to the external device.

Cannot be used via network.

○: Executable, △: Executable (with restrictions)

| Function | Command | Access target | | | | Reference |
|---------------|---------|---------------|---|---|-----|--|
| | | R | L | Q | QnA | |
| Device access | 1 | △*1 | ○ | ○ | ○ | Page 87 Batch read in bit units (command: 0401) |
| | 2 | △*1 | ○ | ○ | ○ | Page 84 Batch read in word units (command: 0401) |
| | 3 | △*1 | ○ | ○ | ○ | Page 92 Batch write in bit units (command: 1401) |
| | 4 | △*1 | ○ | ○ | ○ | Page 89 Batch write in word units (command: 1401) |
| | 5 | △*1 | ○ | ○ | ○ | Page 94 Random read in word units (command: 0403) |
| | 6 | △*1 | ○ | ○ | ○ | Page 105 Random write in bit units (test) (command: 1402) |
| | 7 | △*1 | ○ | ○ | ○ | Page 101 Random write in word units (test) (command: 1402) |
| | 8 | △*1 | ○ | ○ | ○ | Page 115 Register monitor data (command: 0801) |
| | 9 | ○ | ○ | ○ | ○ | Page 116 Monitor (command: 0802) |

*1 Specify the device with the specification method of MELSEC-Q/L series. The added new devices their ranges cannot be specified.

Commands for 1C frame

○: Executable, ×:Not executable

| Function | | | Command | Access target | | | | | | | Reference |
|-------------------------|-----------------------|-----------------|----------|-----------------|-----------------|-----------------|-----------------|------------|-----------------|-----------------|---|
| | | | | CPU module | | | | Remote I/O | | | |
| R | L | Q ^{*1} | QnA | A | QS | H | | | | | |
| Device memory | Batch read | Bit units | BR JR | △ ^{*2} | △ ^{*2} | △ ^{*2} | △ ^{*2} | ○ | △ ^{*2} | △ ^{*2} | Page 348 Batch read (bit units) (command: BR, JR) |
| | | Word units | WR QR | △ ^{*2} | △ ^{*2} | △ ^{*2} | △ ^{*2} | ○ | △ ^{*2} | △ ^{*2} | Page 350 Batch read (word units) (command: WR, QR) |
| | Batch write | Bit units | BW JW | △ ^{*2} | △ ^{*2} | △ ^{*2} | △ ^{*2} | ○ | × | △ ^{*2} | Page 352 Batch write (bit units) (command: BW, JW) |
| | | Word units | WW QW | △ ^{*2} | △ ^{*2} | △ ^{*2} | △ ^{*2} | ○ | × | △ ^{*2} | Page 354 Batch write (word units) (command: WW, QW) |
| | Test (random write) | Bit units | BT JT | △ ^{*2} | △ ^{*2} | △ ^{*2} | △ ^{*2} | ○ | × | △ ^{*2} | Page 356 Test (random write) (bit units) (command: BT, JT) |
| | | Word units | WT QT | △ ^{*2} | △ ^{*2} | △ ^{*2} | △ ^{*2} | ○ | × | △ ^{*2} | Page 358 Test (random write) (word units) (command: WT, QT) |
| | Register monitor data | Bit units | BM JM | △ ^{*2} | △ ^{*2} | △ ^{*2} | △ ^{*2} | ○ | × | △ ^{*2} | Page 361 Register monitor data (bit units) (command: BM, JM) |
| | | Word units | WM QM | △ ^{*2} | △ ^{*2} | △ ^{*2} | △ ^{*2} | ○ | × | △ ^{*2} | Page 362 Register monitor data (word units) (command: WM, QM) |
| | Monitor | Bit units | MB MJ | ○ | ○ | ○ | ○ | ○ | × | ○ | Page 363 Monitor (bit units) (command: MB, MJ) |
| | | Word units | MN MQ | ○ | ○ | ○ | ○ | ○ | × | ○ | Page 364 Monitor (word units) (command: MN, MQ) |
| Extended file register | Batch read | | ER | × | × | × | × | ○ | × | × | Page 369 Batch read (command: ER) |
| | Batch write | | EW | × | × | × | × | ○ | × | × | Page 370 Batch write (command: EW) |
| | Test (random write) | | ET | × | × | × | × | ○ | × | × | Page 371 Test (random write) (command: ET) |
| | Register monitor data | | EM | × | × | × | × | ○ | × | × | Page 373 Register monitor data (command: EM) |
| | Monitor | | ME | × | × | × | × | ○ | × | × | Page 374 Monitor (command: ME) |
| | Direct read | Word units | NR | × | × | × | × | ○ | × | × | Page 375 Direct read (command: NR) |
| | Direct write | Word units | NW | × | × | × | × | ○ | × | × | Page 376 Direct write (command: NW) |
| Special function module | Batch read | | TR | × | × | × | × | ○ | × | × | Page 380 Batch read (command: TR) |
| | Batch write | | TW | × | × | × | × | ○ | × | × | Page 382 Batch write (command: TW) |
| Loopback test | | | TT | — ^{*3} | | | | | | | Page 383 Loopback test (Command: TT) |

*1 When accessing Universal model QCPU, use the module with a serial number whose first five digits are '10102' or later. If using a module with a serial number whose first five digits are '10101' or earlier, access with 2C/3C/4C frame.

*2 Read/write the devices within the range of AnCPU, AnNCPU, AnACPU, or AnUCPU. The added new devices and their ranges cannot be specified.

*3 The command can only be used for C24 (including multidrop connection station) connected to the external device.

Commands for 1E frame

○: Executable, ×:Not executable

| Function | | | Command | Access target | | | | | Reference | |
|-------------------------|-------------------------------------|------------|---------|-----------------|-----------------|-----------------|---|-----------------|---|--|
| | | | | CPU module | | | | | | |
| | | | | L | Q ^{*1} | Qn A | A | QS | | |
| Device memory | Batch read | Bit units | 00H | △ ^{*2} | △ ^{*2} | △ ^{*2} | ○ | △ ^{*2} | Page 397 Batch read in bit units (command: 00) | |
| | | Word units | 01H | △ ^{*2} | △ ^{*2} | △ ^{*2} | ○ | △ ^{*2} | Page 399 Batch read in word units (command: 01) | |
| | Batch write | Bit units | 02H | △ ^{*2} | △ ^{*2} | △ ^{*2} | ○ | × | Page 401 Batch write in bit units (command: 02) | |
| | | Word units | 03H | △ ^{*2} | △ ^{*2} | △ ^{*2} | ○ | × | Page 403 Batch write in word units (command: 03) | |
| | Test (random write) | Bit units | 04H | △ ^{*2} | △ ^{*2} | △ ^{*2} | ○ | × | Page 404 Test in bit units (random write) (command: 04) | |
| | | Word units | 05H | △ ^{*2} | △ ^{*2} | △ ^{*2} | ○ | × | Page 406 Test in word units (random write) (command: 05) | |
| | Register monitor data ^{*3} | Bit units | 06H | △ ^{*2} | △ ^{*2} | △ ^{*2} | ○ | × | Page 409 Register monitor data(command: 06, 07) | |
| | | Word units | 07H | △ ^{*2} | △ ^{*2} | △ ^{*2} | ○ | × | | |
| | Monitor | Bit units | 08H | ○ | ○ | ○ | ○ | × | Page 410 Monitor in bit units (command: 08) Page 411 Monitor in word units (command: 09) | |
| | | Word units | 09H | ○ | ○ | ○ | ○ | × | | |
| Extended file register | Batch read | | 17H | × | × | × | ○ | × | Page 413 Batch read (command: 17) | |
| | Batch write | | 18H | × | × | × | ○ | × | Page 415 Batch write (command: 18) | |
| | Test (random write) | | 19H | × | × | × | ○ | × | Page 417 Test (random write) (command: 19) | |
| | Register monitor data ^{*3} | | 1AH | × | × | × | ○ | × | Page 420 Register monitor data (command: 1A) | |
| | Monitor | | 1BH | × | × | × | ○ | × | Page 421 Monitoring (command: 1B) | |
| | Direct read | | 3BH | × | × | × | ○ | × | Page 422 Direct read (command: 3B) | |
| | Direct write | | 3CH | × | × | × | ○ | × | Page 424 Direct write (command: 3C) | |
| Special function module | Batch read | | 0EH | × | × | × | ○ | × | Page 428 Batch read (command: 0E) | |
| | Batch write | | 0FH | × | × | × | ○ | × | Page 430 Batch write (command: 0F) | |

*1 When accessing Universal model QCPU, use the module with a serial number whose first five digits are '10102' or later. If using a module with a serial number whose first five digits are '10101' or earlier, access with 3E/4E frame.

In addition, when accessing a built-in Ethernet port CPU, refer to the following manual.

QnUCPU User's Manual (Communication via Built-in Ethernet Port)

*2 Read/write the devices within the range of AnACPU. The added new devices and their ranges cannot be specified.

*3 The devices that can be registered to E71 is for 1 command out of the three types of commands (06H, 07H, 1AH).

The specified device recently used by any of the above commands is registered to E71.

Appendix 7 Setting Examples

A

This section explains the setting example of the message.

Setting examples of message for serial communication module

The following shows the setting example when using a command from a serial communication module.

Examples of device access in ASCII code (Format 1, 3C frame)

The following shows the setting example when using a command for format 1 and 3C frame.

■Device reading

- Format 1 (sum check enabled), 3C frame
- Access target: QCPU of the connected station (host station)
- Command: Batch read in word units (command: 0401)
- Device: M100 to M131 (for 2 words)

1. Read values from devices.

Request message

| Control code | Frame ID No. | Access route | | | | | | | | | | | | Self-station No. | | | |
|---|--|-------------------------------------|--|-------------|-------------|--|-------------------------------------|--|--|--|--|-------------------------|-------------------------------------|------------------|--|------------------------------|---|
| | | Station No. | | Network No. | | | PC No. | | | | | | | | | | |
| ENQ | 3C frame | Host station | | | | | | | | | | | | | | | |
| | F 9 0 0 0 0 0 0 F F 0 0 0 0 0 0 0 0 | | | | | | | | | | | | | | | | |
| 05H | 46H 39H 30H 30H 30H 30H 30H 30H 46H 46H 30H 30H | | | | | | | | | | | | | | | | |
| — | The subtotal of the additional value of binary data = 22BH | | | | | | | | | | | | | | | | |
| Request data | | | | | | | | | | | | | | | | Sum check code ^{*1} | |
| Command | | Subcommand | | | Device code | | Head device number | | | | | Number of device points | | | | | |
| 0 4 0 1 0 0 0 0 M * | | 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 | | | | | 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 | | | | | | 0 0 0 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 | | | 0 A | |
| 30H 34H 30H 31H 30H 30H 30H 30H 4DH 2AH 30H 30H 30H 31H 30H 30H 30H 30H 30H 30H 30H 30H 30H 32H 30H 41H | | | | | | | | | | | | | | | | | |
| The subtotal of the additional value of binary data = 3DFH | | | | | | | | | | | | | | | | | — |

*1 The addition result of the sum check range: 22BH+3DF=60AH

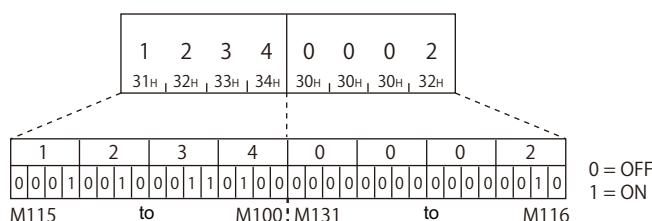
2. Receive the result.

Response message (Normal completion: Response data)

| Control code | Frame ID No. | Access route | | | | | | | | | | | | Self-station No. | | | |
|--|---|--------------|--|-------------|--|--|--------|--|--|--|--|--------------|--|------------------------------|-----|--|---|
| | | Station No. | | Network No. | | | PC No. | | | | | | | | | | |
| STX | 3C frame | Host station | | | | | | | | | | | | | | | |
| | F 9 0 0 0 0 0 0 F F 0 0 0 0 0 0 0 0 | | | | | | | | | | | | | | | | |
| 02H | 46H 39H 30H 30H 30H 30H 30H 30H 46H 46H 30H 30H 30H 31H 30H 30H 30H 30H 30H 30H 30H 30H | | | | | | | | | | | | | | | | |
| — | The subtotal of the additional value of binary data = 22BH | | | | | | | | | | | | | | | | — |
| Response data (read data) ^{*2} | | | | | | | | | | | | Control code | | Sum check code ^{*1} | | | |
| 1 2 3 4 | 0 0 0 2 | | | | | | | | | | | ETX | | B | A | | |
| 31H 32H 33H 34H | 30H 30H 30H 32H | | | | | | | | | | | 03H | | 42H | 41H | | |
| The subtotal of the additional value of binary data = 18FH | | | | | | | | | | | | | | | | | — |

*1 The addition result of the sum check range: 22BH + 18F = 3BAH

*2 The value of read data corresponds to the access target device as follows.



■Device writing

- Format 1 (sum check enabled), 3C frame
- Access target: QCPU of the connected station (host station)
- Command: Batch write in word units (command: 1401)
- Device: M100 to M131 (for 2 words)

1. Write values to devices.

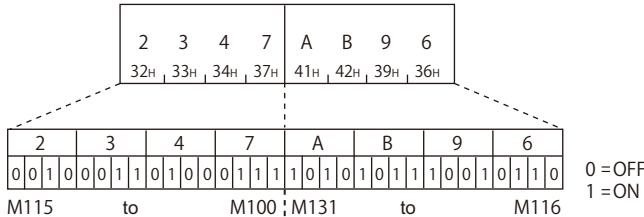
Request message

| Control code | Frame ID No. | Access route | | | | | | | | | | | |
|--------------|--|--------------|--------------|-------------|-----|-----|--------|-----|-----|------------------|-----|-----|-----|
| | | Station No. | | Network No. | | | PC No. | | | Self-station No. | | | |
| ENQ | 3C frame | | Host station | | | | | | | | | | |
| | F | 9 | 0 | 0 | 0 | 0 | F | F | 0 | 0 | 0 | 0 | 0 |
| 05H | 46H | 39H | 30H | 30H | 30H | 30H | 46H | 46H | 30H | 30H | 30H | 30H | 30H |
| — | The subtotal of the additional value of binary data = 22BH | | | | | | | | | | | | |

| Request data | | | | | | | | | | | | | | | | | |
|--|-----|-----|------------|-----|-----|-------------|-----|--------------------|-----|------------------------------|-----|-------------------------|-----|-----|-----|-----|-----|
| Command | | | Subcommand | | | Device code | | Head device number | | | | Number of device points | | | | | |
| 1 | 4 | 0 | 1 | 0 | 0 | 0 | 0 | M | * | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 2 |
| 31H | 34H | 30H | 31H | 30H | 30H | 30H | 30H | 4DH | 2AH | 30H | 30H | 30H | 31H | 30H | 30H | 30H | 32H |
| The subtotal of the additional value of binary data = 3E0H | | | | | | | | | | | | | | | | | |
| Request data | | | | | | | | | | Sum check code ^{*1} | | | | | | | |
| Write data ^{*2} | | | | | | | | | | | | | | | | | |
| 2 | 3 | 4 | 7 | A | B | 9 | 6 | | | C | | D | | | | | |
| 32H | 33H | 34H | 37H | 41H | 42H | 39H | 36H | | | 43H | | 44H | | | | | |
| The subtotal of the additional value of binary data = 1C2H | | | | | | | | | | — | | | | | | | |

*1 The addition result of the sum check range: 22BH + 3E0H + 1C2H = 7CDH

*2 The value of write data corresponds to the access target device as follows.



2. Receive the result.

Response message (Normal completion: No response data)

| Control code | Frame ID No. | Access route | | | | | | | | | | | |
|--------------|--------------|--------------|--------------|-------------|-----|-----|--------|-----|-----|------------------|-----|-----|-----|
| | | Station No. | | Network No. | | | PC No. | | | Self-station No. | | | |
| ACK | 3C frame | | Host station | | | | | | | | | | |
| | F | 9 | 0 | 0 | 0 | 0 | 0 | F | F | 0 | 0 | 0 | 0 |
| 06H | 46H | 39H | 30H | 30H | 30H | 30H | 30H | 46H | 46H | 30H | 30H | 30H | 30H |

Examples of device access in binary code (Format 5, 4C frame)

The following shows the setting example when using a command for format 5 and 4C frame.

■Device reading

- Format 5 (sum check enabled), 4C frame
- Access target: QCPU of the connected station (host station)
- Command: Batch read in word units (command: 0401)
- Device: M100 to M131 (for 2 words)

1. Read values from devices.

Request message

| Control code | | Number of data bytes ^{*1} | Frame ID No. | Access route | | | | | | | Self-station No. | | |
|--|-----|------------------------------------|--------------------|--------------|-------------|-------------------------|------------------------------------|--|-----|------------------------------|------------------|-----|-----|
| | | | | Station No. | Network No. | PC No. | Request destination module I/O No. | Request destination module station No. | | | | | |
| DLE | STX | 18 bytes | 4C frame | Host station | | | | | | | | | |
| 10H | 02H | 12H | 00H | F8H | 00H | 00H | FFH | FFH | 03H | 00H | 00H | | |
| — The subtotal of the additional value of binary data = 30BH | | | | | | | | | | | | | |
| Request data | | | | | | | | | | | | | |
| Command | | Subcommand | Head device number | | Device code | Number of device points | Control code | | | Sum check code ^{*2} | | | |
| 0401 | | 0000 | 100 | | M | 2 | DLE | ETX | 0 | 6 | | | |
| 01H | 04H | 00H | 00H | 64H | 00H | 00H | 90H | 02H | 00H | 10H | 03H | 30H | 36H |
| The subtotal of the additional value of binary data = FBH | | | | | | | | | | | | | |

*1 Frame ID No. (1 byte) + Access route (7 bytes) + Request data (10 bytes)

*2 The addition result of the sum check range: 30BH + FBH = 406H

2. Receive the result.

Response message (Normal completion: Response data)

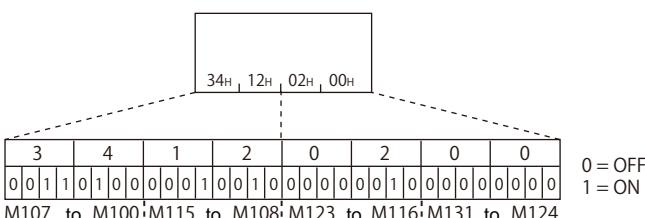
| Control code | | Number of data bytes ^{*1} | Frame ID No. | Access route | | | | | | | Self-station No. |
|--|-----------------|--|---|--------------|--------------|--------|------------------------------------|--|-----|------------------------------|------------------|
| | | | | Station No. | Network No. | PC No. | Request destination module I/O No. | Request destination module station No. | | | |
| DLE | STX | 16 bytes | 4C frame | Host station | | | | | | | |
| 10H | 02H | 10H | 10H | 00H | F8H | 00H | 00H | FFH | FFH | 03H | 00H |
| — | — ^{*2} | The subtotal of the additional value of binary data = 309H | | | | | | | | | |
| Response ID code | | Normal completion code | Response data (read data) ^{*4} | | | | | Control code | | Sum check code ^{*3} | |
| FFFFH | | 0000H | M100 to M115 | | M116 to M131 | | DLE | ETX | 4 | F | |
| FFH | FFH | 00H | 00H | 34H | 12H | 02H | 00H | 10H | 03H | 34H | 46H |
| The subtotal of the additional value of binary data = 246H | | | | | | | | | | | |

*1 Frame ID No. (1 byte) + Access route (7 bytes) + Response ID code (2 bytes) + Normal completion code (2 bytes) + Response data (4 bytes)

*2 Additional code. Calculate the sum check code except for the additional code. (参照 Page 33 Additional code (10H))

*3 The addition result of the sum check range: 309H + 246H = 54FH

*4 The value of read data corresponds to the access target device as follows.



■Device writing

- Format 5 (sum check enabled), 4C frame
- Access target: QCPU of the connected station (host station)
- Command: Batch write in word units (command: 1401)
- Device: M100 to M131 (for 2 words)

1. Write values to devices.

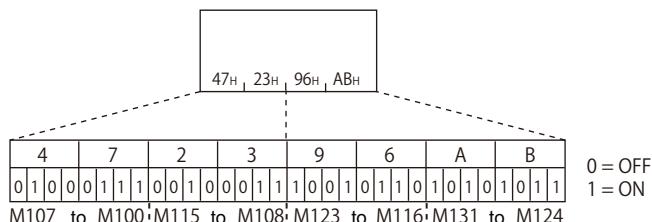
Request message

| Control code | | Number of data bytes ^{*1} | Frame ID No. | Access route | | | | | | | | | | | | | |
|--|-----|------------------------------------|--------------|--------------------|-------------|-------------|-------------------------|------------------------------------|-----|--|-----|------------------|-----|--|--|--|--|
| | | | | Station No. | Network No. | | PC No. | Request destination module I/O No. | | Request destination module station No. | | Self-station No. | | | | | |
| DLE | STX | 22 bytes | 4C frame | Host station | | | | | | | | | | | | | |
| 10H | 02H | 16H 00H | F8H | 00H | 00H | | FFH | FFH | 03H | 00H | 00H | | | | | | |
| — The subtotal of the additional value of binary data = 30FH | | | | | | | | | | | | | | | | | |
| Request data | | | | | | | | | | | | | | | | | |
| Command | | Subcommand | | Head device number | | Device code | Number of device points | Write data ^{*3} | | | | Control code | | | | | |
| 1401 | | 0000 | | 100 | | M | 2 | M100 to M115 | | M116 to M131 | | DLE | ETX | | | | |
| 01H | 14H | 00H | 00H | 64H | 00H | 00H | 90H | 02H | 00H | 47H | 23H | 96H | ABH | | | | |
| The subtotal of the additional value of binary data = 2B6H | | | | | | | | | | | | — | — | | | | |

*1 Frame ID No. (1 byte) + Access route (7 bytes) + Request data (14 bytes)

*2 The addition result of the sum check range: 30FH + 2B6H = 5C5H

*3 The value of write data corresponds to the access target device as follows.



2. Receive the result.

Response message (Normal completion: No response data)

| Control code | | Number of data bytes ^{*1} | Frame ID No. | Access route | | | | | | | | | | | |
|--|-----|------------------------------------|------------------------|--------------|-------------|--------------|--------|------------------------------------|-----|--|-----|------------------|--|--|--|
| | | | | Station No. | Network No. | | PC No. | Request destination module I/O No. | | Request destination module station No. | | Self-station No. | | | |
| DLE | STX | 12 bytes | 4C frame | Host station | | | | | | | | | | | |
| 10H | 02H | 0CH 00H | F8H | 00H | 00H | | FFH | FFH | 03H | 00H | 00H | | | | |
| — The subtotal of the additional value of binary data = 30FH | | | | | | | | | | | | | | | |
| Response ID code | | | Normal completion code | | | Control code | | | | Sum check code ^{*2} | | | | | |
| FFFFH | | | 0000H | | | DLE | ETX | | 0 | 3 | | | | | |
| FFH | FFH | FFH | 00H | 00H | | 10H | 03H | | 30H | 33H | | | | | |
| The subtotal of the additional value of binary data = 1FEH | | | | | | | | — | | | | | | | |

*1 Frame ID No. (1 byte) + Access route (7 bytes) + Response ID code (2 bytes) + Normal completion code (2 bytes)

*2 The addition result of the sum check range: 305H + 1FEH = 503H

Setting examples of message for Ethernet interface module

The following shows the setting example when using a command from an Ethernet interface module.

Examples of device access in ASCII code (3E frame)

The following shows the setting example when using a command for ASCII code and 3E frame.

■Device reading

- ASCII code, 3E frame
- Access target: QCPU of the connected station (host station)
- Command: Batch read in word units (command: 0401)
- Device: M100 to M131 (for 2 words)

1. Read values from devices.

Request message

| Subheader | | Access route | | | | | | | | | | Request data length ^{*1} | | Monitoring timer | | | | | | | |
|--------------------|-----|--------------|--------|------------------------------------|-----|-----|-----|--|-----|-----|-----|-----------------------------------|-----|------------------|-----|-----|-----|-----|-----|-----|-----|
| | | Network No. | PC No. | Request destination module I/O No. | | | | Request destination module station No. | | | | | | | | | | | | | |
| 3E frame (request) | | Host station | | | | | | | | | | 24 | | 4 seconds | | | | | | | |
| 5 | 0 | 0 | 0 | 0 | 0 | F | F | 0 | 3 | F | F | 0 | 0 | 0 | 0 | 1 | 8 | 0 | 0 | 1 | 0 |
| 35H | 30H | 30H | 30H | 30H | 30H | 46H | 46H | 30H | 33H | 46H | 46H | 30H | 30H | 30H | 30H | 31H | 38H | 30H | 30H | 31H | 30H |

| Request data | | | | | | | | | | | | | | | | | | | | |
|--------------|-----|-----|-----|------------|-----|-----|-----|-------------|-----|--------------------|-----|-----|-----|-----|-----|-------------------------|-----|-----|-----|-----|
| Command | | | | Subcommand | | | | Device code | | Head device number | | | | | | Number of device points | | | | |
| 0 | 4 | 0 | 1 | 0 | 0 | 0 | 0 | M | * | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 2 |
| 30H | 34H | 30H | 31H | 30H | 30H | 30H | 30H | 4DH | 2AH | 30H | 30H | 30H | 31H | 30H | 30H | 30H | 30H | 30H | 30H | 32H |

*1 Monitoring timer (4 bytes) + Request data (20 bytes)

2. Receive the result.

Response message (Normal completion: Response data)

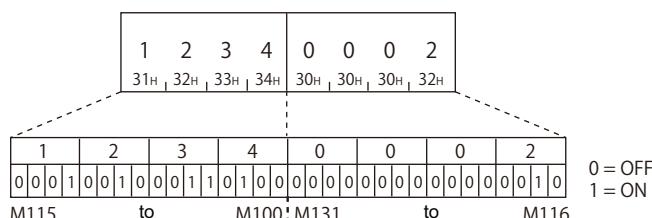
| Subheader | | Access route | | | | | | | | | | Response data length ^{*1} | | End code | | | | | |
|---------------------|-----|--------------|--------|------------------------------------|-----|-----|-----|--|-----|-----|-----|------------------------------------|-----|----------|-----|-----|-----|-----|-----|
| | | Network No. | PC No. | Request destination module I/O No. | | | | Request destination module station No. | | | | | | | | | | | |
| 3E frame (response) | | Host station | | | | | | | | | | 12 | | 0000H | | | | | |
| D | 0 | 0 | 0 | 0 | 0 | F | F | 0 | 3 | F | F | 0 | 0 | 0 | C | 0 | 0 | 0 | 0 |
| 44H | 30H | 30H | 30H | 30H | 30H | 46H | 46H | 30H | 33H | 46H | 46H | 30H | 30H | 30H | 43H | 30H | 30H | 30H | 30H |

Response data (read data)^{*2}

| | | | | | | | | | | | | | | | | | | |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| 1 | 2 | 3 | 4 | 0 | 0 | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 |
| 31H | 32H | 33H | 34H | 30H | 30H | 30H | 32H | 30H | 32H |

*1 End code (4 bytes) + Response data (8 bytes)

*2 The value of read data corresponds to the access target device as follows.



■ Device writing

- ASCII code, 3E frame
 - Access target: QCPU of the connected station (host station)
 - Command: Batch write in word units (command: 1401)
 - Device: M100 to M131 (for 2 words)

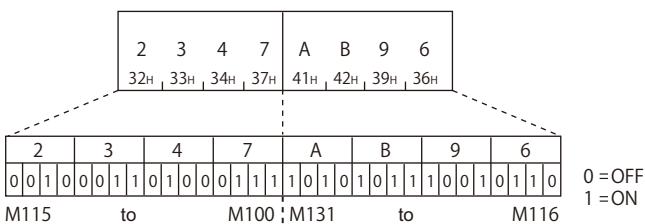
1. Write values to devices.

Request message

| Subheader | | Access route | | | | | | | | | | | | Request data length ^{*1} | | | | Monitoring timer | | | | | |
|--------------------------------|-----|--------------|-----|-------------------|-----|------------------------------------|-----|--------------------|-----|--|-----|---------------------------|-----|-----------------------------------|-----|-----|-----|--------------------------------|-----|-----------|-----|-----|-----|
| | | Network No. | | PC No. | | Request destination module I/O No. | | | | Request destination module station No. | | | | | | | | | | | | | |
| 3E frame (request) | | | | Host station | | | | | | | | | | | | 32 | | | | 4 seconds | | | |
| 5 | 0 | 0 | 0 | 0 | 0 | 0 | F | F | 0 | 3 | F | F | 0 | 0 | 0 | 0 | 0 | 2 | 0 | 0 | 0 | 1 | 0 |
| 35H | 30H | 30H | 30H | 30H | 30H | 30H | 46H | 46H | 30H | 33H | 46H | 46H | 30H | 30H | 30H | 30H | 30H | 32H | 30H | 30H | 30H | 31H | 30H |
| Request data | | | | | | | | | | | | | | | | | | | | | | | |
| Command | | | | Subcommand | | | | Device code | | | | Head device number | | | | | | Number of device points | | | | | |
| 1 | 4 | 0 | 1 | 0 | 0 | 0 | 0 | M | * | | | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 |
| 31H | 34H | 30H | 31H | 30H | 30H | 30H | 30H | 4DH | | 2AH | | 30H | 30H | 30H | 31H | 30H | 30H | 30H | 30H | 30H | 30H | 30H | 32H |
| Request data | | | | | | | | | | | | | | | | | | | | | | | |
| Write data^{*2} | | | | | | | | | | | | | | | | | | | | | | | |
| 2 | | 3 | | | 4 | | | 7 | | | A | | | B | | | 9 | | | 6 | | | |
| 32H | | 33H | | | 34H | | | 37H | | | 41H | | | 42H | | | 39H | | | 36H | | | |

*1 Monitoring timer (4 bytes) + Request data (28 bytes)

*2 The value of write data corresponds to the access target device as follows.



2. Receive the result.

Response message (Normal completion: No response data)

*1 End code (4 bytes)

Examples of device access in binary code (3E frame)

The following shows the setting example when using a command for binary code and 3E frame.

■Device reading

- Binary code, 3E frame
- Access target: QCPU of the connected station (host station)
- Command: Batch read in word units (command: 0401)
- Device: M100 to M131 (for 2 words)

1. Read values from devices.

Request message

| Subheader | | Access route | | | | Request data length ^{*1} | | Monitoring timer | | |
|---------------------|-----|--------------|--------|------------------------------------|--|-----------------------------------|-------------|------------------|-------------------------|-----|
| | | Network No. | PC No. | Request destination module I/O No. | Request destination module station No. | | | | | |
| 3E frame (request) | | Host station | | | | 12 | | 4 seconds | | |
| 50H | 00H | 00H | FFH | FFH | 03H | 00H | 0CH | 00H | 10H | 00H |
| Request data | | | | | | | | | | |
| Command | | Subcommand | | Head device number | | | Device code | | Number of device points | |
| 0401 | | 0000 | | 100 | | | M | | 2 | |
| 01H | 04H | 00H | 00H | 64H | 00H | 00H | 90H | 02H | 00H | |

*1 Monitoring timer (2 bytes) + Request data (10 bytes)

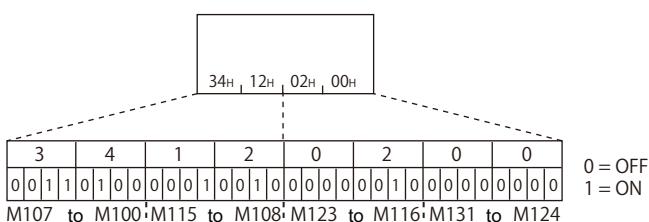
2. Receive the result.

Response message (Normal completion: Response data)

| Subheader | | Access route | | | | Response data length ^{*1} | | End code | | |
|---|-----|--------------|--------|------------------------------------|--|------------------------------------|-----|----------|-----|-----|
| | | Network No. | PC No. | Request destination module I/O No. | Request destination module station No. | | | | | |
| 3E frame (response) | | Host station | | | | 6 | | 0000H | | |
| D0H | 00H | 00H | FFH | FFH | 03H | 00H | 06H | 00H | 00H | 00H |
| Response data (read data)^{*2} | | | | | | | | | | |
| M100 to M115 | | | | | M116 to M131 | | | | | |
| 34H | | | 12H | | 02H | | | 00H | | |

*1 End code (2 bytes) + Response data (4 bytes)

*2 The value of read data corresponds to the access target device as follows.



■Device writing

- Binary code, 3E frame
- Access target: QCPU of the connected station (host station)
- Command: batch write in word units (command: 1401)
- Device: M100 to M131 (for 2 words)

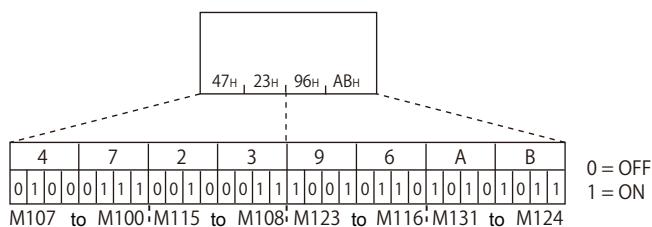
1. Write values to devices.

Request message

| Subheader | | Access route | | | | | Request data length ^{*1} | Monitoring timer | |
|---------------------|-----|--------------|--------------------|------------------------------------|--|-------------------------|-----------------------------------|------------------|--------------|
| | | Network No. | PC No. | Request destination module I/O No. | Request destination module station No. | | | | |
| 3E frame (request) | | Host station | | | | | 16 | 4 seconds | |
| 50H | 00H | 00H | FFH | FFH | 03H | 00H | 10H | 00H | 10H |
| Request data | | | | | | | | | |
| Command | | Subcommand | Head device number | | Device code | Number of device points | Write data ^{*2} | | |
| 1401 | | 0000 | 100 | | M | 2 | M100 to M115 | | M116 to M131 |
| 01H | 14H | 00H | 00H | 64H | 00H | 00H | 90H | 02H | 00H |
| 47H | | 23H | | 96H | | ABH | | | |

*1 Monitoring timer (2 bytes) + Request data (14 bytes)

*2 The value of write data corresponds to the access target device as follows.



2. Receive the result.

Response message (Normal completion: No response data)

| Subheader | | Access route | | | | | Response data length ^{*1} | End code | |
|---------------------|-----|--------------|--------|------------------------------------|--|-----|------------------------------------|----------|-----|
| | | Network No. | PC No. | Request destination module I/O No. | Request destination module station No. | | | | |
| 3E frame (response) | | Host station | | | | | 2 | 0000H | |
| D0H | 00H | 00H | FFH | FFH | 03H | 00H | 02H | 00H | 00H |

*1 End code (2 bytes)

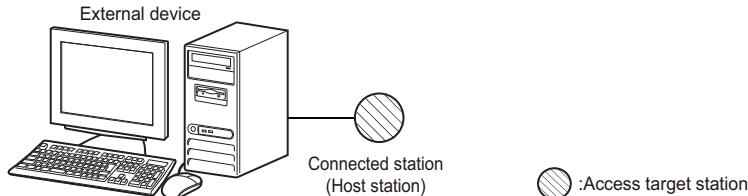
Setting examples of access route

The following shows the setting example of each access route for each frame. For the details of each setting item, refer to the following section.

 Page 48 Detail of data to be set

When accessing the connected station (host station)

The following shows the setting example for accessing the connected station (host station).



—: Unnecessary

| Access target | Frame | Station No. | Network No. | PC No. | Request destination module I/O No. | Request destination module station No. | Self-station No. |
|--|-------|-------------|-------------|--------|------------------------------------|--|------------------|
| Connected station (host station) | 4C | 00 | 00 | FF | 03FF | 00 | 00 |
| | 3C | 00 | 00 | FF | — | — | 00 |
| | 2C | 00 | — | — | — | — | 00 |
| | 1C | 00 | — | FF | — | — | — |
| | 4E/3E | — | 00 | FF | 03FF | 00 | — |
| | 1E | — | — | FF | — | — | — |
| Multiple CPU system (Multiple CPU No.1) | 4C | 00 | 00 | FF | 03E0 | 00 | 00 |
| | 4E/3E | — | 00 | FF | 03E0 | 00 | — |
| Redundant system (Control system CPU) | 4C | 00 | 00 | FF | 03D0 | 00 | 00 |
| | 4E/3E | — | 00 | FF | 03D0 | 00 | — |

Specify the CPU module of an access target in the request destination module I/O No.

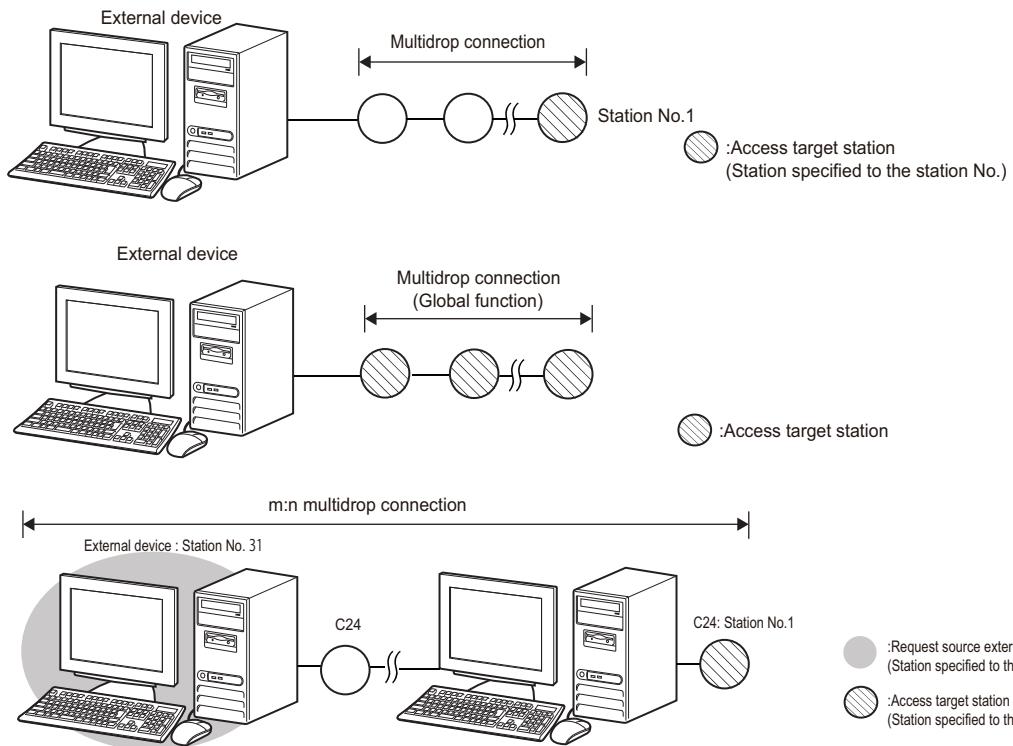
 Page 54 Accessing multiple CPU system, redundant system

When connecting with a multidrop connection

When accessing a multidrop connection station, specify the station No. of an access target to the station No.

When requiring to distinguish request sources in a m:n connection, specify the station No. of an external device to the self-station No.

The following shows setting examples for accessing a multidrop connection station.



—: Unnecessary

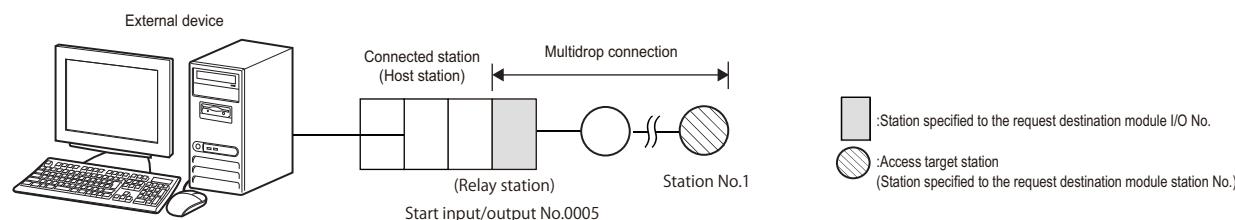
| Access target | Frame | Station No. | Network No. | PC No. | Request destination module I/O No. | Request destination module station No. | Self-station No. |
|---|-------|-------------|-------------|--------|------------------------------------|--|------------------|
| Multidrop connection station • Station No.1 | 4C | 01 | 00 | FF | 03FF | 00 | 00 |
| | 3C | 01 | 00 | FF | — | — | 00 |
| | 2C | 01 | — | — | — | — | 00 |
| | 1C | 01 | — | FF | — | — | — |
| All stations connected with a multidrop connection (Global function) | 4C | FF | 00 | FF | 03FF | 00 | 00 |
| | 3C | FF | 00 | FF | — | — | 00 |
| | 2C | FF | — | — | — | — | 00 |
| Multidrop connection station (At m:n connection) • Station No.1 • Station No.31 (1FH) of the external device | 4C | 01 | 00 | FF | 03FF | 00 | 1F |
| | 3C | 01 | 00 | FF | — | — | 1F |
| | 2C | 01 | — | — | — | — | 1F |

■When accessing from an Ethernet interface module

When accessing a multidrop connection station with the frames (4E frame, 3E frame) for the Ethernet interface module, specify the start input/output number of a multidrop connection source module (relay station) and the station No. of an access target to the request destination I/O No. and the request destination module station No. For the request destination I/O No., specify the value obtained by dividing the start input/output number by 16 in 4 digits (hexadecimal).

The following shows the setting example of the cases below.

- Multidrop connection source module of the relay station: Start input/output number 0050H (request destination module I/O No.0005H)
- Multidrop connection station of the access target: Station No.1



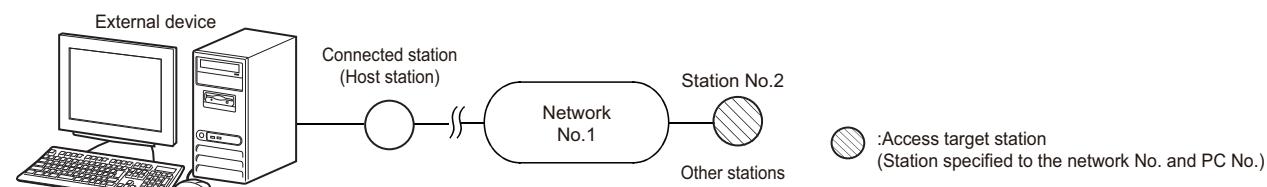
—: Unnecessary

| Access target | Frame | Station No. | Network No. | PC No. | Request destination module I/O No. | Request destination module station No. | Self-station No. |
|------------------------------|-------|-------------|-------------|--------|------------------------------------|--|------------------|
| Multidrop connection station | 4E/3E | — | 00 | FF | 0005 | 01 | — |

When accessing via network

When accessing other stations via network, specify the network No. and station No. of an access target network module to the network No. and PC No.

The following shows setting examples for accessing other stations via network (network No. 1, station No.2) without a multidrop connection.



—: Unnecessary

| Access target | Frame | Station No. | Network No. | PC No. | Request destination module I/O No. | Request destination module station No. | Self-station No. |
|--|-------|-------------|-------------|--------|------------------------------------|--|------------------|
| Other station via network • Network No.1 • Station No.2 | 4C | 00 | 01 | 02 | 03FF | 00 | 00 |
| | 3C | 00 | 01 | 02 | — | — | 00 |
| | 4E/3E | — | 01 | 02 | 03FF | 00 | — |
| Other station accessed via the network module set in the "Valid Module During Other Station Access" setting. • Station No.2 | 4C | 00 | FE | 02 | 03FF | 00 | 00 |
| | 3C | 00 | FE | 02 | — | — | 00 |
| | 1C | 00 | — | 02 | — | — | — |
| | 4E/3E | — | FE | 02 | 03FF | 00 | — |
| | 1E | — | — | 02 | — | — | — |

When connecting with a multidrop link and via network

The following shows setting examples for connecting with a multidrop connection and via network.

■Other stations via network from a multidrop connection

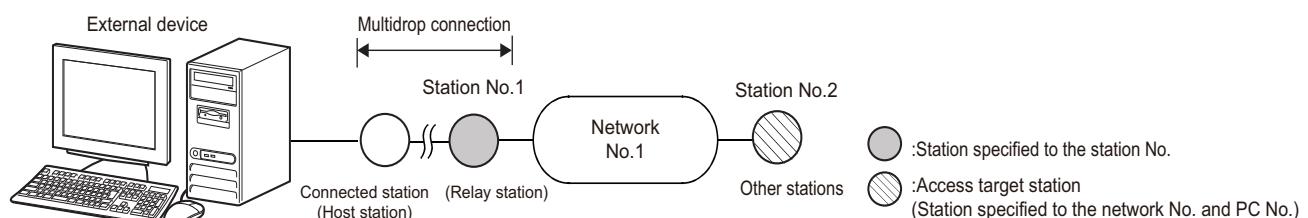
Specify the setting for a multidrop connection including the connected station (host station) and the setting for a connection via network each.

Specify the station No. of a multidrop connection station for a relay to the station No.

Specify the network No. and station No. of an access target to the network No. and PC No.

The following shows the setting example of the cases below.

- Multidrop connection station of the relay station: Station No.1
- Network module of the access target: Network No.1, station No.2



—: Unnecessary

| Access target | Frame | Station No. | Network No. | PC No. | Request destination module I/O No. | Request destination module station No. | Self-station No. |
|--|-------|-------------|-------------|--------|------------------------------------|--|------------------|
| Other stations via network from a multidrop connection | 4C | 01 | 01 | 02 | 03FF | 00 | 00 |
| | 3C | 01 | 01 | 02 | — | — | 00 |

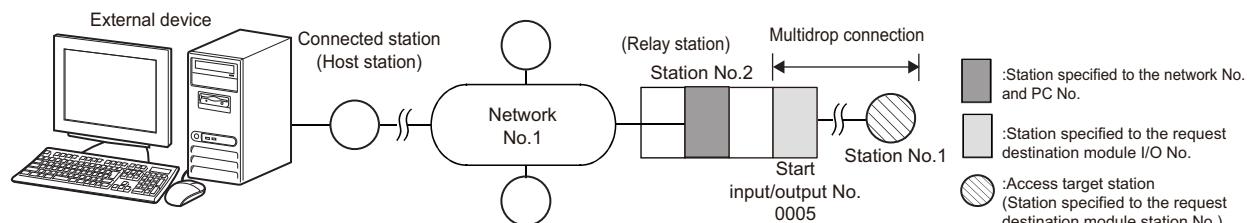
■Multidrop connection station via network

For accessing a multidrop connection station in the target via network, specify the start input/output number of a multidrop connection source module and the station No. of an access target to the request destination module I/O No. and the request destination module station No. For the request destination module I/O No., specify the value obtained by dividing the start input/output number by 16 in 4 digits (hexadecimal).

Specify the network No. and station No. of a network module for a relay station to the network No. and PC No.

The following shows the setting example of the cases below.

- Network module for the relay station: Network No.1, station No.2
- Multidrop connection source module of the relay station: Start input/output number 0050H (request destination module I/O No.0.0005H)
- Multidrop connection station of the access target: Station No.1



—: Unnecessary

| Access target | Frame | Station No. | Network No. | PC No. | Request destination module I/O No. | Request destination module station No. | Self-station No. |
|--|-------|-------------|-------------|--------|------------------------------------|--|------------------|
| Multidrop connection station via network | 4C | 00 | 01 | 02 | 0005 | 01 | 00 |
| | 4E/3E | — | 01 | 02 | 0005 | 01 | — |

MEMO

INDEX

0 to 9

| | |
|--------------------|-------|
| 1C frame | 26,45 |
| 1E frame | 38,47 |
| 2C frame | 26,44 |
| 3C frame | 26,44 |
| 3E frame | 38,46 |
| 4C frame | 26,43 |
| 4E frame | 38,46 |

A

| | |
|---------------------------------|-------|
| Accessible module | 18 |
| Accessible range | 18 |
| ACPU monitoring timer | 388 |
| Additional code | 33 |
| ASCII code | 26,37 |

B

| | |
|-------------------------|-------------|
| Binary code | 26,37 |
| Buffer memory | 147,377,426 |

C

| | |
|--|-----|
| Clear error information | 181 |
| Cluster | 283 |
| Command | 58 |
| Command list | 58 |
| Connected station (host station) | 14 |
| Control code | 32 |

D

| | |
|--|---------------|
| Device | 14,63,344,391 |
| Device code | 64,345,391 |
| Device extension specification | 432 |
| Drive memory | 283 |

E

| | |
|----------------------------------|---------|
| Error code | 36,342 |
| Error information | 42 |
| Extended file register | 365,412 |

F

| | |
|----------------------|---------|
| File | 184,291 |
| Format | 26 |
| Format 1 | 27 |
| Format 2 | 28 |
| Format 3 | 29 |
| Format 4 | 30 |
| Format 5 | 31 |
| Frame | 26,38 |
| Frame ID No. | 34 |

G

| | |
|---------------------------|-----|
| Global function | 249 |
|---------------------------|-----|

H

| | |
|------------------|----|
| Header | 40 |
|------------------|----|

I

| | |
|---------------------------------------|-----|
| Index modification | 440 |
| Intelligent function module | 153 |

K

| | |
|-------------------|-----|
| Keyword | 284 |
|-------------------|-----|

L

| | |
|-------------------------|---------|
| Label | 119 |
| LED OFF | 181,183 |
| Link device | 14 |
| Loopback test | 179,383 |

M

| | |
|-----------------------------------|-------|
| Message format | 27,39 |
| Mode switching function | 253 |
| Monitoring timer | 41 |
| Multiple CPU systems | 455 |

N

| | |
|--|-----|
| Network No. | 50 |
| Number of processing per one communication | 459 |

O

| | |
|---------------------------------|-----|
| On-demand function | 274 |
| Online program change | 457 |
| Other station | 14 |

P

| | |
|---|--------|
| Password | 189 |
| PC No. | 50,387 |
| Processing time | 448 |
| Processing timing of CPU module | 21 |
| Programmable controller CPU monitoring function | 260 |

R

| | |
|---|-----|
| Relay station | 14 |
| Remote operation | 167 |
| Remote password | 174 |
| Request destination module I/O number | 53 |
| Request destination module station No. | 53 |
| Request message | 14 |
| Response message | 14 |

S

| | |
|---|--------|
| Self-station No. | 56 |
| Serial communication function | 19,454 |
| Serial number | 40 |

| | |
|-----------------------------------|---------|
| Special function module | 377,426 |
| Station No. | 48 |
| Subheader | 40 |
| Sum check code | 34 |

T

| | |
|--|-----|
| Transmission sequence initialization function . . . | 252 |
| Transmission time when accessing other stations | 450 |

I

U

| | |
|----------------------|-----|
| User frame | 240 |
|----------------------|-----|

REVISIONS

*The manual number is given on the bottom left of the back cover.

| Revision date | *Manual number | Description |
|----------------------------------|--|---|
| December 1999 to July 2013 | SH(NA)-080008-A to SH(NA)-080008-S | Due to the transition to the e-Manual, the details of revision have been deleted. |
| June 2014 | SH(NA)-080008-T | Complete revision (layout change) ■Added models RJ71C24, RJ71C24-R2, RJ71C24-R4 |
| July 2014 | SH(NA)-080008-U | Minor correction. |
| August 2015 | SH(NA)-080008-V | ■Added or modified parts Section 6.2, Section 8.1, Section 8.2, Section 8.3, Section 8.4, Section 8.5, Chapter 9, Section 12.4, Appendix 7 |

Japanese manual number: SH-080003-AB

This manual confers no industrial property rights or any rights of any other kind, nor does it confer any patent licenses. Mitsubishi Electric Corporation cannot be held responsible for any problems involving industrial property rights which may occur as a result of using the contents noted in this manual.

© 1999 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 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 trademark of Xerox Corp.

SD logo and SDHC logo are trademarks or registered trademarks of SD-3C, LLC.

All other company names and product names used in this manual are trademarks or registered trademarks of their respective companies.



SH(NA)-080008-V(1508)KWIX

MODEL: MC-PROTOCOL-R-E

MODEL CODE: 13JF89

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.