Q Corresponding MELSEC
Communication Protocol

Reference Manual

MITSUBISHI


Mitsubishi Programmable
Logic Controller

MELSEC-Q

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

# • SAFETY PRECAUTIONS •

(Always read these instructions before using this equipment.)

Before using this product, please read this manual and the relevant manuals introduced in this manual carefully and pay full attention to safety to handle the product correctly.
The instructions given in this manual are concerned with this product. For the safety instructions of the programmable controller system, please read the user's manual for the CPU module to use.
In this manual, the safety instructions are ranked as "DANGER" and "CAUTION".

| | |
|---|---|
| ◇! DANGER | Indicates that incorrect handling may cause hazardous conditions, resulting in death or severe injury. |
| ⚠ CAUTION | Indicates that incorrect handling may cause hazardous conditions, resulting in medium or slight personal injury or physical damage. |

Note that the ⚠CAUTION level may lead to a serious consequence according to the circumstances. Always follow the instructions of both levels because they are important to personal safety.

Please save this manual to make it accessible when required and always forward it to the end user.

## [Design Precautions]

### ◇! DANGER

- When controlling a PLC (modifying data) in operation by connecting computer peripheral devices to the CPU module or connecting personal computers to the intelligent function modules, configure an interlocking circuit in a sequence program so that the safety of the overall system is maintained. Also, before performing other control operations (program modifications and operating-status modifications (status control)) on the PLC in operation, be sure to read the manual thoroughly and confirm the safety. Especially in the above-mentioned control operations, which are performed from an external device to a remote PLC, problems arising on the PLC side may not be dealt with immediately due to abnormal data communication. In addition to configuring an interlocking circuit in a sequence program, determine how the system should handle data-communication errors between the PLC CPU and external devices.
- Do not write any data in the "system area" of the buffer memory of the intelligent function module.
  Also, do not output (turn on) the "use prohibited" signal, which is one of the output signals from the PLC CPU to the intelligent function module.
  If data is written to the "system area" or the "use prohibited" signal is output, the PLC system may malfunction.

## [Design Precautions]

### ⚠ CAUTION

- While registering setting values of the buffer memory to the Flash ROM in order to use the serial communication module, do not turn off the power to the station in which the module is loaded or reset the PLC CPU.
  Powering off the station on which the module is loaded or resetting the PLC CPU may result in inconsistent data contents in the Flash ROM. In this case, setting values in the buffer memory must be reset or reregistered to the Flash ROM. This may also lead to module failures and malfunctions.

## [Operation Precautions]

### ⚠ CAUTION

- Please read the User’s Manual carefully and confirm complete safety before performing control operations to the operating PLC (especially modifications of data, programs, and operation status (status control)) by connecting personal computers, etc., to the intelligent function module.
  Improper data modification, program modification or status control may result in a system malfunction, mechanical failure or accident.


A - 2

REVISIONS

＊The manual number is given on the bottom left of the back cover.

| Print Date | ＊Manual Number | Revision |
|---|---|---|
| Dec., 1999 | SH(NA)-080008-A | First edition |
| Oct., 2000 | SH(NA)-080008-B | Reflect the contents of the function version B.<br>Put Windows® base software products together from Mitsubishi Programmable Logic Controller MELSEC series to Mitsubishi integrated FA software MELSOFT series. Standardize the name from software package (GPP function) to product name (GX Developer).<br>Correction<br>Contents, Entire manual (change MELSECNET/10H to MELSECNET/H), About Manuals, About the Generic Terms and Abbreviations, Section 1.1(5), Section 2.2, 2.4, 2.6, 2.7 (g), 2.9, 2.10, Section 3.1.1, 3.1.2 (figure), 3.1.3, 3.1.4 (figure), 3.1.5 (figure), 3.1.6 (5), Remark, 3.2 (table), 3.3.1 (table 6.3), 3.3.8 (POINT (screen)), 3.8.1 (2)(a), 3.8.2 (4), 3.8.5, 3.8.6, Section 4.1 (figure), Section 5.1.1, 5.1.2 (figure), 5.1.3, 5.1.5 (table)*3, 5.2.1 (2) (table), 5.2.8, 5.4.2, 5.4.3, Section 6.1.1, 6.1.2 (figure), 6.1.3, 6.2*3 (table), 6.3.1 (2) (table), 6.3.8 POINT (3), 6.4.9POINT (3), Appendix 1, 2 |
| Jun., 2001 | SH(NA)-080008-C | Standardize the name from utility package (QSCU) to product name (GX Configurator-SC).<br>Correction<br>About Manuals, The Manual's Use and Structure, About the Generic Terms and Abbreviations, Meanings and Descriptions of Terminology, Section 1.1(5), Section 2.2(table), 2.6.1(1)(table), 2.6.2(table), 2.10(4), Section 3.2(table), 3.3.1(table), 3.3.2, 3.3.3, 3.3.7, 3.3.8(2)(3), 3.3.9(2)(3)(4), 3.3.10(4), 3.4.3, 3.5, 3.6, 3.8.5(2), 3.8.16, 3.17.2, Section 4.4, Section 5.2.8(2)POINT, 5.5, Section 6.2 ＊3, 6.3.1(2), Appendix 3.1<br>Addition<br>Section 2.11 |
| Jan., 2003 | SH(NA)-080008-D | Additional model<br>QJ71C24N,QJ71C24N-R2, QJ71C24N-R4<br>Correction<br>The Manual's Use and Structure, About the Generic Terms and Abbreviations, Section 1.1(5), Section 2.5(3), 2.6.1, 2.8, Section 3.1.2(1)(5), 3.1.3(3), 3.2, 3.3.1, 3.3.10(5)POINT, 3.5.2, 3.6.1POINT, 3.6.7, 3.13, Section 5.1.5, 5.2.1POINT, 6.1.2(1) |
| Apr., 2003 | SH(NA)-080008-E | Additional model<br>QJ71E71-B5<br>Deleted model<br>QJ71E71<br>Correction<br>About the Generic Terms and Abbreviations, Section 2.6.1(2), Section 3.5.2, Section 6.1.3(4) |
| Jun., 2004 | SH(NA)-080008-F | Correction<br>About the Generic Terms and Abbreviations, Section 2.6.1, 2.6.3, 2.10(4)(b), 3.1.3(2)(3), 3.1.6(5), 3.2, 3.3.1(3), 3.5.2(1), 3.6.7, 3.8.1, 3.8.11 POINT, Section 6.1.3(4), 6.3.1(2) |

＊ The manual number is given on the bottom left of the back cover.

| Print Date | ＊ Manual Number | Revision |
|---|---|---|
| | | |

Japanese Manual Version SH-080003-I

INTRODUCTION

Thank you for purchasing the MELSEC-Q series PLC.
Before using the equipment, please read this manual carefully to develop full familiarity with the functions and performance of the Q series PLC you have purchased, so as to ensure correct use.
Please forward a copy of this manual to the end user.

CONTENTS (This manual)

(Related Manual-1) ··· Q corresponding Ethernet Interface Module User's Manual (Basic)

SH-080009-I




A - 10

(Related Manual-1) ··· Q corresponding Ethernet Interface Module User's Manual (Basic)

SH-080009-I



(Related Manual-2) ··· Q Corresponding Ethernet Interface Module User's Manual (Web function)

SH-080180-B




A - 11

(Related Manual-3) … Q Corresponding Serial Communication Module User's Manual (Basic)
SH-080006-G

(Related Manual-4) … Q Corresponding Serial Communication Module User's Manual (Application)
SH-080007-F

(Related Manual-4) … Q Corresponding Serial Communication Module User's Manual (Application)
SH-080007-F

About Manuals

The following manuals are also related to this product.

If necessary, order them by quoting the details in the tables below.

Related Manuals

| Manual Name | Manual Number (Model Code) |
|---|---|
| Q Corresponding Serial Communication Module User's Manual (Basic)<br>This manual provides an 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 the Q-Corresponding Serial Communication Module. (Sold separately) | SH-080006<br>(13JL86) |
| Q Corresponding Serial Communication Module User's Manual (Application)<br>This manual contains information on how to perform data communication with external devices using the module's special functions. (Sold separately) | SH-080007<br>(13JL87) |
| Q Corresponding Ethernet Interface Module User's Manual (Basic)<br>This manual contains information on the specifications of the Ethernet module, the procedures for data communications with external devices, circuit connection (open/close), fixed buffer exchange, random access buffer exchange, and the troubleshooting. (Sold separately) | SH-080009<br>(13JL88) |
| Q Corresponding Ethernet Interface Module User's Manual (Web function)<br>This manual explains how to use the Web function of the Ethernet module. (Sold separately) | SH-080180<br>(13JR40) |

- How to use this manual

This manual explains the communication functions via the MC protocol, with each section covering a specific function.

## (1) To learn about the communication functions using the MC protocol

- A summary of the data communication using the MC protocol is explained in Section 2.1 (2)
- The main data communication functions using the MC protocol are explained in Section 2.1 (3)

## (2) To learn about the types and access ranges of frames for the MC protocol

(a) To learn how to read the frame names for the MC protocol

- How to read the frame names is explained in Chapter 1.
- The correspondence between the communication frames of the conventional modules and those for the MC protocol are explained in Section 2.1.

(b) To learn about the types and access ranges of frames

- The types of frames that can be used for the serial communication modules and Ethernet Interface modules are explained in Section 2.2.1.
- The application and access range of each frame are explained in Section 2.2.2 and succeeding sections.

## (3) When communicating through the MC protocol

- The common information on communication through the MC protocol is explained in Chapter 2, Section 2.3 and succeeding sections.
- The commands, message formats, and control procedures for A compatible 1C frames are explained in Chapter 5.
- The commands, message formats, and control procedures for A compatible 1E frames are explained in Chapter 6.
- The commands, message formats, and control procedures for QnA compatible 2C frames are explained in Chapter 4.
- The commands, message formats, and control procedures for QnA compatible 3E, QnA compatible 3C, and QnA compatible 4C frames are explained in Chapter 3.


A - 16

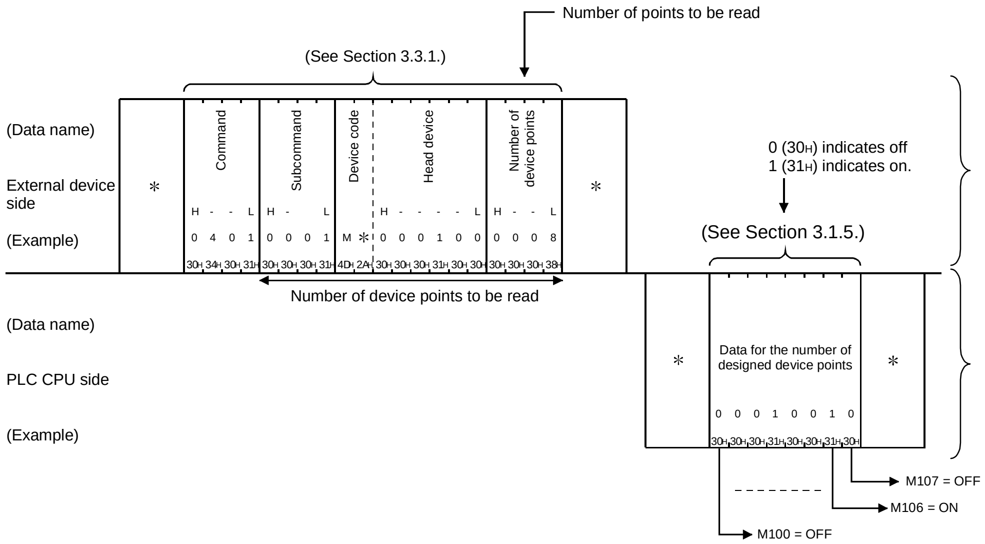
• Structure of this manual

The explanations of the message formats and the control procedures while communicating through the MC protocol are given in the following format:

[Control procedure]

(1) Reading eight points of data from internal relays M100 to M107 in communication in ASCII code

This is a description of the request from the external device side for the control procedure explained in the example.




This diagram illustrates the sequence of data items in the command message to be sent by the external device.

This diagram illustrates the sequence of data items in the response message (the message providing the processing result) to be sent by the PLC CPU side for the request sent from the external device side.

An asterisk (*) in both the command message and the response message indicates the same order of data items as in the control procedures when other commands are used.

(The head and end sections of the command and response messages.)

Confirm the order of data items in the fields marked with "*" by referring to the reference sections listed below.

The order of data items in the fields marked with "*" differs between Ethernet interface modules and serial communication modules.

(1) When communicating via an Ethernet Interface module
A compatible 1E frames : See Section 6.1.
QnA compatible 3E frames : See Section 3.1.2.

(2) When communicating via a serial communication module
A compatible 1C frames : See Section 5.1.
QnA compatible 2C frames : See Section 4.1.
QnA compatible 3C frames : See Section 3.1.4.
QnA compatible 4C frames : See Section 3.1.5.

About the Generic Terms and Abbreviations

This manual uses the following generic terms and abbreviations to explain the serial communication modules, unless otherwise specified.

## (1) Generic terms and abbreviations of relevant modules

In this manual, the following generic terms and abbreviations are used to indicate the serial communication modules and PLC CPU modules, etc. The model names of target modules are used to identify the specific models.

<table>
<tr><th>Generic term<br>/abbreviation</th><th colspan="2">Description of generic term/abbreviation</th></tr>
<tr><td>ACPU<br>A series CPU</td><td colspan="2">Generic term for AnNCPU, AnACPU, AnUCPU.<br>＊ A series PLC CPUs accessible from external devices using the MC protocol communication functions.</td></tr>
<tr><td>AnACPU</td><td colspan="2">Generic term for A2ACPU, A2ACPU-S1, A2ACPUP21/R21, A2ACPUP21/R21-S1, A3ACPU, A3ACPUP21/R21.</td></tr>
<tr><td>AnA/AnU/QnACPU</td><td colspan="2">Generic term for AnACPU, AnUCPU, QnACPU.</td></tr>
<tr><td>AnNCPU</td><td colspan="2">Generic term for A1NCPU, A1NCPUP21/R21, A2NCPU, A2NCPU-S1, A2NCPUP21/R21, A2NCPUP21/R21-S1, A3NCPU, A3NCPUP21/R21.</td></tr>
<tr><td>AnUCPU</td><td colspan="2">Generic term for A2UCPU, A2UCPU-S1, A2ASCPU, A2ASCPU-S1, A3UCPU, A4UCPU.</td></tr>
<tr><td>AnU/QnACPU</td><td colspan="2">Generic term for AnUCPU, QnACPU.</td></tr>
<tr><td>LP25/BR15</td><td colspan="2">Generic term for AJ72LP25, AJ72BR15.</td></tr>
<tr><td>QC24</td><td colspan="2">Generic term for AJ71QC24, AJ71QC24-R2, AJ71QC24-R4, A1SJ71QC24, A1SJ71QC24-R2.</td></tr>
<tr><td>QC24N</td><td colspan="2">Generic term for AJ71QC24N, AJ71QC24N-R2, AJ71QC24N-R4, A1SJ71QC24N, A1SJ71QC24N-R2.</td></tr>
<tr><td>QC24(N)</td><td colspan="2">Generic term for QC24, QC24N.</td></tr>
<tr><td>QCPU<br>Q series CPU</td><td>Q Mode</td><td>Generic term for Q00JCPU, Q00CPU, Q01CPU, Q02CPU, Q02HCPU, Q06HCPU, Q12HCPU, Q25HCPU, Q12PHCPU, Q25PHCPU, Q12PRHCPU, Q25PRHCPU.</td></tr>
<tr><td>QCPU-A</td><td>A Mode</td><td>Generic term for Q02CPU-A, Q02HCPU-A, Q06HCPU-A.</td></tr>
<tr><td>QCPU station</td><td colspan="2">Abbreviation for the PLC with QCPU installed.</td></tr>
<tr><td>QE71</td><td colspan="2">Generic term for AJ71QE71N-T, AJ71QE71N-B5, AJ71QE71N-B2, A1SJ71QE71N-T, A1SJ71QE71N-B5, A1SJ71QE71N-B2.</td></tr>
<tr><td>QLP21/QBR11</td><td colspan="2">Generic term for AJ71QLP21, AJ71QBR11.</td></tr>
<tr><td>QLP25/QBR15</td><td colspan="2">Generic term for AJ72QLP25 (G), AJ72QBR15, A1SJ72QLP25, A1SJ72QBR15.</td></tr>
<tr><td>QnACPU<br>QnA series CPU</td><td colspan="2">Generic term for Q2ACPU, Q2ACPU-S1, Q2ASCPU, Q2ASCPU-S1, Q2ASHCPU, Q2ASHCPU-S1, Q3ACPU, Q4ACPU, Q4ARCPU.</td></tr>
<tr><td>QnACPU station</td><td colspan="2">Abbreviation for the PLC with QnACPU installed.</td></tr>
<tr><td>Q/QnACPU</td><td colspan="2">Generic term for QCPU, QnACPU</td></tr>
<tr><td>Q series C24 (C24)</td><td colspan="2">Abbreviation for QJ71C24N, QJ71C24N-R2, QJ71C24N-R4, QJ71C24 and QJ71C24-R2 serial communication modules (shown as "C24" in diagrams)</td></tr>
<tr><td>Q series E71 (E71)<br>Ethernet module</td><td colspan="2">Abbreviation for QJ71E71-100, QJ71E71-B5 and QJ71E71-B2 Ethernet interface modules (shown as "E71" in diagrams)</td></tr>
<tr><td>Q series C24/E71</td><td colspan="2">Generic term for Q series C24, Q series E71.</td></tr>
<tr><td rowspan="3">Serial communication module</td><td colspan="2">Generic term for the modules below.</td></tr>
<tr><td>QnA series</td><td>AJ71QC24, AJ71QC24-R2, AJ71QC24-R4, A1SJ71QC24, A1SJ71QC24-R2, AJ71QC24N, AJ71QC24N-R2, AJ71QC24N-R4, A1SJ71QC24N, A1SJ71QC24N-R2.</td></tr>
<tr><td>Q series</td><td>QJ71C24N, QJ71C24N-R2, QJ71C24N-R4, QJ71C24, QJ71C24-R2.</td></tr>
<tr><td>UC24<br>Computer link module</td><td colspan="2">Generic term for AJ71UC24, A1SJ71UC24-R2, A1SJ71UC24-R4, A1SJ71UC24-PRF, A1SJ71C24-R2, A1SJ71C24-R4, A1SJ71C24-PRF, A2CCPUC24, A2CCPUC24-PRF<br>＊ A series computer link module</td></tr>
</table>

(2) Other generic terms and abbreviations

In this manual, the following generic terms and abbreviations are used to explain the data communication devices and other devices of the serial communication module and Ethernet Interface module. The names and/or model names of the serial communication modules are used to identify the specific models.

| Generic term/abbreviation | Description of generic term/abbreviation |
|---|---|
| A compatible 1E/1C frame | A compatible 1E frame, A compatible 1C frame |
| Buffer memory | Generic term for the memory of the intelligent function module or special function module for storing the transmission/reception data when communicating with the PLC CPU (such as setting values and monitor values). |
| Computer | Generic term for a unit in the external device that can communicate data through the MC protocol or bidirectional protocol. |
| Data communication function | Generic term for MC protocol, non-procedure protocol, or bidirectional protocol |
| External device<br>Opposite device | Generic term for computer, indicator, measuring instrument, ID module, barcode reader, regulator, the serial communication modules and others, Ethernet Interface modules and others which are connected to this serial communication module to perform data communication. |
| GX Configurator-SC | Abbreviation for GX Configurator-SC (SW0D5C-QSCU-E or later).<br>• Initial settings for the module, monitoring and testing can be performed with a PLC program without having to consider the I/O signals or buffer memory. (Intelligent function utility)<br>• Converting PLC programs necessary for data communication processing into FB can shorten program production man-hours.<br>In addition, the monitoring and analysis of the transmitted/received data by the communication network can shorten the system start-up time. (Protocol FB support function) |
| GX Developer | Abbreviation for GX Developer (SWnD5C-GPPW-E). (n in the model should be 4 or greater) |
| I/F | Abbreviation for Interface |
| Intelligent function module | Generic term for a Q series PLC module that operates by the commands from the PLC CPU.<br>(Equivalent to a special function module of the A series PLC.)<br>[Examples]<br>• CC-Link Interface module<br>• A/D, D/A conversion module<br>• Ethernet Interface module<br>• Serial communication module |
| Intelligent function module device | Generic term for the buffer memory of the intelligent function module for storing the transmission/reception data when communicating with the PLC CPU (such as setting values and monitor values). |
| MELSECNET/10 | Abbreviation for MESECNET/10 Network System |
| MELSECNET/H | Abbreviation for MESECNET/H Network System |
| MX Component | Abbreviation for MX Component (SWnD5C-ACT-E or later) |
| QnA compatible 2C/3C/4C frame | QnA compatible 2C frame, QnA compatible 3C frame, QnA compatible 4C frame |
| QnA compatible 3C/4C frame | QnA compatible 3C frame, QnA compatible 4C frame |
| QnA compatible 3E/3C frame | QnA compatible 3E frame, QnA compatible 3C frame |
| QnA compatible 3E/3C/4C frame | QnA compatible 3E frame, QnA compatible 3C frame, QnA compatible 4C frame |
| Reference Manual | MELSEC Communication Protocol Reference Manual |
| RS-232 (interface) | Abbreviation for Interface that conforms to the RS-232 interface |
| RS-422/485 (interface) | Abbreviation for Interface that conforms to either the RS-422 or RS-485 interface |

| Generic term/abbreviation | Description of generic term/abbreviation |
|---|---|
| Special function module | Generic term for an A/QnA series PLC module that operates by the commands from the PLC CPU.<br>(Equivalent to an intelligent function module of the Q series PLC)<br>[Examples]<br>• CC-Link Interface module<br>• A/D, D/A conversion module<br>• High-speed counter module<br>• Ethernet Interface module<br>• Computer link module, serial communication module |
| Switch setting | Generic term for intelligent function module switch setting |
| User's Manual (Application) | • Serial communication module<br>Q Corresponding Serial Communication Module User's Manual (Application)<br>• Ethernet Interface module<br>Q Corresponding Ethernet Interface Module User's Manual (Application) |
| User's Manual (Basic) | • Serial communication module<br>Q Corresponding Serial Communication Module User's Manual (Basic)<br>• Ethernet Interface module<br>Q Corresponding Ethernet Interface Module User's Manual (Basic) |
| User's Manual (Web function) | • Ethernet interface module<br>Q Corresponding Ethernet Interface Module User's Manual (Web function) |


A - 20

Meanings and Descriptions of Terminology

The following table outlines the meanings and descriptions of the terms used in this and related manuals of the Q series serial communication module and Ethernet Interface module.

| Term | Description |
|---|---|
| A compatible 1C frame (formats 1 to 4) | One of the message formats for the serial communication module used to communicate ASCII code data through the MC protocol.<br>This is the same message format as when communicating through the dedicated protocol for A series computer link modules.<br>For a QCPU, reading from and writing to device memories are allowed within the same device range as for an AnACPU.<br>For more details, see Chapter 5. |
| A compatible 1E frame | One of the message formats for the Ethernet interface module used to communicate ASCII or binary code data through the MC protocol.<br>This is the same message format as when reading/writing data from/to the PLC CPU of an A series Ethernet interface module.<br>For a QCPU, reading from and writing to device memories are allowed within the same device range as for an AnACPU.<br>For more details, see Chapter 6. |
| Bidirectional protocol | A communication procedure of the serial communication module and one of the data communication functions of the serial communication module that allow communication of arbitrary data between the PLC CPU and external devices.<br>For more details, see Chapter 7. |
| Independent operation | Operation of two interfaces of the serial communication module in which each interface performs data communication with an external device independently of each other, using a function designated in each transmission protocol setting. |
| Linked operation | Operation of two interfaces of the serial communication module when they are linked to perform data communication with external devices that are connected to each of the two interfaces.<br>The two interfaces perform communication using the same data communication function (MC protocol (same format) or non-procedure protocol) or the same transmission specifications. (Linked operation cannot be performed using the bidirectional protocol.) |
| MELSEC communication protocol (MC protocol) | Name of a communication system for accessing the PLC CPU from an external device using the communication procedure for Q series serial communication modules or Ethernet interface modules. (Throughout this manual, this is referred to as the MC protocol.)<br>Two types of communication systems are available; one using ASCII code data and the other using binary code data. |
| Message transmission function (printer function) | A function that preregisters character data (messages) to be sent to an external device (mainly printers) in the serial communication module as a user frame, and sends registered data of multiple user frames using the non-procedure protocol (sent by instruction from the PLC CPU). |
| Multidrop connection | A mode of connection using the RS-422/485 interface of the serial communication module in which multiple external devices and other serial communication modules are connected in 1:n or m:n mode. |
| Non-procedure protocol | A user's communication procedure, and one of the data communication functions of the serial communication module for communicating arbitrary data between the PLC CPU and an external device.<br>For more details, see Chapter 6 of the User's Manual (Basic). |
| QnA compatible 2C frame (formats 1 to 4) | One of the message formats for the serial communication modules used to communicate ASCII code data through the MC protocol.<br>This is the same message format as the frames for communication through the dedicated protocol of QnA series serial communication modules.<br>• QnA compatible 2C frame (formats 1 to 4): QnA simplified frame (formats 1 to 4)<br>For more details, see Chapter 4. |

| Term | Description |
| --- | --- |
| QnA compatible 3C frame (formats 1 to 4)<br>QnA compatible 4C frame (formats 1 to 4) | One of the message formats for the serial communication modules used to communicate ASCII code data through the MC protocol.<br>This is the same message format as the frame for communication through the dedicated protocol of QnA series serial communication modules.<br>• QnA compatible 3C frame (formats 1 to 4): QnA frame (formats 1 to 4)<br>• QnA compatible 4C frame (formats 1 to 4): QnA extension frame (formats 1 to 4)<br>For more details, see Chapter 3. |
| QnA compatible 3E frame | One of the message formats for the Ethernet interface modules used to communicate ASCII or binary code data through the MC protocol.<br>This is the same message format as when reading/writing data from/to the PLC CPU of a QnA series Ethernet interface module.<br>For more details, see Chapter 3. |
| QnA compatible 4C frame (format 5) | One of the message formats for the serial communication modules used to communicate binary code data through the MC protocol.<br>This is the same message format as the frame for communication through the dedicated protocol of a QnA series serial communication module.<br>• QnA compatible 4C frame (format 5): QnA extension frame (format 5)<br>For more details, see Chapter 3. |
| User frame | Data name used when registering the fixed format section of a message to be communicated between the serial communication module and an external device using the functions listed below, in order to use it for data transmission and reception. (The contents of data in a user frame should be consistent with the specifications of the external device.)<br>Used for registering the sequence of data of each of the head and ending sections in a communication message (transmission control code, C24 station number, sum check, fixed data, etc.) to the serial communication module.<br>• The on-demand function of the MC protocol<br>• Data transmission and reception functions through the non-procedure protocol.<br>For more details, see Chapter 9 of the User's Manual (Application). |

# MEMO


A - 23

1

# 1 OVERVIEW

This manual provides information on the methods that are employed by external devices to read and write data from/to the PLC CPU using the data communication functions of the MELSEC communication protocol (referred to as the MC protocol in this manual) via the serial communication modules or Ethernet interface modules.
When applying the following program examples to the actual system, make sure to examine the applicability and confirm that it will not cause system control problems.

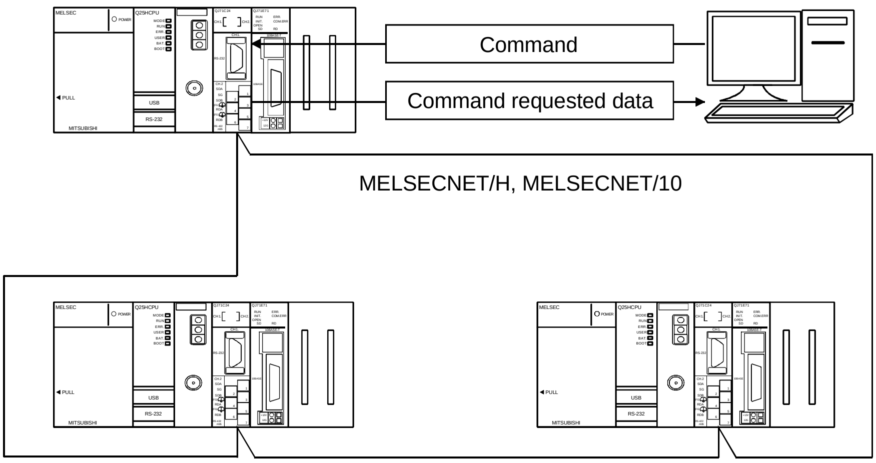
When performing data communication using the MC protocol, please read Chapter 2.

## 1.1 Overview of the MELSEC Communication Protocol

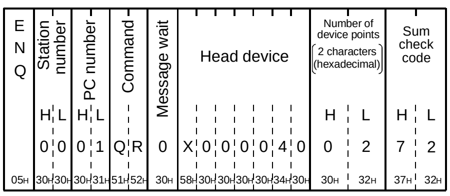
The following is an overview of the MC protocol:

(1) The MC protocol is the name of a communication method for Q series PLCs and is used by external devices to read and write device data and programs of the PLC CPU via the Q series C24 or Q series E71.
Communications through the MC protocol can be performed if a device can incorporate application programs and send/receive data using the control procedures of the MC protocol.

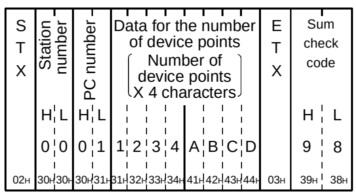
(2) The message formats and control procedures for an external device to access the PLC CPU are defined separately for the Q series C24 and Q series E71.




(3) The message formats and control procedures during data communication using the MC protocol are the same as when accessing the PLC CPU via conventional A/QnA series modules.




(Command message of the A compatible 1C frame)




(Response message of the A compatible 1C frame)

An external device can access a Q series PLC using a program with which the PLC is accessed via one of the following A/QnA series modules.

1) Accessing the Q series PLC via the Q series C24
   It is possible to access the PLC using a program on the external device via the following A/QnA series modules.
   - A series computer link module
   - QnA series serial communication module

2) Accessing the Q series PLC via the Q series E71
   It is possible to access the PLC using a program on the external device via the following A/QnA series modules.
   - A series Ethernet interface module
   - QnA series Ethernet interface module

∗ The data communication functions using the MC protocol correspond to the following data communication function of the conventional modules.
   - In case of the Q series C24
     Corresponds to the data communication functions using the dedicated protocol supported by A series computer link modules and QnA series serial communication modules.
   - In case of the Q series E71
     Corresponds to the functions for reading/writing data from/to the PLC CPU supported by A series/QnA series Ethernet interface modules.

(4) At the PLC CPU side, the Q series C24/Q series E71 sends/receives data according to the commands from an external device.
Thus, a sequence program for data communication is not required at the PLC CPU side.
∗ In case of the Q series C24, a sequence program for data communication is required when using the on-demand function for sending data from the PLC CPU.

(5) If the external device is a PC running one of the basic operation systems below, it is possible to create a communication program for the external device without considering the detailed MC protocol (transmission/reception procedures) using one of the following separately sold communication support tools.
(Supported basic operation systems)
- Microsoft® Windows® 95 Operating System
- Microsoft® Windows® 98 Operating System
- Microsoft® WindowsNT® Workstation Operating System Version 4.0
- Microsoft® Windows® Millennium Edition Operating System
- Microsoft® Windows® 2000 Professional Operating System
- Microsoft® Windows® XP Professional Operating System
- Microsoft® Windows® XP Home Edition Operating System

∗ Depending on the version of MX Component used, different operating systems are supported.
For details, see the MX Component manual.

(Separately sold communication support tools)
- MX Component (SW0D5C-ACT-E or later, hereinafter abbreviated as MX Component.)

## 1.2 Features of the MELSEC Communication Protocol

This section explains the features of the MC protocol.

(1) The data communication using the MC protocol is a function that enables the reading/writing of data from/to the PLC CPU in order to control and monitor the PLC equipment from the external device side.
Reading/writing device data and program files for the PLC CPU and controlling the PLC CPU status (remote RUN/STOP) can be performed from the external devices (personal computer, indicator, etc.)

(a) Reading and writing data
By reading/writing data to/from the device memory of the PLC CPU and the buffer memory of the intelligent function modules, the following control operations can be performed:
Data read and write operations can also be performed for A/QnA series PLC CPUs (other stations) and intelligent function modules.

1) Reading data
Operation monitoring, data analysis, production control, etc. can be performed from the external device side.

2) Writing data
Production instructions, etc. can be issued from the external device side.

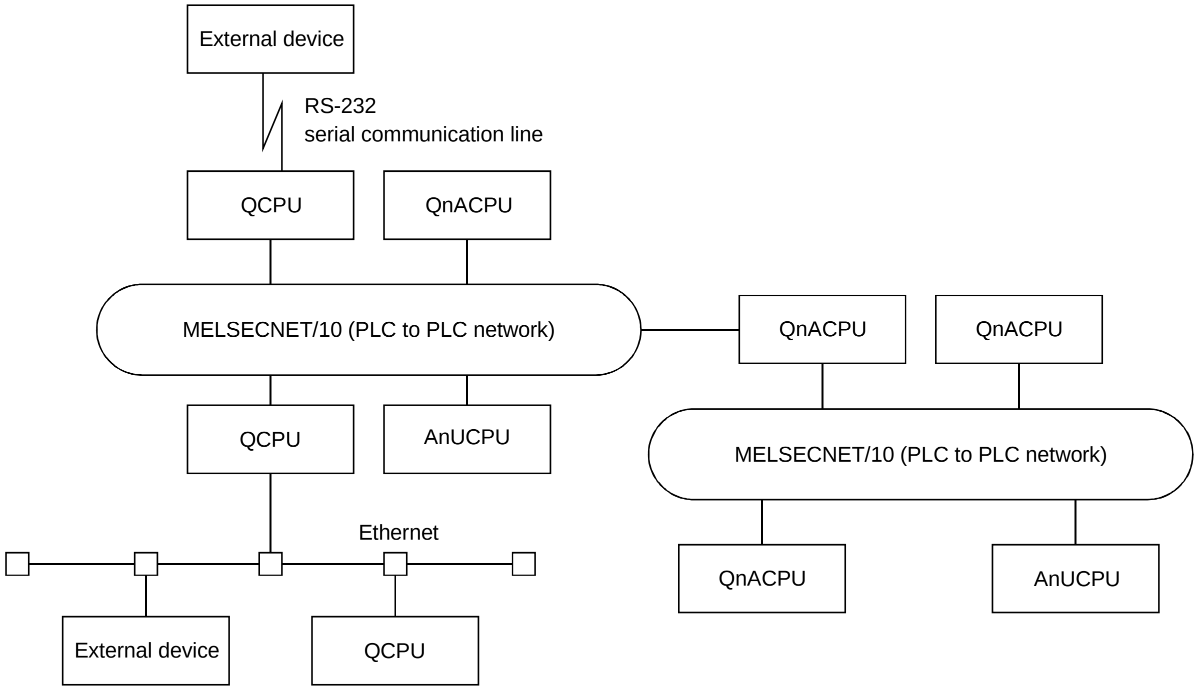
(b) Reading and writing files
By reading and writing files for sequence programs and parameters stored in the PLC CPU, the following control operations can be performed:

1) Reading files
File management for the local station’s QCPU and other station' QCPU/QnACPU can be performed from the external device side.

2) Writing files
Programs such as execution programs can be modified (replaced) by writing file data stored in the external device to the PLC CPU as needed.

(c) Remote control of the PLC CPU
The PLC CPU can be remotely controlled from the external device by performing remote RUN/STOP/PAUSE/LATCH CLEAR/RESET operations.

(2) When performing data communication using the MC protocol, the external devices can access PLCs (including A/QnA series PLCs) of other stations via a network system (MELSECNET/H, MELSECNET/10, Ethernet).




(3) The Q series C24 is equipped with the PLC CPU monitoring function.
By using this function, the PLC CPU status 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.
This helps reduce the number of data read processes performed by the external device.
✻ The Q series E71 also has the PLC CPU monitoring function that uses e-mail transmission.
(See the User's Manual (Application).)

**POINT**

To check what functions can be performed by external devices when communicating data through the MC protocol, see the "Commands and function list" reference sections in Chapters 3 to 6.

# 2 DATA COMMUNICATION USING THE MELSEC COMMUNICATION PROTOCOL

This chapter explains the data communication using the MC protocol when an external device reads data from and writes data to a PLC CPU using the Q series C24/E71.



## 2.1 Types and Applications of Data Communication Frames

This section provides information about the types and applications of frames (data communication messages) used by the external device to access the PLC CPU using the MC protocol.
When the external device accesses the PLC via the Q series C24/E71, the data communication is performed by transmission/reception of command messages (access requests) and response messages (responses) of one of the following frames listed below.
Choose the frame to be used by considering the accessible range of each frame shown in Section 2.2.

| Target module | Available communication frame | | Code of communication data | Control procedure reference section |
|---|---|---|---|---|
| Q series C24 | QnA compatible 3C frame | Formats 1 to 4 | ASCII code | Chapter 3 |
| | QnA compatible 4C frame | Format 5 | Binary code | |
| | QnA compatible 2C frame | Formats 1 to 4 | ASCII code | Chapter 4 |
| | A compatible 1C frame | | | Chapter 5 |
| Q series E71 | QnA compatible 3E frame | | ASCII code or binary code | Chapter 3 |
| | A compatible 1E frame | | | Chapter 6 |

### (1) QnA compatible 3C, QnA compatible 4C and QnA compatible 3E frames

(a) These frames are mainly used to access all devices and files of the Q/QnACPU from the external device.
(b) It is also possible to access A series PLC CPU devices.

### (2) QnA compatible 2C frame

(a) This frame can access the device memory of QCPU stations on which the Q series C24 is loaded and Q/QnACPU stations linked by multidrop connection.
(b) The message format is simplified compared to the QnA compatible 3E/3C/4C frames.
(c) Since smaller amounts of transmission data are used, the messages are easier to be processed by the external device and the transmission times of the messages are shortened.

### (3) A compatible 1C and A compatible 1E frames

(a) These frames have the same message structure as when accessing the PLC CPU using an A series computer link module or Ethernet interface module.
(b) By utilizing the software for data communication on the external device that has been created for the A series PLCs, Q/QnACPUs linked by multidrop connection or network connection, and PLC CPUs other than Q/QnACPU can be accessed using the same frame.
With respect to the Q/QnACPUs, only those devices with the same names as those existing in the AnCPUs, AnNCPUs, AnACPUs and AnUCPUs can be accessed within the AnACPU device range.
- When using Q series C24 : See Section 5.2.1(2)
- When using Q series E71 : See Section 6.3.1(2)

Devices that have been newly added to Q/QnACPUs cannot be accessed.

REMARK

(1) The following explains how to read the data communication frame names when performing data communication using the MC protocol.
A data communication frame name indicates a relevant PLC CPU series to show command compatibility with conventional modules, a frame designation of the corresponding conventional module, and the target module.

xxx compatible n m frame
a) b) c)
(Examples: QnA compatible 3C frame, QnA compatible 3E frame)

(a) Relevant PLC CPU series, showing command compatibility with conventional modules
A : A series PLC CPU
QnA : QnA series PLC CPU

(b) Frames of the corresponding conventional modules
1 : Corresponds to the communication frames of commands supported by A series computer link modules and Ethernet interface modules.
2 : Corresponds to the QnA simplified frames supported by QnA series serial communication modules.
3 : Corresponds to the QnA frames supported by QnA series serial communication modules and the communication frames supported by QnA series Ethernet interface modules.
4 : Corresponds to the QnA extension frames supported by QnA series serial communication modules.

(c) Modules described in this manual with which data communication can be performed using the applicable frames.
C : Q series C24
E : Q series E71

(2) When accessing via the Q series C24
The external device accesses the PLC CPU using frames with a format number selected in the "Communication protocol setting" in the switch settings using GX Developer.
✻ When any of Formats 1 to 4 is selected, access is enabled using each of the four types of frames shown above and data communication is performed by the transmission/reception of command messages and response messages in the selected format.

(3) When accessing via the Q series E71
The external device can access using any of the two types of frames shown above; it accesses the PLC CPU by sending/receiving command messages and response messages using frames corresponding to the setting selected in the operation settings using GX Developer.

(4) Time required for communication in binary code is shortened because the amount of the communication data is approximately a half of what is required for communication in ASCII code data.

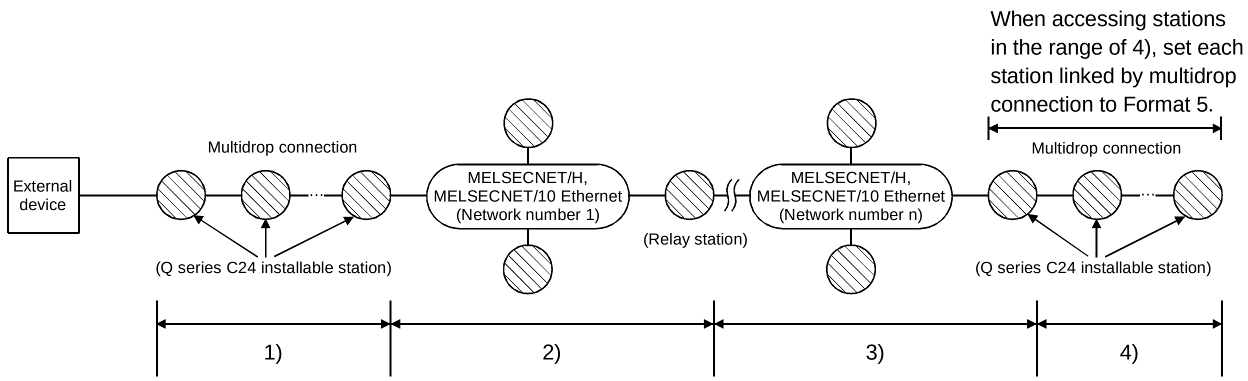
## 2.2 Accessible Range of Each Data Communication Frames

In a communication using the MC protocol, access to PLC stations on which the Q series C24/E71 is not loaded is possible via MELSECNET/H, MELSECNET/10 and Ethernet.
This section explains the range of PLCs of other stations that can be accessed from an external communication device when the access destination PLC is not directly connected by a serial communication line or Ethernet.

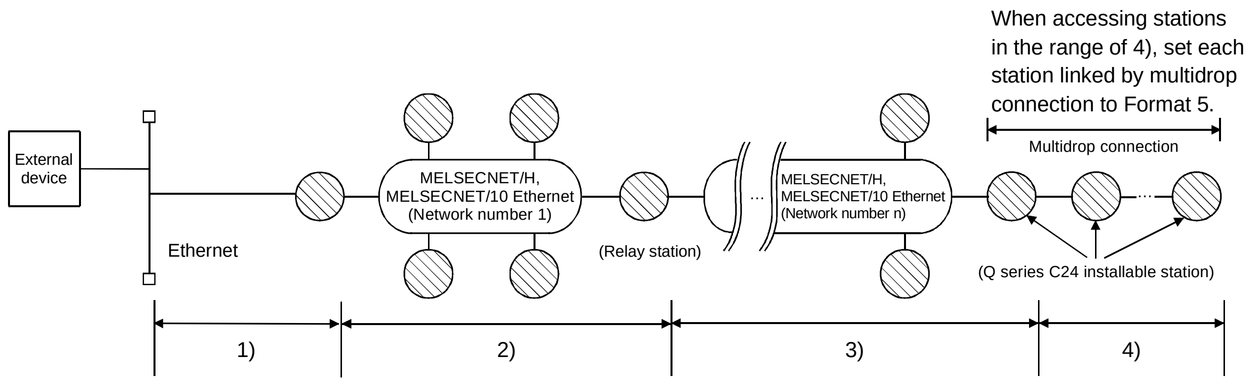
○: Accessible, ×: Not accessible

| Target module | Frame used | | Range of accessible stations 1) | 2) | 3) | 4) |
|---|---|---|---|---|---|---|
| Q series C24 | QnA compatible 3C frame | Formats 1 to 4 | ○ | ○ | ○ | × |
| | QnA compatible 4C frame | Formats 1 to 4 | ○ | ○ | ○ | ○ |
| | | Format 5 | ○ | ○ | ○ | ○ |
| | QnA compatible 2C frame | Formats 1 to 4 | ○ | × | × | × |
| | A compatible 1C frame | | ○ | ○ | × | × |
| Q series E71 | QnA compatible 3E frame | | ○ | ○ | ○ | ○ |
| | A compatible 1E frame | | ○ | ○ | × | × |

### (1) When using the Q series C24




### (2) When using the Q series E71




**POINT**

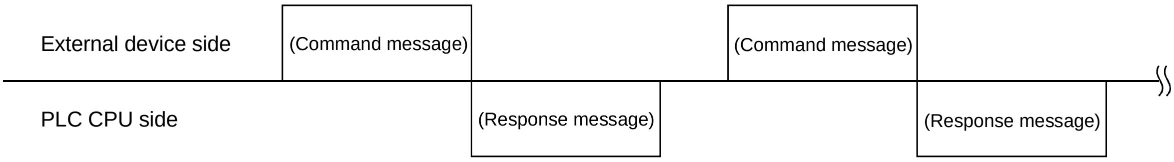
For other MELSECENT/H, MELSECNET/10 or Ethernet, it is possible to access each PLC on other MELSECNET/H, MELSECNET/10 or Ethernet designated by a user with the routing parameter setting to a relay station within the range of network system specifications.

## 2.3 How to Read the Control Procedures of the MC Protocol

This section explains the control procedures when an external device accesses the PLC CPU using the MC protocol.

### (1) Transmission of command messages

Data communication through the MC protocol is performed using half-duplex communication. (∗1)
When accessing the PLC CPU, send the next command message after receiving a response message from the PLC CPU side for the previous command message transmission.




∗1 When accessing via the Q series C24, full-duplex communication is performed by user setting when the on-demand function is being used.
When the system between external devices and PLC CPUs is configured with a m:n connection, the next command message transmission cannot be performed until data communication between either of the external devices and PLC CPUs is completed.

### (2) When a normal completion response message to a command message cannot be received

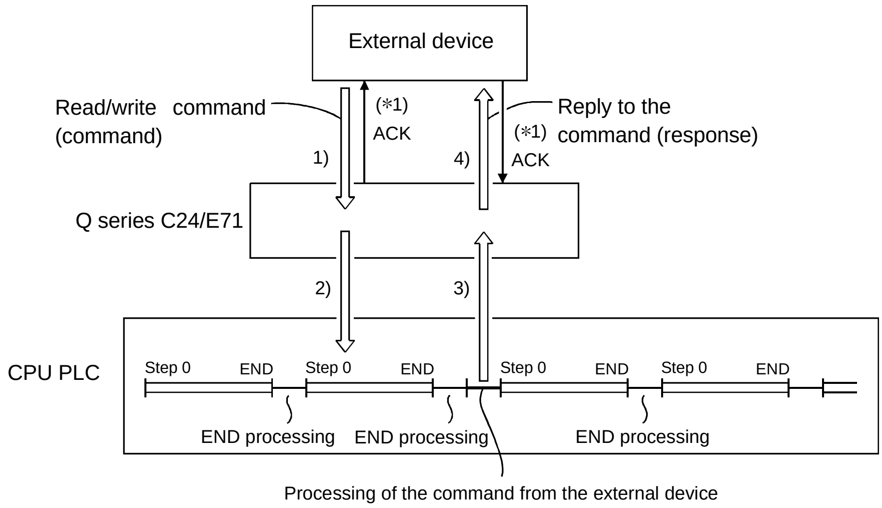
(a) If an abnormal completion response message is received
Handle the case according to the error code in the response message.

(b) If the response message cannot be received or nothing can be received
- Via the Q series E71
  Send the command message again after the monitoring time of the response monitor timer value has elapsed.
- Via the Q series C24
  Send the command message again after the monitoring time of the response monitor timer (timer 1) has elapsed.

∗ Modify the setting value of the monitoring time as needed.
In case of via the Q series C24, try to modify the message wait time also.

## 2.4 Access Timing of the PLC CPU Side

The following diagram illustrates the access timing of the PLC CPU side when an external device accesses the PLC CPU via the Q series C24/E71.




1) A command message is transmitted from the external device to the PLC CPU side.
2) Upon receiving a command message from the external device, the Q series C24/E71 of the PLC CPU side sends a data read/write request to the PLC CPU according to the content of the command.
3) The PLC CPU reads/writes data according to the request from the external device at the time of executing the sequence program's END instruction, and passes the processing result to the Q series C24/E71.
4) Upon receiving the processing result from the PLC CPU, the Q series C24/E71 sends a response message including the processing result to the external device that originated the request.

*1 The ACK response shown in the diagram is sent/received between the Q series E71 and the external device when the access via the Q series E71 is performed by TCP/IP communication.
It is different from the response to the processing requested by the external device in the command message (the processing result).
When access via the Q series E71 is performed by UDP/IP or when access is performed via the Q series C24, the ACK response indicated by *1 is not sent.

| POINT |
| --- |
| (1) Note that when data is read/written from/to the external device while the PLC CPU is in operation, the scan time of the PLC CPU becomes longer, according to the amount of time it takes to process the command from the external device.<br>If this has an influence on the extension of scan time, access the PLC CPU a small amount at a time, breaking it up over many accesses.<br><br>(2) An access request from the external device to the buffer memory of the Q series C24/E71 is immediately acknowledged. Also, it does not affect the scan time of the PLC CPU. |

REMARK

Regarding the scan time of the PLC CPU

1) While the PLC CPU is performed by required command, access command to the Q series C24/E71 and the PLC CPU is processed during every END processing.
   (The scan time becomes longer according to the amount of time it takes to process the command.)
2) When multiple external devices issue access requests to a single station at the same time, the processing requested from the external devices may have to wait until END processing has been executed multiple times depending on the timing of the requests.
   * By specifying the amount of time it takes to process the communication (1 to 100 ms) in the special register (SD315), multiple command requests can be processed within one scan. (However, the scan time becomes longer according to the amount of time it takes to process communication).
     Also the group of multiple command requests can be processed within one scan by executing a COM instruction in a sequence program. (However, the scan time becomes longer according to the amount of time it takes to process the COM instruction.)

## 2.5 Setting Method for Writing to the PLC CPU during RUN

This section explains the setting for writing data from an external device to the PLC CPU while in the RUN status via the Q series C24/E71 (write during RUN).

### (1) Setting of the Q series C24

Set the "Write during RUN" switch of the transmission settings within the intelligent function module switch settings by GX Developer to "Allow."

＊ The GX Developer switch setting is explained in Section 4.5.2 of the Q series C24 User’s Manual (Basic).

### (2) Setting of the Q series E71

Place a check mark for the "Enable write at RUN time" in the Ethernet operations setting with GX Developer.

＊ The GX Developer Ethernet operations setting is explained in Section 4.7 of the Q series E71 User’s Manual (Basic).

### (3) Setting on the PLC CPU side

Set the system protect switch to the "Enable write at PLC CPU RUN time. "
(In case of a QCPU, set the system protection switch, SW1, to "OFF.")

**POINT**

When controlling the PLC (modifying data) while it is in operation by connecting computer peripheral devices to the CPU module or connecting personal computers to the intelligent function modules, configure an interlocking circuit in a sequence program so that the safety of the overall system is always maintained.
Also, perform other control operations (program and operation status modifications (status control)) on the PLC in operation only after reading the manual carefully and thoroughly confirming the safety.
Especially in the above mentioned control operations, which are performed from an external device to a remote PLC, any problems on the PLC side may not be dealt with promptly due to abnormal data communication. In addition to configuring an interlocking circuit in a sequence program, determine how the system handles data communication problem between the external devices and the PLC CPU.

## 2.6 Accessing Other Stations

This section provides an overview on how to access PLCs of other stations that are connected via MELSECNET/H, MELSECNET/10 and Ethernet.
For more detailed information on MELSECNET/H, refer to the Q corresponding MELSECNET/H network system reference manual.

### 2.6.1 Accessible PLCs of other stations

The following tables summarize the accessible PLC modules and modules that can act as message relays in a network when accessing PLCs of other stations (applicable stations whose designated PC number in the data communication frames is not FFH.)

#### (1) Accessible PLC modules

(a) CPU PLC modules

| | | | Model name |
|---|---|---|---|
| QCPU (Q mode) | PLC CPU | | - |
| | | Basic model | Q00JCPU, Q00CPU, Q01CPU |
| | | High performance model | Q02CPU, Q02HCPU, Q06HCPU, Q12HCPU, Q25HCPU |
| | Process CPU | | Q12PHCPU, Q25PHCPU |
| | Redundant CPU | | Q12PRHCPU, Q25PRHCPU |
| QnACPU | PLC CPU | | Q2ACPU, Q2ACPU-S1, Q2ASCPU, Q2ASCPU-S1, Q2ASHCPU, Q2ASHCPU-S1, Q3ACPU, Q4ACPU |
| | Redundant CPU | | Q4ARCPU |
| ACPU<br>QCPU (A mode) (＊1) | | | A1NCPU, A2NCPU, A2NCPU-S1, A3NCPU, A2ACPU, A2ACPU-S1, A3ACPU, A2UCPU, A2UCPU-S1, A3UCPU, A4UCPU, A1SCPU, A1SJCPU(-S3), A1SHCPU, A1SJHCPU, A2SCPU, A2SHCPU, A2ASCPU, A2ASCPU-S1, A0J2HCPU, Q02CPU-A, Q02HCPU-A, Q06HCPU-A |

＊1 Other than the PLC CPUs above, it is possible to access A2CCPUC24 and A2CCPUC24-PRF modules that are linked to an external device by multidrop connection.

(b) Remote I/O station module

| | Model name |
|---|---|
| MELSECNET/H remote I/O station | QJ72LP25-25, QJ72LP25GE, QJ72BR15 |
| MELSECNET/10 remote I/O station (＊2) | AJ72QLP25, AJ72QBR15, A1SJ72QLP25, A1SJ72QBR15, AJ72LP25 (G), AJ72BR15 |

＊2 Only reading/writing of the intelligent function module's buffer memory is available for the MELSECNET/10 remote I/O stations.

**POINT**

When writing data to the device in the remote I/O station or the intelligent function module (special function module) loaded on the remote I/O station module inside the network system, specify the instruction "Write allow during RUN" with the GX Developer setting indicated in Section 2.5 (1) (2).

(2) Modules that can act as relays between the networks

| | Model name |
|---|---|
| MELSECNET/H | QJ71LP21, QJ71LP21-25, QJ71LP21S-25, QJ71LP21GE, QJ71BR11 (MELSECNET/H mode) |
| MELSECNET/10 | QJ71LP21, QJ71LP21-25, QJ71LP21S-25, QJ71LP21GE, QJ71BR11 (MELSECNET/10 mode) |
| | AJ71QLP21 (S/G), AJ71QBR11, A1SJ71QLP21, A1SJ71QBR11 |
| | AJ71LP21 (G), AJ71LR21, AJ71BR11, A1SJ71LP21, A1SJ71BR11 |
| Ethernet (＊1) | QJ71E71-100, QJ71E71-B5, QJ71E71-B2, QJ71E71 |
| | AJ71QE71N-T, AJ71QE71N-B5, AJ71QE71N-B2, AJ71QE71N-B5T, A1SJ71QE71N-T, A1SJ71QE71N-B5, A1SJ71QE71N-B2, A1SJ71QE71N-B5T, AJ71QE71, AJ71QE71-B5, A1SJ71QE71-B2, A1SJ71QE71-B5 |

＊1 QnA series Ethernet interface modules can act as relays between the networks if the function version is B or later.
Verify the function version by the production date shown in the "Date column of the rated plate," which is attached to the side of the module.
(Manufactured date: Year (last two digits), Month (two digits), function version (one digit))

POINT

(1) When accessing other stations using A compatible 1E/1C frames, the following parameter should be set with GX Developer in the PLC CPU of the Q series C24/E71 installable station.
- The "Valid module during other station access" setting: Sets the routing module when accessing other stations.

(2) When multiple network modules with the same network number are loaded on the Q series C24/E71 installable station, access to other stations is performed via the network module installed on the slot that has the smallest base unit number.

(3) For the required number of scans when multiple external devices and GX Developer request accesses to the same station simultaneously, see Section 2.4.

(4) For the details of the accessible range of PLCs of other stations on a network system, refer to the reference manual for the network system being used.

(5) If a computer link module is included in a multidrop connection when connecting to the Q series C24, it should be accessed using ASCII code communication frames.
The QnA compatible 4C frame (format 5) for binary code communication cannot be used when accessing stations linked via multidrop connection, including when accessing stations connected to an external device.

(6) When connecting to the Q series E71, it is possible to communicate with the PLC CPU on other Ethernet via MELSECNET/H, MELSECNET/10 by using the MELSECNET/H, MELSECNET/10 relay communication function.
For more details, see Chapter 3 of the User's Manual (Application).

## 2.6.2 Example of accessible station when each frame is used

The following examples show some other stations whose PLCs are accessible.
In the diagrams, the following symbols are used for each station.

MELSECNET/H, MELSECNET/10

| 1 | Mp | 1 | (Network number 1, control station, station number 1)

Station number (the "00" of a master station of a remote I/O network is omitted.)

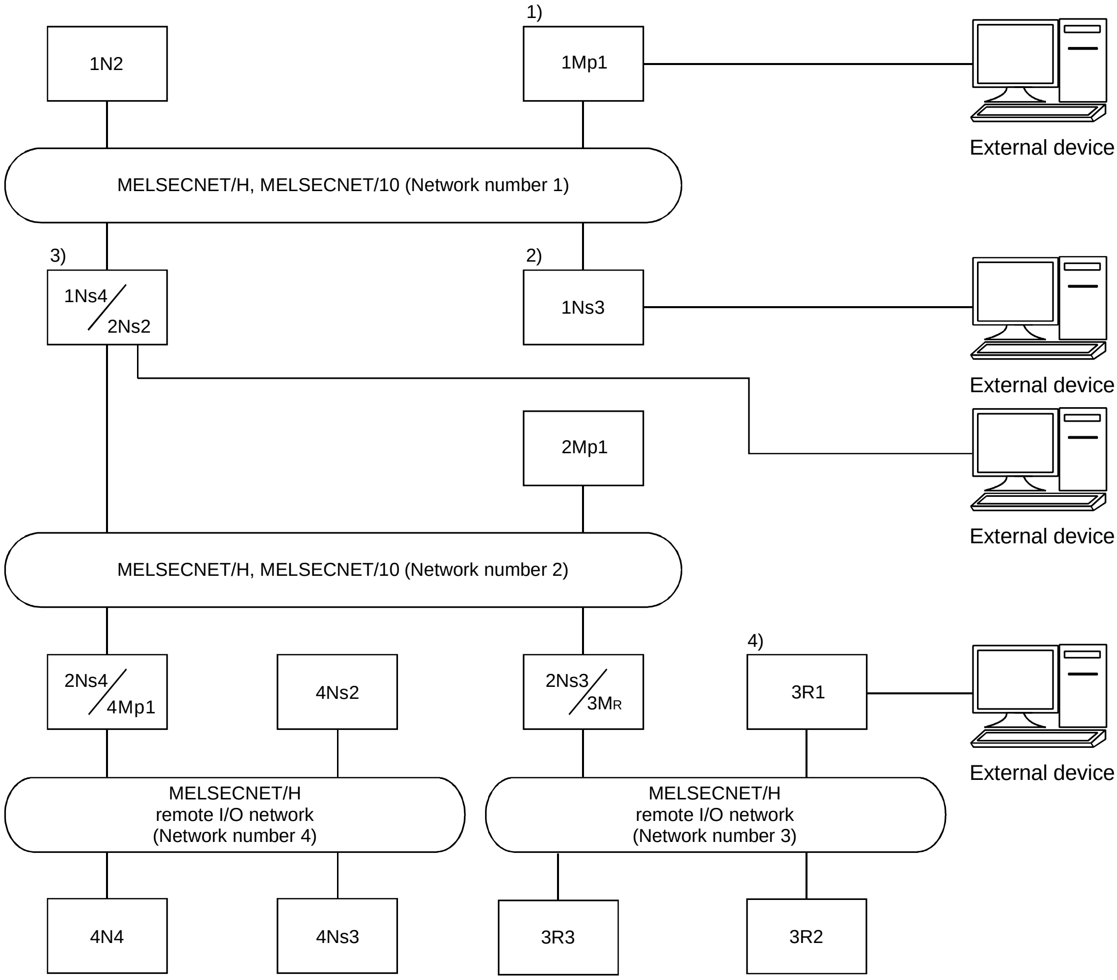
MELSECNET/H, MELSECNET/10
Control station/normal station
Mp…………..Control station
Ns…………..Normal station (AnUCPU, QnACPU or QCPU)
N……………..Normal station (other than AnUCPU, QnACPU or QCPU)

Remote I/O network
Master station/remote station
MR…………..Master station
R……………..Remote station

Network number

∗ When the Q series E71 is included in a network system, it is configured as a normal station (the "N" symbol above).

(Control stations and normal stations connected to external devices are all Q/QnACPUs)



<table>
<tr><th colspan="2" rowspan="2">External device connection station</th><th colspan="13">Station accessible from an external device</th></tr>
<tr><th>1Mp1</th><th>1N2</th><th>1Ns3</th><th>1Ns4<br>2Ns2</th><th>2Mp1</th><th>2Ns3<br>3MR</th><th>2Ns4<br>4Mp1</th><th>3R1</th><th>3R2</th><th>3R3</th><th>4Ns2</th><th>4Ns3</th><th>4N4</th></tr>
<tr><td rowspan="7">1)</td><td>A compatible 1E frame</td><td colspan="4">△</td><td colspan="9">×</td></tr>
<tr><td>A compatible 1C frame</td><td colspan="4">△</td><td colspan="9">×</td></tr>
<tr><td>QnA compatible 2C frame</td><td>○</td><td colspan="12">×</td></tr>
<tr><td>QnA compatible 3E frame</td><td colspan="13">○</td></tr>
<tr><td>QnA compatible 3C frame</td><td colspan="13">○</td></tr>
<tr><td>QnA compatible 4C frame<br>(formats 1 to 4)</td><td colspan="13">○</td></tr>
<tr><td>QnA compatible 4C frame<br>(format 5)</td><td colspan="13">○</td></tr>
<tr><td rowspan="7">2)</td><td>A compatible 1E frame</td><td colspan="4">△</td><td colspan="9">×</td></tr>
<tr><td>A compatible 1C frame</td><td colspan="4">△</td><td colspan="9">×</td></tr>
<tr><td>QnA compatible 2C frame</td><td colspan="2">×</td><td>○</td><td colspan="10">×</td></tr>
<tr><td>QnA compatible 3E frame</td><td colspan="13">○</td></tr>
<tr><td>QnA compatible 3C frame</td><td colspan="13">○</td></tr>
<tr><td>QnA compatible 4C frame<br>(formats 1 to 4)</td><td colspan="13">○</td></tr>
<tr><td>QnA compatible 4C frame<br>(format 5)</td><td colspan="13">○</td></tr>
<tr><td rowspan="7">3)</td><td>A compatible 1E frame</td><td colspan="4">△</td><td colspan="9">×</td></tr>
<tr><td>A compatible 1C frame</td><td colspan="4">△</td><td colspan="9">×</td></tr>
<tr><td>QnA compatible 2C frame</td><td colspan="3">×</td><td>○</td><td colspan="9">×</td></tr>
<tr><td>QnA compatible 3E frame</td><td colspan="13">○</td></tr>
<tr><td>QnA compatible 3C frame</td><td colspan="13">○</td></tr>
<tr><td>QnA compatible 4C frame<br>(formats 1 to 4)</td><td colspan="13">○</td></tr>
<tr><td>QnA compatible 4C frame<br>(format 5)</td><td colspan="13">○</td></tr>
<tr><td rowspan="7">4)</td><td>A compatible 1E frame</td><td colspan="13">×</td></tr>
<tr><td>A compatible 1C frame</td><td colspan="4">×</td><td colspan="2">△</td><td>×</td><td colspan="3">△</td><td colspan="3">×</td></tr>
<tr><td>QnA compatible 2C frame</td><td colspan="7">×</td><td>○</td><td colspan="5">×</td></tr>
<tr><td>QnA compatible 3E frame</td><td colspan="13">○</td></tr>
<tr><td>QnA compatible 3C frame</td><td colspan="13">○</td></tr>
<tr><td>QnA compatible 4C frame<br>(formats 1 to 4)</td><td colspan="13">○</td></tr>
<tr><td>QnA compatible 4C frame<br>(format 5)</td><td colspan="13">○</td></tr>
</table>

○: Accessible, △: (＊1), ×: Not accessible

＊1 There are limits to the devices and their ranges that can be accessed to the Q/QnACPU using the A compatible 1E frame or A compatible 1C frame.
With respect to the Q/AnACPU, only those devices with the same name as those existing in the AnCPU, AnNCPU, AnACPU and AnUCPU can be accessed within the AnACPU device range.
- When using the Q series C24: See section 5.2.1 (2)
- When using the Q series E71: See section 6.3.1 (2)

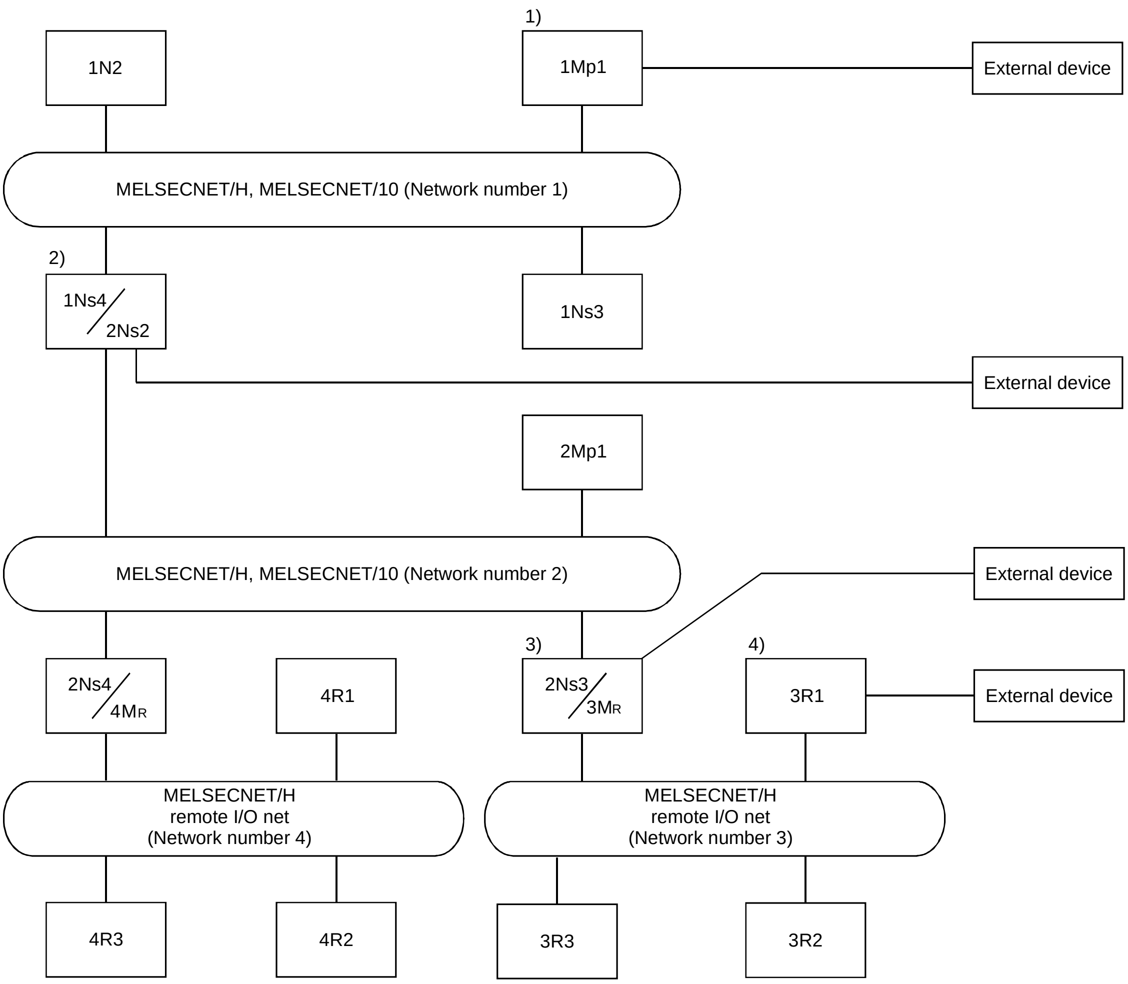
Devices that have been newly added with the Q/QnACPU cannot be accessed.

### 2.6.3 Example of designating data items for accessing other station designated within each data communication frame

This section shows examples of designating the "Network number", "PLC number", "Request destination module I/O number" and "Request destination module station number" that are designated in each frame when accessing to the PLC station on the MELSECNET/H, MELSECNET/10 using the MC protocol.
For items to be specified by data, refer to Section 3.1.
(How to understand the diagram and tables)

- See Section 2.6.2 for the meaning of symbols used in explanations.
- The numbers in the "Designated value when accessing from an external device" columns indicate the values designated for the data items shown in the left side of the table when accessing the relevant PLC station on the network system shown in the upper row of the table.

(Example of designation)
Control stations and normal stations connected to external devices are all Q/QnACPUs



When the external device connection station is 1)

| Data name | Designated value when accessing from an external device | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 1Mp1 | 1N2 | 1Ns3 | 1Ns4/ 2Ns2 | 2Mp1 | 2Ns3/ $3M_R$ | 2Ns4/ $4M_R$ | 3R1 | 3R2 | 3R3 | 4R1 | 4R2 | 4R3 |
| Network number Mp | $00_H$ | $01_H$ | | | $02_H$ | | | $03_H$ | | | $04_H$ | | |
| PC number Mp | $FF_H$ | $02_H$ | $03_H$ | $04_H$ | $01_H$ | $03_H$ | $04_H$ | $01_H$ | $02_H$ | $03_H$ | $01_H$ | $02_H$ | $03_H$ |
| Request destination module I/O No. Spm | $03FF_H$ | | | | | | | | | | | | |
| Request destination module station No. Spm | $00_H$ | | | | | | | | | | | | |

When the external device connection station is 2)

| Data name | Designated value when accessing from an external device | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 1Mp1 | 1N2 | 1Ns3 | 1Ns4/ 2Ns2 | 2Mp1 | 2Ns3/ $3M_R$ | 2Ns4/ $4M_R$ | 3R1 | 3R2 | 3R3 | 4R1 | 4R2 | 4R3 |
| Network number Mp | $01_H$ | | | $00_H$ | $02_H$ | | | $03_H$ | | | $04_H$ | | |
| PC number Mp | $01_H$ | $02_H$ | $03_H$ | $FF_H$ | $01_H$ | $03_H$ | $04_H$ | $01_H$ | $02_H$ | $03_H$ | $01_H$ | $02_H$ | $03_H$ |
| Request destination module I/O No. Spm | $03FF_H$ | | | | | | | | | | | | |
| Request destination module station No. Spm | $00_H$ | | | | | | | | | | | | |

When the external device connection station is 3)

| Data name | Designated value when accessing from an external device | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 1Mp1 | 1N2 | 1Ns3 | 1Ns4/ 2Ns2 | 2Mp1 | 2Ns3/ $3M_R$ | 2Ns4/ $4M_R$ | 3R1 | 3R2 | 3R3 | 4R1 | 4R2 | 4R3 |
| Network number Mp | $01_H$ | | | $02_H$ | | $00_H$ | $02_H$ | $03_H$ | | | $04_H$ | | |
| PC number Mp | $01_H$ | $02_H$ | $03_H$ | $02_H$ | $01_H$ | $FF_H$ | $04_H$ | $01_H$ | $02_H$ | $03_H$ | $01_H$ | $02_H$ | $03_H$ |
| Request destination module I/O No. Spm | $03FF_H$ | | | | | | | | | | | | |
| Request destination module station No. Spm | $00_H$ | | | | | | | | | | | | |

When the external device connection station is 4)

| Data name | Designated value when accessing from an external device | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 1Mp1 | 1N2 | 1Ns3 | 1Ns4/ 2Ns2 | 2Mp1 | 2Ns3/ $3M_R$ | 2Ns4/ $4M_R$ | 3R1 | 3R2 | 3R3 | 4R1 | 4R2 | 4R3 |
| Network number Mp | $01_H$ | | | $02_H$ | | $03_H$ | $02_H$ | $00_H$ | $03_H$ | | $04_H$ | | |
| PC number Mp | $01_H$ | $02_H$ | $03_H$ | $02_H$ | $01_H$ | $7D_H$ | $04_H$ | $FF_H$ | $02_H$ | $03_H$ | $01_H$ | $02_H$ | $03_H$ |
| Request destination module I/O No. Spm | $03FF_H$ | | | | | | | | | | | | |
| Request destination module station No. Spm | $00_H$ | | | | | | | | | | | | |

## 2.7 Precautions on Data Communication

This section explains some precautions that should be observed when performing data communication between an external device and the Q series C24/E71.

### (1) Precautions that should be observed when performing on data communication with the Q series E71 is used

(a) Perform read/write only when the following signals on the Q series E71 are on.
1) When using the automatic open UDP port
The initial normal completion signal (X19)
2) When using a port opened by a user
The initial normal completion signal (X19) and the open completion signal of the connection used (X10 to X17, buffer memory address $5000_H$)

If these signals are on, it is possible to communicate from an external device using the MC protocol regardless of whether or not a sequence program is used.

(b) When writing data while the PLC CPU is in the RUN status, perform the setting procedure according to the instructions in Section 2.5.

(c) When issuing a remote STOP instruction to the PLC CPU, use the automatic open UDP port. Alternatively, use the passive open connection set to "Always wait for OPEN" in the initial timing setting in the "Ethernet operation settings" with GX Developer.

(d) When the application setting of the open connection is "non-procedure," communication using the MC protocol cannot be performed.

(e) When the automatic open UDP port is used, communication in ASCII code data cannot be performed.

(f) Replacing the PLC CPU of other station with which data is communicated
The Q series E71 retrieves and keeps information of the PLC CPUs of other stations after being started up.
When replacing the PLC CPU of other station with which data is communicated after starting up the Q series E71, reboost the Q series E71 if the model name of the PLC CPU is changed (power supply of PLC CPU of local station reset/CPU reset).

(g) Replacing the Q series E71
The Ethernet address (MAC address) differs depending on the device.
When the Q series E71 is replaced due to breakdown, reboost any external devices also.
Similarly, reboot the Ethernet module even when an external device is replaced (such as a personal computer).

(h) When controlling the PLC in operation (especially modifications of data, programs and operation status (status control)) by connecting a personal computer or other device to the intelligent function module, read the manual thoroughly and confirm the safety before proceeding.
Incorrect data modification, program modification, and status control may lead to system malfunctions, damage to the machines, or fatal errors.

(i) When the Q series E71 is loaded on the MELSECNET/H remote I/O station
1) Communicate using the QnA compatible 3E frame. Communication cannot be performed using the A compatible 1E frame.
2) Accessing other stations relaying through the Q series E71 loaded on the remote I/O station, or accessing other stations relaying between Q series E71 cannot be performed from the MELSECNET/H remote master station.

(2) Precautions that should be observed when performing on data communication with the Q series C24

(a) Conditions under which the transmission sequence of the Q series C24 goes into the initial status

The transmission sequence of the Q series C24 returns to the initial status under the following conditions.

- When turning the power on, pressing the reset switch on the front of the CPU, and switching modes.
- When the transmission of a response message to 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" to Yes in the full-duplex communication of the RS-232 side.

(b) NAK responses from the Q series C24

A NAK response using the MC protocol is sent to an external device whenever an error is detected in the request sent to the local station.
Therefore, a NAK response may be sent even when the external device is sending a request in the full-duplex communication.

(c) Replacing PLC CPUs of other stations with which data is communicated

The Q series C24 retrieves and keeps the information of PLC CPUs of other stations after started up.
When replacing the PLC CPU of other station with which data is communicated after starting up the Q series C24, reboost the Q series C24 if the model name of the PLC CPU is changed (power supply of PLC CPU of local station reset/CPU reset).

(d) Framing error generation of an external device side

A framing error may occur at the external device side when nothing is sent from the Q series 24 to the external device side via RS-422/485 (see Section 3.3.3 of the User’s Manual (Basic)).
Set the external device so that it will skip to read data until the Q series C24 sends either STX, ACK, or NAK.
Refer to the interface specifications for the Q series C24 described in Section 3.3.3 of User's Manual (Basic), before performing data communication.

(e) Installed multiple Q series C24s

When multiple external devices that are connected to each Q series C24 request access to the PLC CPU at the same time, the PLC CPU decides the order of access. The user cannot decide this access priority order.

(f) When controlling the PLC in operation (especially modifications of data, programs and operation status (status control)) by connecting a personal computer or other device to the intelligent function module, read the manual thoroughly and confirm the safety before proceeding.

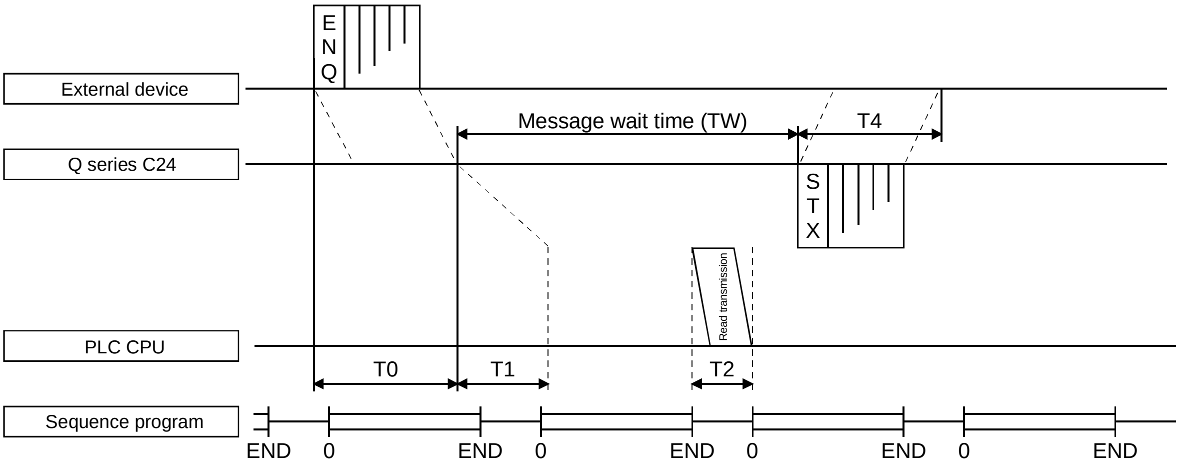
Incorrect data modification, program modification, and status control may cause system malfunctions, damages to machines, and/or fatal errors.

## 2.8 Time Chart and Communication Time of the Transmission Sequence of the Serial Communication Module

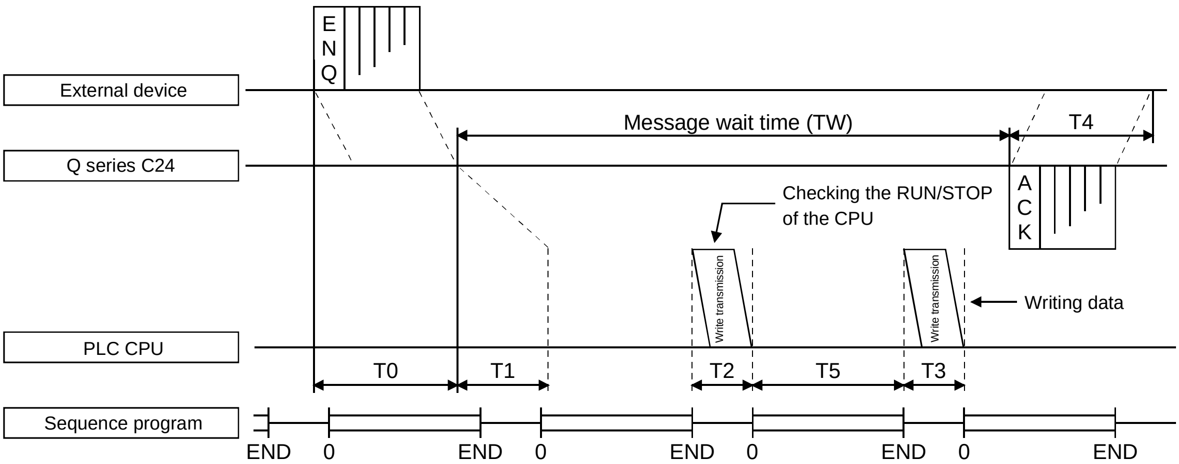
The diagram below shows a communication time chart when an external device accesses the PLC CPU via the Q series C24.

### (1) An external device accesses the PLC CPU on which the Q series C24 is loaded

(a) Reading data (In this example, the massage wait time is set)




(b) Writing data (In this example, massage wait time is set)




REMARK

(1) The communication between the Q series C24 and the PLC CPU is always performed after END. Therefore, the scan time becomes longer according to the communication time (the time to interrupt to the PLC CPU).

(2) The number of scans required for processing when requesting read/write operations to the PLC CPU differs depending on the content of the request. For example, when a read operation that requires two scans is requested, it takes an extra time of 1 scan + T2.

(3) For details on how many scans are required for processing each read/write request, see Appendix 3.
For the number of points processed per communication, see the command list reference sections in Chapters 3 through 6.

### (2) Transmission time of transmission sequence

This section explains how to roughly estimate the time from when an external device starts data transmission till when the result is returned from the Q series C24.
See (1) in the previous page for the contents of T0 to T5.

Use the average values of the above-mentioned rough estimate time described below as a standard for processing time.

(a) When an external device reads data from the PLC (Unit: ms)
Communication time = request message transmission time (T0) + ([T1 + {1.5 × scan time} + T2] and TW with longer time) + response message transmission time (T4)

(b) When an external device writes data in the PLC
Communication time = request message transmission time (T0) + ([T1 + {1.5 × scan time} + T2 + T3 + T5] and TW with longer time) + response message transmission time (T4)

T0, T4 = (1/Transmission rate) × number of bits for 1 byte at transmission $(\underbrace{1}_{\text{Start bit}} + \underbrace{7/8}_{\text{Data length (7 or 8)}} + \underbrace{0/1}_{\text{Parity bit (0 or 1)}} + \underbrace{1/2}_{\text{Stop bit (1 or 2)}}) \times$ number of bytes

T1 = Q series C24 processing time
Depending on the access points (MIN to MAX), the following ranges.

| | |
|---|---|
| QJ71C24N (-R2/R4): | 12.5 to 45.0 ms |
| QJ71C24 (-R2): | 13.0 to 70.0 ms |

T2, T3 = CPU intervention time (See Appendix 3.) *1

T5 = Scan time *1

TW = The set time when message wait time is set

*1 For functions that can process one scan, T3 and T5 equal 0.

## 2.9 Transmission Time When Accessing Other Stations Via MELSECNET/H, MELSECNET/10

This section explains the transmission time (T1) when accessing the PLC of other station on MELSECNET/H, MELSECNET/10.

### (1) PLC to PLC network

Transmission time (T1) = (Transmission delay time [∗1] + 1 scan time of the station on whitch the Q series C24/E71 is loaded or internal processing time) × (n [∗2] + 1 [∗3])
(When this value is more than the internal processing time.)

Internal processing time Q series C24 : Approximately 50 ms, Q series E71 : Approximately 30 ms

∗1 Refer to the applicable section for the transmission delay time in the reference manual for the network system.

∗2
- When communicating with the applicable station for the first time after the power supply is turned on or the CPU is reset.
- When communicating with stations other than the 16 last stations communicated with.

} n=6

- When the number of stations communicated with is less than 16 and communication is performed for the second time.
- When communicating for the second time to one the 16 last stations communicated with.

} n=1

∗3 Add only when writing data from the external device if "Write prohibit during RUN" is set in the Q series C24/E71.
(Set in the switch setting with the GX Developer for the Q series C24 and set in the operation setting with the GX Developer for the Q series E71.)

(Example)

Loading the Q series C24 on a station on MELSECNET/H and reading the device memory of other station on the same MELSECNET/H.
(When the number of stations communicated with is eight and communicating for the second time at the following processing time/settings.)

- $S_T$ : Scan time of transmission side 12 ms
- $S_R$ : Scan time of reception side 10 ms
- LS : Link scan time 9 ms
- $\alpha_T$ : Link refresh time of transmission side 5 ms
- $\alpha_R$ : Link refresh time of reception side 5 ms
- Simultaneous transient request : 2
- Maximum transient request : 1

(Normal transmission delay time: MELSECNET/H mode)

Transmission time (T1) = {(12 + 5 + 10 + 5) × 2 + 9 × 4 + ((2/1) -1) × 9 × 2) + 50} × 1

(168 ms) ($S_T \cdot \alpha_T \cdot S_R \cdot \alpha_R$) ($L_S$) ($L_S$)

Simultaneous transient request → 2
Maximum transient request → 1
Q series C24 internal processing time → 50

(Worst-case transmission delay time: MELSECNET/H mode)

Transmission time (T1) = {(12 + 5 + 10 + 5) × 2 + 9 × 6 + ((2/1) -1) × 9 × 2) + 50} × 1

(186 ms) (ST · $\alpha_T$ · $S_R$ · $\alpha_R$) ($L_S$) ($L_S$)

Simultaneous transient request

Maximum transient request

Q series C24 internal processing time

* Causes of transmission time (T1) delay
  When a command that takes two scans to send is issued, the transmission time is twice the value obtained by the above expression.
  For more details on when other modules and GX Developer request access to the Q/QnACPU at the same time, see REMARK in Section 2.4.
* For more detailed information about the network system, refer to the Q corresponding MELSECNET/H Network System Reference Manual (PLC to PLC network).

**POINT**

Data transmission to the PLC CPU of stations on MELSECNET/H, MELSECNET/10 other than the local station may be delayed considerably depending on conditions.
The transmission delay time can be kept short by limiting communication between an external device and the PLC CPU to stations on which the Q series C24/E71 is loaded and communicating with PLCs on other stations by data link (LB, LM).

(2) Remote I/O network

Transmission time (T1) = (Transmission delay time + 1 link scan time or internal processing time) × (n + 1)

∗1 (When this value is more than the internal processing time.) ∗2 ∗3

Internal processing time Q series C24 : Approximately 50 ms, Q series E71 : Approximately 30 ms

∗1 Refer to the applicable section for the transmission delay time in the reference manual (remote I/O network) for the network system.

∗2
- When performing initial communication for the applicable station after link begins.
- When communicating with stations other than the 16 last stations communicated with.

} n=6

- When the number of stations communicated with is less than 16 and communication is performed for the second time.
- When communicating for the second time to one the 16 last stations communicated with.

} n=1

∗3 Added only when data is written from the external device where "Write prohibit during RUN" has been set in the Q series C24/E71.
(Set in the switch setting with the GX Developer for the Q series C24 and set in the operation setting with the GX Developer for the Q series E71.)

(Example)

Loading the Q series C24 on a station on MELSECNET/H (remote I/O network) and reading the device memory of other station on the same MELSECNET/H.
(When the number of stations communicated with is eight and communicating for the second time at the following processing time/settings.)

- Sm : Scan time for the sequence program in the remote master station 12 ms
- αm : Link refresh time for the remote master station 5 ms
- LS : Link scan time 9 ms

In the (Example), transmission time (T1) is calculated as follows since (Sm) > (LS).
(When the master station is station 1)

Transmission time (T1) = {(12 + 5) × 3 + 50} × 1
(101 ms) (Sm · αm)

∗ Causes of transmission time (T1) delay
When a command that takes two scans to send is issued, the transmission time is twice the value obtained by the above expression.
For more details on when other modules and GX Developer request access to the Q/QnACPU at the same time, see Remark in Section 2.4.

∗ For more detailed information about the network system, refer to the Q corresponding MELSECNET/H Network System Reference Manual (Remote I/O network).

**POINT**

Data transmission to the PLC CPU of stations on MELSECNET/H other than the local station may be delayed considerably depending on conditions.
The transmission delay time can be kept short by limiting communication between an external device and the PLC CPU to stations on which the Q series C24/E71 is loaded and communicating with PLCs on other stations by data link (LB, LM).

## 2.10 Compatibility with Multiple CPU Systems

This section explains communication using the MC protocol when the QCPU is configured as a multiple CPU system.

**POINT**

Read this section if the QCPU is configured as a multiple CPU system.
See the QCPU User's Manual (Multiple CPU System) first when using the Q series C24/E71 with a multiple CPU system QCPU.

### (1) Accessing the local station multiple CPU system

(a) When using the Q series C24/E71with a multiple CPU system, use the function version B Q series C24/E71.

(b) When accessing the local station multiple CPU system from an external device, both the control PLC (set using GX Developer) and non control PLC for the Q series C24/E71 can be accessed.

* The commands that can be used will differ depending on the control PLC and non control PLC accessed.

* The function version A Q series C24/E71 can also be used so PLC No. 1 of the QCPU will be the control PLC of the Q series C24/E71.
In this case, only the control PLC of the Q series C24/E71 can be accessed with accessing the local station QCPU from an external device.

### (2) Accessing a multiple CPU system in another station

(a) Use a module with function version B for the applicable modules of each station indicated below:

| | Module | Remarks |
|---|---|---|
| Local station | QCPU | — |
| | Q series C24/E71 connected to an external device | The control PLC for the Q series C24/E71 and network module should be the same |
| | Network module connected to the relay station/access station | |
| Relay station | QCPU | — |
| | Two modules connecting the local station and access station network | |
| Access station | QCPU | |
| | Network module connected to the relay station | |

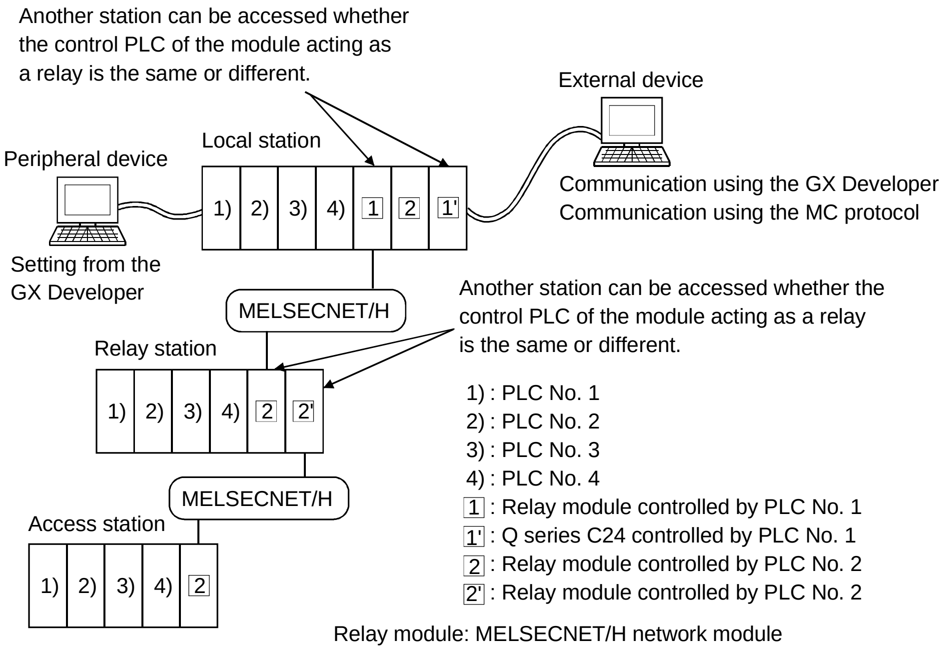
(b) When accessing a multiple CPU system in another station (access station), both the control PLC and non control PLC of the network module connected to the relay station can be accessed.

* The commands that can be used differ depending the on the QCPU (control PLC, non control PLC) of the access target.

* When accessing another station from an external device, only the control PLC of the network module connected to the relay station can be accessed if a function version A module is included in one of the above modules of the local station, relay station or access station. Also, another station that relays via a module controlled by the same control PLC can be accessed.

* The following are the modules that can relay via the network when accessing other station.
- MELSECNET/H, MELSECNET/10 network module
- Q series C24 • Q series E71

(Example)




**POINT**

By setting the routing parameter to relay stations within the scope of the network system specifications, PLCs in other MELSECNET/H, MELSECNET/10 and Ethernet specified by the user, can be accessed.

### (3) QCPU processing time when accessing a multiple CPU system

(a) Accessing the control PLC
The PLC CPU side processing time when accessing the following control PLC will be the same as the processing time for accessing a single CPU system as indicated in Addendum Section 3. (∗1)
- Control PLC for the local station Q series C24/E71
- Control PLC for the network module of another station

(b) Accessing a non control PLC
More PLC CPU processing time is required (approximately 9 ms per command) when accessing the following non control PLC as compared with the processing time when accessing a single CPU system. (∗1)
- Non control PLC for the local station Q series C24/E71
- Non control PLC for the network module of another station

∗1 The processing time per command when accessing a multiple CPU system will be further increased for the following reasons. See the QCPU (Q Mode) User's Manual (Function Explanation, Program Fundamentals) for more details.
- QCPU operating status (during RUN)
- User and execution status of the automatic refresh function between QCPUs
- Status of access and automatic refresh between the QCPU and intelligent device function module

(4) Stations that can access to the communication frame for the MC protocol

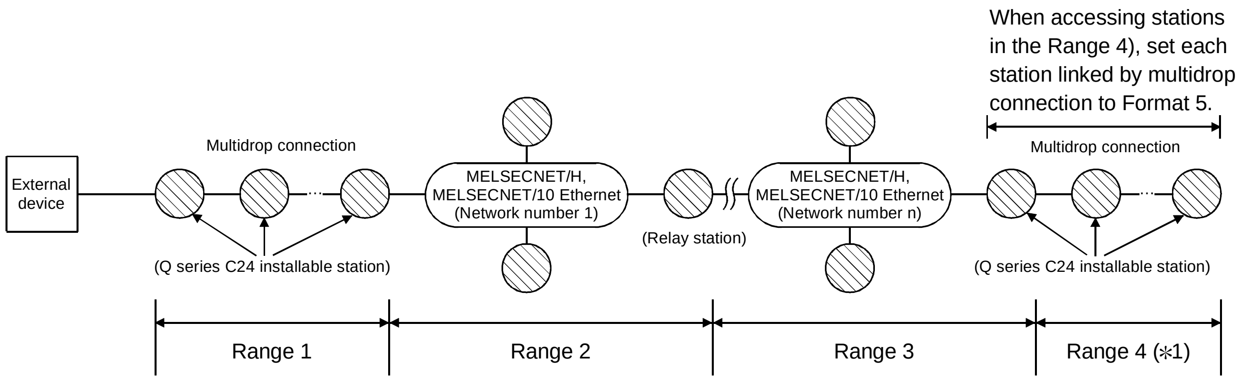
(a) The stations that can access the MC protocol communication frame are indicated, including access with respect to a multiple CPU system.
Use the following frames when accessing a non control PLC in a multiple CPU system.
- Q series C24: QnA compatible 4C frame (Formats 1 to 5)
- Q series E71: QnA compatible 3E frame

Ranges 1 through 4 in the table correspond to ranges 1 through 4 shown in the illustration below.
The control/non control in the table indicates the QCPU when the access station is a multiple CPU system and the meaning is as follows:
- Control : Indicates the QCPU controlling the Q series C24/E71 connected to the external device or the relaying network module.
- Non control : Indicates the QCPU that does not control the Q series C24/E71 connected to the external device or the relaying network module

| Applicable module | Frame used | | Stations that can be accessed | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | Range 1 | | Range 2 | | Range 3 | | Range 4 | |
| | | | Control | Non control | Control | Non control | Control | Non control | Control | Non control |
| Q series C24 | QnA compatible 3C frame | Formats 1 to 4 | ○ | × | ○ | × | ○ | × | × | |
| | QnA compatible 4C frame | Formats 1 to 4 | ○ | | ○ | | ○ | | ○ | × |
| | | Format 5 | ○ | | ○ | | ○ | | ○ | × |
| | QnA compatible 2C frame | Formats 1 to 4 | ○ | × | × | | × | | × | |
| | QnA compatible 1C frame | | ○ | × | ○ | × | × | | × | |
| Q series E71 | QnA compatible 3E frame | | ○ | | ○ | | ○ | | ○ | × |
| | A compatible 1E frame | | ○ | × | ○ | × | × | | × | |

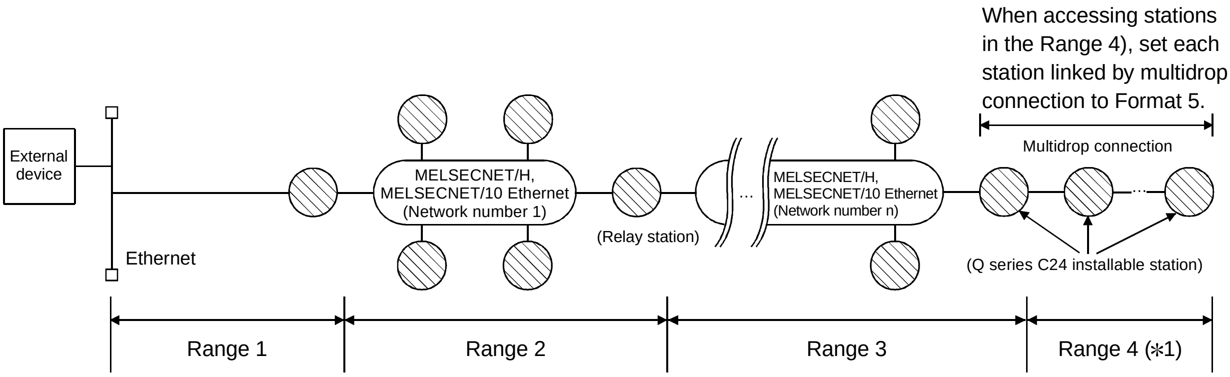
○: Accessible, ×: Not accessible

[When using the Q series C24]




*1 In the above illustration, only the control PLC for the applicable station's Q series C24 can be accessed when the station in range 4 is a multiple CPU system.

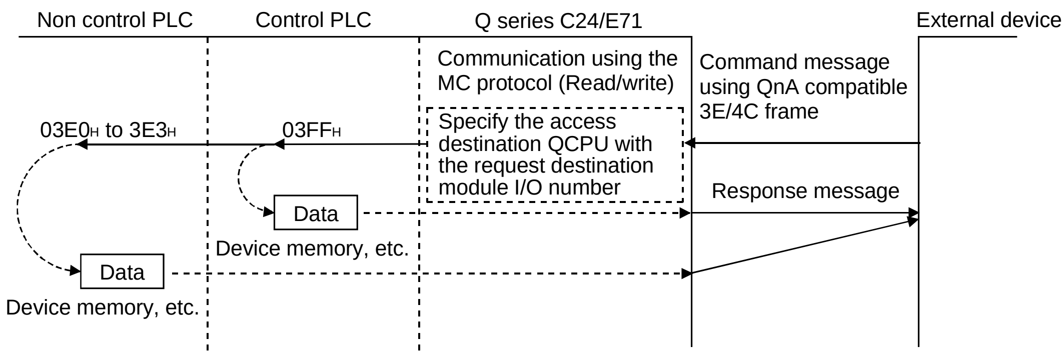
[When using the Q series E71]




*1 In the above illustration, only the control PLC for the applicable station's Q series C24 can be accessed when the station in range 4 is a multiple CPU system.

(b) The QCPU to be accessed in a multiple CPU system when the MC protocol QnA compatible 3E/4C frame is used, is specified using the data item "Request target module I/O number in the QnA compatible 3E/4C frame.

| | External device access station | Request destination module I/O number |
|---|---|---|
| 1 | Control PLC | 03FFH |
| 2 | CPU No.1 | 03E0H |
| 3 | CPU No.2 | 03E1H |
| 4 | CPU No.3 | 03E2H |
| 5 | CPU No.4 | 03E3H |

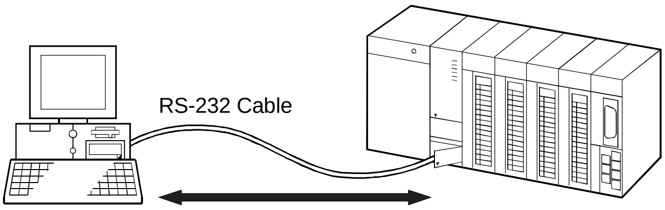



(c) The control PLC/non control PLC and buffer memory of the intelligent function module can be accessed using the MC protocol.
Access functions that can be used vary depending on the control PLC and non control PLC.
For details on each command, see the list of commands and functions for the QnA compatible 3E/3C/4C frames found in Section 3.2 of this manual.

## 2.11 Compatibility with the Q00CPU, Q01CPU Serial Communications Function

### (1) Serial Communications Function

(a) The serial communications function is a function for connecting the RS-232 interface of the Q00/Q01CPU with an external device, and for monitoring and controlling the operation of the Q00/Q01CPU from the external device by communications through the MC protocol.




(b) With the serial communications function, the host Q00/Q01CPU connected to the RS-232 interface can be accessed.

Other stations cannot be accessed via the following modules.

- MELSECNET/H, MELSECNET/10 network modules
- Q Series C24
- Ethernet Module

(c) The serial communications function can be set by the GX Developer and used.

See the QCPU User's Manual (Function Explanation, Program Fundamentals) concerning the serial communications function.

### (2) About the MC protocol communications frame

(a) If the Q00/Q01CPU is being accessed by the serial communications function, the following frames can be accessed.
See Section 3.1.4 and Section 3.1.5 for the format of each frame.

| Function | | Type 4 | Type 5 |
|---|---|---|---|
| Communications by ASCII code | QnA compatible 3C Frame | ○ | × |
| | QnA compatible 4C Frame | ○ | × |
| Communications by binary code | QnA compatible 4C Frame | × | ○ |

○: Can be used ×: Cannot be used

(b) Commands used when accessing the Q00/Q01CPU from an external device using the serial communications function, and their functions, are shown in the following table.
See Section 3.3.1(3) for Q00/Q01CPU devices that be accessed and the device No. range.

| Function | | | Command (Subcommand) | Description, processing | Number of points accessed |
|---|---|---|---|---|---|
| Device memory | Batch read | Bit units | 0401(00□1) | Read bit devices (X, Y, M, etc.) in one-point units. | ASCII: 3584 points<br>Binary: 7168 points |
| | | Word units | 0401(00□0) | Read bit devices (X, Y, M, etc.) in 16-point units. | 480 Words (7680 points) |
| | | | | Read word devices (D, R, T, C, etc) in one-point units. | 480 points |
| | Batch Write | Bit units | 1401(00□1) | Write to bit devices (X, Y, M, etc.) in one-point units. | ASCII: 3584 points<br>Binary: 7168 points |
| | | Word units | 1401(00□0) | Write to bit devices (X, Y, M, etc.) in 16-point units. | 480 words (7680 points) |
| | | | | Write to word devices (D, R, T, C, etc.) in one-point units. | 480 points |
| | Random read | Word units | 0403(00□0) | Read bit devices (X, Y, M, etc.) in 16-point and 32-point units by designating the device and device number at random. | 96 points |
| | | | | Read word devices, (D, R, T, C, etc.) in one-point and two-point units by designating the device and device number at random. | |
| | Test (random write) | Bit units | 1402(00□1) | Set/reset bit devices (X, Y, M, etc.) in one-point units by designating the device and device number at random. | 94 points |
| | | Word units | 1402(00□0) | Set/reset bit devices (X, Y, M, etc.) in 16-point units by designating the device and device number at random. | (∗1) |
| | | | | Write to word devices (D, R, T, C, etc.) in one-point and two-point units by designating the device and device number at random. | |
| | Monitor data registra-tion | Word units | 0801(00□0) | Registers the bit devices (X, Y, M, etc.) to be monitored in 16-point units. | 96 points |
| | | | | Registers the word devices (D, R, T, C, etc.) to be monitored in one-point and two-point units. | |
| | Monitor | Word units | 0802(0000) | Monitors the device registered for monitoring. | (The number of points registered) |

∗1 Set the number of points accessed within the range shown below.
(Word access points) × 12 + (Double word access points) × 14 ≦ 960
- The bit device is accessed 16 bits at a time for one point during word access and 32 bits at a time for one point during double word access.
- The word device is accessed one word at a time for 1 point during word access and at 2 words at a time for one point during double word access.

# 3 WHEN COMMUNICATING USING THE QnA COMPATIBLE 3E/3C/4C FRAMES

This chapter explains how to specify the formats and data of messages, and the restrictions when performing MC protocol data communication with the Q series C24/E71 using the frames shown below.
The table below lists the frames that can be used for data communication with the Q series C24 and E71.

| Frame type | Q series E71 | Q series C24 | Note |
|---|---|---|---|
| QnA compatible 3E frame | Communication possible | Communication not possible | Same as the frames for the QnA series QE71 |
| QnA compatible 3C frame | Communication not possible | Communication possible | Same as the QnA frames for the QnA series QC24 (N) |
| QnA compatible 4C frame | Communication not possible | Communication possible | Same as the QnA extension frames for the QnA series QC24 (N) |

## 3.1 Message Formats

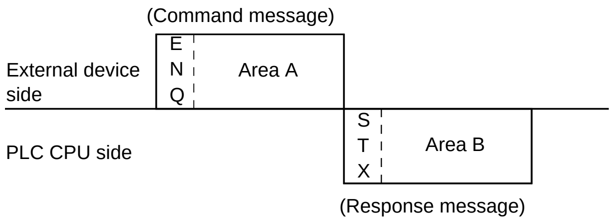
This section describes the message format for each command when data communication is performed using QnA compatible 3E/3C/4C frames.

### 3.1.1 How to read the command description sections

3

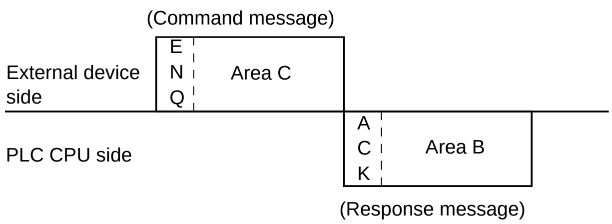
The following explains how to read the message explanation diagrams shown in each of the command description sections in Sections 3.3.2 to 3.18.
The following example of communication with the Q series C24 demonstrates how to read the transmission data illustrated in the succeeding diagrams that explain each control procedure.

(1) When an external device reads data from the PLC




1) Area A indicates transmission from the external device to the PLC CPU.
2) Area B indicates transmission from the PLC CPU to the external device.
3) The program of the external device is generated so that the data is transmitted sequentially from left to right. (For example: in case of area A, data should be sequentially sent from ENQ to the right.)

(2) When data is written from the external device to the PLC




1) Area C indicates transmission from the external device to the PLC CPU.
2) Area B indicates transmission from the PLC CPU to the external device.
3) The program of the external device is generated so that the data is transmitted sequentially from left to right. (For example: in case of area C, data should be sequentially sent from ENQ to the right.)

**POINT**

When the PLC CPU receives a command message from an external device, it completes processing of the data in area A/C, then sends a response message and waits for the next command message (neutral state).

### 3.1.2 Message format and control procedures of QnA compatible 3E frames

This section explains the message format and control procedures when data communication is performed using QnA compatible 3E frames for the Q series E71.

(1) Data format

When communicating data between the Q series E71 and an external device, the data format described below is used.
The communication data consists of a "header" and "application data" as shown below.

| | |
|---|---|
| Header | Application data |

(2) Header

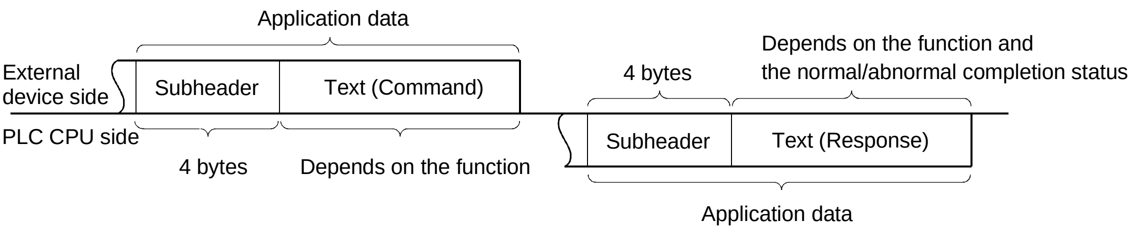
The header for TCP/IP or UDP/IP is used. It is not required for the user to set it at the PLC CPU side since it is added by the Q series E71.

(3) Application data

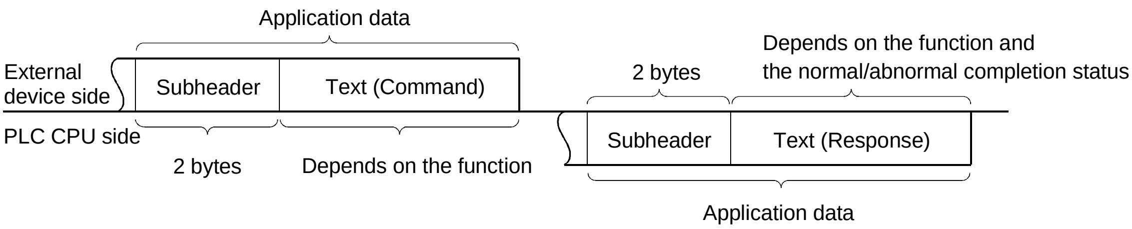
The application data is largely divided into a "subheader" and "text" as shown below. The subheader represents a command/response and the setting value is predetermined. The text contains the request data (command) and response data (response) for each function and its data is set in the prescribed format (for more details, see Section 3.2 and succeeding sections).

(4) Format

Communication in ASCII code




Communication in binary code




REMARK

When communicating using the MC protocol, it is not required for the user to set a response for a command from an external device; the Q series E71 generates and returns it.

## (5) Subheader

During data communication, the following codes and line are transmitted/received.

| | Command | | | | Response | | | |
|---|---|---|---|---|---|---|---|---|
| ASCII codes | 5<br>$35_H$ | 0<br>$30_H$ | 0<br>$30_H$ | 0<br>$30_H$ | D<br>$44_H$ | 0<br>$30_H$ | 0<br>$30_H$ | 0<br>$30_H$ |
| Binary codes | $50_H$ | $00_H$ | | | $D0_H$ | $00_H$ | | |

## (6) Control procedures

The control procedures and formats of the application data section are as follows when QnA compatible 3E frames are used.
The □ section shown in the message explanation diagram of this section are items common to all commands and correspond to the * portion of the message explanation diagrams indicated in sections 3.3.2 and after of this chapter.
See section 3.1.3 regarding the contents and specification method of the data found in □.

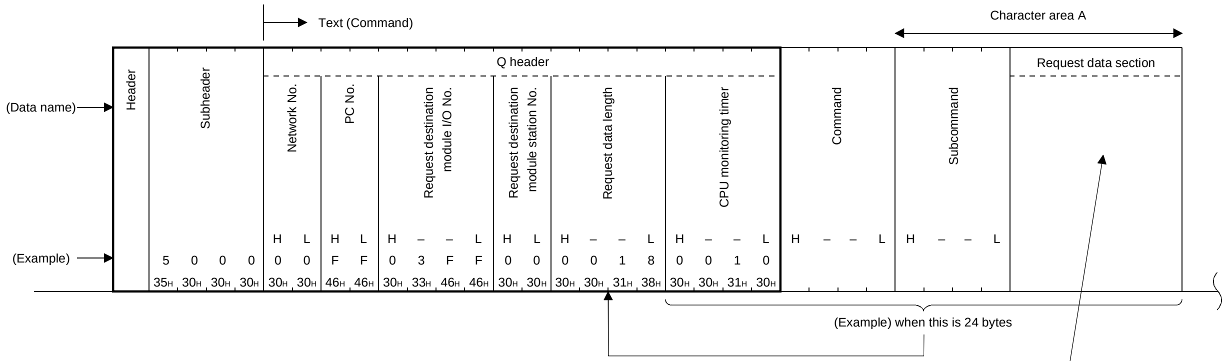
**POINT**

The format of the QnA compatible 3E frame is determined by the communication data code setting in the GX Developer "Ethernet operation setting."

(a) Communication using ASCII code

1) When reading data from the local station PLC CPU at the external device

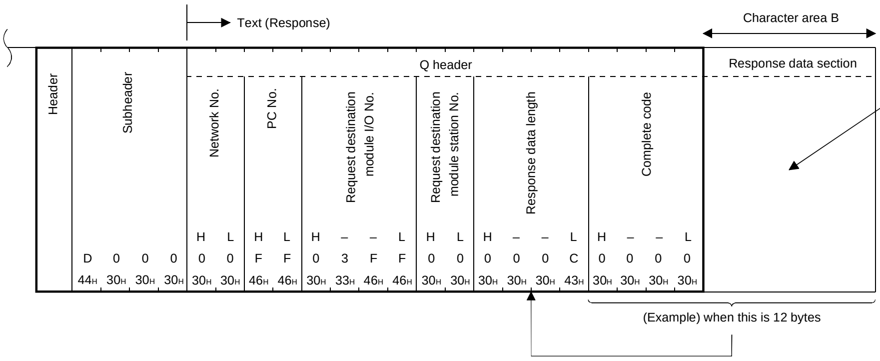
External device side → PLC CPU side (Command message)



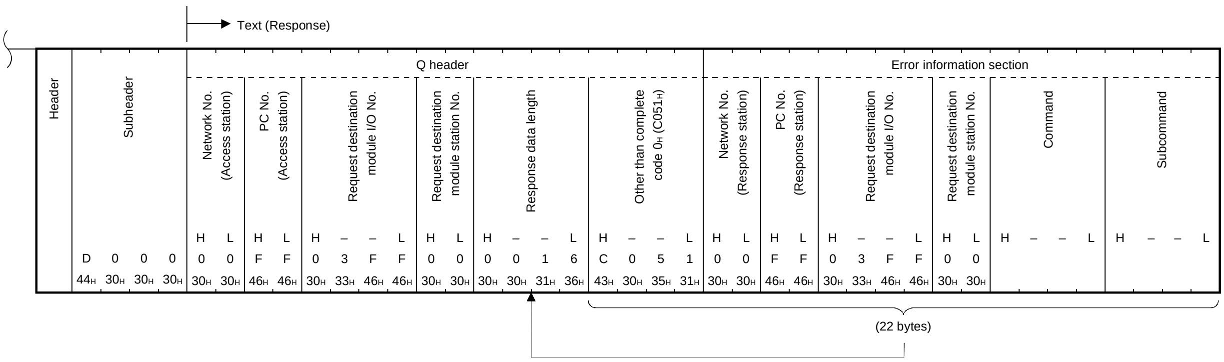

PLC CPU side → External device side (Response message)

The arrangement of the data items differs according to the command and sub command.
See the sections explaining commands found in Section 3.3.2 and after for details.

(Normal completion)




(Abnormal completion)



2) When writing data from the local station PLC CPU at the external device

**External device side → PLC CPU side (Command message)**

Text (Command) →; Q header; Character area C

| (Data name) | Header | Subheader | Network No. | PC No. | Request destination module I/O No. | Request destination module station No. | Request data length | CPU monitoring timer | Command | Subcommand | Request data section |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | H L | H L | H – – L | H L | H – – L | H – – L | H – – L | H – – L | |
| (Example) | | 5 0 0 0 | 0 0 | F F | 0 3 F F | 0 0 | 0 0 2 0 | 0 0 1 0 | | | |
| | | 35H 30H 30H 30H | 30H 30H | 46H 46H | 30H 33H 46H 46H | 30H 30H | 30H 30H 32H 30H | 30H 30H 31H 30H | | | |

(Example) when this is 32 bytes

The arrangement of the data items differs according to the command and sub command.
See the sections explaining commands found in Section 3.3.2 and after for details.

**PLC CPU side → External device side (Response message)**

(Normal completion)

Text (Response) →; Q header

| Header | Subheader | Network No. | PC No. | Request destination module I/O No. | Request destination module station No. | Response data length | Complete code |
|---|---|---|---|---|---|---|---|
| | | H L | H L | H – – L | H L | H – – L | H – – L |
| | D 0 0 0 | 0 0 | F F | 0 3 F F | 0 0 | 0 0 0 4 | 0 0 0 0 |
| | 44H 30H 30H 30H | 30H 30H | 46H 46H | 30H 33H 46H 46H | 30H 30H | 30H 30H 30H 34H | 30H 30H 30H 30H |

(4 bytes)

(Abnormal completion)

Text (Response) →; Q header; Error information section

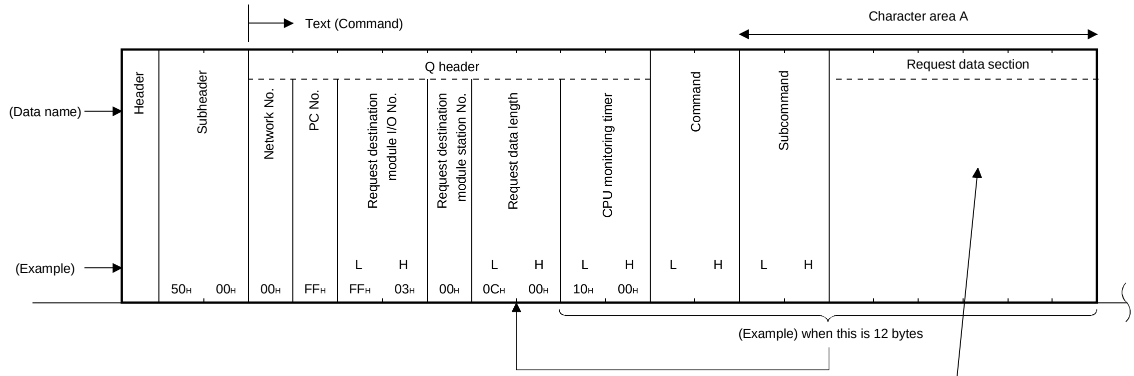
| Header | Subheader | Network No. (Access station) | PC No. (Access station) | Request destination module I/O No. | Request destination module station No. | Response data length | Other than complete code 0H (C051H) | Network No. (Response station) | PC No. (Response station) | Request destination module I/O No. | Request destination module station No. | Command | Subcommand |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | H L | H L | H – – L | H L | H – – L | H – – L | H L | H L | H – – L | H L | H – – L | H – – L |
| | D 0 0 0 | 0 0 | F F | 0 3 F F | 0 0 | 0 0 1 6 | C 0 5 1 | 0 0 | F F | 0 3 F F | 0 0 | | |
| | 44H 30H 30H 30H | 30H 30H | 46H 46H | 30H 33H 46H 46H | 30H 30H | 30H 30H 31H 36H | 43H 30H 35H 31H | 30H 30H | 46H 46H | 30H 33H 46H 46H | 30H 30H | | |

(22 bytes)

(b) Communication using binary code

1) When reading data from the local station PLC CPU at the external device

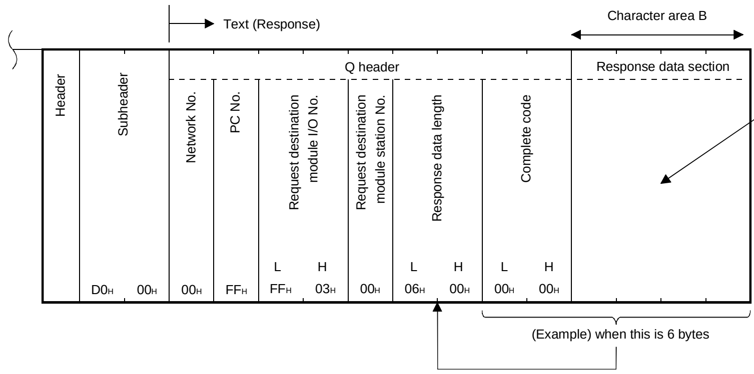
External device side → PLC CPU side (Command message)



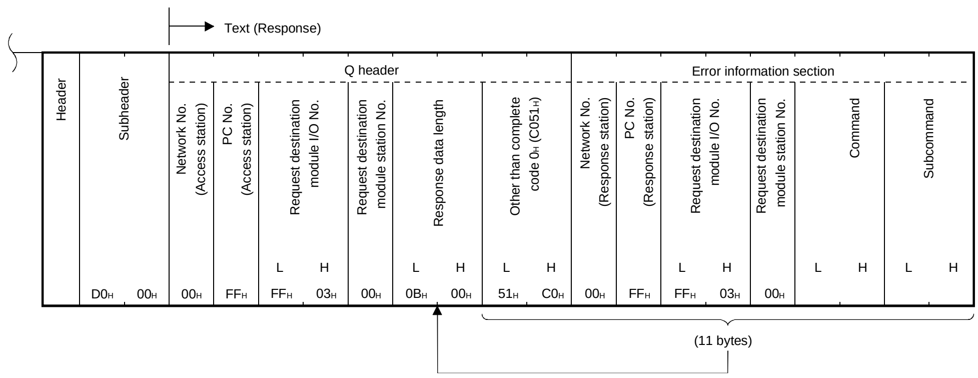

PLC CPU side → External device side (Response message)

(Normal completion)

The arrangement of the data items differs according to the command and sub command.
See the sections explaining commands found in Section 3.3.2 and after for details.

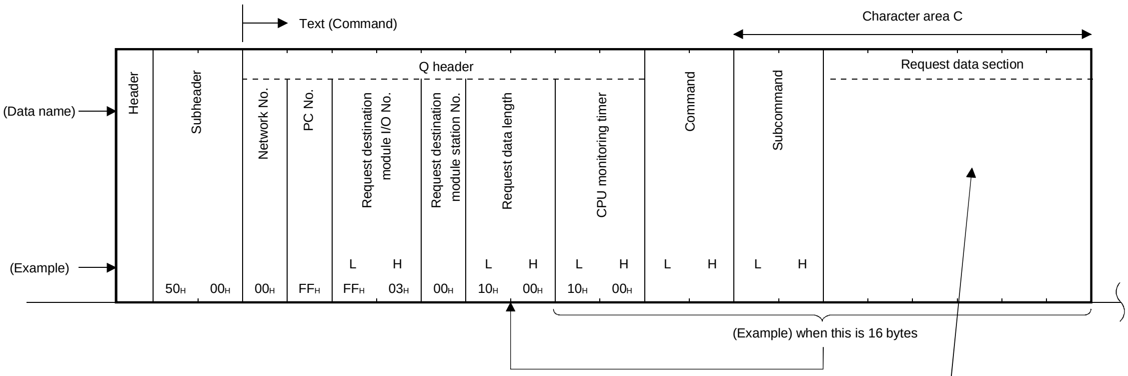



(Abnormal completion)



2) When writing data from the local station PLC CPU at the external device

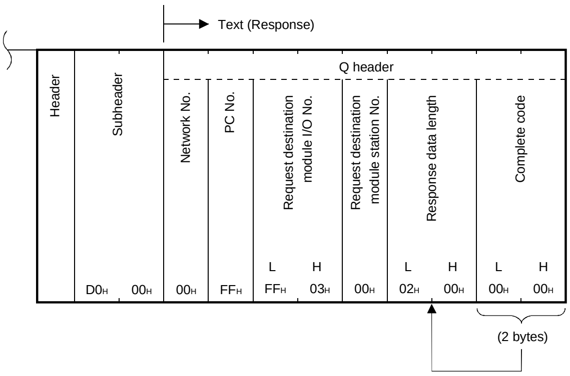
External device side → PLC CPU side (Command message)



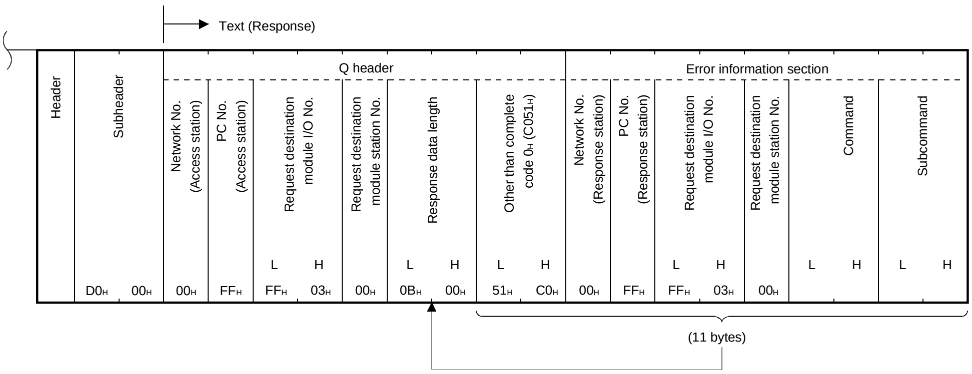

The arrangement of the data items differs according to the command and sub command.
See the sections explaining commands found in Section 3.3.2 and after for details.

PLC CPU side → External device side (Response message)

(Normal completion)




(Abnormal completion)



### 3.1.3 Data designation items for QnA compatible 3E frames

This section explains common data items in the application data section in each message and how to specify them when data communication is performed using QnA compatible 3E frames for the Q series E71.

#### (1) Network numbers and PC numbers

When accessing a PLC of another station, specify the number of the network system that is routed last and the PC number (PLC number) of the access station on the corresponding system by setting numbers of network modules, etc. as follows:

| | External device access station | | Number designated by external device: Network number | Number designated by external device: PC number |
|---|---|---|---|---|
| 1 | Q series E71 installable station (local station) | | $00_H$ | $FF_H$ (*1) |
| 2 | Control station on MELSECNET/H, MELSECNET/10 (When Q series E71 is installed to the normal station) | Excluding 1 above | $01_H$ to $EF_H$ (1 to 239) (*2) | $7D_H$: Designated control station/Master station<br>$7E_H$: Current control station/ Master station |
| 3 | The remote master station in the MELSECNET/H (when the Q series E71 is connected to the remote I/O station) | Excluding 1 above | $01_H$ to $EF_H$ (1 to 239) (*2) | $7D_H$: Designated control station/Master station<br>$7E_H$: Current control station/ Master station |
| 4 | Station on MELSECNET/H, MELSECNET/10 | Excluding 1, 2 and 3 above | $01_H$ to $EF_H$ (1 to 239) (*2) | $01_H$ to $40_H$ (1 to 64) (*3) |
| 5 | Network module routing station, set in "Valid module during other station access" | | $FE_H$ (254) (*4) | $01_H$ to $40_H$ (1 to 64) (*3) |

*1 PLC number $FF_H$ is valid only when the network number is $00_H$.
*2 Designate the access station network number.
*3 Designate the access station number.
*4 The network number $FE_H$ will be disregarded when accessing another station via a Q series C24/E71 loaded on the MELSECNET/H remote I/O station.
If the network number $FE_H$ is specified, the other station specified by the PC number of the MELSECNET/H remote I/O station will be accessed.

**POINT**

(1) The network numbers and the station numbers of a network module are set as decimal numbers, but in communication using ASCII code each number is designated as a hexadecimal number.
(Example)
Network number "10": Designated network number "0A"

(2) No access can be performed via MELSECNET/H, MELSECNET/10 with network numbers from 240 to 255.

(3) When designating network number $FE_H$ and accessing another station via a Q series E71 installable station in a network system, set the following parameters using GX Developer in the PLC CPU of the Q series E71 loaded station.
The "Valid module during other station access" setting:
* Sets the routing module when other stations are accessed. Specify this parameter with the number of MNET/10H Ethernet cards setting.

(2) Request destination module I/O No., Request destination module station No.

Specify this when the PLC CPU for the access station is as follows:

- A PLC CPU for a multiple CPU system
- A PLC CPU connected via a multidrop connection using the Q series C24.
- A PLC CPU for a redundant system

The specification method is the same as when a QnA compatible 4C frame is used. Refer to the remarks at Section 3.1.6 when specifying.

∗ A fixed value is specified when the access destination PLC CPU is other than as described above.

- Request destination module I/O number : 03FFH
- Request destination module station number : 00H

(3) CPU monitoring timer

This is a timer for setting the period of time that the Q series E71 (that has received request data from an external device) should wait after outputting a read/write request to a PLC CPU until the result is returned.

(a) Specify the value as shown below.

0000H (0) : Waits infinitely (∗1)

0001 to FFFFH (1 to 65535) : Wait time (Unit: 250ms)

∗1 It keeps waiting until a response is returned from the PLC CPU.

(b) For normal data communications, using a value within the following setting range is recommended.

| Setting range | Communication destination |
|---|---|
| 1 to 40H (0.25 to 10s) | Local station |
| 2 to 240H (0.5 to 60s) | Other stations accessed via MELSECNET/H or MELSECNET/10, or by router relay |

POINT

Accessing the QnACPU or ACPU for the first time requires full time duration preset for the CPU monitoring timer before receiving a response because the CPU type must be identified. Be sure to set a value within the setting range shown in the above (b).

(4) Request data length, response data length

In request data length, designate the number of bytes from the CPU monitoring timer item in the text to the end of the request data section.

The byte size from the complete code item in the text to the response data section /error data section is returned as the response data length.

(5) Command and subcommand

Specify the command and subcommand that indicate the content of a request when an external device reads data from and writes data to the PLC CPU.

Specify the command and subcommand according to each function shown in Section 3.2 and succeeding sections so that it reflects the content of the read/write request.

(6) Request data section, response data section

In the request data section, designate the corresponding data (head device, read/write range, write data, etc.) for when the external device has designated the commands and sub-commands shown above using communication based on MC protocol.
The read data/write data results, etc. corresponding to the request contents from the external device are returned as response data.
Designate the data corresponding to the commands and sub-commands for each function indicated in and following Section 3.2 and read the data.

(7) Complete code

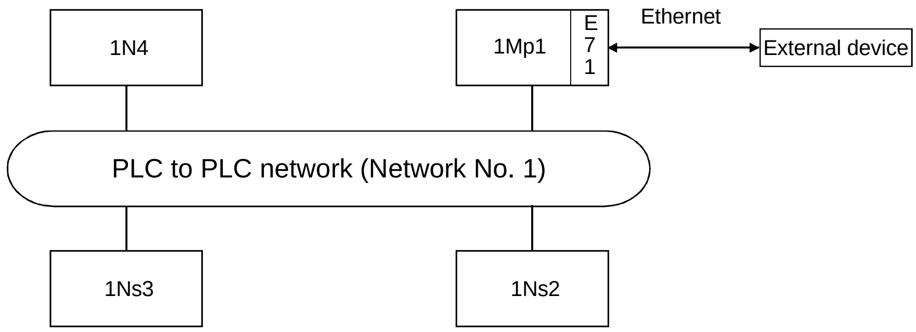
The command process result is returned. In a normal completion, the value shown in the Figure is returned. In an abnormal completion, the error code is returned. (Refer to Chapter 11 of the User's Manual (Basic).)

(8) Error information section

The sequencer with the error response or the command during an error, etc., is returned.

- Network number, PC number
  Network number of sequencer with the error response, PC nubmer.
- Command, sub-command
  Command and sub-command when error is generated.

(9) The following diagram shows an example of network numbers and PC numbers when the station to which the Q series E71 is installed is the control station/normal station of the PLC to PLC network.




Example of network numbers and PC numbers

| When installing the Q series E71 to the control station (1Mp1) | Designated values of PLC to which communication is possible and corresponding items (hexadecimal) | | | | |
|---|---|---|---|---|---|
| | Local station | 1Mp1 | 1Ns2 | 1Ns3 | 1N4 |
| Network number | 00 | – | 01 | | |
| PC number | FF | – | 02 | 03 | 04 |

✻ For details on accessible stations, see Section 2.6.2.

### 3.1.4 Message formats and control procedures of QnA compatible 3C frames

This section explains the control procedures and message formats of each protocol format when data communication is performed using QnA compatible 3C frames for the Q series C24.
The □ section shown in the message explanation diagram of this section corresponds to the ✻ portion of the message explanation diagrams indicated in sections 3.3.2 and after of this chapter.
See Section 3.1.6 regarding the content of data in the □ section and data specification method.

(1) Basic formats for data communication

Five formats of control procedures (command message, response message structure and transmission/reception procedure) are used by an external device to access the PLC using the MC protocol.
Data communications using each frame are possible in a designated format by setting the communication protocol of the PLC CPU's target interface to either "1" through "5" with GX Developer according to the format to be used.

| Communication protocol setting value | Format | Frame with which communication is possible | | | |
|---|---|---|---|---|---|
| | | QnA compatible 3C frame | QnA compatible 4C frame | QnA compatible 2C frame | A compatible 1C frame |
| 1 | Format 1 | ○ | ○ | ○ | ○ |
| 2 | Format 2 | ○ | ○ | ○ | ○ |
| 3 | Format 3 | ○ | ○ | ○ | ○ |
| 4 | Format 4 | ○ | ○ | ○ | ○ |
| 5 | Format 5 | × | ○ | × | × |

✻ Formats 1 to 4 : For communication in ASCII code
Format 5 : For communication in binary code

The differences among the four ASCII code formats are shown below, using format 1 as a reference format:

Format 2 : Format with block number added to each message
Format 3 : Format with each message enclosed by STX and ETX
Format 4 : Format with CR and LF added to each message

For frames other than QnA compatible 3C frames, see the following sections:

- QnA compatible 4C frame : Section 3.1.5
- QnA compatible 2C frame : Section 4.1
- A compatible 1C frame : Section 5.1.2

POINT

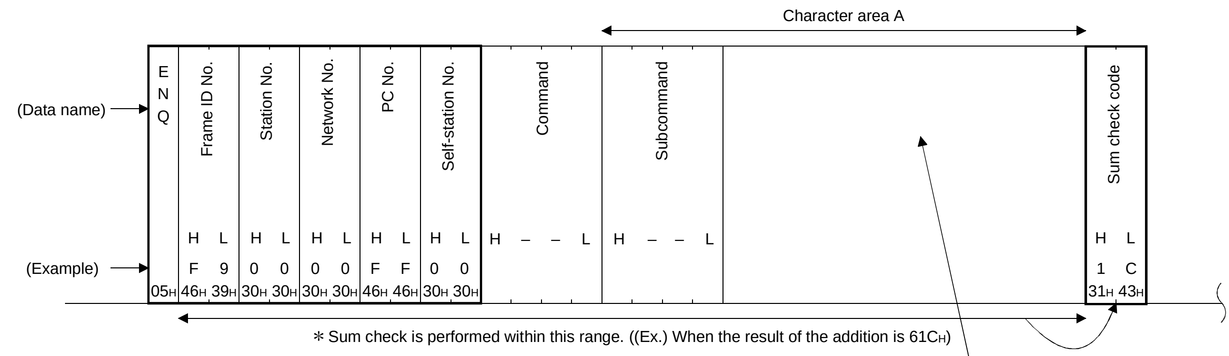
It shows the form 1 to 4 ((2) to (5)) for the QnA compatible 4C frame shown in this chapter.

(a) The sum check code is added only when sum check is enabled. When sum check is disabled, the sum check code is not added.
(b) When sum check is enabled, the sum check is performed only for the characters in the areas indicated by an asterisk (∗) in the diagrams (2) to (5).
(c) The contents of "Character area A," "Character area B," and "Character area C" in the diagrams (2) to (5) depend on the processing contents. See the description of each command for more information. The contents of each character area are the same for all four formats.
(d) The time from the command message reception to the response message transmission by a module can be set between 0 ms to 150 ms. (Set by GX Configurator-SC)
(e) Control procedure format 3 cannot be used when data communication is performed in a system configured with an m:n connection between the external devices and the PLC CPUs.

## (2) Communication in format 1 (for communication in ASCII code)

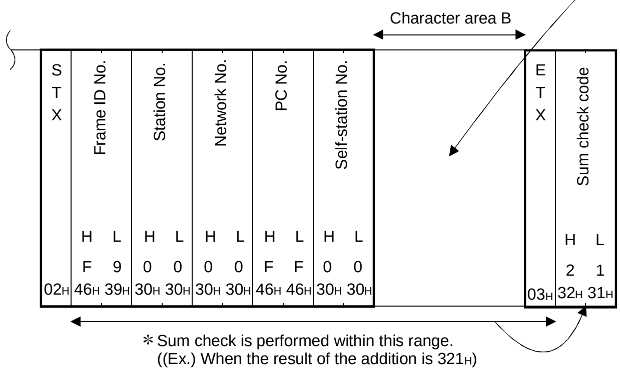
### (a) When reading data from the local station PLC CPU at the external device side

External device side → PLC CPU side (Command message)



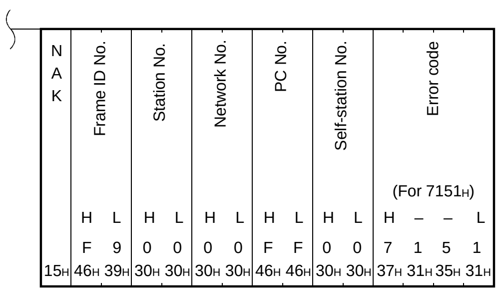

PLC CPU side → External device side (Response message)

(Normal completion)



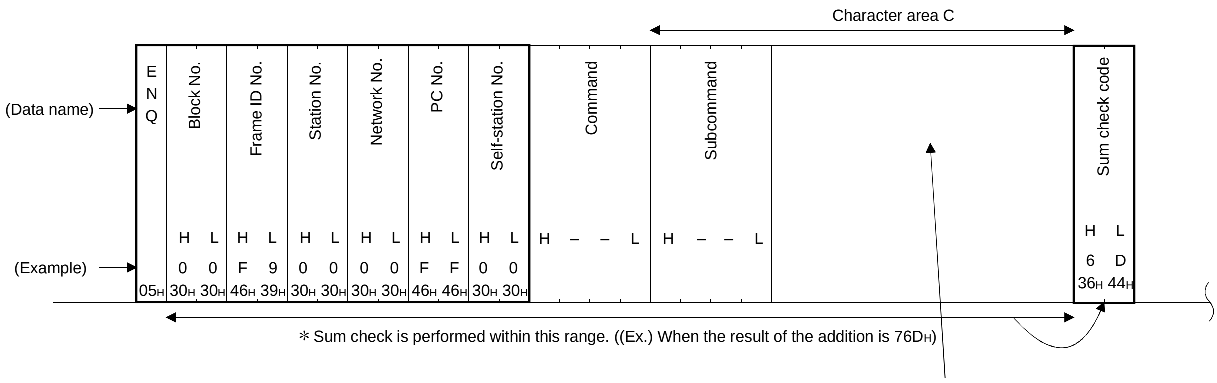

* The arrangement of the data items differs according to the command and sub command. See the sections explaining commands found in Section 3.3.2 and after for details.

(Abnormal completion)



(b) When writing data to the local station PLC CPU from the external device side

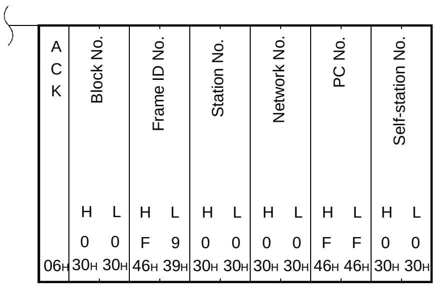
External device side → PLC CPU side (Command message)



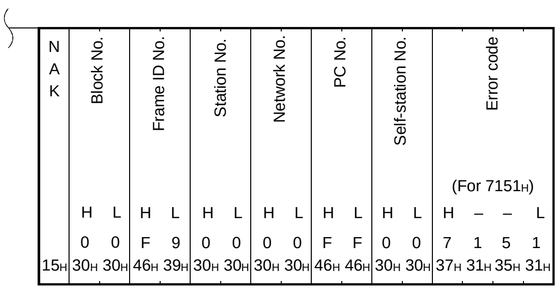

* The arrangement of the data items differs according to the command and sub command. See the sections explaining commands found in Section 3.3.2 and after for details.

PLC CPU side → External device side (Response message)

(Normal completion)



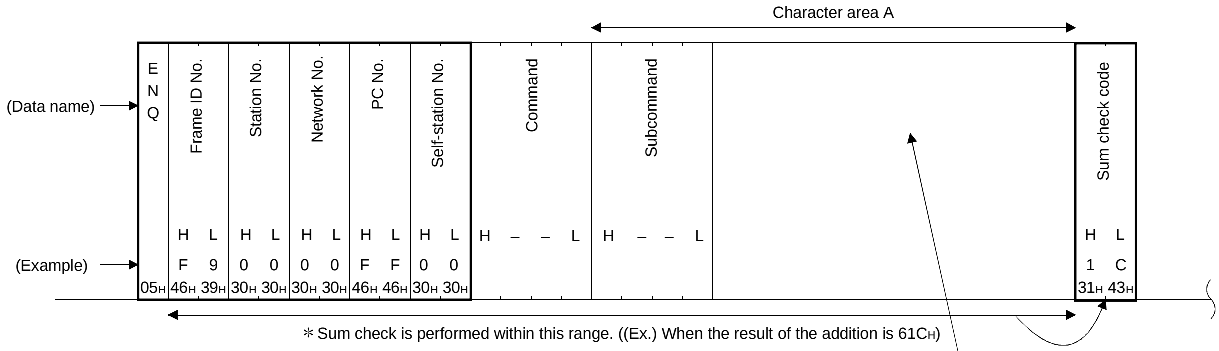

(Abnormal completion)



## (3) Communication in format 2 (for communication in ASCII code)

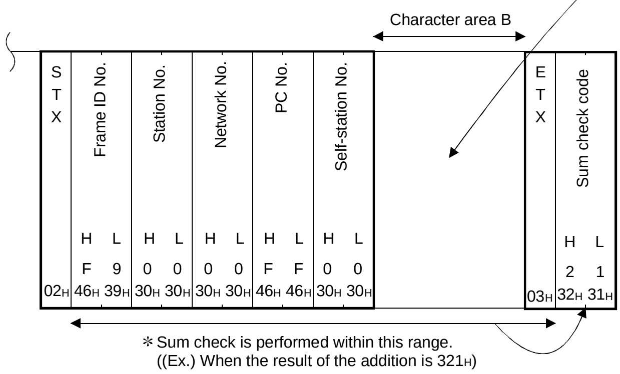
(a) When reading data from the local station PLC CPU at the external device side

External device side → PLC CPU side (Command message)



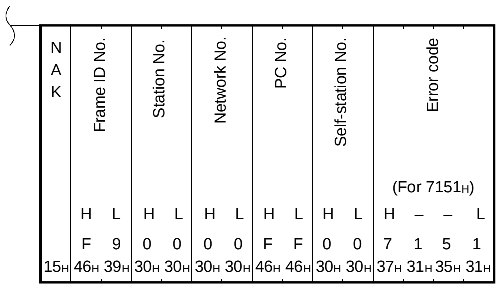

PLC CPU side → External device side (Response message)

(Normal completion)



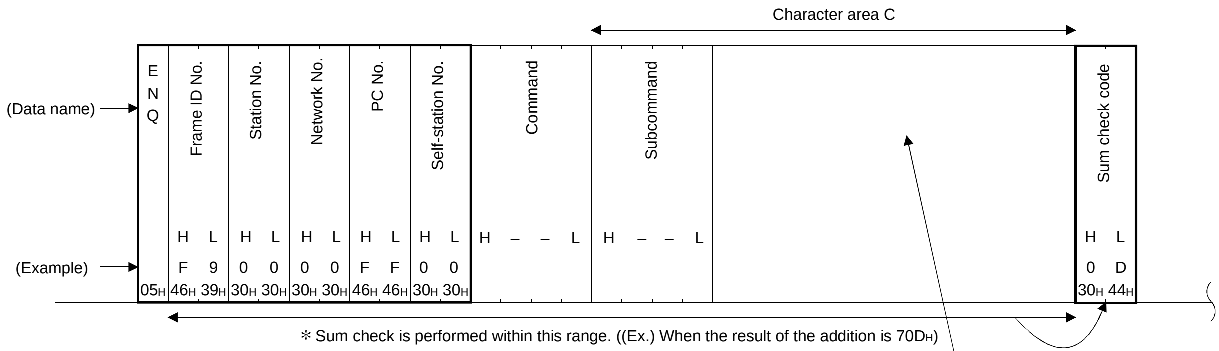

✻ The arrangement of the data items differs according to the command and sub command. See the sections explaining commands found in Section 3.3.2 and after for details.

(Abnormal completion)



(b) When writing data to the local station PLC CPU from the external device side

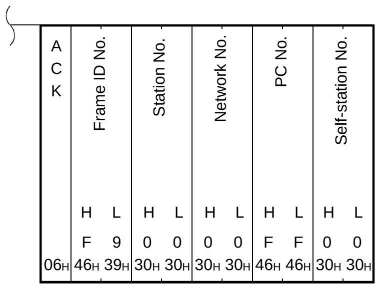
External device side → PLC CPU side (Command message)



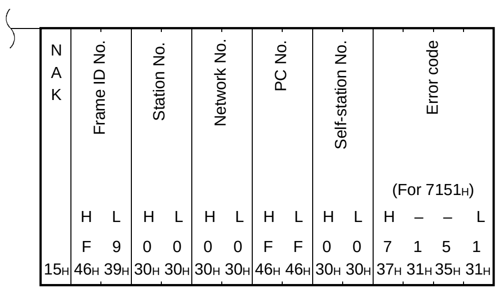

PLC CPU side → External device side (Response message)

(Normal completion)

✻ The arrangement of the data items differs according to the command and sub command. See the sections explaining commands found in Section 3.3.2 and after for details.



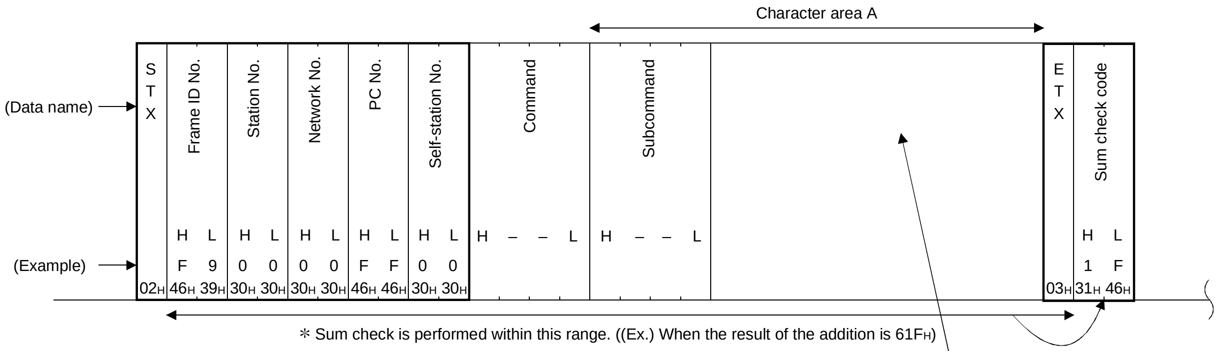

(Abnormal completion)



(4) Communication in format 3 (for communication in ASCII code)

(a) When reading data from the local station PLC CPU at the external device side

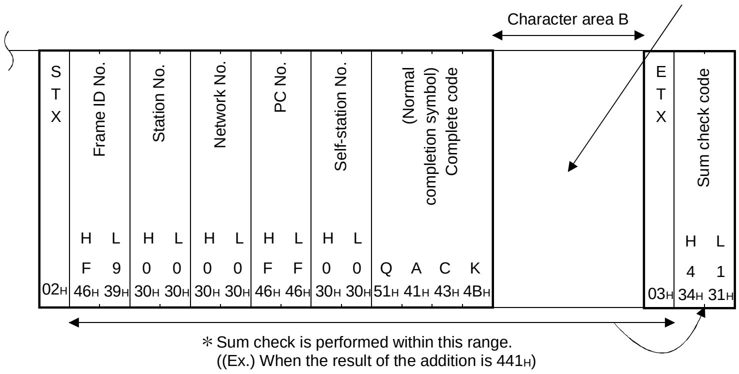
External device side → PLC CPU side (Command message)



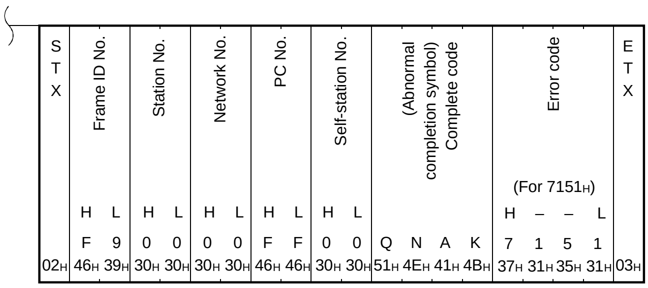

PLC CPU side → External device side (Response message)

(Normal completion)

* The arrangement of the data items differs according to the command and sub command. See the sections explaining commands found in Section 3.3.2 and after for details.




(Abnormal completion)



(b) When writing data to the local station PLC CPU from the external device side

External device side → PLC CPU side (Command message)

Character area C

| (Data name) | STX | Frame ID No. | | Station No. | | Network No. | | PC No. | | Self-station No. | | Command | | | | Subcommand | | | | | ETX | Sum check code | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | H | L | H | L | H | L | H | L | H | L | H | – | – | L | H | – | – | L | | | H | L |
| (Example) | | F | 9 | 0 | 0 | 0 | 0 | F | F | 0 | 0 | | | | | | | | | | | 1 | 0 |
| | 02H | 46H | 39H | 30H | 30H | 30H | 30H | 46H | 46H | 30H | 30H | | | | | | | | | | 03H | 31H | 30H |

* Sum check is performed within this range. ((Ex.) When the result of the addition is 710H)

* The arrangement of the data items differs according to the command and sub command. See the sections explaining commands found in Section 3.3.2 and after for details.

PLC CPU side → External device side (Response message)

(Normal completion)

| STX | Frame ID No. | | Station No. | | Network No. | | PC No. | | Self-station No. | | Complete code (Normal completion symbol) | | | | ETX |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | H | L | H | L | H | L | H | L | H | L | | | | | |
| | F | 9 | 0 | 0 | 0 | 0 | F | F | 0 | 0 | Q | A | C | K | |
| 02H | 46H | 39H | 30H | 30H | 30H | 30H | 46H | 46H | 30H | 30H | 51H | 41H | 43H | 4BH | 03H |

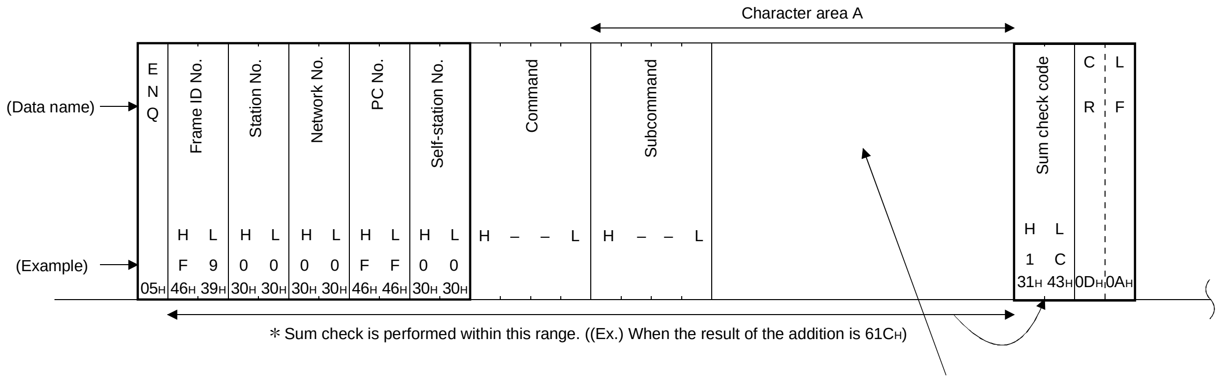
(Abnormal completion)

| STX | Frame ID No. | | Station No. | | Network No. | | PC No. | | Self-station No. | | Complete code (Abnormal completion symbol) | | | | Error code (For 7151H) | | | | ETX |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | H | L | H | L | H | L | H | L | H | L | | | | | H | – | – | L | |
| | F | 9 | 0 | 0 | 0 | 0 | F | F | 0 | 0 | Q | N | A | K | 7 | 1 | 5 | 1 | |
| 02H | 46H | 39H | 30H | 30H | 30H | 30H | 46H | 46H | 30H | 30H | 51H | 4EH | 41H | 4BH | 37H | 31H | 35H | 31H | 03H |

(5) Communication in format 4 (for communication in ASCII code)

(a) When reading data from the local station PLC CPU at the external device side

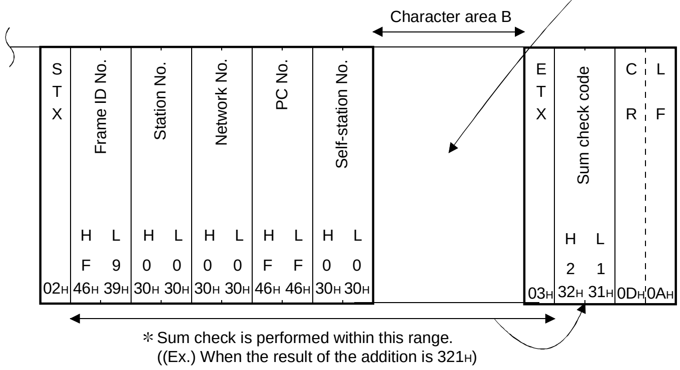
External device side → PLC CPU side (Command message)



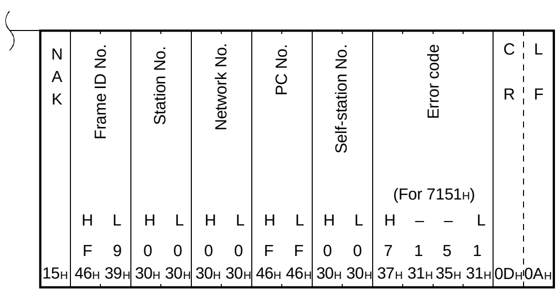

PLC CPU side → External device side (Response message)

* The arrangement of the data items differs according to the command and sub command. See the sections explaining commands found in Section 3.3.2 and after for details.

(Normal completion)




(Abnormal completion)



(b) When writing data to the local station PLC CPU from the external device side

External device side → PLC CPU side (Command message)

Character area C

| (Data name) | ENQ | Frame ID No. | | Station No. | | Network No. | | PC No. | | Self-station No. | | Command | | | | Subcommand | | | | | Sum check code | | CR | LF |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | H | L | H | L | H | L | H | L | H | L | H | – | – | L | H | – | – | L | | H | L | | |
| (Example) | | F | 9 | 0 | 0 | 0 | 0 | F | F | 0 | 0 | | | | | | | | | | 0 | D | | |
| | 05H | 46H | 39H | 30H | 30H | 30H | 30H | 46H | 46H | 30H | 30H | | | | | | | | | | 30H | 44H | 0DH | 0AH |

* Sum check is performed within this range. ((Ex.) When the result of the addition is 70DH)

* The arrangement of the data items differs according to the command and sub command. See the sections explaining commands found in Section 3.3.2 and after for details.

PLC CPU side → External device side (Response message)

(Normal completion)

| ACK | Frame ID No. | | Station No. | | Network No. | | PC No. | | Self-station No. | | CR | LF |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | H | L | H | L | H | L | H | L | H | L | | |
| | F | 9 | 0 | 0 | 0 | 0 | F | F | 0 | 0 | | |
| 06H | 46H | 39H | 30H | 30H | 30H | 30H | 46H | 46H | 30H | 30H | 0DH | 0AH |

(Abnormal completion)

| NAK | Frame ID No. | | Station No. | | Network No. | | PC No. | | Self-station No. | | Error code (For 7151H) | | | | CR | LF |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | H | L | H | L | H | L | H | L | H | L | H | – | – | L | | |
| | F | 9 | 0 | 0 | 0 | 0 | F | F | 0 | 0 | 7 | 1 | 5 | 1 | | |
| 15H | 46H | 39H | 30H | 30H | 30H | 30H | 46H | 46H | 30H | 30H | 37H | 31H | 35H | 31H | 0DH | 0AH |

### 3.1.5 Message formats and control procedures of QnA compatible 4C frames

This section explains the control procedures and message formats of each protocol format when data communication is performed using QnA compatible 4C frames for the Q series C24.

The ☐ section shown in the message explanation diagram of this section corresponds to the ✻ portion of the message explanation diagrams indicated in sections 3.3.2 and after of this chapter.

See Section 3.1.6 regarding the content of data in the ☐ section and data specification method.

For the following items, see the sections listed below:

- How to read the MC protocol control procedure : Section 3.1.1
- Basic formats for data communication : Section 3.1.4.

**POINT**

It shows the form 1 to 5 ((1) to (5)) for the QnA compatible 4C frame shown in this chapter.

(a) The sum check code is added only when sum check is enabled. When sum check is disabled, the sum check code is not added.

(b) When sum check is enabled, the sum check is performed only for the characters in the areas indicated by an asterisk (∗) in the diagrams (1) to (4). When transmitting (diagram (5)) by a binary code, the sum check is performed only for the data (exeluding "$10_H$" added by (e) below) in the area indicated by an asterisk (∗), and is sent after converting to the ASCII code.

(c) The contents of "Character area A," "Character area B," and "Character area C" in the diagrams (1) to (5) depend on the processing contents. See the description of each command for more information. The contents of each character area are the same for all four formats.

(d) The time from the command message reception to the response message transmission by a module can be set between 0 ms to 150 ms. (set by GX Configurator-SC)

(e) When "$10_H$" data is included in the areas marked with an asterisk (∗) in the diagram (5), "$10_H$" (DLE code) is added immediately in front of the data before the data is sent. (The data is sent as "$10_H$" → "$10_H$" + "$10_H$".)
However, the added "$10_H$" is not included in the "Number of data bytes" to be sent.

(f) Control procedure formats 3 and 5 cannot be used when data communication is performed in a system configured with an m:n connection between the external devices and the PLC CPUs.

## (1) Communication in format 1 (for communication in ASCII code)

### (a) When reading data from the local station PLC CPU at the external device side

External device side → PLC CPU side (Command message)

Character area A

| (Data name) | ENQ | Frame ID No. | Station No. | Network No. | PC No. | Request destination module I/O No | Request destination module station No. | Self-station No. | Command | Subcommand | | Sum check code |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | H L | H L | H L | H L | H – – L | H L | H L | H – – L | H – – L | | H L |
| (Example) | | F 8 | 0 0 | 0 0 | F F | 0 3 F F | 0 0 | 0 0 | | | | 6 A |
| | 05H | 46H 38H | 30H 30H | 30H 30H | 46H 46H | 30H 33H 46H 46H | 30H 30H | 30H 30H | | | | 36H 41H |

* Sum check is performed within this range. ((Ex.) When the result of the addition is 76AH)

PLC CPU side → External device side (Response message)

* The arrangement of the data items differs according to the command and sub command. See the sections explaining commands found in Section 3.3.2 and after for details.

(Normal completion)

Character area B

| STX | Frame ID No. | Station No. | Network No. | PC No. | Request destination module I/O No | Request destination module station No. | Self-station No. | | ETX | Sum check code |
|---|---|---|---|---|---|---|---|---|---|---|
| | H L | H L | H L | H L | H – – L | H L | H L | | | H L |
| | F 8 | 0 0 | 0 0 | F F | 0 3 F F | 0 0 | 0 0 | | | 6 F |
| 02H | 46H 38H | 30H 30H | 30H 30H | 46H 46H | 30H 33H 46H 46H | 30H 30H | 30H 30H | | 03H | 36H 46H |

* Sum check is performed within this range.
((Ex.) When the result of the addition is 46FH)

(Abnormal completion)

| NAK | Frame ID No. | Station No. | Network No. | PC No. | Request destination module I/O No | Request destination module station No. | Self-station No. | Error code (For 7151H) |
|---|---|---|---|---|---|---|---|---|
| | H L | H L | H L | H L | H – – L | H L | H L | H – – L |
| | F 8 | 0 0 | 0 0 | F F | 0 3 F F | 0 0 | 0 0 | 7 1 5 1 |
| 15H | 46H 38H | 30H 30H | 30H 30H | 46H 46H | 30H 33H 46H 46H | 30H 30H | 30H 30H | 37H 31H 35H 31H |

(b) When writing data to the local station PLC CPU from the external device side

External device side → PLC CPU side (Command message)

Character area C

| (Data name) | ENQ | Frame ID No. | Station No. | Network No. | PC No. | Request destination module I/O No | Request destination module station No. | Self-station No. | Command | Subcommand | | Sum check code |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | H L | H L | H L | H L | H – – L | H L | H L | H – – L | H – – L | | H L |
| (Example) | | F 8 | 0 0 | 0 0 | F F | 0 3 F F | 0 0 | 0 0 | | | | 5 B |
| | 05H | 46H 38H | 30H 30H | 30H 30H | 46H 46H | 30H 33H 46H 46H | 30H 30H | 30H 30H | | | | 35H 42H |

∗ Sum check is performed within this range. ((Ex.) When the result of the addition is 85BH)

PLC CPU side → External device side (Response message)

∗ The arrangement of the data items differs according to the command and sub command. See the sections explaining commands found in Section 3.3.2 and after for details.

(Normal completion)

| ACK | Frame ID No. | Station No. | Network No. | PC No. | Request destination module I/O No | Request destination module station No. | Self-station No. |
|---|---|---|---|---|---|---|---|
| | H L | H L | H L | H L | H – – L | H L | H L |
| | F 8 | 0 0 | 0 0 | F F | 0 3 F F | 0 0 | 0 0 |
| 06H | 46H 38H | 30H 30H | 30H 30H | 46H 46H | 30H 33H 46H 46H | 30H 30H | 30H 30H |

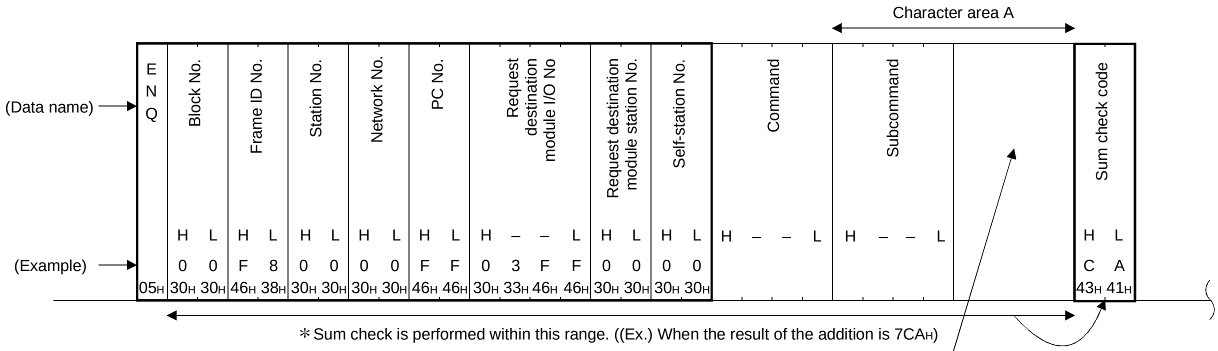
(Abnormal completion)

| NAK | Frame ID No. | Station No. | Network No. | PC No. | Request destination module I/O No | Request destination module station No. | Self-station No. | Error code |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | (For 7151H) |
| | H L | H L | H L | H L | H – – L | H L | H L | H – – L |
| | F 8 | 0 0 | 0 0 | F F | 0 3 F F | 0 0 | 0 0 | 7 1 5 1 |
| 15H | 46H 38H | 30H 30H | 30H 30H | 46H 46H | 30H 33H 46H 46H | 30H 30H | 30H 30H | 37H 31H 35H 31H |

## (2) Communication in format 2 (for communication in ASCII code)

### (a) When reading data from the local station PLC CPU at the external device side

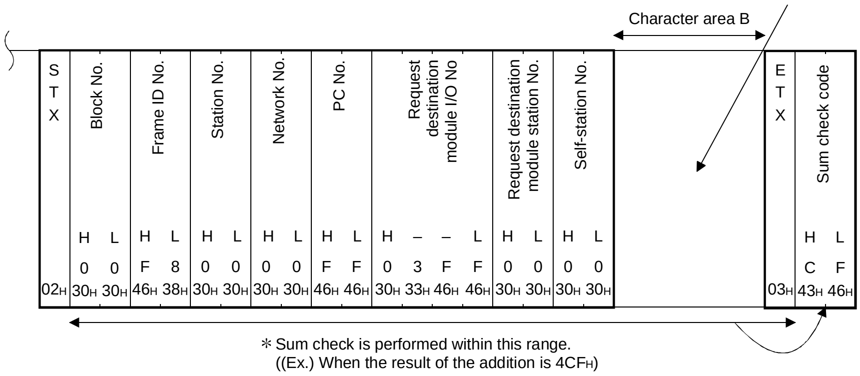
External device side → PLC CPU side (Command message)



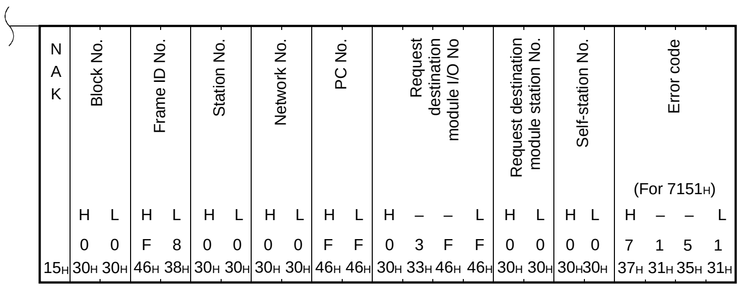

PLC CPU side → External device side (Response message)

* The arrangement of the data items differs according to the command and sub command. See the sections explaining commands found in Section 3.3.2 and after for details.

(Normal completion)




(Abnormal completion)



(b) When writing data to the local station PLC CPU from the external device side

External device side → PLC CPU side (Command message)

Character area C

| (Data name) | ENQ | Block No. | Frame ID No. | Station No. | Network No. | PC No. | Request destination module I/O No | Request destination module station No. | Self-station No. | Command | Subcommand | | Sum check code |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | H L | H L | H L | H L | H L | H – – L | H L | H L | H – – L | H – – L | | H L |
| (Example) | | 0 0 | F 8 | 0 0 | 0 0 | F F | 0 3 F F | 0 0 | 0 0 | | | | B B |
| | $05_H$ | $30_H$ $30_H$ | $46_H$ $38_H$ | $30_H$ $30_H$ | $30_H$ $30_H$ | $46_H$ $46_H$ | $30_H$ $33_H$ $46_H$ $46_H$ | $30_H$ $30_H$ | $30_H$ $30_H$ | | | | $42_H$ $42_H$ |

＊ Sum check is performed within this range. ((Ex.) When the result of the addition is $8BB_H$)

PLC CPU side → External device side (Response message)

＊ The arrangement of the data items differs according to the command and sub command. See the sections explaining commands found in Section 3.3.2 and after for details.

(Normal completion)

| ACK | Block No. | Frame ID No. | Station No. | Network No. | PC No. | Request destination module I/O No | Request destination module station No. | Self-station No. |
|---|---|---|---|---|---|---|---|---|
| | H L | H L | H L | H L | H L | H – – L | H L | H L |
| | 0 0 | F 8 | 0 0 | 0 0 | F F | 0 3 F F | 0 0 | 0 0 |
| $06_H$ | $30_H$ $30_H$ | $46_H$ $38_H$ | $30_H$ $30_H$ | $30_H$ $30_H$ | $46_H$ $46_H$ | $30_H$ $33_H$ $46_H$ $46_H$ | $30_H$ $30_H$ | $30_H$ $30_H$ |

(Abnormal completion)

| NAK | Block No. | Frame ID No. | Station No. | Network No. | PC No. | Request destination module I/O No | Request destination module station No. | Self-station No. | Error code |
|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | | (For $7151_H$) |
| | H L | H L | H L | H L | H L | H – – L | H L | H L | H – – L |
| | 0 0 | F 8 | 0 0 | 0 0 | F F | 0 3 F F | 0 0 | 0 0 | 7 1 5 1 |
| $15_H$ | $30_H$ $30_H$ | $46_H$ $38_H$ | $30_H$ $30_H$ | $30_H$ $30_H$ | $46_H$ $46_H$ | $30_H$ $33_H$ $46_H$ $46_H$ | $30_H$ $30_H$ | $30_H$ $30_H$ | $37_H$ $31_H$ $35_H$ $31_H$ |

### (3) Communication in format 3 (for communication in ASCII code)

(a) When reading data from the local station PLC CPU at the external device side

**External device side → PLC CPU side (Command message)**

Character area A

| (Data name) → | STX | Frame ID No. | Station No. | Network No. | PC No. | Request destination module I/O No | Request destination module station No. | Self-station No. | Command | Subcommand | | ETX | Sum check code |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | H L | H L | H L | H L | H – – L | H L | H L | H – – L | H – – L | | | H L |
| (Example) → | | F 8 | 0 0 | 0 0 | F F | 0 3 F F | 0 0 | 0 0 | | | | | 6 D |
| | 02H | 46H 38H | 30H 30H | 30H 30H | 46H 46H | 30H 33H 46H 46H | 30H 30H | 30H 30H | | | | 03H | 36H 44H |

＊ Sum check is performed within this range. ((Ex.) When the result of the addition is 76DH)

**PLC CPU side → External device side (Response message)**

＊ The arrangement of the data items differs according to the command and sub command. See the sections explaining commands found in Section 3.3.2 and after for details.

(Normal completion)

Character area B

| STX | Frame ID No. | Station No. | Network No. | PC No. | Request destination module I/O No | Request destination module station No. | Self-station No. | (Normal completion symbol) Complete code | | ETX | Sum check code |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | H L | H L | H L | H L | H – – L | H L | H L | | | | H L |
| | F 8 | 0 0 | 0 0 | F F | 0 3 F F | 0 0 | 0 0 | Q A C K | | | 9 2 |
| 02H | 46H 38H | 30H 30H | 30H 30H | 46H 46H | 30H 33H 46H 46H | 30H 30H | 30H 30H | 51H 41H 43H 4BH | | 03H | 39H 32H |

＊ Sum check is performed within this range.
((Ex.) When the result of the addition is 592H)

(Abnormal completion)

| STX | Frame ID No. | Station No. | Network No. | PC No. | Request destination module I/O No | Request destination module station No. | Self-station No. | (Abnormal completion symbol) Complete code | Error code | ETX |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | | (For 7151H) | |
| | H L | H L | H L | H L | H – – L | H L | H L | | H – – L | |
| | F 8 | 0 0 | 0 0 | F F | 0 3 F F | 0 0 | 0 0 | Q N A K | 7 1 5 1 | |
| 02H | 46H 38H | 30H 30H | 30H 30H | 46H 46H | 30H 33H 46H 46H | 30H 30H | 30H 30H | 51H 4EH 41H 4BH | 37H 31H 35H 31H | 03H |

(b) When writing data to the local station PLC CPU from the external device side

External device side → PLC CPU side (Command message)

Character area C (Command, Subcommand and the data area after them)

| (Data name) | STX | Frame ID No. | | Station No. | | Network No. | | PC No. | | Request destination module I/O No | | | | Request destination module station No. | | Self-station No. | | Command | | | | Subcommand | | | | * | ETX | Sum check code | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | H | L | H | L | H | L | H | L | H | – | – | L | H | L | H | L | H | – | – | L | H | – | – | L | | | H | L |
| (Example) | | F | 8 | 0 | 0 | 0 | 0 | F | F | 0 | 3 | F | F | 0 | 0 | 0 | 0 | | | | | | | | | | | 5 | E |
| | 02H | 46H | 38H | 30H | 30H | 30H | 30H | 46H | 46H | 30H | 33H | 46H | 46H | 30H | 30H | 30H | 30H | | | | | | | | | | 03H | 35H | 45H |

* Sum check is performed within this range. ((Ex.) When the result of the addition is 85EH)

* The arrangement of the data items differs according to the command and sub command. See the sections explaining commands found in Section 3.3.2 and after for details.

PLC CPU side → External device side (Response message)

(Normal completion)

| STX | Frame ID No. | | Station No. | | Network No. | | PC No. | | Request destination module I/O No | | | | Request destination module station No. | | Self-station No. | | Complete code (Normal completion symbol) | | | | ETX |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | H | L | H | L | H | L | H | L | H | – | – | L | H | L | H | L | | | | | |
| | F | 8 | 0 | 0 | 0 | 0 | F | F | 0 | 3 | F | F | 0 | 0 | 0 | 0 | Q | A | C | K | |
| 02H | 46H | 38H | 30H | 30H | 30H | 30H | 46H | 46H | 30H | 33H | 46H | 46H | 30H | 30H | 30H | 30H | 51H | 41H | 43H | 4BH | 03H |

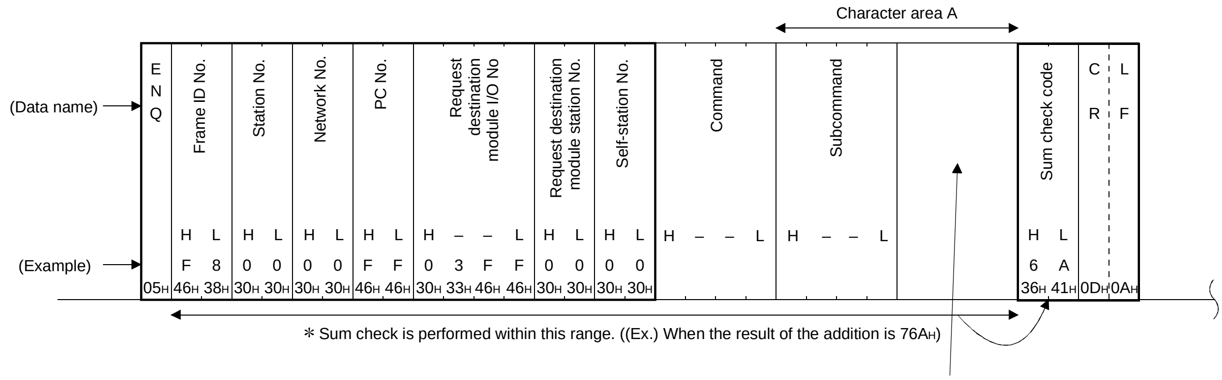
(Abnormal completion)

| STX | Frame ID No. | | Station No. | | Network No. | | PC No. | | Request destination module I/O No | | | | Request destination module station No. | | Self-station No. | | Complete code (Abnormal completion symbol) | | | | Error code (For 7151H) | | | | ETX |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | H | L | H | L | H | L | H | L | H | – | – | L | H | L | H | L | | | | | H | – | – | L | |
| | F | 8 | 0 | 0 | 0 | 0 | F | F | 0 | 3 | F | F | 0 | 0 | 0 | 0 | Q | N | A | K | 7 | 1 | 5 | 1 | |
| 02H | 46H | 38H | 30H | 30H | 30H | 30H | 46H | 46H | 30H | 33H | 46H | 46H | 30H | 30H | 30H | 30H | 51H | 4EH | 41H | 4BH | 37H | 31H | 35H | 31H | 03H |

## (4) Communication in format 4 (for communication in ASCII code)

### (a) When reading data from the local station PLC CPU at the external device side

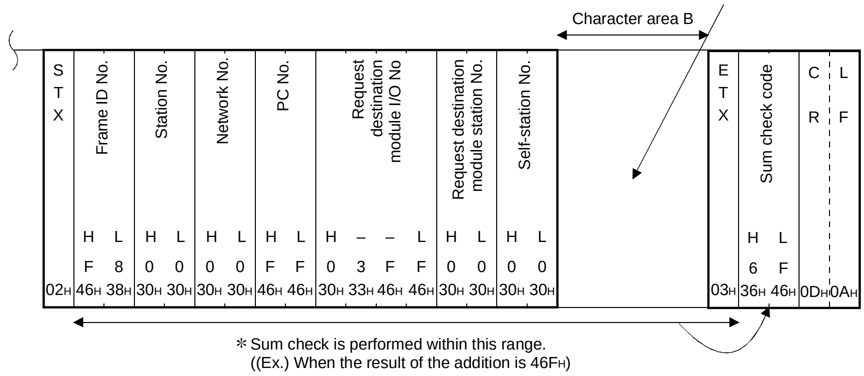
External device side → PLC CPU side (Command message)



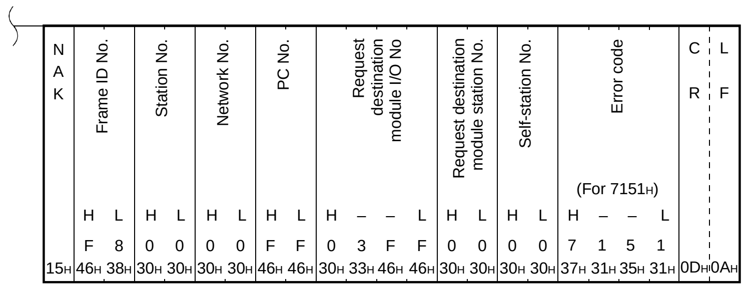

PLC CPU side → External device side (Response message)

* The arrangement of the data items differs according to the command and sub command. See the sections explaining commands found in Section 3.3.2 and after for details.

(Normal completion)




(Abnormal completion)



(b) When writing data to the local station PLC CPU from the external device side

External device side → PLC CPU side (Command message)

Character area C (Command, Subcommand and following area)

| (Data name) | ENQ | Frame ID No. | Station No. | Network No. | PC No. | Request destination module I/O No | Request destination module station No. | Self-station No. | Command | Subcommand | ✻ | Sum check code | CR | LF |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | H L | H L | H L | H L | H – – L | H L | H L | H – – L | H – – L | | H L | | |
| (Example) | | F 8 | 0 0 | 0 0 | F F | 0 3 F F | 0 0 | 0 0 | | | | 5 B | | |
| | 05H | 46H 38H | 30H 30H | 30H 30H | 46H 46H | 30H 33H 46H 46H | 30H 30H | 30H 30H | | | | 35H 42H | 0DH | 0AH |

✻ Sum check is performed within this range. ((Ex.) When the result of the addition is 85BH)

PLC CPU side → External device side (Response message)

✻ The arrangement of the data items differs according to the command and sub command. See the sections explaining commands found in Section 3.3.2 and after for details.

(Normal completion)

| ACK | Frame ID No. | Station No. | Network No. | PC No. | Request destination module I/O No | Request destination module station No. | Self-station No. | CR | LF |
|---|---|---|---|---|---|---|---|---|---|
| | H L | H L | H L | H L | H – – L | H L | H L | | |
| | F 8 | 0 0 | 0 0 | F F | 0 3 F F | 0 0 | 0 0 | | |
| 06H | 46H 38H | 30H 30H | 30H 30H | 46H 46H | 30H 33H 46H 46H | 30H 30H | 30H 30H | 0DH | 0AH |

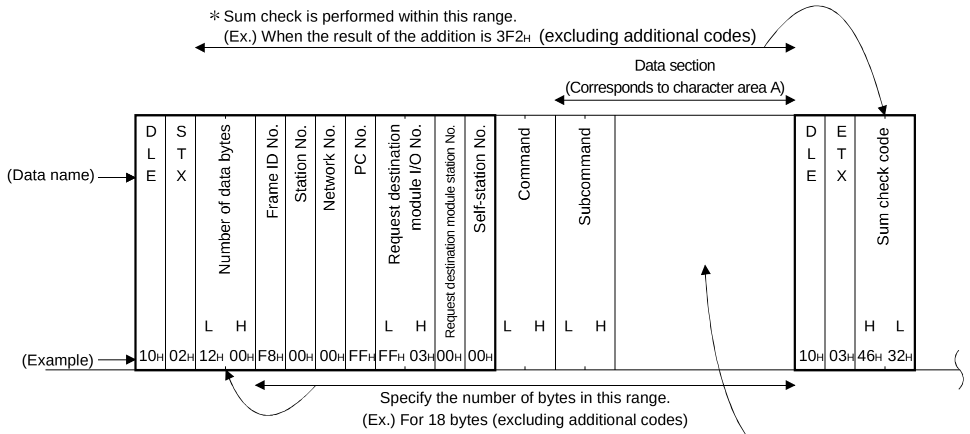
(Abnormal completion)

| NAK | Frame ID No. | Station No. | Network No. | PC No. | Request destination module I/O No | Request destination module station No. | Self-station No. | Error code | CR | LF |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | (For 7151H) | | |
| | H L | H L | H L | H L | H – – L | H L | H L | H – – L | | |
| | F 8 | 0 0 | 0 0 | F F | 0 3 F F | 0 0 | 0 0 | 7 1 5 1 | | |
| 15H | 46H 38H | 30H 30H | 30H 30H | 46H 46H | 30H 33H 46H 46H | 30H 30H | 30H 30H | 37H 31H 35H 31H | 0DH | 0AH |

## (5) Communication in format 5 (for communication in binary code)

### (a) When reading data from the local station PLC CPU at the external device side

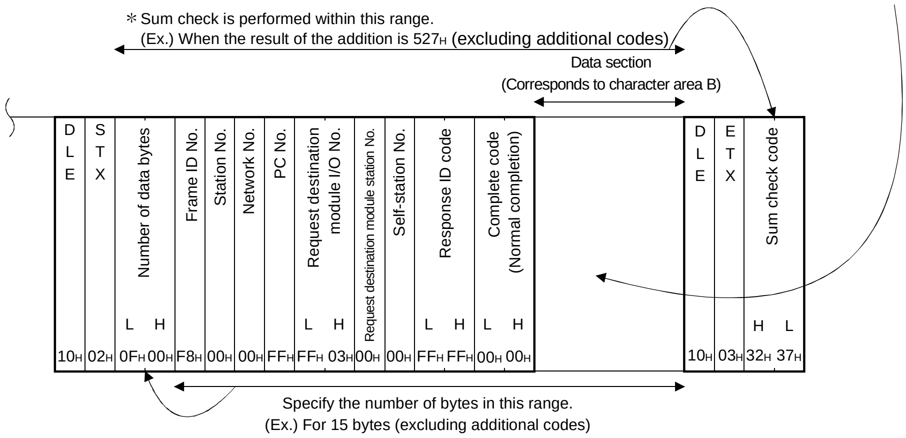
External device side → PLC CPU side (Command message)



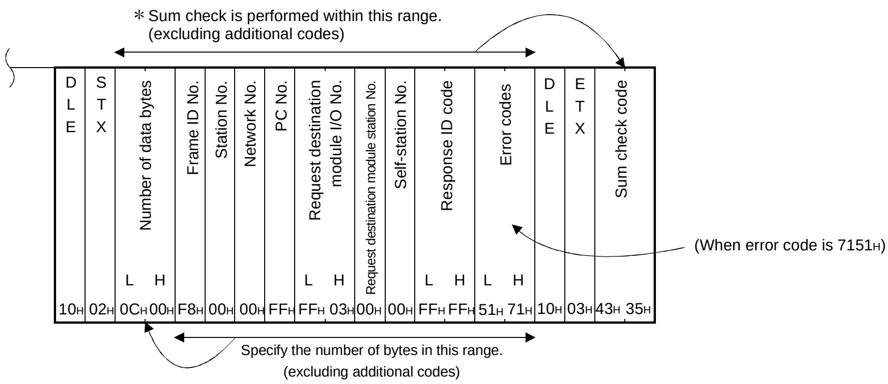

PLC CPU side → External device side (Response message)

* The arrangement of the data items differs according to the command and sub command. See the sections explaining commands found in Section 3.3.2 and after for details.

(Normal completion)

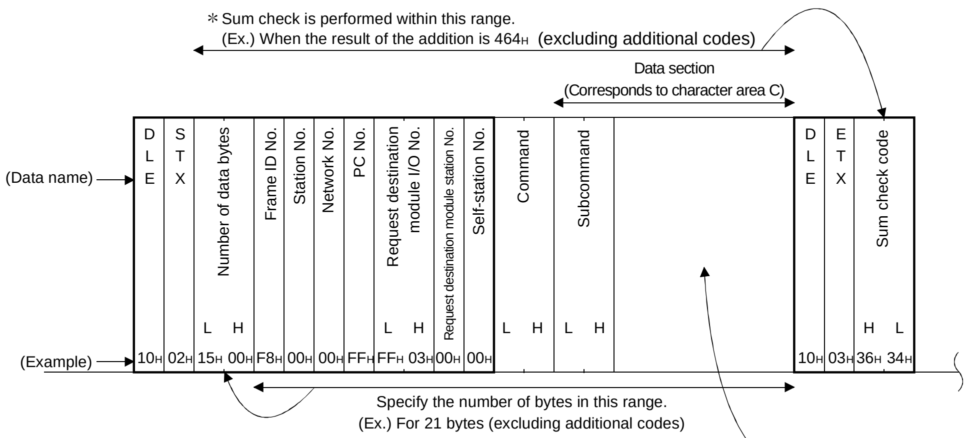



(Abnormal completion)



(b) When writing data to the local station PLC CPU from the external device side

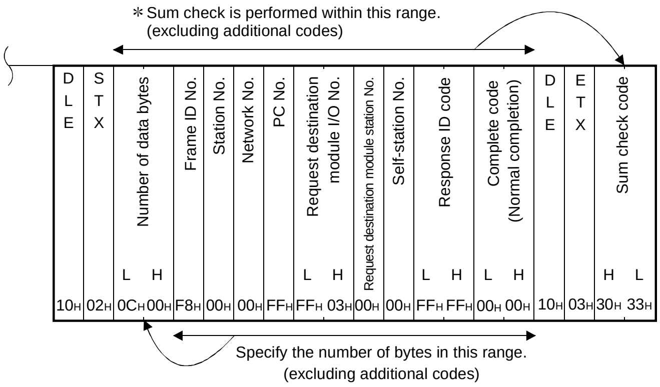
External device side → PLC CPU side (Command message)



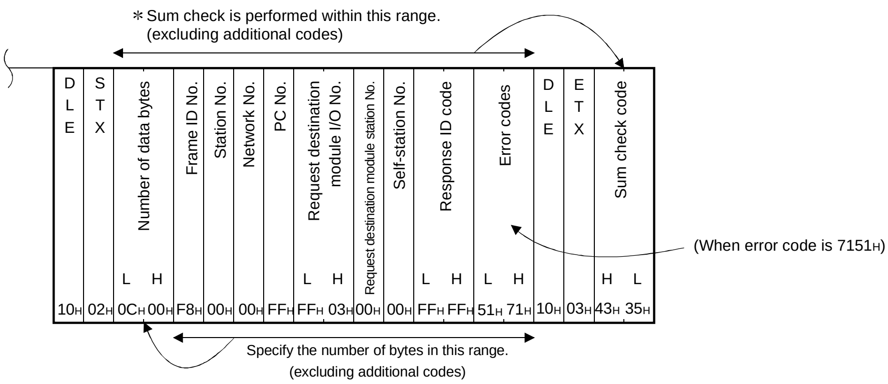

* The arrangement of the data items differs according to the command and sub command. See the sections explaining commands found in Section 3.3.2 and after for details.

PLC CPU side → External device side (Response message)

(Normal completion)




(Abnormal completion)



### 3.1.6 Data designation items for QnA compatible 3C/4C frames

This section explains common data items in each message and how to specify them when data communication is performed using QnA compatible 3C/4C frames for the Q series C24.

#### (1) Control codes

The following shows the codes and contents of the data (message head data, etc.) that have special meanings for the Q series C24 transmission control in each control procedure.
The control codes used by each ASCII mode frame are indicated by an ○ mark in the ASCII code column in the table.
The control codes used by the binary mode QnA compatible 4C frames are indicated by an ○ mark in the binary code column in the table.
The control codes are used in the same way in each message.

| Symbol name | Code (hexadecimal) | Description | ASCII code | Binary code | Symbol name | Code (hexadecimal) | Description | ASCII code | Binary code |
|---|---|---|---|---|---|---|---|---|---|
| NUL | $00_H$ | Null | ○ | | CL | $0C_H$ | Clear | ○ | |
| STX | $02_H$ | Start of Text | ○ | ○ | CR | $0D_H$ | Carriage Return | ○ | |
| ETX | $03_H$ | End of Text | ○ | ○ | DLE | $10_H$ | Data Link Escape | | ○ |
| EOT | $04_H$ | End of Transmission | ○ | | NAK | $15_H$ | Negative Acknowledge | ○ | |
| ENQ | $05_H$ | Enquiry | ○ | | QnA compatible frame ID number | $F6_H$ | (For system) | | ○ |
| ACK | $06_H$ | Acknowledge | ○ | | | $F8_H$ | (QnA compatible 4C frame ID code) | ○ | ○ |
| LF | $0A_H$ | Line Feed | ○ | | | $F9_H$ | (QnA compatible 3C frame ID code) | ○ | |
| – | – | – | | | | $FA_H$ | (For system) | ○ | ○ |

**POINT**

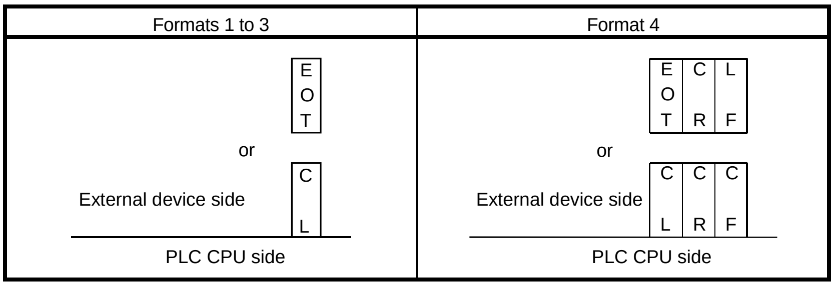
When the fields "Number of data bytes" to "Data area" in a message include "$10_H$", the "$10_H$" DLE code (expressed as additional code in the document) is added immediately before the data, after which the data communication is performed (as "$10_H$" → "$10_H$" + "$10_H$."). This is valid when data communication is performed in binary code using QnA 4C frame format 5
The Q series C24 adds the response message.
An example of the message structure is shown in Section 3.1.7 (2).

(a) Null codes ($00_H$) are ignored in all messages. That means that even if there are Null codes in a message, they are not processed.

(b) EOT and CL are codes for initializing the transmission sequence for data communications in ASCII code using the MC protocol and for placing the Q series C24 in wait status to receive commands from an external device.
According to the format used, EOT or CL is sent to the Q series C24 when giving the following commands from the external device side.

1) Cancel a read/write request issued by the immediately preceding command. (When a write request was issued, and data was already written to the PLC CPU, the write request cannot be canceled.)
2) Before commands are sent, place the Q series C24 in the command receive wait status.
3) If data communications cannot be carried out normally, place the Q series C24 in the same status as when it was started up.

The following shows the message structure when sending EOT or CL.




✻ When EOT or CL is sent, only the data shown in the table to the left is sent.
The station number, PLC number, etc. do not have to be sent.

When the Q series C24 receives EOT or CL, it proceeds as follows:

- Any read/write processing performed for the PLC CPU upon request from an external device is terminated.
  In this case, the Q series C24 does not send a response message for the command received last.
- The MC protocol transmission sequence of the interface that received the EOT or CL is initialized and the Q series C24 enters the wait status to wait for further commands from an external device.
- There is no response message for receiving EOT and CL. (No message is sent to the external device.)
- When EOT or CL is received while the on-demand function (the function that sends data from the PLC CPU to an external device, see Section 3.11) is being executed, the Q series C24 terminates the on-demand function data transmission to the external device.

**POINT**

During data communication in binary code using the MC protocol, the transmission sequence can be initialized through the transmission sequence initialize command (1615).

## (2) Block number (For QnA compatible 3C/4C frame format 2)

The block number is a (context-sensitive) arbitrary number that is given meaning by the relevant message on the external device side. It is used as a data management number, etc.

The block numbers range is $00_H$ to $FF_H$. Block numbers are converted to a 2-digit (hexadecimal) ASCII code and sequentially transmitted beginning from the most significant digit.

The Q series C24 only checks if the block number is designated within the correct range.

It does not check whether or not the block numbers sent by command messages are sequential.

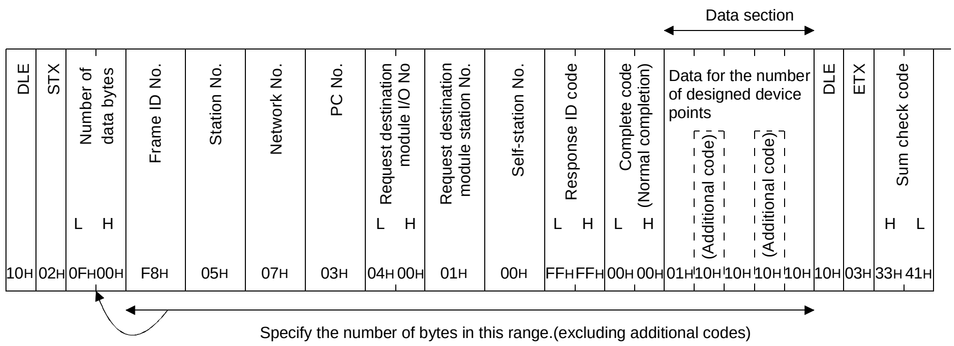
## (3) Number of data bytes (for QnA compatible 4C frame format 5)

The number of data bytes is used to inform the external device of the total number of bytes of data, excluding additional codes, between the frame ID number and data section or the frame ID number and complete code (error code). (Since the PLC CPU does not check the number of data bytes in command messages received from an external device, it can also be designated as dummy data $00_H$, $00_H$.)

Number of data bytes is transmitted as a 2-byte value in ascending order, low byte (L: bit 0 to 7) to high byte (H: bit 8 to 15).

(Example)

A response message after data was read from the PLC CPU




## (4) Frame ID number

The frame ID number is used to identify if the message to be communicated is a QnA compatible 3C frame or a QnA compatible 4C frame.

The following shows the frame ID numbers that are designated during data communications.

| | Frame | Frame ID Number. | Note |
|---|---|---|---|
| Communication in ASCII code | QnA compatible 3C frame | "F9" | Two characters, "F" and "9," are sent in that order. |
| | QnA compatible 4C frame | "F8" | Two characters, "F" and "8," are sent in that order. |
| Communication in binary code | QnA compatible 4C frame | $F8_H$ | One byte of $F8_H$ data is sent. |

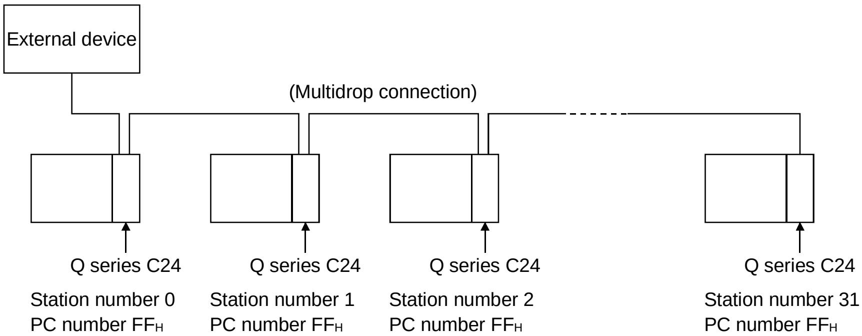
## (5) Station number to self-station number

The values in the station number to local station number fields designate the PLC station to be accessed by an external device. The four sets of data identify the route until the access destination.

An example of designating the network number to request destination module station number while accessing other stations is shown in Section 2.6.3.

- Station number
  The station number designates to which Q series C24 an external device will access or through which Q series C24 access is made to other station on MELSECNET/H, MELSECNET/10, using the station number of the Q series C24 that has been set with GX Developer.

[Example]




**REMARK**

It is not necessary to set the station numbers in ascending order as in the diagram above. Even when station number 31 is set first, the communication is performed normally.

- Network number and PC number
  These values identify the number of the network system that is passed through last and the PC number (PLC station number) on the relevant system according to the setting number of the network module when an external device accesses another station's PLC.
- Request destination module I/O number and request destination module station number
  Designate the connection source PLC CPU and access destination PLC CPU of the multidrop connection when an external device accesses another station's PLC through a network system and multidrop connection PLC CPU.
  The request destination module I/O number designates the multidrop connection source Q series C24 by the head number of the input/output signals.
  The request destination module station number designates the Q series C24 to be accessed according to the station number of the Q series C24 in the settings.
- Self-station number
  Designates the external device's station number (local station number) by the number "0" (preset value).

a) Station number
  1) Data communication in ASCII code
     Must be in the $00_H$ to $1F_H$ (0 to 31) range. The data is converted to 2-digit (hexadecimal) ASCII code and sequentially transmitted beginning from the most significant digit.
  2) Data communication in binary mode
     A 1-byte value from "$00_H$ " to "$1F_H$" (0 to 31) is used for transmission.

**POINT**

(1) When the global function described in Section 3.10 is used, the designation of the station number in the protocol is "FF"/$FF_H$.
When a number from 0 to 31 ($00_H$ to $1F_H$) is designated, X1A/X1B of the designated station only is turned ON; those of other stations are not turned on.

(2) When the external device and PLC CPU system configuration is m:n and data communication is performed using QnA compatible 3C frames or QnA compatible 4C frames, the external device station number must be a number in a range between $00_H$ and $1F_H$ (0 to 31) that does not overlap with the Q series C24 side station numbers.
For the description of the external device station numbers, see Chapter 14 of the User's Manual (Application).

(b) Network number

1) Data communication in ASCII code
A value within the range shown below is converted to 2-digit (hexadecimal) ASCII code and sequentially transmitted beginning from the most significant digit.

2) Data communication in binary code
The 1-byte value shown below is used for transmission.

3) The network numbers designated during data communication are shown in (c) below.

**POINT**

(1) The PLC CPU cannot be accessed through a MELSECNET/H, MELSECNET/10 with a network number from 240 to 255.
(2) When designating the network number $FE_H$ in order to access another station through a PLC CPU station incorporated in a network system, set the following parameter in the PLC CPU of the PLC CPU loaded station using GX Developer.
"Valid module during other station access" setting:
Sets the module in the MNET/10H Ethernet number setting through which an external device accesses another station
(3) Designate the network number using the numbers shown in the table. Sometimes a response will not be returned when a number other than the ones shown in the table is designated.

(c) PC number

1) Data communication in ASCII code
A value within the range shown below is converted to a 2-digit (hexadecimal) ASCII code and sequentially transmitted beginning from the most significant digit.

2) Data communication in binary code
The 1-byte value shown below is used for transmission.

3) The following table shows the PC numbers that can be designated during data communication.

| | External device access station | Network number designated by external device | |
|---|---|---|---|
| | | Network number | PC number |
| 1 | External device connection station (local station) | $00_H$ | $FF_H$ ＊1 |
| 2 | Multidrop connection station from external device connection station | $00_H$ | $FF_H$ ＊1 |
| 3 | Control station on MELSECNET/H, MELSECNET/10 (When the Q series C24 is installed to the normal station.) | $01F$ to $EF_H$ (1 to 239) (Access station network number.) | $7D_H$ : Designated Control station/Master station<br>$7E_H$ : Current Control station/Master station |
| 4 | Remote master station on MELSECNET/H. (When the Q series C24 is installed to remote I/O station) | | $01_H$ to $40_H$ (1 to 64) (Access station number) (Excluding 1, 2 and 3 above) |
| 5 | Station on MELSECNET/H, MELSECNET/10 | | |
| 6 | Pass through the network module set in "Valid module during other station access" | $FE_H$ (254) ＊2 | $01_H$ to $40_H$ (1 to 64) |
| 7 | Multidrop connection station via MELSECNET/H, MELSECNET/10 connection station | $01_H$ to $EF_H$ (1 to 239) (Network Number. passed through last) | $01_H$ to $40_H$ (1 to 64) (Station number passed through last) |

＊1 PC number $FF_H$ can be designated only when (b) Network number is $00_H$.
＊2 The network number $FE_H$ will be disregarded when accessing another station via a Q series C24/E71 loaded on the MELSECNET/H remote I/O station. If the network number $FE_H$ is specified, the other station specified by the PC number of the MELSECNET/H remote I/O station will be accessed.

**POINT**

When the on-demand function is used, the PLC CPU sends "FE"/$FE_H$ as the PC number on the protocol.

(d) Request destination module I/O number

1) Data communication in ASCII code

The three higher digits of the four-digit input/output signals of the objective PLC CPU, or 03FFH, are converted to a 4-digit ASCII (hexadecimal) code and sequentially transmitted from the most significant digit.

(Example)

When the PLC CPU input/output signals are 0080H to 009FH; the request destination module I/O number "0008" is sequentially transmitted from the first "0".

2) Data communication in binary mode

The higher three digits of the four-digit input/output signals of the objective PLC CPU, or two bytes of 03FFH data, are transmitted in the order, low byte (L: bits 0 to 7) to high byte (H: bits 8 to 15).

(Example)

When the PLC CPU input/output signals are 0080H to 009FH; the request destination module I/O number 0008H is transmitted in the order 08H, then 00H.

3) The following table shows the request destination module I/O numbers that are designated during data communication.

| | External device access station | Request destination module I/O number |
|---|---|---|
| 1 | Control system CPU | 03D0H |
| 2 | Standby system CPU | 03D1H |
| 3 | System A CPU | 03D2H |
| 4 | System B CPU | 03D3H |
| 5 | CPU No.1 | 03E0H |
| 6 | CPU No.2 | 03E1H |
| 7 | CPU No.3 | 03E2H |
| 8 | CPU No.4 | 03E3H |
| 9 | Local station CPU, control CPU, CPU in the system directly connected to the local station | 03FFH |
| 10 | Control CPU of Q series C24 on a multidrop connected station (Specify any of the above 1 to 9 when accessing the PLC that is located on the last relay station connected to MELSECNET/H or MELSECNET/10.) | 0000H to 01FFH |

(e) Request destination module station number

1) Data communication in ASCII code

A value within the range shown below is converted to a 2-digit (hexadecimal) ASCII code and sequentially transmitted beginning from the most significant digit.

2) Data communication in binary code

The 1-byte value shown below is transmitted.

3) The following table shows the request destination module station numbers that are designated during data communication.

| | External device access station | Request destination module station number |
|---|---|---|
| 1 | Station other than the station below. | 00H (0) |
| 2 | Station on multidrop connection (Conforms to the above when accessing the Q/QnACPU of the MELSECNET/H, MELSECNET/10 connection station passed through last.) | 00H to 1FH (0 to 31) |

(f) Local station number
1) During communication in ASCII code, "00" is used for transmission.
2) During communication in binary code, the 1-byte value "00H" (0) is used for transmission.

**POINT**

When the external device and PLC CPU system configuration is m:n and data communication is performed by QnA compatible 3C frames or QnA compatible 4C frames, the external device station number must be a number within the range 00H to 1FH (0 to 31) that does not duplicate the Q series C24 side station number.
For the description of the external device station number, see Chapter 14 of the User's Manual (Application).

## (6) Command

A command designates the purpose of the access when an external device accesses the PLC, such as read and write.

1) Data communication in ASCII code
The commands are converted to 4-digit (hexadecimal) ASCII code and sequentially transmitted beginning from the most significant digit.
(Example)
Bit units batch read command
The command "0401" is sequentially transmitted from the first "0."

2) Data communication in binary code
The commands are used as 2-byte values (hexadecimal) and are transmitted in the order, low byte (L: bits 0 to 7) to high byte (H: bits 8 to 15).
(Example)
Bit units batch read command
The command 0401H is sent in the order of 01H and 04H.

## (7) Character area ⌜⌟ (data section)

The following describes the application of each character area.

| | | |
|---|---|---|
| Character area A | : | Data that will instruct the PLC CPU to execute a read request as designated by a command. |
| Character area B | : | Data returned to the external device in response to a request designated by a command. |
| Character area C | : | Data to instruct the PLC CPU to execute a write request as designated by a command. |

The contents of the character areas (data section) depend on the command sent from the external device.

1) During data communication in ASCII code, the character areas are converted to ASCII code and transmitted.
2) During data communication in binary code, the character areas are transmitted in binary code.
3) See Section 3.1.7 about the approach to the transmission data in the character area.

## (8) Sum check code

The sum check code represents the low byte (8 bits) of the result of binary adding up (summing) the data within the sum check range (see Sections 3.1.4 to 3.1.5) in the message.
When the "Sum Check Enable/Disable Setting" is set to "Enable" in the transmission specifications setting in GX Developer, the sum check code must be added.
When sum check is enabled, the PLC CPU generates the sum check code and adds it to the transmission message. The PLC CPU also checks the sum check code in the reception message.
When sum check is disabled, the PLC CPU does not add a sum check code to the transmission message. The PLC CPU also processes reception messages as if they did not contain a sum check code.

1) During communication in either ASCII or binary code, the sum check code is converted to a 2-digit (hexadecimal) ASCII code and sequentially transmitted beginning from the most significant digit.
2) During data communication in binary code, the sum check is calculated after removing the sum check objective range additional code (see Section 3.1.6 (1), Point).
3) The examples below show the contents of the sum check code.

(Example 1)

When data is read using QnA compatible 3C frame format 1.

Character area A (Subcommand through Number of device points)

| (Data name) External device side | ENQ | Frame ID No. | Station No. | Network No. | PC No. | Self-station No. | Command | Subcommand | Device code | Head device | Number of device points | Sum check code |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | H L | H L | H L | H L | H L | H – – L | H – – L | | H – – – – L | H – – L | H L |
| (Example) | | F 9 | 0 0 | 0 0 | F F | 0 0 | 0 4 0 1 | 0 0 0 1 | X * | 0 0 0 0 4 0 | 0 0 0 5 | 1 C |
| | 05H | 46H 39H | 30H 30H | 30H 30H | 46H 46H | 30H 30H | 30H 34H 30H 31H | 30H 30H 30H 31H | 58H 2AH | 30H 30H 30H 30H 34H 30H | 30H 30H 30H 35H | 31H 43H |

Sum check is performed within this range.

46H+39H+30H+30H+30H+30H+46H+46H+30H+30H+30H+34H+30H+31H+30H+30H+
30H+31H+58H+2AH+30H+30H+30H+30H+34H+30H+30H+30H+30H+35H=61CH

The sum check becomes “1C”(ASCII code 31H, 43H).

(Example 2)

When data is read using QnA compatible 4C frame format 5

Data section (Corresponds to character area A)

| (Data name) External device side | DLE | STX | Number of data bytes | Frame ID No. | Station No. | Network No. | PC No. | Request destination module I/O No. | Request destination module station No. | Self-station No. | Command | Subcommand | Head device | Device code | Number of device points | DLE | ETX | Sum check code |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | L H | L H | | | | L H | | | L H | L H | L – H | | L H | | | H L |
| (Example) | 10H | 02H | 12H 00H | F8H | 05H | 07H | 03H | 04H 00H | 01H | 00H | 01H 04H | 01H 00H | 40H 00H 00H | 9CH | 05H 00H | 10H | 03H | 30H 35H |

Sum check is performed within this range.(excluding additional codes)

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

The sum check becomes “05”(ASCII code 30H, 35H).

## (9) Response ID code (QnA compatible 4C frame format 5)

The response ID code notifies to the external device that the message is a response to the request it sent earlier. The 2-byte value, "FFFFH " is used for transmission.

## (10) Complete code (for normal completion of QnA compatible 4C frame format 5)

The complete code notifies to the external device that the PLC CPU processed the request it sent normally. The 2-byte value, "0000H " is used for transmission.

## (11) Error code

An error code tells the external device that the PLC CPU processing of the request it sent ended abnormally.

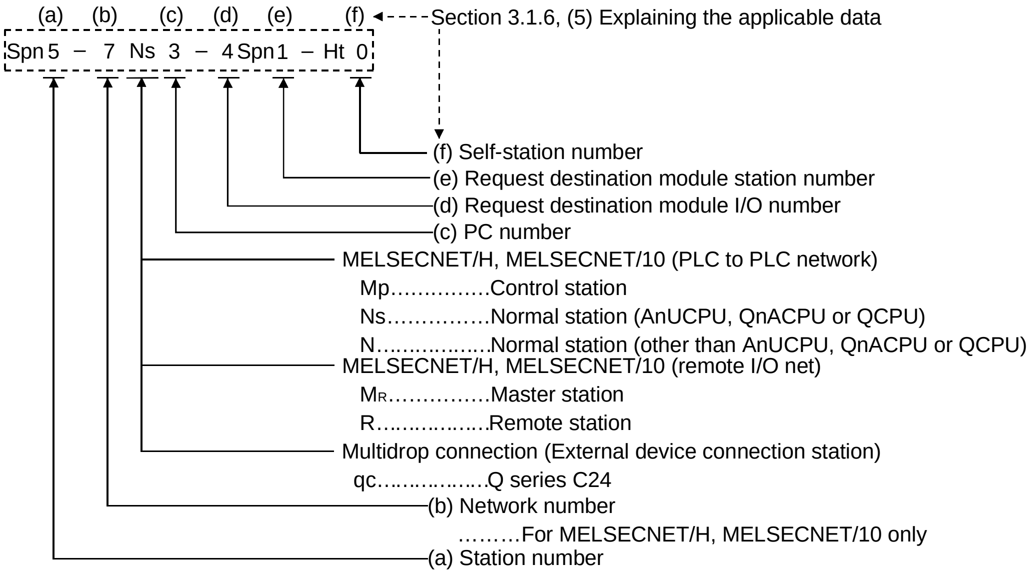
1) Data communication in ASCII code
   The error code is converted to a 4-digit (hexadecimal) ASCII code and sequentially transmitted beginning from the most significant digit.
   (Example)
   Error code 7151H
   The error code sent to the external device is "7151," and is sequentially transmitted beginning from "7."
2) Data communications in binary code
   The error code is immediately used as a 2-byte value and is transmitted in the order, low byte (L: bits 0 to 7) to high byte (H: bits 8 to 15).
   (Example)
   Error code 7151H
   The error code is sent to the external device in the order of first 51H, then 71H.
3) If two or more errors occurred at the same time, the PLC CPU sends the error code detected first.
4) See Chapter 10 of the User's Manual (Basic) for the detailed description of the error codes.

REMARK

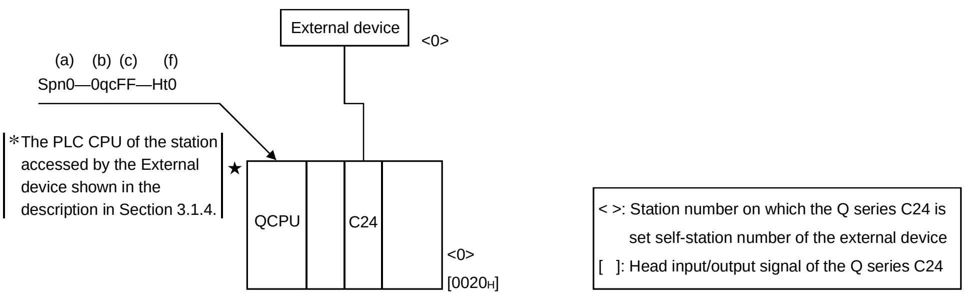
(1) The "message wait" time cannot be designated in the control procedure of the QnA compatible 3C, QnA compatible 4C, and QnA compatible 2C frames shown in Chapter 4. When communicating data using these frames, the message wait time should be designated in the "Message Wait Time Designation" in GX Configurator-SC. (Set on the "MC protocol system setting" screen.)
A message wait time which is as long as the external device hardware gate off-time or longer should be designated when communicating data with an external devices connected to the RS-422/485 interface of the PLC CPU, including during A compatible 1C frame data communication,

(2) The following shows some designation examples of "Station number" to "Self-station number." These values are the data that is designated in each control procedure during data communication using the QnA compatible 3C and 4C frames
(How to read the figures and tables in 3) and 4) below)

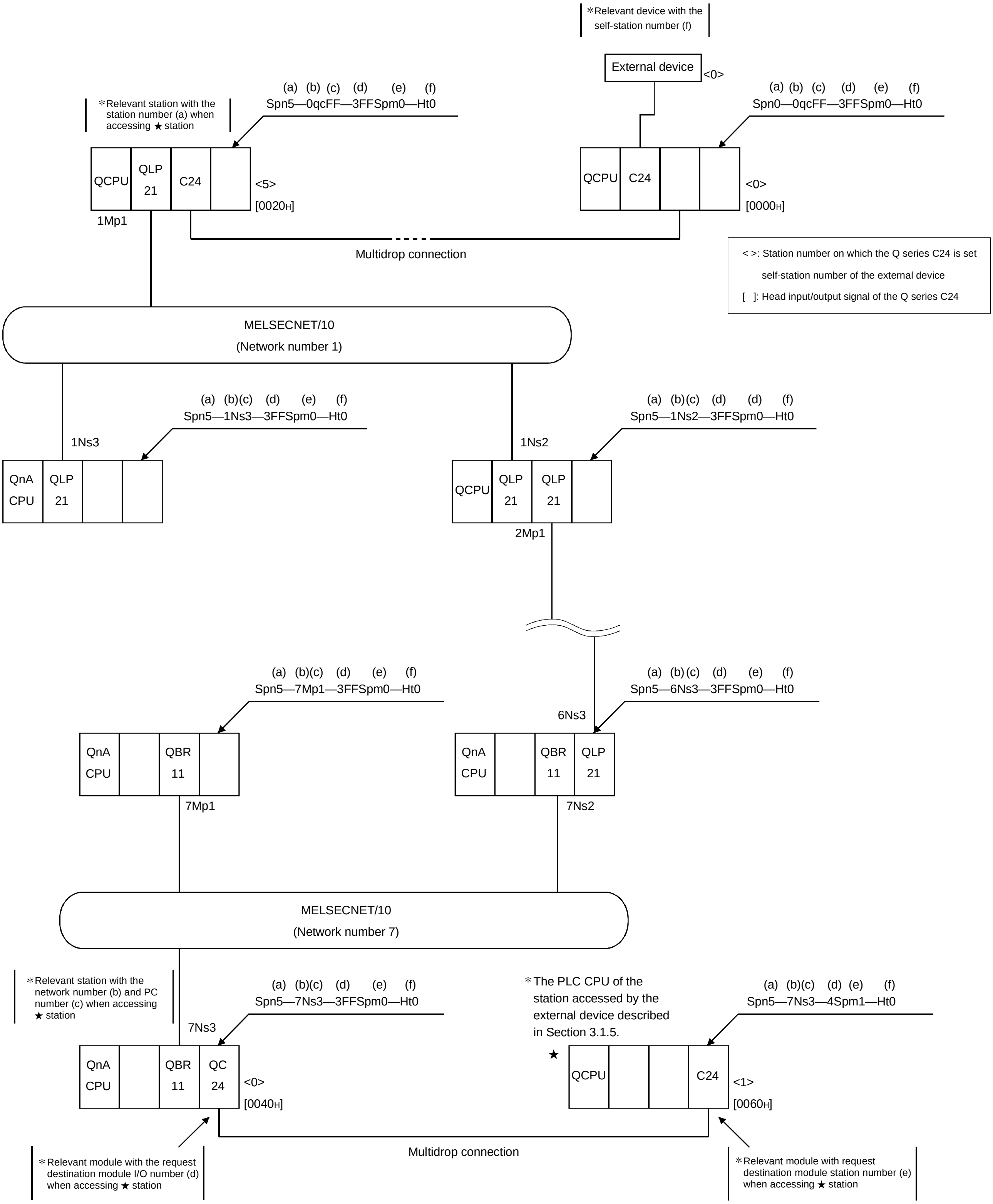
1) The numerical values in the symbols (< >, [ ] ) shown at the right side of the station in the figures provide the following information.
< >: Station number of the Q series C24 loaded to the corresponding station
[ ]: Head input/output signal of the Q series C24 loaded to the corresponding station
2) Meaning of the symbols indicated by the arrows to each station in the diagrams
The alphanumeric characters before or after the symbol is the contents (value) of the relevant data in the message when an external device accesses the station.




3) The external device connection station and access destination are explained in the section that describes the data communication formats in the QnA compatible 3C frames
The following diagram shows an example of an external device connection station sending the transmission data shown in the description of each format in Section 3.1.4 and the PLC CPU accessed by the external device.



4) The external device connection station and access destination are explained in the section that describes the data communications formats in the QnA compatible 4C frames.
The following diagrams show examples of external device connection stations sending the transmission data shown in the description of each format in Section 3.1.5 and the PLC CPU accessed by the external devices.



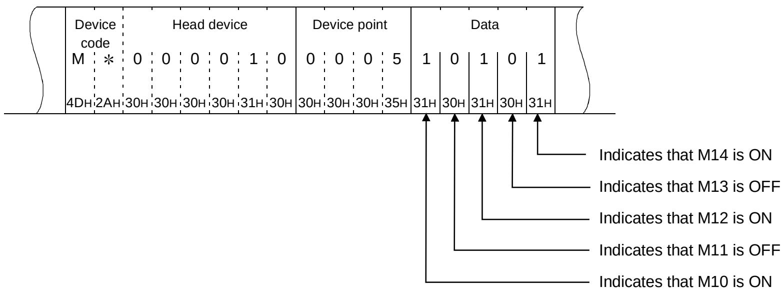
### 3.1.7 Character area transmission data

This section explains how to transmit the bit device data and word device data handled in the character areas, as well as the order of transmission when communicating data between an external device and the PLC CPU by each command.
The transmission data shown in the examples is contained in character area B in case of reading and monitoring, and in character area C in case of writing, testing, and monitor data registration.

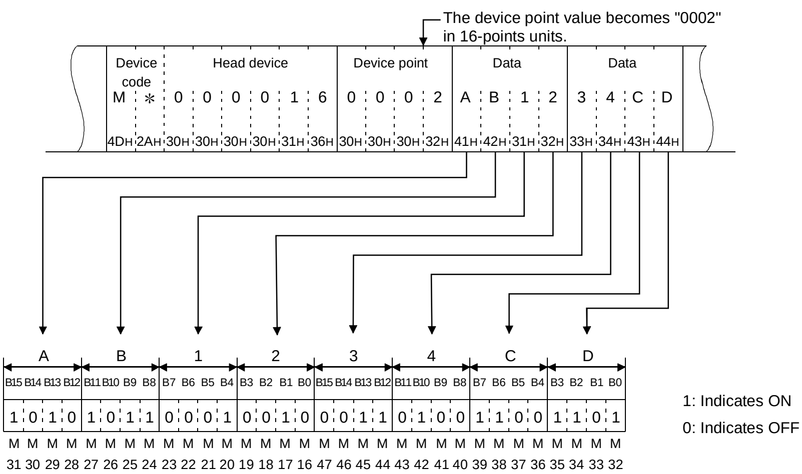
#### (1) Data communication using ASCII code

(a) When reading to or writing from a bit device memory
The bit device memory can be read and written in bit units (one device point) or word units (16 device points).
How data is transmitted in each case is explained below.

1) Bit units (one point)
In case of bit units, the bit device memory is handled from the designated head device for the number of designated device points sequentially from the left. They are expressed as "1" ($31_H$) if the device is on or "0" ($30_H$) if the device is off.
(Example)
Indicating the ON/OFF status of five points from M10




2) Word units (16 points)
When the bit device memory is handled as word units, each word is expressed in hexadecimal values in 4-bit units sequentially from the higher bit.
(Example)
Indicating the ON/OFF status of 32 points from M16

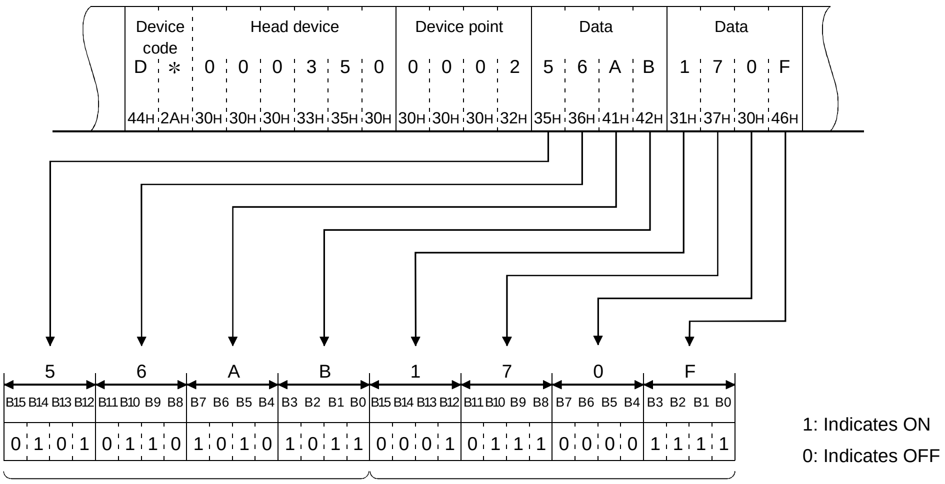


(b) When reading from or writing to a word device memory

Each word of the word device memory is expressed in hexadecimal values in 4-bit units sequentially from the higher bit.

(Example)

Indicating the storage contents of data registers D350 and D351




The content of D350 indicates 56ABH (22187 in decimal)

The content of D351 indicates 170FH (5903 in decimal)

**POINT**

(1) When designating alphabet letters in the character area, use upper case code.

(2) When data other than an integer value (real number, character string, etc.) was stored in a word device memory to be read, the Q series C24/E71 reads the stored value as an integer value.

(Example 1)

When the real number (0.75) is stored in D0 and D1, it is read as the following integer values:

D0 = 0000H, D1 = 3F40H

(Example 2)

When the character string ("12AB") is stored in D2 and D3, it is read as the following integer values:

D2 = 3231H, D3 = 4241H

(3) The same transmission method for bit device data and word device data and data arrangement at transmission shown in item (1) on the previous page apply to the following frames as well.

- QnA compatible 2C frames (explained in Chapter 4)
- A compatible 1C frames (explained in Chapter 5)
- A compatible 1E frames (explained in Chapter 6)

**REMARK**

Word unit data handled by the buffer memory read and write functions, etc. are also treated the same way as a word device memory.

(2) Data communication using binary code

(a) When reading to or writing from the bit device memory

The bit device memory can be read and written in bit units (one device point) or word units (16 device points).

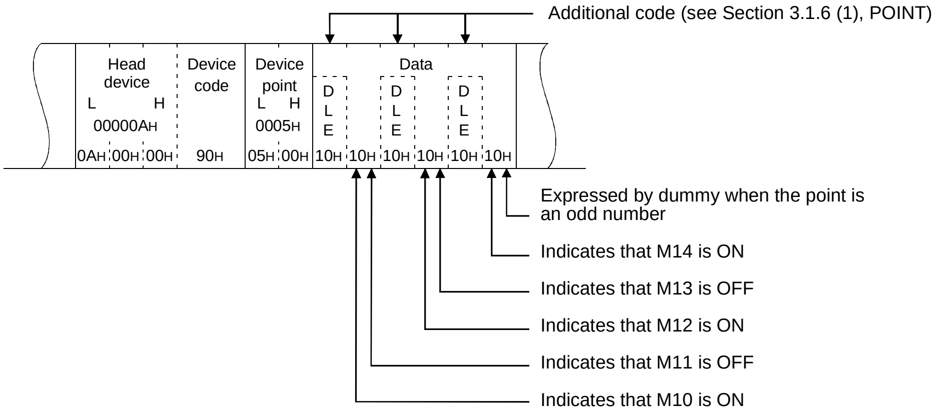
How data is transmitted in each case is explained below.

1) Bit units (one point)

In case of bit units, four bits designate one point and the bit device memory is handled from the designated head device for the number of designated device points sequentially from the left. They are expressed as "1" if the device is ON or "0" if the device is OFF.

(Example)

Indicating the ON/OFF status of five points from M10

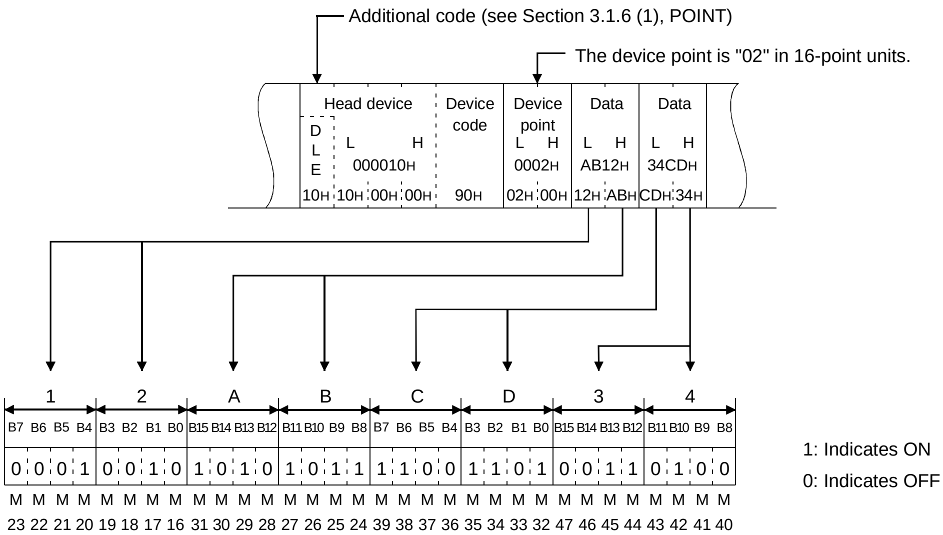



2) Word units (16 points)

In case of word units, one bit designates one point and the bit device memory is handled from the designated head device for the number of designated device points sequentially from the left. They are expressed in 16-point units in the order, low byte (L: bits 0 to 7) to high byte (bits 8 to 15).

(Example)

Indicating the ON/OFF status of 32 points from M16

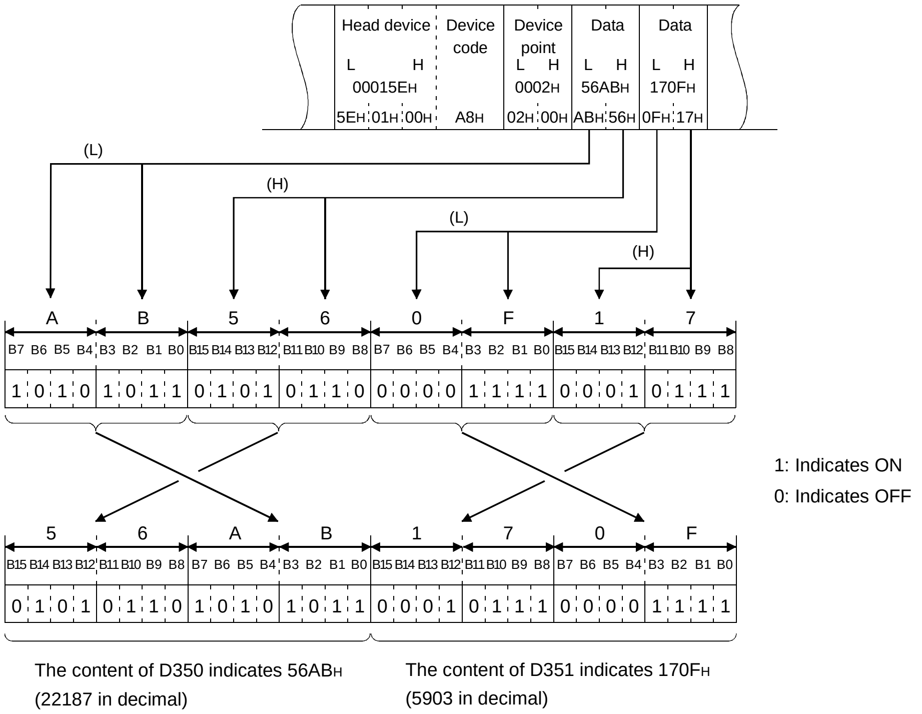


(b) Reading from or writing to a word device memory

Each word of a word device memory is designated by 16 bits and the designated number of points from the designated head device are sequentially expressed in one-point units in the order, low byte (L: bits 0 to 7) to high byte (H: bits 8 to 15).

(Example)

Indicating the storage contents of data registers D350 and D351




**POINT**

(1) When data other than an integer value (real number, character string, etc.) was stored in a word device memory to be read, the module reads the stored value as an integer value.

(Example 1)

When the real number (0.75) is stored in D0 and D1, it is read as the following integer values:

D0 = 0000H, D1 = 3F40H

(Example 2)

When the character string ("12AB") is stored in D2 and D3, it is read as the following integer values:

D2 = 3231H, D3 = 4241H

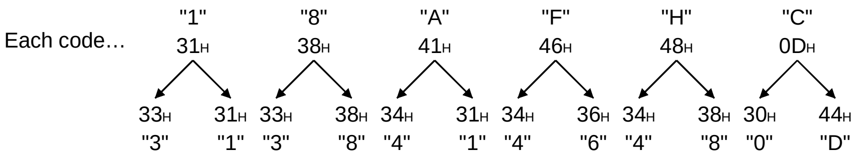
(2) The same transmission method for bit device data and word device data and data arrangement at transmission shown in this section also apply when communicating using the A compatible 1E frames (explained in Chapter 6) as well.

**REMARK**

(1) Reading/writing of the extension file register and reading/writing of the buffer memory On-demand data is handled the same way as a word device memory in the word designation.

(2) When communicating using ASCII data, follow the steps below to pass character strings from an external device to the PLC CPU and output them using the PR command.

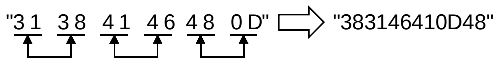
1) Decompress each character of a character string sent from the external device into a 2-byte code.




2) Rearrange the decompressed 2-byte codes for every two characters and send them to the Q series C24/E71.

(Example)

In case of the character string created in 1) above.

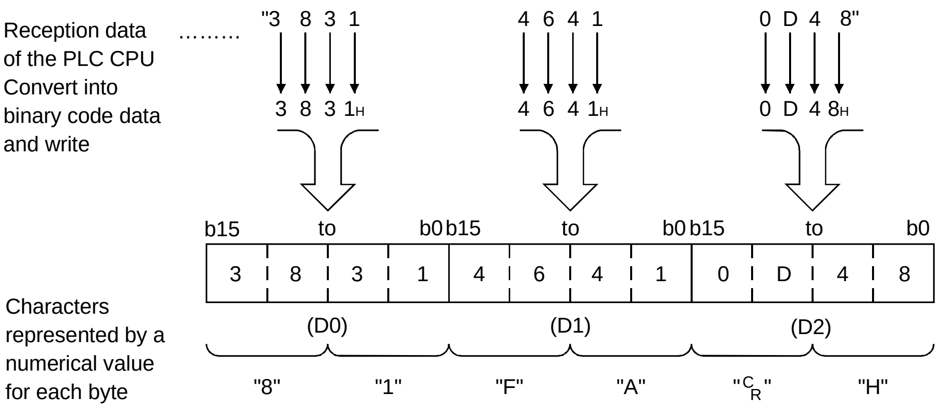



Send "383146410D48" from the external device to the PLC CPU.

The Q series C24/E71 converts the data received from the external device into binary code data and writes to the designated device.

(Example)

In case of writing the data created in 2) above to D0 to D2 of the PLC CPU.



## 3.2 List of Commands and Functions for the QnA Compatible 3E/3C/4C Frames

The following table lists the commands and functions used by an external device to access the PLC CPU with commands using the QnA compatible 3E/3C/4C frames.
The □ (in subcommands) in the table differ depending on the devices specified, usage of command functions, etc. (see the section explaining the corresponding command following Section 3.3 in this manual.)
The commands found in the □ section are those that can be used when accessing a non control PLC of a multiple CPU system.
Access commands for device memory can also be used to access PLC CPUs other than the Q/QnACPU.

| Function | | — | Command (subcommand) | Description of processing | Number of points processed per: Access station – 1 (＊7) | Access station – 2 (＊8) | Access station – 3 (＊9) |
|---|---|---|---|---|---|---|---|
| Device memory | Batch read | Bit units | 0401 (00□1) | Reads bit devices (X, Y, M, etc.) in one-point units. | 3584 points<br>7168 points<br>7904 points | 1792 points<br>3584 points<br>3952 points | 256 points |
| | | Word units | 0401 (00□0) | Reads bit devices (X, Y, M, etc.) in 16-point units. | 960 words (15360 points) | 480 words (7680 points) | 32 words (512 points) |
| | | | | Reads word devices (D, R, T, C, etc.) in one-point units. | 960 points | 480 points | 64 points |
| | Batch write (＊4) | Bit units | 1401 (00□1) | Writes to bit devices (X, Y, M, etc.) in one-point units. | 3584 points<br>7168 points<br>7904 points | 1792 points<br>3584 points<br>3952 points | 160 points |
| | | Word units | 1401 (00□0) | Writes to bit devices (X, Y, M, etc.) in 16-point units | 960 words (15360 points) | 480 words (7680 points) | 10 words (160 points) |
| | | | | Writes to word devices (D, R, T, C, etc.) in one-point units. | 960 points | 480 points | 64 points |
| | Random read | Word units | 0403 (00□0) | Reads bit devices (X, Y, M, etc.) in units of 16 points and 32 points by designating the devices and device number at random. | 192 points | 96 points | 10 words (160 points) |
| | | | | Reads word device (D, R, T, C, etc.) in units of 1 point and 2 points by designating the devices and device number at random. | | | 10 points |
| | Test [Random write] (＊4) | Bit units | 1402 (00□1) | Sets/resets bit devices (X, Y, M, etc.) in one-point unit by designating the devices and device number at random. | 188 points | 94 points | 20 points |
| | | Word units | 1402 (00□0) | Sets/resets bit devices (X, Y, M, etc.) in units of 16 points by designating the device and device number at random. | 1920 points | 960 points | 10 words (160 points) |
| | | | | Writes to word devices (D, R, T, C, etc.) in units of 1 point and 2 points by designating the devices and device number at random. However, for PLC CPUs other than Q/QnACPU, only one-point units can be used. | | | 10 points |
| | Monitor data registration | Word units | 0801 (00□0) | Registers the bit devices (X, Y, M, etc.) to be monitored in 16-point units. (＊2) | 192 points | 96 points | (prohibited) |
| | | | | Registers the word devices (D, R, T, C, etc.) to be monitored in units of 1 point and 2 points. However, for PLC CPUs other than Q/QnACPU, only one-point units can be used. | | | |
| | Monitor | Word units | 0802 (0000) | Monitors the devices registered for monitoring. | (for the number of registered points) | | |
| | Multiple block batch read (＊4) (＊10) | Word units | 0406 (00□0) | Reads and writes data by treating n points of word devices or bit devices (one point is equivalent to 16 bits) as one block and specifying multiple blocks randomly. | 960 points | 480 points | (prohibited) |
| | Multiple block batch write (＊4) (＊10) | Word units | 1406 (00□0) | | | 480 points | (prohibited) |

| Function | — | Access station (＊11) | | | | | | Status of the PLC CPU (＊1) | | | Command executable module | | Reference section |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | A series CPU | QnA series CPU | Q series CPU | MELSECNET/10 remote station | | MELSECNET/H remote station | During STOP | During RUN | | C24 | E71 | |
| | | | | | A series | QnA series | Q series | | Write allow setting | Write prohibit | | | |
| Batch read | Bit units | ○ | ○ | ○ | × | × | ○ | ○ | ○ | ○ | ○ | ○ | Sections 3.3.2 and 3.3.5 |
| | Word units | ○ | ○ | ○ | × | × | ○ | | | | | | |
| | | ○ | ○ | ○ | × | × | ○ | | | | | | |
| Batch write | Bit units | ○ | ○ | ○ | × | × | ○ | ○ | ○ | × | ○ | ○ | Sections 3.3.3 and 3.3.6 |
| | Word units | ○ | ○ | ○ | × | × | ○ | | | | | | |
| | | ○ | ○ | ○ | × | × | ○ | | | | | | |
| Random read | Word units | × | ○ | ○ | × | × | ○ | ○ | ○ | ○ | ○ | ○ | Section 3.3.8 |
| | | × | ○ | ○ | × | × | ○ | | | | | | |
| Test [random write] | Bit units | ○ | ○ | ○ | × | × | ○ | ○ | ○ | × | ○ | ○ | Sections 3.3.4 and 3.3.7 |
| | Word units | ○ | ○ | ○ | × | × | ○ | | | | | | |
| | | ○ | ○ | ○ | × | × | ○ | | | | | | |
| Monitor data registration | Word units | × | ○ | ○ | × | × | ○ | ○ | ○ | ○ | ○ | ○ | Section 3.3.9 |
| | | × | ○ | ○ | × | × | ○ | | | | | | |
| Monitor | Word units | × | ○ | ○ | × | × | ○ | ○ | ○ | ○ | | | |
| Multiple block batch read | Word units | × | ○ | ○ | × | × | ○ | ○ | ○ | ○ | ○ | ○ | Section 3.3.10 |
| Multiple block batch write | Word units | × | ○ | ○ | × | × | ○ | ○ | ○ | × | | | |

| Function | | | Command (subcommand) | Description of processing | | Number of points processed per Access station – 1 (* 7) | Access station – 2 (* 8) | Access station – 3 (* 9) |
|---|---|---|---|---|---|---|---|---|
| Buffer memory (* 3) | Batch read | | 0613 (0000) | Reads data in the buffer memory of the Q series C24/E71 connected to an external device. | This can also be used for data communication between the PLC CPU and an external device. | 960 words (1920 bytes) | 480 words (960 bytes) | (prohibited) |
| | Batch write | | 1613 (0000) | Writes data to the buffer memory of the Q series C24/E71 connected to an external device. | | | | |
| Intelligent function module | Batch read | | 0601 (0000) | Reads data in the buffer memory of an intelligent function module. | | 1920 bytes | 960 bytes | |
| | Batch write | | 1601 (0000) | Writes data to the buffer memory of an intelligent function module. | | | | |
| PLC CPU (* 4) | Remote RUN | | 1001 (0000) | Requests remote RUN to the PLC CPU. | | (for one station) | (for one station) | (prohibited) |
| | Remote STOP | | 1002 (0000) | Requests remote STOP to the PLC CPU. | | (for one station) | (for one station) | |
| | Remote PAUSE | | 1003 (0000) | Requests remote PAUSE to the PLC CPU. | | (for one station) | (for one station) | |
| | Remote latch clear | | 1005 (0000) | Requests remote latch clear to the PLC CPU when it is in the STOP status. | | (for one station) | (for one station) | |
| | Remote RESET | | 1006 (0000) | Request remote RESET to the PLC CPU in order to cancel the PLC CPU's error stop status. | | (for one station) | (for one station) | |
| | CPU model name read | | 0101 (0000) | Reads the model name from the PLC CPU | | (for one station) | (for one station) | |
| Drive memory | Memory usage status read | | 0205 (0000) | Reads the drive cluster usage status. | | (for 256 clusters) | | |
| | Memory defragmentation (* 4) (* 5) | | 1207 (0000) | Increases the continuous free area by defragmenting the drive memory (defragmentation of file storage location). | | (for one station) | | |
| File | File information table read | No header statement | 0201 (0000) | Reads the file table (filename, data and time of last update, and file size). | | (prohibited) | (for 36) | |
| | | Header statement | 0202 (0000) | Reads the file table (header statement attached to a file, file name, date and time of last update, and file size). | | | (for 16) | |
| | | File No. usage status | 0204 (0000) | Reads the file number usage status. | | | (for 256) | |
| | File information modification (* 4) (* 6) | Modification of time and data of last update | 1204 (0000) | Changes the date and time of the last update of the file. | | | (for 1) | |
| | | File name, file size modification | 1204 (0001) | Changes the filename or file size. | | | (for 1) | |
| | | Batch modification | 1204 (0002) | Changes the filename, file size, and date and time of last update. | | | (for 1) | |
| | File search | | 0203 (0000) | Reads whether or not the designated file exist, and the file number and file size. | | | (for 1) | |
| | File contents read (* 6) | | 0206 (0000) | Reads the contents of a file. | | | 960 bytes | |
| | New registration (File name registration) (* 4) (* 6) | | 1202 (0000) | Reserves a file area for the designated filename. | | | (for 1) | |
| | File contents write (* 4) (* 6) | Aribitary data | 1203 (0000) | Writes the designated data (n bytes) to a file. | | | 960 bytes | |
| | | Identical data (FILL) | 1203 (0001) | Writes n bytes of the designated data (one word) to a file. | | | For file size | |
| | File lock register/cancel | | 0808 (000□) | Registers a file lock so that the contents cannot be modified from another device while the designated file is being accessed, or cancels the same registration. | | (for 1) | | (prohibited) |
| | File copy (* 4) (* 6) | | 1206 (0000) | Writes the contents of an existing file to a newly registered file. (Copy) | | 960 bytes | 480 bytes | |
| | File delete (* 4) (* 6) | | 1205 (0000) | Deletes a file. | | (for 1) | | |

| Function | | Access station (＊11) | | | | | | Status of the PLC CPU (＊1) | | | Command execu-table module | | Reference section |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | A series CPU | QnA series CPU | Q series CPU | MELSECNET/10 remote station | | MELSECNET/H remote station | During STOP | During RUN | | | | |
| | | | | | A series | QnA series | Q series | | Write allow setting | Write prohibit | C24 | E71 | |
| Buffer memory batch read | | – | – | – | – | – | – | ○ | ○ | ○ | ○ | ○ | Section 3.4 |
| Buffer memory batch write | | – | – | – | – | – | – | ○ | ○ | ○ | | | |
| Intelligent function module batch read | | × | ○ | ○ | × | ○ | ○ | ○ | ○ | ○ | ○ | ○ | Section 3.5 |
| Intelligent function module batch write | | × | ○ | ○ | × | ○ | ○ | ○ | ○ | ○ | | | |
| Remote RUN | | × | ○ | ○ | × | × | × | ○ | ○ | ○ | ○ | × | Section 3.6 |
| Remote STOP | | × | ○ | ○ | × | × | × | | | | | | |
| Remote PAUSE | | × | ○ | ○ | × | × | × | | | | | | |
| Remote latch clear | | × | ○ | ○ | × | × | × | ○ | × | × | | | |
| Remote RESET | | × | ○ | ○ | × | × | × | ○ | × | × | | | |
| CPU model name read | | × | × | ○ | × | × | × | ○ | ○ | ○ | | | |
| Memory usage status read | | × | ○ | × | × | × | × | ○ | ○ | ○ | ○ | ○ | Section 3.7 |
| Memory defragmentation | | × | ○ | × | × | × | × | ○ | × | × | | | |
| File information table read | No header statement | × | ○ | × | × | × | × | ○ | ○ | ○ | ○ | ○ | Section 3.8.16 |
| | Header state-ment | × | ○ | × | × | × | × | | | | | | |
| | File No. usage status | × | ○ | × | × | × | × | | | | | | |
| File information modification | Modifica-tion of time and data of last update | × | ○ | × | × | × | × | ○ | ○ | × | ○ | ○ | Section 3.8.24 |
| | File name, file size modifica-tion | × | ○ | × | × | × | × | | | | | | |
| | Batch modifica-tion | × | ○ | × | × | × | × | | | | | | |
| File search | | × | ○ | × | × | × | × | ○ | ○ | ○ | ○ | ○ | Section 3.8.17 |
| File contents read | | × | ○ | × | × | × | × | ○ | ○ | ○ | ○ | ○ | Section 3.8.19 |
| New registration (File name registration) | | × | ○ | × | × | × | × | ○ | ○ | × | ○ | ○ | Section 3.8.20 |
| File contents write | Arbitrary data | × | ○ | × | × | × | × | ○ | ○ | × | ○ | ○ | Section 3.8.21 |
| | Identical data (FILL) | × | ○ | × | × | × | × | | | | | | |
| File lock register/cancel | | × | ○ | × | × | × | × | ○ | ○ | ○ | ○ | ○ | Section 3.8.18 |
| File copy | | × | ○ | × | × | × | × | ○ | ○ | ○ | ○ | ○ | Section 3.8.23 |
| File delete | | × | ○ | × | × | × | × | ○ | ○ | × | ○ | ○ | Section 3.8.22 |

| Function | — | Command (subcommand) | Description of processing | Number of points processed per communication | | |
|---|---|---|---|---|---|---|
| | | | | Access station-1 (* 7) | Access station-2 (* 8) | Access station-3 (* 9) |
| File | Directory file information read | 1810 (0000) | Reads file list information | (for 36) | | |
| | Directory file information search (* 6) | 1811 (0000) | Reads file number of the designated file | (for 16) | | |
| | New file creation (* 4) (* 6) | 1820 (0000) | Reserves storage area for the designated file | (for 256) | | |
| | File delete (* 4) (* 6) | 1822 (0000) | Deletes a file | (for 1) | | |
| | File copy (* 4) (* 6) | 1824 (0000) | Copies the designated file | (for 1) | | |
| | File attribute modification (* 4) (* 6) | 1825 (0000) | Changes file attributes | (for 1) | | |
| | File creation date modification (* 4) | 1826 (0000) | Changes file creation date | (for 1) | | |
| | File open (* 6) | 1827 (0000) | Locks a file so that the content is not changed by other devices | (for 1) | | |
| | File read (* 6) | 1828 (0000) | Reads the contents of a file | 1920 bytes | | |
| | File write (* 4) (* 6) | 1829 (0000) | Writes the content in a file | 1920 bytes | | |
| | File close | 182A (0000) | Cancels the file lock by Open processing | (for 1) | | |
| User frame (* 3) | Registered data read | 0610 (0000) | Reads registered data of designated frame number | 80 bytes | | |
| | Data registration | 1610 (0000) | Registers (writes) data order of head frame/last frame when communicating data in a user formatted message | (for 1) * 7 1) 5) | | |
| | Registered data delete | 1610 (0001) | Deletes registered data of designated frame number | (for one station/all stations) | | |
| Global (* 3) | | 1618 (000□) | Turns ON/OFF the global signals (X1A/X1B) of a Q/QnACPU on which the Q series C24 is installed | Same as above | | |
| On-demand | | 2101 (——) | Issues transmission request from the PLC CPU and sends the data to the external device.<br>It is possible to send data equivalent to the size of the maximum continuous unused area in the user-defined area of the Q series C24 buffer memory.<br>(System configuration 1: possible for 1) | (for one station)<br>* 7 possible only for 1) station | (Prohibited) | |
| Transmission sequence initialization (only in binary mode) (* 3) | | 1615 (0000) | Stops the current processing request and instructs the Q series C24 to wait to receive a command | (for one station)<br>* 7 possible only for 1) and 3) station | | |
| Mode switching (* 3) | | 1612 (0000) | Switches operation mode and transmission specification of the designated interface | (for one station) | | |
| LED off, Error code initialization (* 3) | | 1617 (000□) | Turns off the displayed error LED and initializes any error codes | (for one station)<br>* 7 possible only for 1) and 3) station<br>(Prohibited) | | |
| Loopback test (* 3) | | 0619 (0000) | Checks that the data communication between the Q series C24/E71 and the external device is performed normally (for checking connection status and communication function) | 960 bytes (communication is possible only with a connected station) | | |
| PLC CPU monitoring | Registration | 0630 (0000) | Registers monitoring of device memory and CPU status and starts the operation of the PLC CPU monitoring | 960 points | | |
| | Cancel | 0631 (0000) | Ends the operation of the PLC CPU monitoring | – | | |
| Remote password | Unlock | 1630 (0000) | Specify the remote password and change the lock status to unlock status. (This makes it possible to communicate with the PLC CPU.) | – | | |
| | Lock | 1631 (0000) | Specify the remote password and change the unlock status to lock status. (This makes it impossible to communicate with the PLC CPU) | – | | |

| Function | — | Access station (* 11) | | | | | | Status of the PLC CPU (* 1) | | | Command executable module | | Reference section |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | A series CPU | QnA series CPU | Q series CPU | MELSECNET/10 remote station | | MELSECNET/H remote station | During STOP | During RUN | | | | |
| | | | | | A series | QnA series | Q series | | Write allow setting | Write prohibit | C24 | E71 | |
| File | Directory file information read | × | × | ○ | × | × | × | ○ | ○ | ○ | ○ | ○ | Section 3.8.5 |
| | Directory file information search | × | × | ○ | × | × | × | ○ | ○ | ○ | ○ | ○ | Section 3.8.6 |
| | New file creation | × | × | ○ | × | × | × | ○ | ○ | × | ○ | ○ | Section 3.8.10 |
| | File delete | × | × | ○ | × | × | × | ○ | ○ | × | ○ | ○ | Section 3.8.12 |
| | File copy | × | × | ○ | × | × | × | ○ | ○ | C24: ×<br>E71: ○ | ○ | ○ | Section 3.8.13 |
| | File attribute modification | × | × | ○ | × | × | × | ○ | ○ | × | ○ | ○ | Section 3.8.15 |
| | File creation date modification | × | × | ○ | × | × | × | ○ | ○ | × | ○ | ○ | Section 3.4.14 |
| | File open | × | × | ○ | × | × | × | ○ | ○ | ○ | ○ | ○ | Section 3.8.7 |
| | File read | × | × | ○ | × | × | × | ○ | ○ | ○ | ○ | ○ | Section 3.8.9 |
| | File write | × | × | ○ | × | × | × | ○ | ○ | × | ○ | ○ | Section 3.8.11 |
| | File close | × | × | ○ | × | × | × | ○ | ○ | ○ | ○ | ○ | Section 3.8.8 |
| User frame | Registered data read | – | – | – | – | – | – | ○ | ○ | ○ | | | |
| | Data registration | – | – | – | – | – | – | ○ | ○ | ○ | ○ | × | Section 3.9 |
| | Registered data delete | – | – | – | – | – | – | ○ | ○ | ○ | | | |
| Global | | × | ○ | ○ | × | × | × | × | ○ | ○ | ○ | × | Section 3.10 |
| On-demand | | × | ○ | ○ | × | × | × | × | ○ | ○ | ○ | × | Section 3.11 |
| Transmission sequence initialization (only in binary mode) | | – | – | – | – | – | – | ○ | ○ | ○ | ○ | × | Section 3.12 |
| Mode switching | | – | – | – | – | – | – | ○ | ○ | ○ | ○ | × | Section 3.13 |
| LED off, Error code initialization | | – | – | – | – | – | – | ○ | ○ | ○ | ○ | × | Section 3.14 |
| | | | | | | | | ○ | ○ | ○ | × | ○ | Section 3.15 |
| Loopback test | | – | – | – | – | – | – | ○ | ○ | ○ | ○ | ○ | Section 3.16 |
| PLC CPU monitoring registration | | – | – | – | – | – | – | ○ | ○ | ○ | ○ | × | Section 3.17 |
| PLC CPU monitoring cancel | | – | – | – | – | – | – | ○ | ○ | ○ | | | |
| Remote password unlock | | – | – | – | – | – | – | ○ | ○ | ○ | ○ | ○ | Section 3.1.18 |
| Remote password lock | | – | – | – | – | – | – | ○ | ○ | ○ | × | | |

*1 The write allow/prohibit during RUN setting for the PLC CPU should be set using the following GX Developer screens.
- In case of the Q series C24:
  The "Switch setting for I/O and intelligent functional module" screen
- In case of the Q series E71:
  The "Ethernet operations" screen

*2 In modules other than AnA/AnU/QnA/QCPUs, device X (input) is processed for two points per point.
When X is included in the designated device, set the processed points as follows:
((points designated for X $\times$ 2) + points of other designated devices) $\leq$ the number of points that can be processed in one communication
When only X is specified, the number of points that can be processed in one communication is 1/2 of the values shown in the table.

*3 These commands can be issued by the external device to connected the Q series C24 (including a multidrop connection station) as well as to a Q/QnACPU station on which the Q series C24 is installed.
The commands cannot be issued to PLC CPUs of other stations by passing them through the network system.

*4 When there is a system protection on a Q/QnACPU to which one of these commands is issued, an error occurs and a NAK message is returned.

*5 When a data read/write keyword (password) is registered for a Q/QnACPU to which a command is issued, that same keyword should be specified in the command message. If the keywords do not match, an error occurs and a NAK message is returned.

*6 When a data read/write keyword (password) is registered for a Q/QnACPU when a command is issued on its program files and parameter files, that keyword should be specified in the command message.
If the keywords do not match, an error occurs and a NAK message is returned.

*7 "Access station-1" indicates the access which one of the following stations.
1) Q series C24/E71 installed station (local station)
2) A QCPU station (other station)/MELSECNET/H remote I/O station, passing through a network system (MELSECNET/H, MELSECNET/10, Ethernet) supporting the Q series
3) The Q series C24 of 1) and 2) above and QCPU stations with multidrop connection

*8 "Access station-2" indicates the acess which one of the following stations.
1) The Q/QnACPU stations (other station)/MELSECNET/10 remote I/O stations, passing through a network system (MELSECNET/H, MELSECNET/10, Ethernet) supporting the QnA series
2) The Q series C24 installed on 1) above and Q/QnACPU stations (other stations) with multidrop connection
3) The Q/QnACPUs of 1) and 2) above and Q/QnACPU stations (other stations)/MELSECNET/H, MELSECNET/10 remote I/O stations, passing through a network system (MELSECNET/H, MELSECNET/10, Ethernet)

*9 "Access station-3" indicates the access which stations other than *7 and *8 above.
(Example)
PLC CPUs other than Q/QnACPU stations (other stations)

*10 For details on the QnACPU to which multi-block batch read/write can be performed, see Section 3.3.10.

*11 See Section 2.6.1 for details on the access station shown in the table.

## 3.3 Device Memory Read/Write

This section describes what to designate in the control procedure when reading and writing to the device memory by explaining with some examples.

### 3.3.1 Commands, character area contents and device range

The following describes the commands, character areas (in binary code, data fields) of the control procedures, and the range of accessible devices when reading and writing to device memory.

(1) Commands

| Function | — | Command (subcommand) | Description of processing | Number of points processed per communication (see *7 to *9 in Section 3.2) | | | PLC CPU status (see *9 in Section 3.2) | | | Reference section |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | | Access station-1 | Access station-2 | Access station-3 | During STOP | During RUN: Write allow setting | During RUN: Write prohibit setting | |
| Batch read | Bit units | 0401 (00□1) | Reads bit devices in 1-point units. | 3584 points<br>7168 point<br>7904 point | 1792 points<br>3584 point<br>3952 point | 256 points | ○ | ○ | ○ | Sections 3.3.2 and 3.3.5 |
| | Word units | 0401 (00□0) | Reads bit devices in 16-point units. | 960 words (15360 points) | 480 words (7680 points) | 32 words (512 points) | | | | |
| | | | Reads word devices in 1-point units. | 960 points | 480 points | 64 points | | | | |
| Batch write (see *4 in Section 3.2) | Bit units | 1401 (00□1) | Writes to bit devices in 1-point units. | 3584 points<br>7168 point<br>7904 point | 1792 points<br>3584 point<br>3952 point | 160 points | ○ | ○ | × | Sections 3.3.3 and 3.3.6 |
| | Word units | 1401 (00□0) | Writes to bit devices in 16-point units | 960 words (15360 points) | 480 words (7680 points) | 10 words (160 points) | | | | |
| | | | Writes to word devices in 1-point units. | 960 points | 480 points | 64 points | | | | |
| Random read (*1) | Word units | 0403 (00□0) | Reads bit devices in units of 16 points and 32 points by designating the devices at random. | 192 points | 96 points | 10 words (160 points) | ○ | ○ | ○ | Section 3.3.8 |
| | | | Reads word devices in 1-point and 2-point units by designating the devices at random. | | | 10 points | | | | |
| Test (*1) [Random write] (see *4 in Section 3.2) | Bit units | 1402 (00□0) | Sets/resets bit devices in 1-point units by designating the devices at random. | 188 points | 94 points | 20 points | ○ | ○ | × | Sections 3.3.4 and 3.3.7 |
| | Word units | 1402 (00□0) | Sets/resets bit devices in units of 16 points and 32 points by designating the units at random. | (see Section 3.3.7) | | 10 words (160 points) | | | | |
| | | | Writes to word devices in 1-point and 2-point units by designating the devices at random. (see *2 in Section 3.2) | | | 10 points | | | | |

| Function | — | Command (subcommand) | Description of processing | Number of points processed per communication (see ＊7 to ＊9 in Section 3.2) | | | PLC CPU status (see ＊1 in Section 3.2) | | | Reference section |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | | Access station-1 | Access station-2 | Access station-3 | During STOP | During RUN: Write allow setting | During RUN: Write prohibit setting | |
| Monitor data registration (＊1) | Word units | 0801 (00□0) | Registers the bit devices to be monitored in units of 16 points and 32 points. | 192 points | 96 points | 20 words (320 points) | ○ | ○ | ○ | Section 3.3.9 |
| | | | Registers the word devices to be monitored in 1-point and 2-point units. | | | 20 points | | | | |
| Monitor | Word units | 0802 (0000) | Monitors the devices registered for monitoring. | (for number of registered points) | | | ○ | ○ | ○ | |
| Multiple block batch read | Word units | 0406 (00□0) | Reads data by treating n points of word devices or bit devices (one point is equivalent to 16 bits) as one block by specifying multiple blocks randomly. | 960 points | 480 points | (Prohibited) | ○ | ○ | ○ | Section 3.3.10 |
| Multiple block batch write | Word units | 1406 (00□0) | Writes data by treating n points of word devices or bit devices (one point is equivalent to 16 bits) as one block by specifying multiple blocks randomly. | 960 points | 480 points | (Prohibited) | ○ | ○ | × | |

○ in the PLC CPU status column of the table above indicates that the corresponding function is executable.

＊1 The number of processing points for other than Q/QnACPU is in 1-point units for word devices and 16-point units for bit devices.

## (2) Character areas (data fields in binary mode)

This section describes the common data in the character areas in each control procedure when the external device reads and writes PLC data.

**POINT**

(1) The character area depends on the command used and the content specified. This section describes the data common to the character areas when the device memory to be read or written is designated directly.
(2) Character area data handled only by a certain command and not by others, is described in the section that describes the corresponding command.
(3) The following sections describes data handled by some of the functions and data for specifying device memory extension using a special expression.

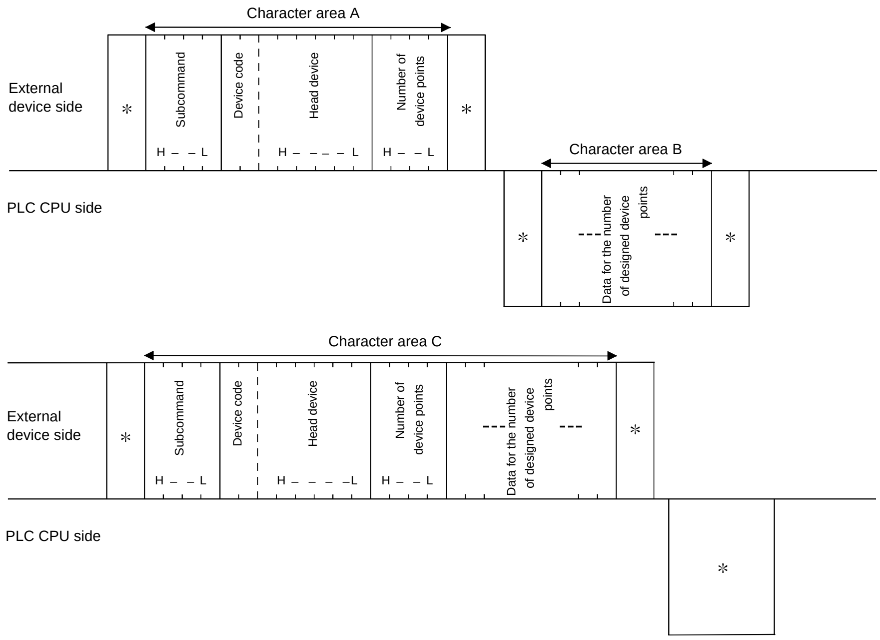
- Section 3.3.8 : Data designated by the random read word units (command: 0403) and monitor data registration (command: 0801) functions.
- Appendix 1 : Data used to designate the extension of the device memory to be read or written.

(a) Data in the character areas when communicating in ASCII code

The data order and contents of character areas A, B, and C are identical when the same command is used under the same conditions in the control procedure when communicating using ASCII code.

(Example)

Reading data

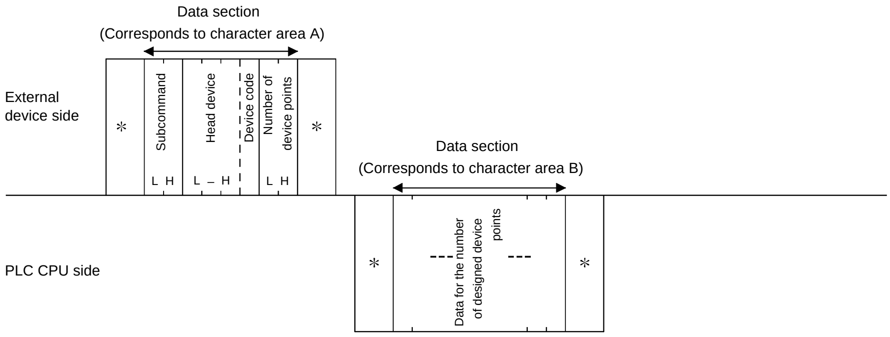



| |
|---|
| The data arrangement and content of the area marked with "✻" are shown in Section 3.1. |

(b) Data in data fields when communicating in binary code

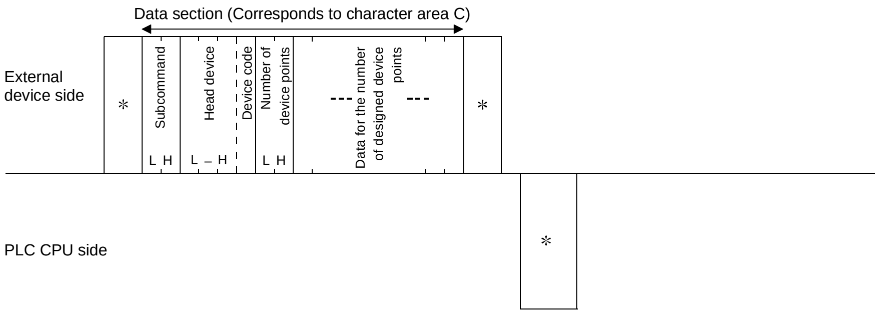
(Example)

Reading data




(Example)

Writing data




| |
|---|
| The data arrangement and content of the area marked with "✻" are shown in Section 3.1. |

(c) Contents of data common to character areas

1) Subcommand

This value designates the read/write units, type of device designated, conditions for reading data, etc.

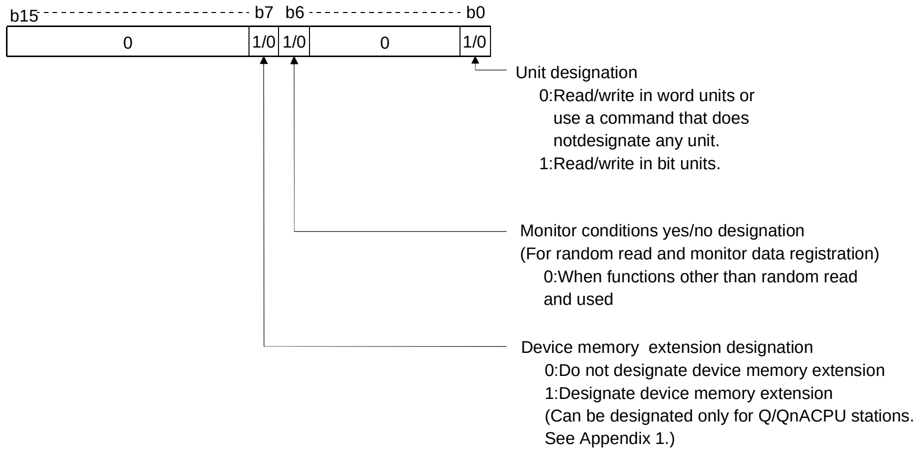
a) When data is communicated in ASCII code

The value 0000H (0), or the following value, is converted to a 4 digit (hexadecimal) ASCII code and sequentially transmitted beginning from the most significant digit ("0").

b) When data is communicated in binary code

The value 0000H, or the following 2-byte value, is used for transmission.

c) The following items are designated by subcommands.




d) In the following cases, the subcommand is 0000H/0001H.
- When neither monitor condition and device memory extension are designated.
- When using a command that cannot select monitor condition designation and device memory extension designation.

**POINT**

The following device memories of stations on which the Q series C24/E71 is mounted and other station's Q/QnACPUs can be accessed by setting the "Device memory extension designation" in bit 7 of a subcommand. See Appendix 1.

1) The device memory for direct input/output to the Q/QnACPU
2) The buffer register of an intelligent functional module (buffer memory)

2) Device code

This value identifies the device memory to be read or written.

a) The device codes are shown in the table in (3).

b) When data is communicated in ASCII code

The device code is converted to a 2-digit ASCII code and sequentially transmitted beginning from the most significant digit.

(Example)

In case of an input relay

The input relay device code "X✻" is sequentially transmitted from "X".

The second character "✻" can be designated a blank space (code: 20H).

c) When data is communicated in binary code

The 1-byte value shown in the table in (3) is transmitted.

3) Head device (device)

This value designates the number of the device memory to read data from or write data to.

When a range of contiguous device memory numbers are designated, the head number of that device memory range is designated.

The head device number is designated by the data representation (decimal or hexadecimal) shown in the "Representation" column of the table shown in (3) according to the objective device memory.

a) When data is communicated in ASCII code

The device number shown in the table is converted to a 6-digit ASCII code and sequentially transmitted beginning from the most significant digit.

The "0" column of the most significant digit (in for example "001234", this refers to "0" of the first two characters) can also be designated by a blank space (code: 20H).

(Example)

In case of internal relay M1234 and link relay B1234

Internal relay M1234 and link relay B1234 become both "001234" or "␣␣1234" and are sequentially transmitted from the first "0 "or "␣".

b) When data is communicated in binary code

The 3-byte value shown in the table is sequentially transmitted beginning from the low byte (L: bits 0 to 7).

(Example)

In case of internal relay M1234 and link relay B1234

Internal relay M1234 becomes 0004D2H and is transmitted in the order of D2H, 04H, and 00H.

Link relay B1234 becomes 001234H and is transmitted in the order of 34H, 12H, and 00H.

4) Device points

This value designates the number of points to be read or written when each command is executed. It should be designated within the limits to the number of points that can be processed per communication given in the table in (1).

a) When data is communicated in ASCII code
The number of processed points is converted to a 4-digit (hexadecimal) ASCII code and sequentially transmitted beginning from the most significant digit.
(Example)
5 points : Becomes "0005" and is sequentially transmitted beginning from the first "0".
20 points : Becomes "0014" and is sequentially transmitted beginning from the first "0".

b) When data is communicated in binary code
A 2-byte value representing the number of points to be processed is sequentially transmitted beginning from the low byte (L: bits 0 to 7).
(Example)
5 points : Becomes $0005_H$ and is transmitted in the order of $05_H$, then $00_H$.
20 points : Becomes $0014_H$ and is transmitted in the order of $14_H$, then $00_H$.

5) Data for the designated number of device points
This field holds the actual contents of the data written to the designated device memory, or the contents of the data read from the designated device memory. The data order changes depending on the processing units (words/bytes).
See Section 3.1 for the data contents and order (transmission order).

6) Number of bit access points
This value designates the number of points to be accessed in units of bits. It should be designated within the limits to the number of points processed per communication given in the table in Section 3.3.1 (1).

a) When data is communicated in ASCII code
The number of points is converted into a 2-digit (hexadecimal) ASCII code and sequentially transmitted beginning from the most significant digit.
(Example)
5 points : Becomes "05" and is sequentially transmitted beginning from the first "0".
20 points : Becomes "14" and is sequentially transmitted beginning from the first "1".

b) When data is communicated in binary code
A 1-byte value representing the number of points is used for transmission.
(Example)
5 points : $05_H$ is transmitted.
20 points : $14_H$ is transmitted.

7) Set/reset
This is the data to be written to a bit device. It is designated by the values shown below.

| | Data to be written | | Remarks |
|---|---|---|---|
| | ON | OFF | |
| ASCII code | “01” | “00” | Two characters are sequentially transmitted from the “0” |
| Binary code | $01_H$ | $00_H$ | A 1-byte value is transmitted as shown |

## (3) Device range

This indicates the device and device number range of the PLC CPU that can be accessed.
Specify the device and device number range existing in the modules to be read or written.

(a) In the case of Q/QnACPU

Table 3.1 Accessible devices list (Q/QnACPU)

| Classification | Device | | Device type: Bit | Device type: Word | Device code: ASCII code | Device code: Binary code | Device No. range (Default allocation): Q02(H), Q06H, Q12H, Q25H, Q12PH, Q25PH, Q12PRH, Q25PRH, Q2A, Q2A-S1, Q2AS, Q2AS-S1, Q2ASH, Q2ASH-S1, Q3A, Q4A, Q4AR | Device No. range (Default allocation): Q00J, Q00, Q01 | Representation: Decimal | Representation: Hexa-decimal | Remarks |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Internal system | Function input | | ○ | | – | – | 000000 to 00000F | 000000 to 00000F | | ○ | Cannot be accessed |
| | Function output | | ○ | | – | – | 000000 to 00000F | 000000 to 00000F | | ○ | |
| | Function register | | | ○ | – | – | 000000 to 000004 | 000000 to 000004 | ○ | | |
| | Special relay | | ○ | | SM | 91H | 000000 to 002047 | 000000 to 001023 | ○ | | – |
| | Special register | | | ○ | SD | A9H | 000000 to 002047 | 000000 to 001023 | ○ | | |
| Internal user | Input relay | | ○ | | X＊ | 9CH | 000000 to 001FFF | 000000 to 0007FF | | ○ | See the manual for the sequencer CPU you are using for the number of points that can be changed in the device allocation. Devices up to the maximum device No. can be accessed after the change when changing the allocation. Local devices can not be accessed. |
| | Output relay | | ○ | | Y＊ | 9DH | 000000 to 001FFF | 000000 to 0007FF | | ○ | |
| | Internal relay ＊2 | | ○ | | M＊ | 90H | 000000 to 008191 | 000000 to 008191 | ○ | | |
| | Latch relay ＊2 | | ○ | | L＊ | 92H | 000000 to 008191 | 000000 to 002047 | ○ | | |
| | Annunciator | | ○ | | F＊ | 93H | 000000 to 002047 | 000000 to 001023 | ○ | | |
| | Edge relay | | ○ | | V＊ | 94H | 000000 to 002047 | 000000 to 001023 | ○ | | |
| | Link relay | | ○ | | B＊ | A0H | 000000 to 001FFF | 000000 to 0007FF | | ○ | |
| | Data register | | | ○ | D＊ | A8H | 000000 to 012287 | 000000 to 011135 | ○ | | |
| | Link register | | | ○ | W＊ | B4H | 000000 to 001FFF | 000000 to 0007FF | | ○ | |
| | Timer ＊3 | Contact | ○ | | TS | C1H | 000000 to 002047 | 000000 to 000511 | ○ | | |
| | | Coil | ○ | | TC | C0H | | | ○ | | |
| | | Current value | | ○ | TN | C2H | | | ○ | | |
| | Retentive timer ＊3 | Contact | ○ | | SS | C7H | | | ○ | | |
| | | Coil | ○ | | SC | C6H | | | ○ | | |
| | | Current value | | ○ | SN | C8H | | | ○ | | |
| | Counter ＊3 | Contact | ○ | | CS | C4H | 000000 to 001023 | 000000 to 000511 | ○ | | |
| | | Coil | ○ | | CC | C3H | | | ○ | | |
| | | Current value | | ○ | CN | C5H | | | ○ | | |
| | Special link relay | | ○ | | SB | A1H | 000000 to 0007FF | 000000 to 0003FF | | ○ | |
| | Special link register | | | ○ | SW | B5H | 000000 to 0007FF | 000000 to 0003FF | | ○ | |
| | Step relay ＊2 | | ○ | | S＊ | 98H | 000000 to 008191 | 000000 to 002047 | ○ | | The Q00J/Q00/Q01 CPU cannot be accessed. |
| | Direct input | | ○ | | DX | A2H | 000000 to 001FFF | 000000 to 0007FF | | ○ | Same as input relay and output relay (for direct access) |
| | Direct output | | ○ | | DY | A3H | 000000 to 001FFF | 000000 to 0007FF | | ○ | |
| | Index register | | | ○ | Z＊ | CCH | 000000 to 000015 | 000000 to 000009 | ○ | | – |
| – | File register ＊4 ＊5 ＊6 | | | ○ | R＊ | AFH | 000000 to 032767 | 000000 to 032767 | ○ | | For normal access by block switching |
| | | | | | ZR | B0H | 000000 to 0FE7FF | 000000 to 00FFFF | | ○ | For serial No. access |

Table 3.2 Accessible Devices List (MELSECNET/H remote I/O station)

| Device | Device type: Bit | Device type: Word | Device code: ASCII code | Device code: Binary code | Device No. range: QJ72LP25-25, QJ72LP25GE, QJ72BR15 | Representation: Decimal | Representation: Hexadecimal | Remarks |
|---|---|---|---|---|---|---|---|---|
| Special relay | ○ | | SM | 91H | 000000 to 002047 | ○ | | – |
| Special register | | ○ | SD | A9H | 000000 to 002047 | ○ | | |
| Input relay | ○ | | X＊ | 9CH | 000000 to 001FFF | | ○ | Cannot change the allocation |
| Output relay | ○ | | Y＊ | 9DH | 000000 to 001FFF | | ○ | |
| Internal relay | ○ | | M＊ | 90H | 000000 to 008191 | ○ | | |
| Link relay | ○ | | B＊ | A0H | 000000 to 003FFF | | ○ | |
| Data register | | ○ | D＊ | A8H | 000000 to 012287 | ○ | | |
| Link register | | ○ | W＊ | B4H | 000000 to 003FFF | | ○ | |
| Special link relay | ○ | | SB | A1H | 000000 to 0001FF | | ○ | |
| Special link register | | ○ | SW | B5H | 000000 to 0001FF | | ○ | |

∗1 The device memory in the Q/QnACPU is accessed.
The device range varies depending on the Q/QnACPU version.
For device range, see the User's Manual for the CPU which is used.
When the device range was modified by parameter settings, devices within the new device range after modification can be accessed.
However, local devices, such as programs, cannot be accessed.

∗2 For Q/QnACPUs, the internal user's internal relay (M), latch relay (L), and step relay (S), are different devices.

∗3 The Q/QnACPU for which each contact and coil for the timer, retentive timer and counter can be designated during random read is shown in the following table.
For function version A of the QnACPU, however, each contact and coil for the timer, retentive timer and counter cannot be designated.
In monitor data registration, the Q/QnACPU's timer, retentive timer and counter contacts are well as coils cannot be designated.

| Function | Command | QCPU | | QnACPU | |
|---|---|---|---|---|---|
| | | Function Version | | Function Version | |
| | | A | B | A | B |
| Random read | 0403 | ○ | | × | ○ |
| Monitor data registration | 0801 | × | | × | |

○: Can be designated ×: Cannot be designated

When each contact and coil for the timer, retentive timer and counter are designated for a QnACPU that cannot be designated, the error 4032H is returned.

∗4 Designate the following device code according to the PLC CPU file register configuration when accessing the file register from the external device.
1) When the file register is made up of multiple blocks
   - Designate the "ZR" serial number access device code in ASCII mode, and designate the "B0H" serial number access device code in binary code.
2) When the file register is made up of block 0 only
   - Designate the "ZR"/"R∗" serial number access device code in ASCII mode, and designate the "B0H"/"AFH" serial number access device code in binary mode.
   - The device number can be specified by a decimal number by designating the normal access code of "R∗" or "AFH."
   - In serial number access, a hexadecimal number specifies the device number.

   For device numbers of file registers for serial access and normal access, as well as the access restrictions, refer to the user's manuals and programming manuals for the respective CPU modules.

REMARK

Example of device numbers for serial number access (for QCPU)

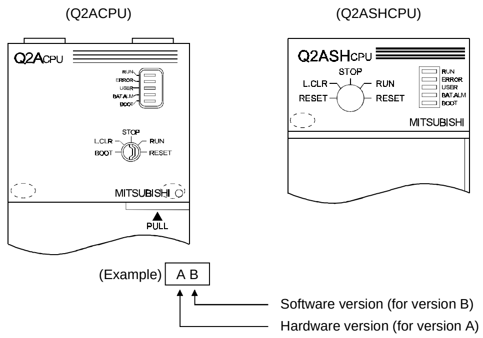
| | |
|---|---|
| ZR000000( 0) to ZR032767(32767) | File register block number 0 area (R0 to R32767) |
| ZR032768(32768) to ZR065535(65535) | File register block number 1 area (R0 to R32767) |
| ZR065536(65536) to | File register block number 2 area |

The device number for serial number access is allocated automatically in order from the device with the smallest block No. of the existing blocks.

＊5 Writing data to the file register defined in the QnACPU EEPROM can only be performed when all of the following restrictions are cleared.
If any of the following restrictions is not cleared, an abnormal completion message will be returned at the point when writing data to the file register is attempted.
(Restrictions when writing data to EEPROM file register)
- Writing is allowed only through the use of the batch write function (command: 1401).
- Writing is allowed only when the operation status of the target QnACPU is stopped or paused.
- Writing is allowed only for the QnACPU types listed below, and all QnACPU types manufactured in January 1999 or later.

| Types of QnACPU | Manufactured date | Software version |
|---|---|---|
| Q2ACPU (-S1), Q3ACPU, Q4ACPU | September to December, 1998 | Version L and later |
| Q4ARCPU | | Version S and later |
| Q2ASCPU (-S1), Q2ASHCPU (-S1) | | Version T and later |

See the applicable manuals for QnACPU models not listed above.
＊ The manufactured date (last 2 digits of the year, 2-digit month) can be verified in the "DATE column of the rated plate" located on the side of the module.
The module's software version can be verified on the seal (located in either of the dotted areas below) attached to the front of the module.




＊6 In the parameters, file registers for each program with a file name that is the same as that of the program cannot be accessed from an external device.

(b) In the case of other than Q/QnACPU

Table 3.3 Accessible Devices List (PLC CPUs Other Than the Q/QnACPU) (＊7)

| Device | | | Device type | | Device code | | Device No. range (Default allocation) | | Representation | | Remarks |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | Bit | Word | ASCII code | Binary code | QnA compatible 3C/4C frame | QnA compatible 3E frame | Decimal | Hexadecimal | |
| Input | | X | ○ | | X＊ | $9C_H$ | 000000 to 001FFF | 000000 to 0007FF | | ○ | Extension designation is not possible. Access these devices by designating the device and device range in the access destination. |
| Output | | Y | ○ | | Y＊ | $9D_H$ | 000000 to 001FFF | 000000 to 0007FF | | ○ | |
| Internal relay | | M | ○ | | M＊ | $90_H$ | 000000 to 008191 | 000000 to 008191 | ○ | | |
| Latch relay | | L | ○ | | L＊ | $92_H$ | 000000 to 008191 | 000000 to 008191 | ○ | | |
| Step relay | | S | ○ | | S＊ | $98_H$ | 000000 to 008191 | 000000 to 008191 | ○ | | |
| Link relay | | B | ○ | | B＊ | $A0_H$ | 000000 to 001FFF | 000000 to 000FFF | | ○ | |
| Annunciator | | F | ○ | | F＊ | $93_H$ | 000000 to 002047 | 000000 to 002047 | ○ | | |
| Special relay | | M | ○ | | M＊ | $90_H$ | 009000 to 009255 | 009000 to 009255 | ○ | | |
| Timer T | Contact | | ○ | | TS | $C1_H$ | 000000 to 002047 | 000000 to 002047 | ○ | | |
| | Coil | | ○ | | TC | $C0_H$ | 000000 to 002047 | 000000 to 002047 | ○ | | |
| | Current value | | | ○ | TN | $C2_H$ | 000000 to 002047 | 000000 to 002047 | ○ | | |
| Counter C | Contact | | ○ | | CS | $C4_H$ | 000000 to 001023 | 000000 to 001023 | ○ | | |
| | Coil | | ○ | | CC | $C3_H$ | 000000 to 001023 | 000000 to 001023 | ○ | | |
| | Current value | | | ○ | CN | $C5_H$ | 000000 to 001023 | 000000 to 001023 | ○ | | |
| Data register | | D | | ○ | D＊ | $A8_H$ | 000000 to 008191 | 000000 to 008191 | ○ | | |
| Link register | | W | | ○ | W＊ | $B4_H$ | 000000 to 001FFF | 000000 to 000FFF | | ○ | |
| File register ＊8 | | R | | ○ | R＊ | $AF_H$ | 000000 to 008191 | 000000 to 008191 | ○ | | For normal access by block switching. |
| | | | | ○ | ZR | $B0_H$ | 000000 to 07FFFF | 000000 to 07FFFF | | ○ | For serial No. access |
| Special register | | D | | ○ | D＊ | $A8_H$ | 009000 to 009255 | 009000 to 009255 | ○ | | – |

＊7 Access is to device memory inside the designated CPU.
Take the following precautions when accessing a PLC CPU other than a Q/QnACPU.

1) Access the PLC CPU within the device number range that can be used by the access destination PLC CPU.
2) When accessing a PLC CPU, except the external device connecting station's Q/QnACPU and a Q/QnACPU that is accessed over MELSECNET/H, MELSECNET/10, in word units, always make sure the bit device number is a multiple of 16 (for decimal, 0, 16....).
   Special relays M beginning from M9000 can be designated by (9000 + multiple of 16).
3) When accessing the PLC CPU via the QnA compatible 3C/4C frame, the M, L, and S ranges are designated. However, if the number range of M is designated by L and S or vise versa, they are processed identically.
   When accessing the PLC CPU via the QnA compatible 3E frame, designate M by L and S.
4) Special relays (M9000 to M9255) and special registers (D9000 to D9255) are divided into read only, write only and system use.
   Writing data to outside the write enable range may result in a PLC CPU error.
   See the ACPU Programming Manual for a detailed description of the special relays and special registers.

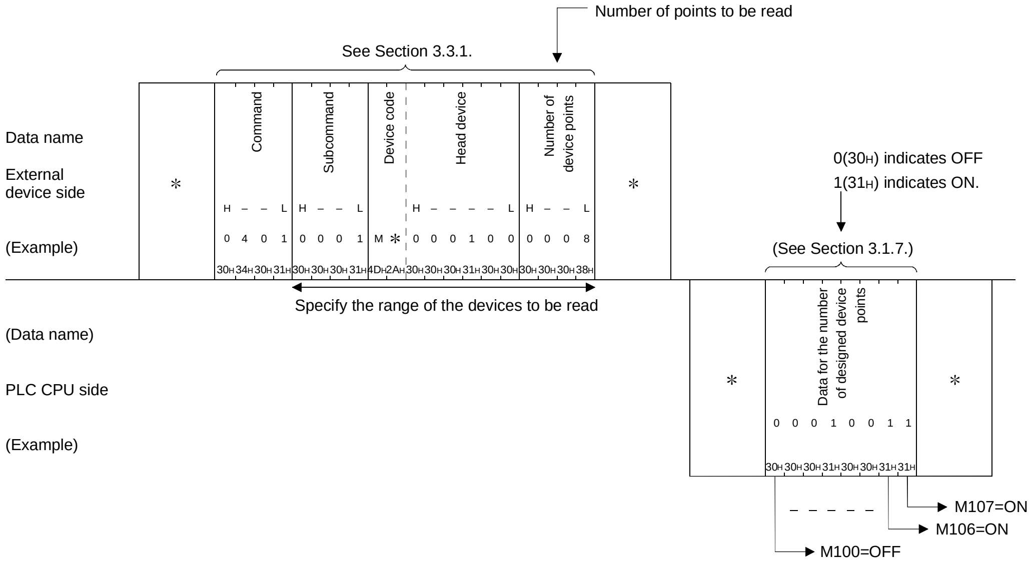
＊8 See(a) ＊4.

### 3.3.2 Batch read in bit units (command: 0401)

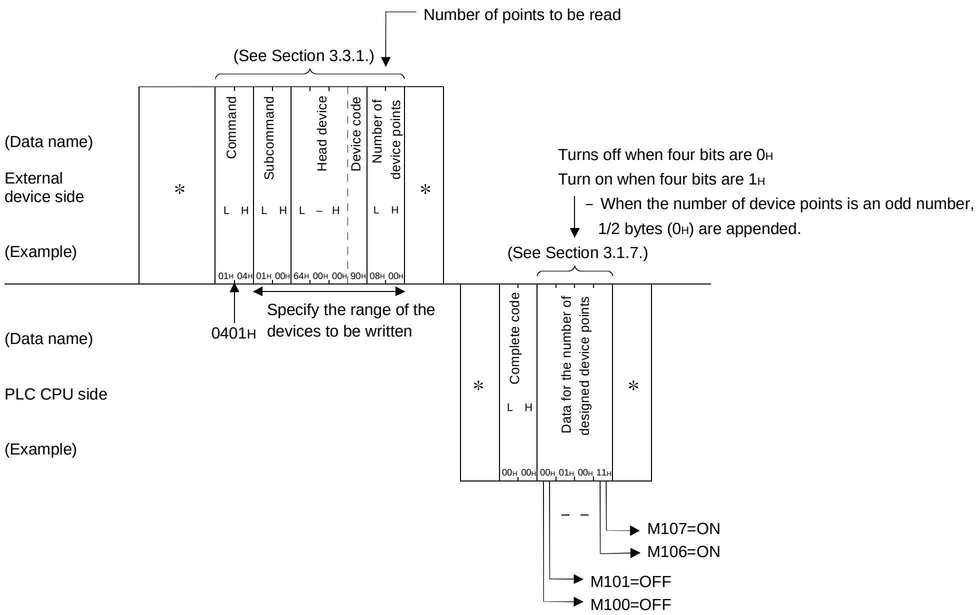
The examples shown in this section explain the control procedure of a batch read command executed on the bit device memory.
The order and content of data items of the areas marked by "*" shown in the control procedure diagram differ depending on the module used as well as the frame and format used for communication.
See the detailed information described in Section 3.1.

[Control procedure]

(1) Reading eight points of internal relays M100 to 107 while communicating in ASCII code



(2) Reading eight points from internal relays M100 to M107 while communicating in binary code




**POINT**

(1) Designate the number of device points within the following range:

(a) When accessing a QCPU of a station on which the Q series C24/E71 is installed (local station) or a QCPU (other station) via a network system (MELSECNET/H, MELSECNET/10 and Ethernet) supporting the Q series

Number of device points : 1 ≤ number of device points ≤ 3584, 7168 or 7904

1) Via the Q series E71 : Communication in ASCII code : 3584 points
Communication in binary code : 7168 points

2) Via the Q series C24 : 7904 points

(b) When accessing a QnACPU (other station) or a Q/QnACPU (other station) via a network system (MELSECNET/10 and Ethernet) supporting the QnA series

Number of device points : 1 ≤ number of device points ≤ 1792, 3584 or 3952

1) Via the Q series E71 : Communication in ASCII code : 1792 points
Communication in binary code : 3584 points

2) Via the Q series C24 : 3952 points

(c) When accessing a PLC CPU other than the above (other station)

Number of device points : 1 ≤ number of device points ≤ 256

(2) Set the access range as follows

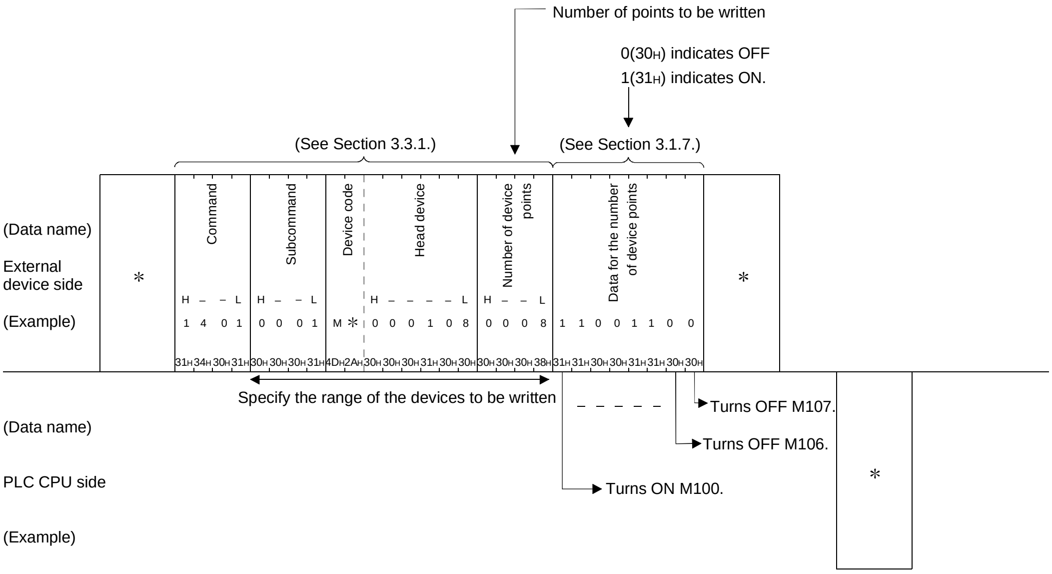
Access range : (Head device number + number of device points - 1) ≤ maximum device number

### 3.3.3 Batch write in bit units (command: 1401)

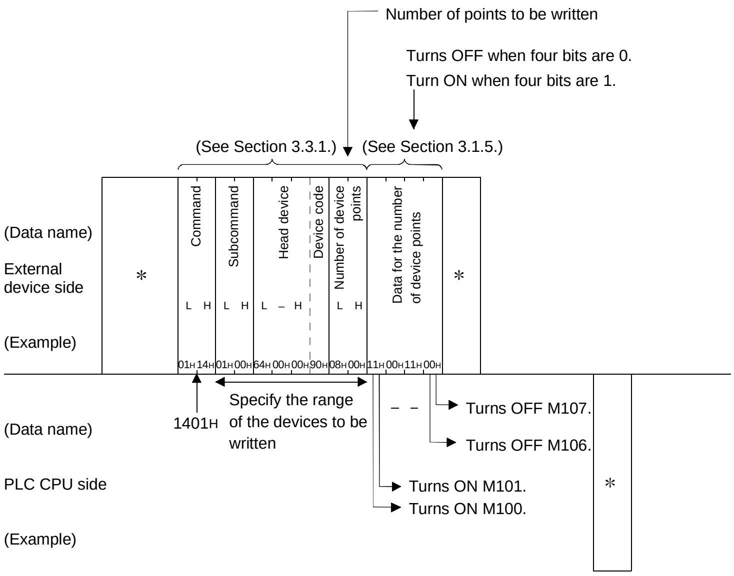
The examples shown in this section explain the control procedure of a batch write command executed on the bit device memory.
The order and content of data items of the areas marked by "∗" shown in the control procedure diagram differ depending on the module used as well as the frame and format used for communication.
See the detailed information described in Section 3.1.

[Control procedure]

(1) Writing eight points of internal relays M100 to 107 while communicating in ASCII code



(2) Writing eight points from internal relays M100 to M107 while communicating in binary code




| POINT |
|---|

(1) Designate the number of device points within the following range:

(a) When accessing a QCPU of a station on which the Q series C24/E71 is installed (local station) or a QCPU (other station) via a network system (MELSECNET/H, MELSECNET/10 and Ethernet) supporting the Q series

Number of device points : 1 ≤ number of device points ≤ 3584, 7168 or 7904

1) Via the Q series E71 : Communication in ASCII code : 3584 points
Communication in binary code : 7168 points

2) Via the Q series C24 : 7904 points

(b) When accessing a QnACPU (other station) or a Q/QnACPU (other station) via a network system (MELSECNET/10 and Ethernet) supporting the QnA series

Number of device points : 1 ≤ number of device points ≤ 1792, 3584 or 3952

1) Via the Q series E71 : Communication in ASCII code : 1792 points
Communication in binary code : 3584 points

2) Via the Q series C24 : 3952 points

(c) When accessing a PLC CPU other than the above (other station)

Number of device points : 1 ≤ number of device points ≤ 160

(2) Set the access range as follows

Access range : (Head device number + number of device points - 1) ≤ maximum device number

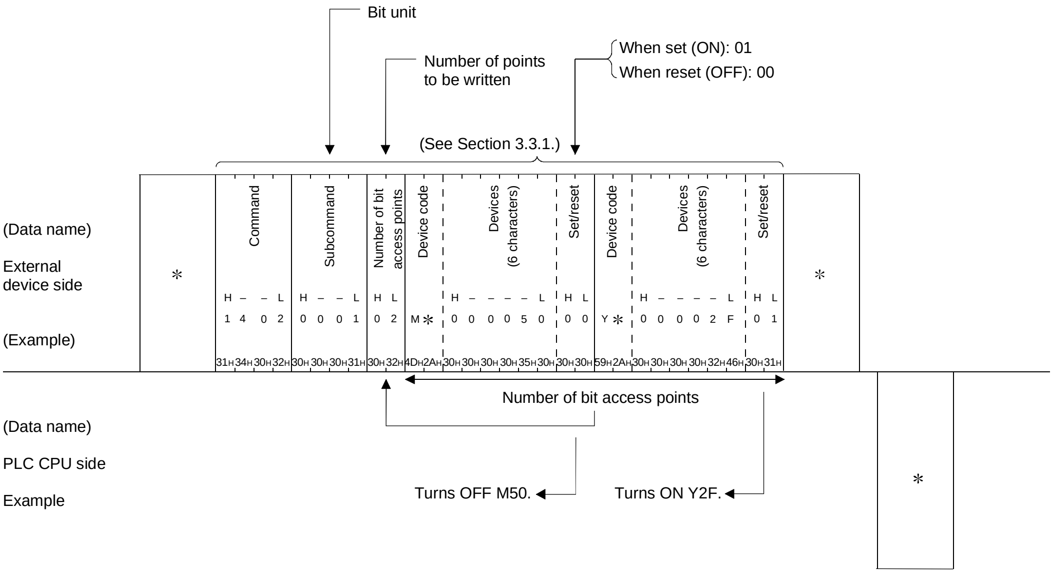
(3) If there is system protection on a Q/QnACPU when attempting to write to it, an error occurs and an error completion code is returned.

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

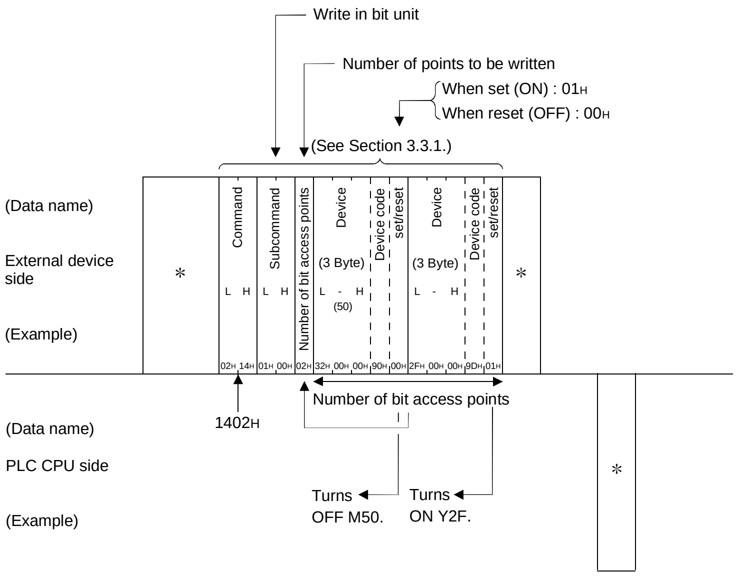
The examples shown in this section explain the control procedure for writing by designating bit device memories at random.
The order and content of data items of the areas marked by "✻" shown in the control procedure diagram differ depending on the module used as well as frame and format used for communication.
See the detailed information described in Section 3.1.

[Control procedure]

(1) Turning OFF internal relay M50 and turning ON output relay Y2F while communicating in ASCII code



(2) Turning OFF internal relay M50 and turning ON output relay Y2F while communicating in binary code




**POINT**

(1) Designate the number of access points within the following range:

(a) When accessing a QCPU of a station on which the Q series C24/E71 is installed (local station) or a QCPU (other station) via a network system (MELSECNET/H, MELSECNET/10 and Ethernet) supporting the Q series

Number of device points : 1 ≤ number of device points ≤ 188

(b) When accessing a QnACPU (other station) or a Q/QnACPU (other station) via a network system (MELSECNET/10 and Ethernet) supporting the QnA series

Number of device points : 1 ≤ number of device points ≤ 94

(c) When accessing a PLC CPU other than the above (other station)

Number of device points : 1 ≤ number of device points ≤ 20

(2) If there is system protection on a Q/QnACPU when attempting to write to it, an error occurs and an error completion code is returned.

### 3.3.5 Batch read in word units (command: 0401)

The examples shown in this section explain the control procedure of batch read commands executed on the bit device memory (16bit units) and word device memory (word units)

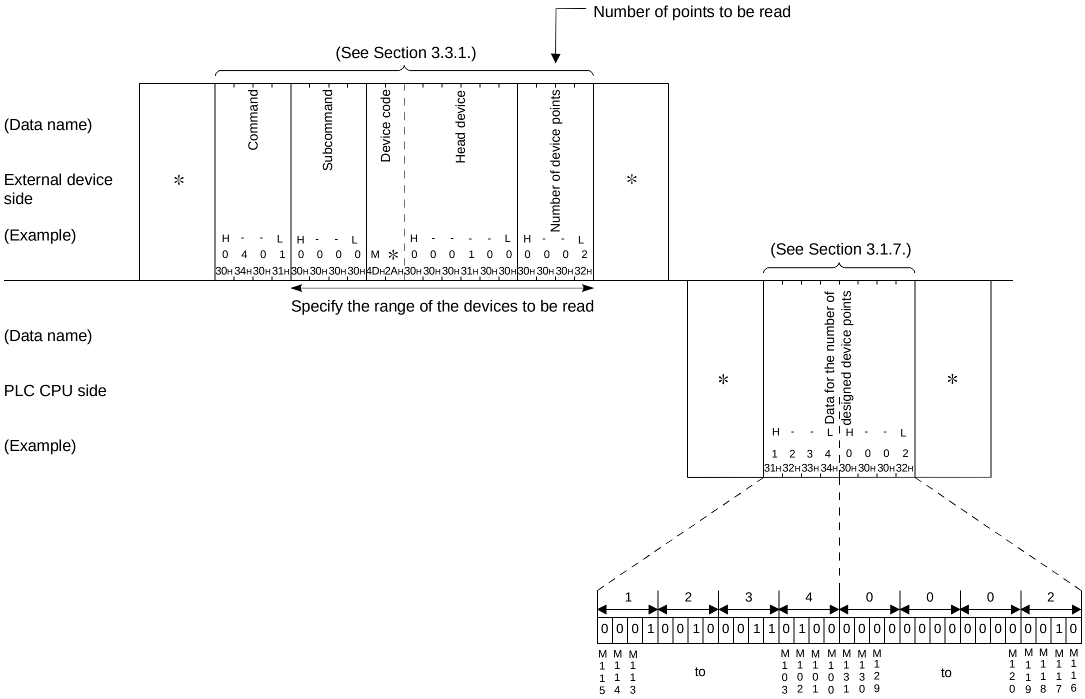
The order and content of data items of the areas marked by "✻" shown in the control procedure diagram differ depending on the module used as well as frame and format used for communication.

See the detailed information described in Section 3.1.

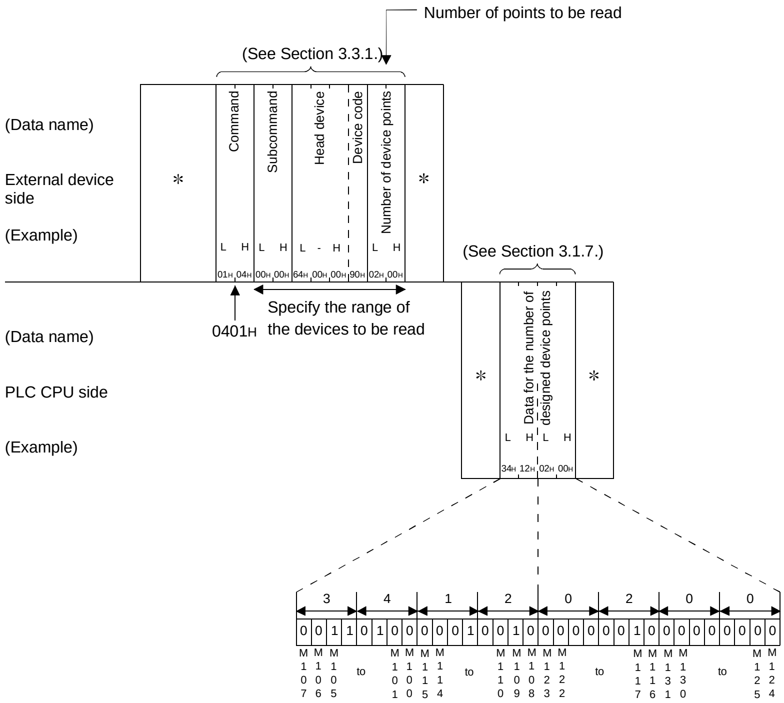
[Control procedure]

#### (1) Reading bit device memory

(a) Reading the two points (32 bits) from internal relays M100 to M131 while communicating in ASCII code



(b) Reading the two points (32 bits) from internal relays M100 to M131 while communicating in binary code




| POINT |
|---|

(1) Designate the number of device points within the following range (for bit devices, one point equals to 16 bits).

(a) When accessing the QCPU of a station on which the Q series C24/E71 is installed (local station) or a QCPU (other station) via a network system (MELSECNET/H, MELSECNET/10 and Ethernet) supporting the Q series

Number of device points : 1 ≤ number of device points ≤ 960

(b) When accessing a QnACPU (other station) or a Q/QnACPU (other station) via a network system (MELSECNET/10 and Ethernet) supporting the QnA series

Number of device points : 1 ≤ number of device points ≤ 480

(c) When accessing a PLC CPU other than the above (other stations)

Number of device points : 1 ≤ number of device points ≤ 32

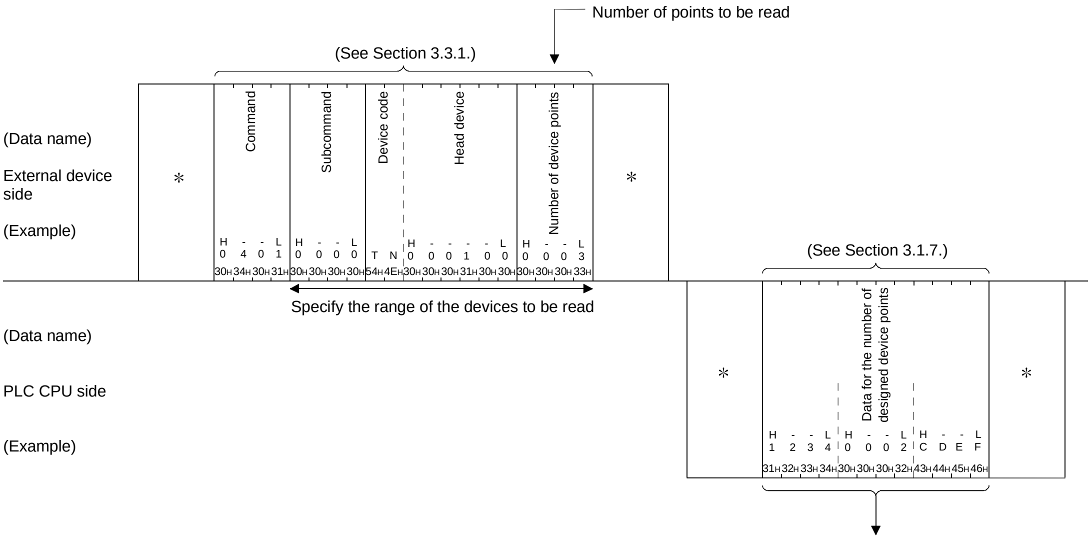
(2) Set the access range as follows

Access range" : (Head device number + number of device points × 16 - 1) ≤ maximum device number

✻ Make sure the head device number of the PLC CPU in (c) above is a multiple of 16.

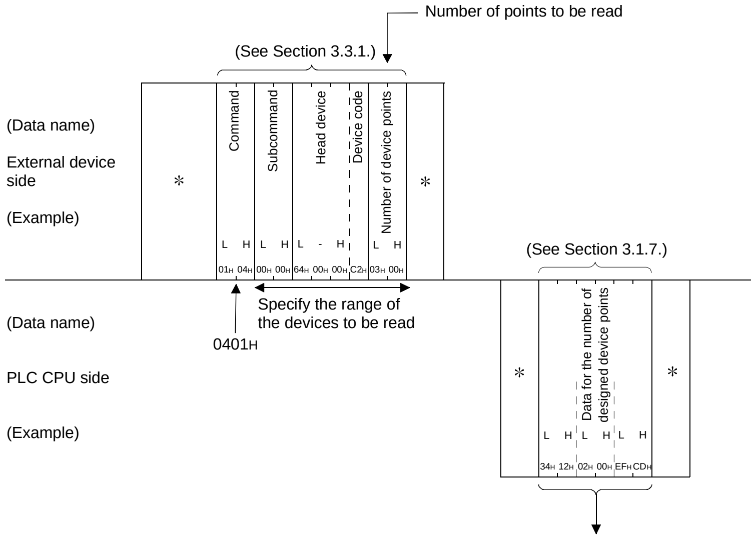
## (2) Reading a word device memory

### (a) Reading the current values of the three points from timers T100 to T102 while communicating in ASCII code




Indicate the current value of T100, 1234H hexadecimal (4660 decimal)
Indicate the current value of T101, 0002H hexadecimal (2 decimal)
Indicate the current value of T102, CDEFH hexadecimal (-12817 decimal)

### (b) Reading the current values of the three points from timers T100 to T102 while communicating in binary code




Indicate the current value of T100, 1234H hexadecimal (4660 decimal)
Indicate the current value of T101, 0002H hexadecimal (2 decimal)
Indicate the current value of T102, CDEFH hexadecimal (-12817 decimal)

**POINT**

(1) Designate the number of device points within the following range (for bit devices, one point equals to 16 bits).

(a) When accessing the QCPU of a station on which the Q series C24/E71 is installed (local station) or a QCPU (other station) via a network system (MELSECNET/H, MELSECNET/10 and Ethernet) supporting the Q series

Number of device points : 1 ≤ number of device points ≤ 960

(b) When accessing a QnACPU (other station) or a Q/QnACPU (other station) via a network system (MELSECNET/10 and Ethernet) supporting the QnA series

Number of device points : 1 ≤ number of device points ≤ 480

(c) When accessing a PLC CPU other than the above (other stations)

Number of device points : 1 ≤ number of device points ≤ 64

(2) Set the access range as follows

Access range" : (Head device number + number of device points - 1) ≤ maximum device number

### 3.3.6 Batch write in word units (command: 1401)

The examples shown in this section explain the control procedure of batch write commands executed on the bit device memory (16bit units) and word device memory (word units)

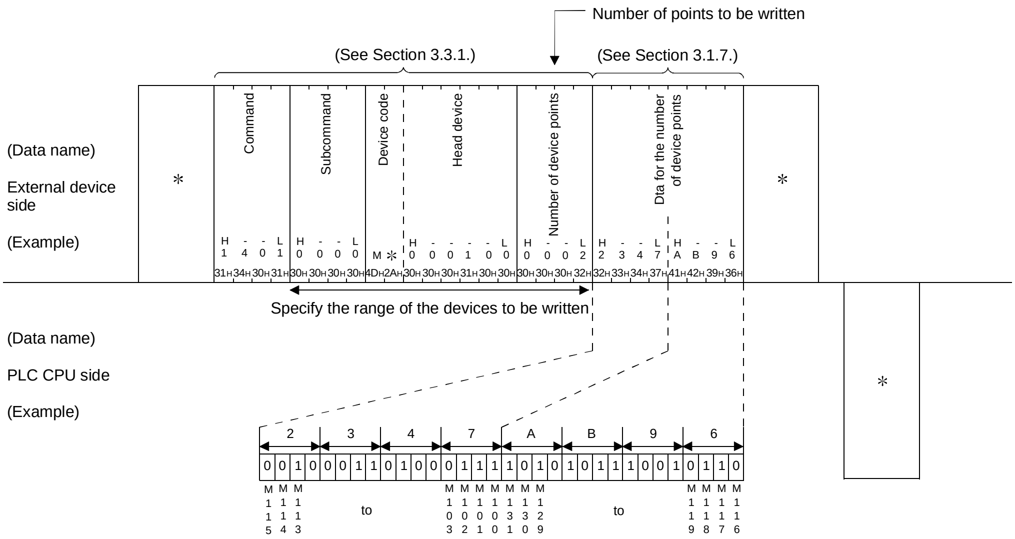
The order and content of data items of the areas marked by "✻" shown in the control procedure diagram differ depending on the module used as well as frame and format used for communication.

See the detailed information described in Section 3.1.

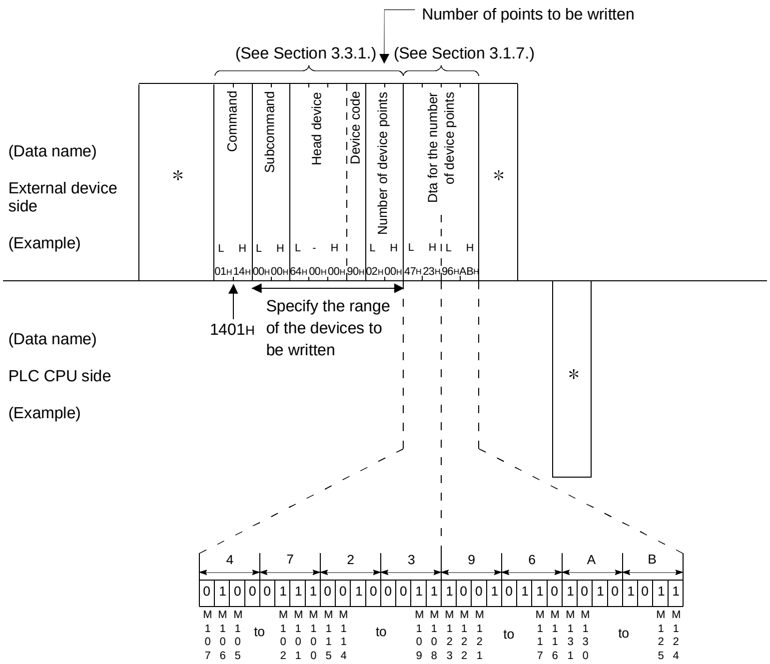
[Control procedure]

(1) Writing to a bit device memory

(a) Writing two points (32 bits) from internal relays M100 to M131 while communicating in ASCII code



(b) Writing two points (32 bits) from internal relays M100 to M131 while communicating in binary code




| POINT |
|---|

(1) Designate the number of device points within the following range. For bit devices, one point equals to 16 bits.

(a) When accessing the QCPU of a station on which the Q series C24/E71 is loaded (local station) or a QCPU (other station) via a network system (MELSECNET/H, MELSECNET/10 and Ethernet) supporting the Q series

Number of access points : 1 ≤ number of device points ≤ 960

(b) When accessing a QnACPU (other station) or a Q/QnACPU (other station) via a network system (MELSECNET/10 and Ethernet) supporting the QnA series

Number of access points : 1 ≤ number of device points ≤ 480

(c) When accessing a PLC CPU other than the above (other stations)

Number of access points : 1 ≤ number of device points ≤ 10

(2) Set the access range as follows

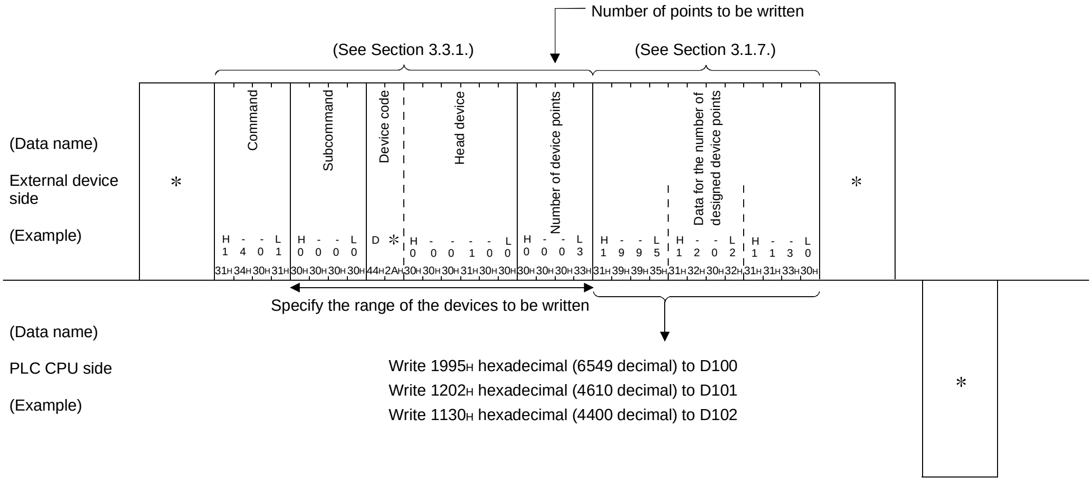
Access range" : (Head device number + number of device points × 16 - 1) ≤ maximum device number

(3) When accessing a bit device of a PLC CPU other than (c) above, make sure the head device number is a multiple of 16 (for decimal, 0,16...).

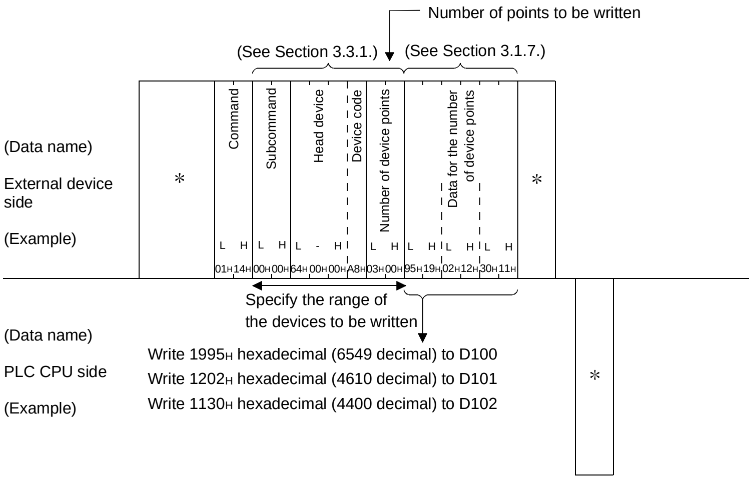
(4) If there is system protection on a Q/QnACPU when attempting to write to it, an error occurs and an error completion code is returned.

(2) Writing to a word device memory

(a) Writing three points to D100 to D102 while communication in ASCII code




(b) Writing three points to D100 to D102 while communicating in binary code



POINT

(1) Designate the number of device points within the following range.

(a) When accessing the QCPU of a station on which the Q series C24/E71 is installed (local station) or a QCPU (other station) via a network system (MELSECNET/H, MELSECNET/10 and Ethernet) supporting the Q series

Number of access points : 1 ≤ number of device points ≤ 960

(b) When accessing a QnACPU (other station) or a Q/QnACPU (other station) via a network system (MELSECNET/10 and Ethernet) supporting the QnA series

Number of access points : 1 ≤ number of device points ≤ 480

(c) When accessing a PLC CPU other than the above (other stations)

Number of access points : 1 ≤ number of device points ≤ 64

(2) Set the access range as follows

Access range" : (Head device number + number of device points × - 1) ≤ maximum device number

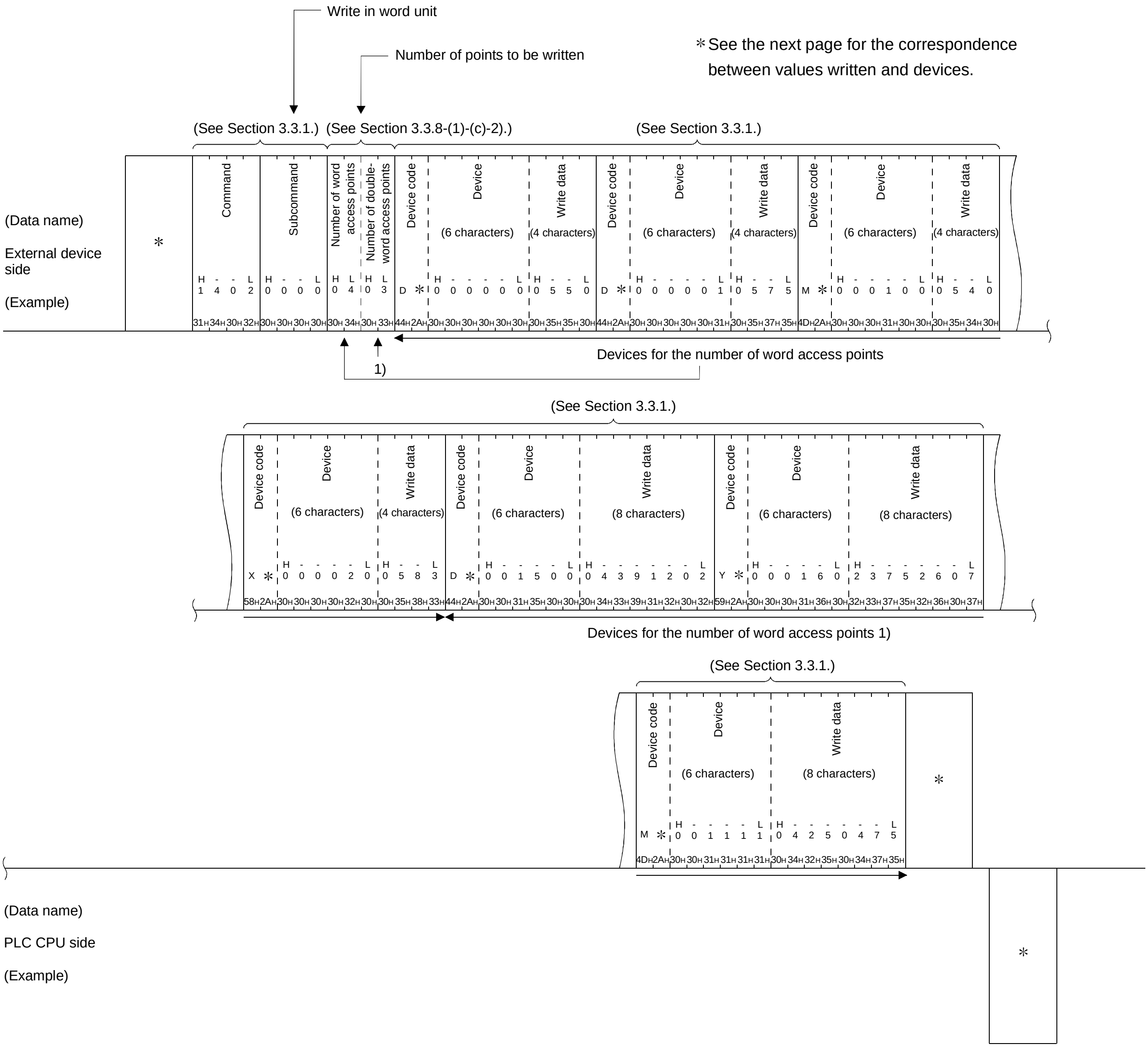
(3) If there is system protection on a Q/QnACPU when attempting to write to it, an error occurs and an error completion code is returned.

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

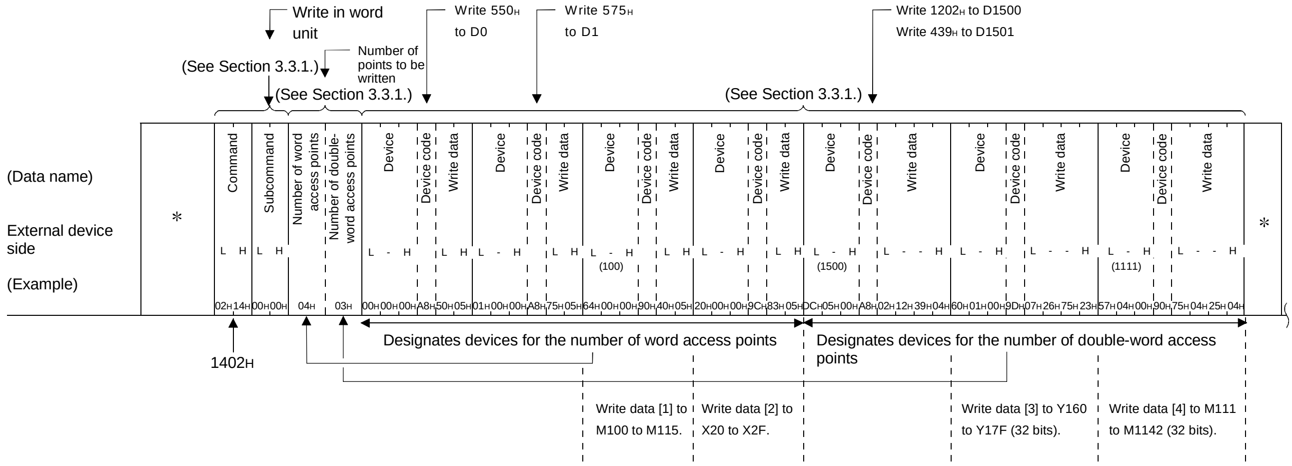
The examples shown in this section explain the control procedure of writing data by randomly designating bit device memories (16/32 bit units) and word device memories (1/2 word units).
The order and content of data items of the areas marked by "✻" shown in the control procedure diagram differ depending on the module used as well as frame and format used for communication.
See the detailed information described in Section 3.1.

[Control procedure]

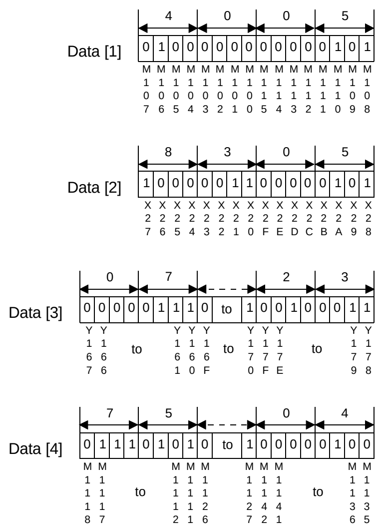
(a) Writing to the following device memories while communicating in ASCII code
- Word access : D0, D1 M100 to M115, X20 to X2F
- Double word access : D1500 to D1501, Y160 to Y17F, M1111 to M1142



(b) Writing to the following device memories while communicating in binary code
- Word access : D0, D1, M100 to M115 X20 to X2F
- Double word access : D1500 to D1501, Y160 to Y17F, M1111 to M1142







**POINT**

(1) Designate the number of device points within the following ranges.
For a bit device, one word access point is 16 bits and one double word access point is 32 bits.

(a) When accessing the QCPU of a station on which the Q series C24/E71 is installed (local station) or a QCPU (other station) accessed via a network system (MELSECNET/H, MELSECNET/10 and Ethernet) supporting the Q series

Number of access points : $1 \leq$ (number of word access points $\times$ 12 + number of double word access points $\times$ 14) $\leq$ 1920

(b) When accessing a QnACPU (other station) or a Q/QnACPU (other station) via a network system (MELSECNET/10 and Ethernet) supporting the QnA series

Number of access points : $1 \leq$ (number of word access points $\times$ 12 + number of double word access points $\times$ 14) $\leq$ 960

(c) When accessing a PLC CPU other than the above (other stations)

Number of access points : $1 \leq$ number of word access points $\leq$ 10

(2) When accessing bit devices of a PLC CPU other than (c) above, make sure the head device number is a multiple of 16 (for decimal, 0,16...).

(3) If there is system protection on a Q/QnACPU when attempting to write to it, an error occurs and an error completion code is returned.

### 3.3.8 Random read in word units (command: 0403)

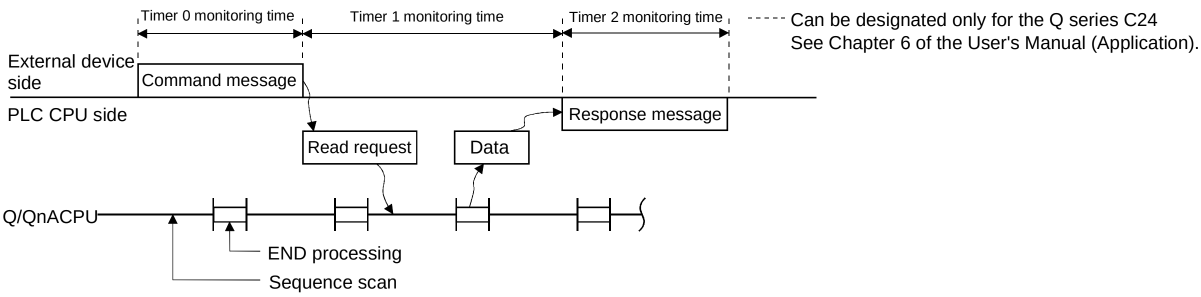
The examples shown in this section explain the control procedure of reading data by randomly designating bit device memories (16/32 bit units) and word device memories (1/2 word units).
In the random read in word units command, the following conditions (hereafter called monitor conditions) concerning when to read data, can be designated. (Combined designation is also possible.)

- Read the device memory when the designated device is turned ON or OFF during the PLC CPU END processing.
- Read the device memory when the value of the designated device memory reaches the monitor condition value during the PLC CPU END processing. (A mask value can also be designated as the monitor condition.)
- Read the device memory when a designated step of a designated file is executed during the PLC CPU END processing. (A block number and step number of MELSAP3 can also be designated.)

The following shows the device memory read timing by device condition designation.

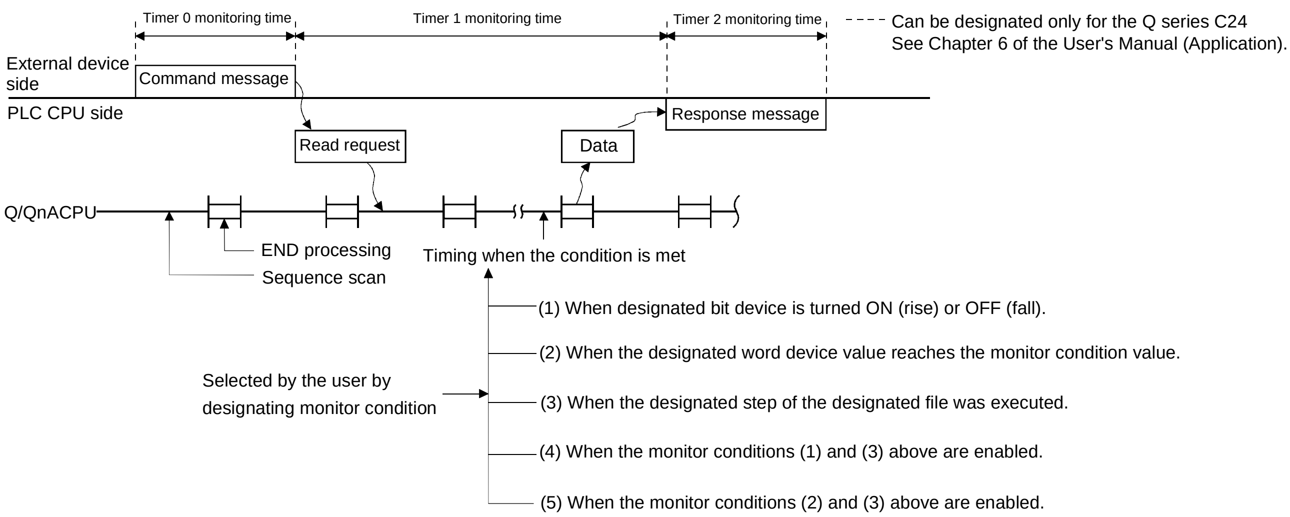
When monitor condition is not designated

The device memory is read during the END processing when the Q series C24/E71 issued a read request to the PLC CPU.




When monitor condition is designated

The device memory is read during the END processing when the monitor condition designated by the user is satisfied after the Q series C24/E71 issued a read request to the PLC CPU.



**POINT**

An intelligent function module and GX Developer cannot monitor the same Q/QnACPU device memory using monitor conditions at the same time.
If the Q series C24/E71 receives one of the command messages shown below from an opposite device, and the same Q/QnACPU is being conditionally monitored simultaneously from an intelligent functional module and a GX Developer, the Q series C24/E71 will send an abnormal complete code to the opposite device.
(When unconditional monitoring is being performed, the Q series C24/E71 can perform conditional/unconditional monitoring.)

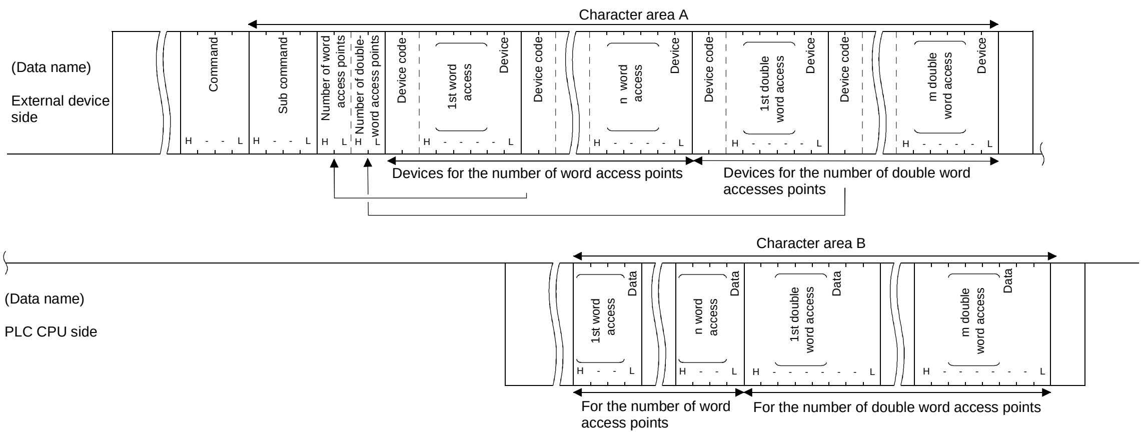
| Command | Function | Reference section |
|---|---|---|
| 0403 | Random read in word units function | This section |
| 0802 | Registered device memory monitor function | Section 3.3.9 (4) |

## (1) Data order and contents of character areas during random read

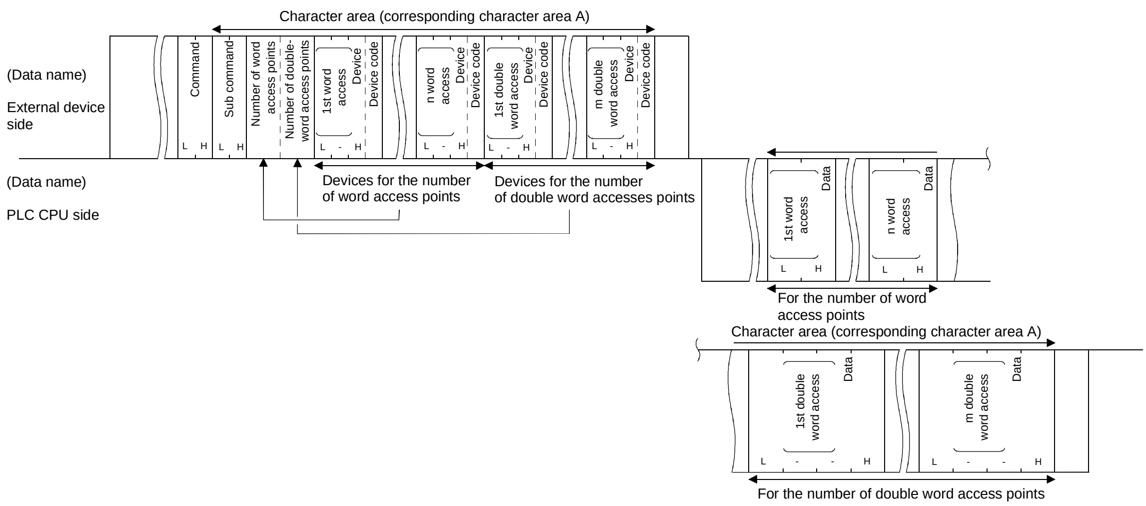
The following section explains the data order and contents of character areas during random read.
The order and contents of the character areas are different from when other commands are used

(a) Data order of character areas when a monitor condition is not designated

1) Data order while communicating in ASCII code

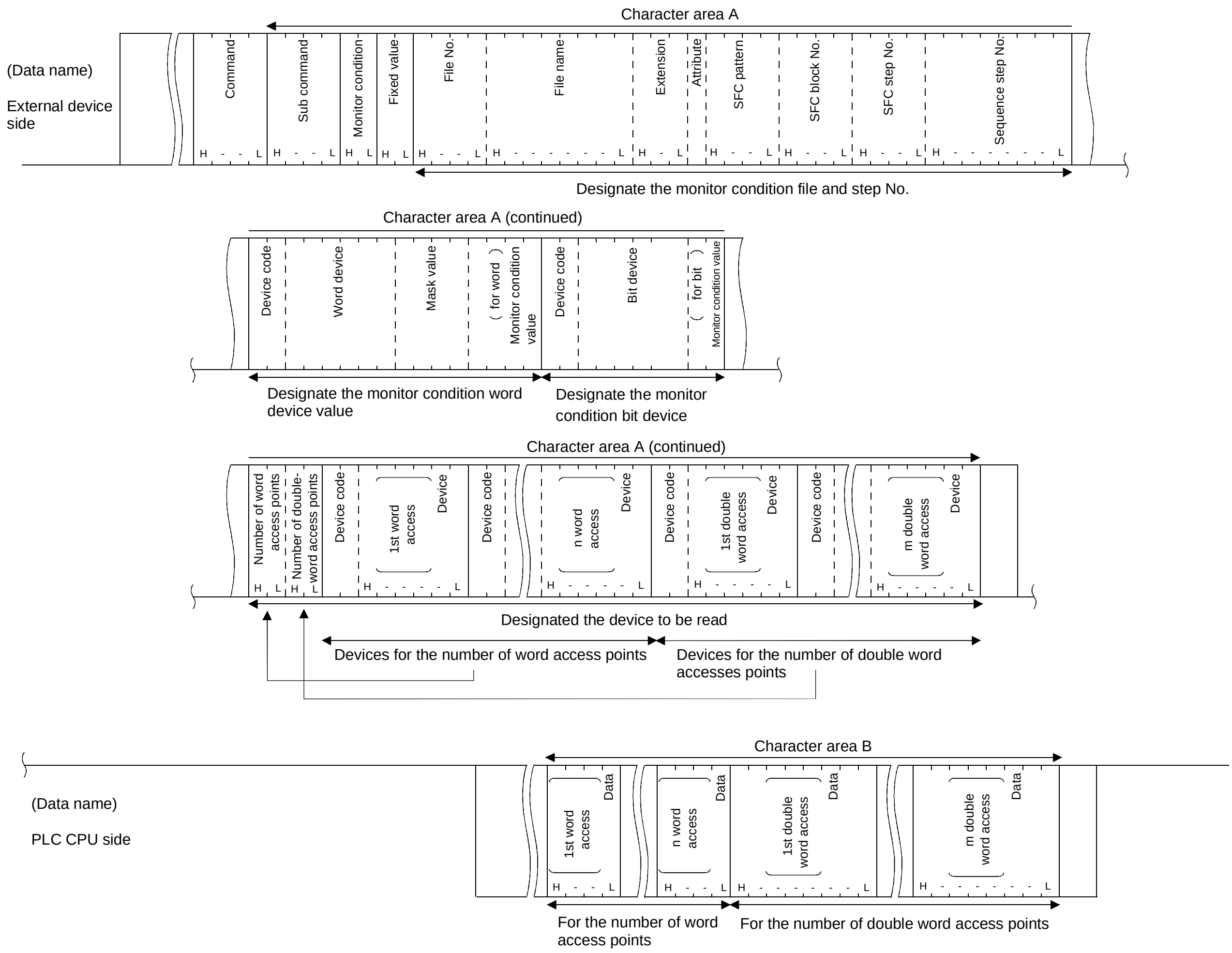



2) Data order while communicating in binary code

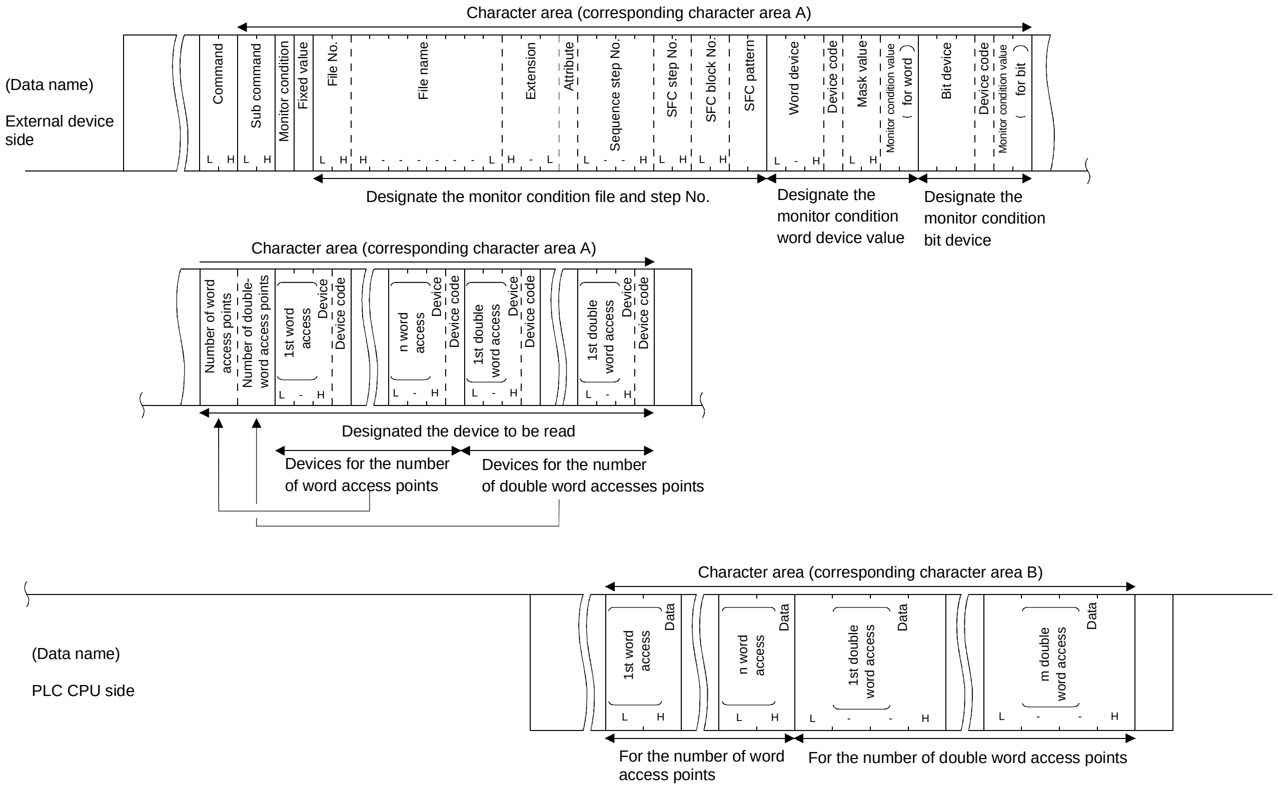


(b) Data order of character areas when monitor conditions are designated

1) Data order while communicating in ASCII code



2) Data order while communicating in binary code




(c) Contents of character areas

The contents of the character areas are the same as the contents when other commands are used, except for the following data:

1) Subcommand

This value designates the read/write units, type of device to be designated, data read conditions (timing), etc.

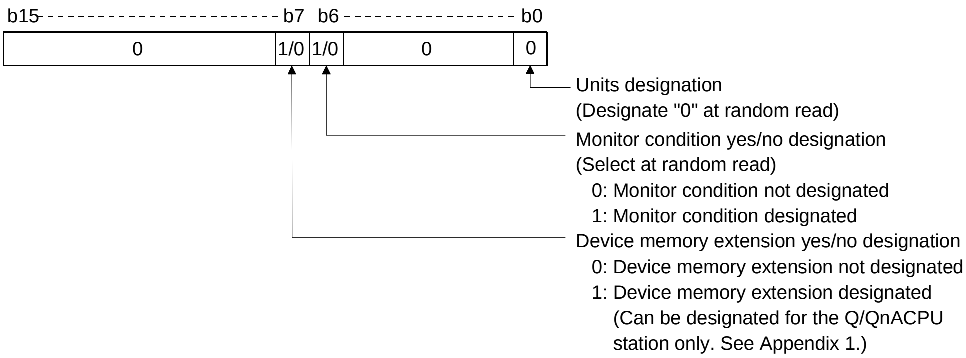
a) Data communication in ASCII code

The value 0000H (0), or the value shown below, is converted to a 4-digit (hexadecimal) ASCII code and transmitted beginning from the most significant digit.

b) Data communication in binary code

The value 0000H, or the 2-byte value shown below, is transmitted.

c) The following is designated for subcommands:




d) When neither monitor conditions nor device memory extensions are designated, the subcommand is "0000H."

REMARK

The following table summarizes the subsequent data that must be designated to the character area depending on whether the subcommand monitor condition is designated or not.

| Condition designation / Data name | When monitor condition is not designated | When monitor condition is designated | | |
|---|---|---|---|---|
| | – | When file and step numbers are designated | When word device value is designated | When bit device is designated |
| Number of word access points<br>Number of double word access points | ● | | | |
| (For designating read device) | – | – | – | – |
| (For word read) Device code<br>Device | ●<br>(Designation is unnecessary when the number of word access points is 0.) | | | |
| (For double word read) | – | – | – | – |
| Device code<br>Device | ●<br>(Designation is unnecessary when the number of double word access points is 0.) | | | |
| Monitor condition<br>Fixed value | × | ● | | |
| File number to Attribute | × | ● | △ | △ |
| SFC pattern to SFC step number | × | △ | △ | △ |
| Sequence step number | × | ● | △ | △ |
| (For designating word device value) | – | – | – | – |
| Device code to Monitor condition value (for word) | × | △ | ● | △ |
| (For designating bit device) | – | – | – | – |
| Device code to Monitor condition value (for bit) | × | △ | △ | ● |

●: Designation necessary △: Selective (When not designated, a default value is designated.) ×: Designation unnecessary

2) Number of word access points and number of double word access points
These data items are for designating the number of points to be read in word units and in double word units.
The total number of points of each should be designated within the limit on the number of points processed per communication shown in Section 3.2 (1).
a) Data communication in ASCII code
Each number of points is converted to a 2-digit (hexadecimal) ASCII code and transmitted beginning from the most significant digit.
(Examples)
5 points : Converted to "05" and transmitted from the "0."
20 points : Converted to "14" and transmitted from the "1."
b) Data communication in binary mode
The 1-byte value representing the number of points is used for transmission.
(Examples)
5 points : $05_H$ is transmitted.
20 points : $14_H$ is transmitted.
c) When either of the number of access points set is 0, the relevant device whose data is read and the device code do not have to be designated.

3) Monitor condition

This value designates the condition under which data is read. It can be designated for the Q/QnACPU only.

The table below summarizes the combinations of monitor conditions that can be designated.

(Three conditions cannot be designated at the same time.)

| Combination \ Monitor condition | File and step number designation | Word device value designation | Bit device designation |
|---|---|---|---|
| Single designation | ○ | ○ | ○ |
| Combined designation | ○ | ○ | |
| | ○ | | ○ |

○: Can be designated

a) Data communication in ASCII code

The value shown below is converted to a 2-digit (hexadecimal) ASCII code and transmitted beginning from the most significant digit.

b) Data communication in binary code

The 1-byte value shown below is transmitted.

c) The following data is designated as monitor conditions.

| b7 - - - - - - b3 | b2 | b1 | b0 |
|---|---|---|---|
| 0 | 1/0 | 1/0 | 1/0 |

- b0: File and step number yes/no designation
  - 0: File and step numbers are not designated
  - 1: File and step numbers are designated
- b1: Word device value yes/no designation
  - 0: Word device value is not designated
  - 1: Word device value is designated
    (Can be selected when bit device is not designated.)
- b2: Bit device yes/no designation
  - 0: Bit device is not designated
  - 1: Bit device is designated
    (Can be selected when word device value is not designated.)

d) When the designated monitor condition was selected by a subcommand, the permitted monitor conditions are "$01_H$" to "$05_H$" "$00_H$" cannot be used.

e) When a monitor condition was selected not to be designated by a subcommand, it is not required to specify the monitor conditions.

4) Fixed value

a) Data communication in ASCII code

The value "0F" is transmitted beginning from the most significant digit ("0").

b) Data communication in binary code

The 1-byte value "$0F_H$" is used for transmission.

c) When a monitor condition was selected not to be designated by a subcommand, it is not required to specify the fixed values.

5) File number

This value designates the registration number when a file designated by the subsequent filename and extension is registered (written) to the PLC CPU. One of the following can be designated.

| File No. | Description | Designation contents |
| --- | --- | --- |
| 0000H | Filename not designated | Designated when file and step numbers are not designated in the monitor conditions. |
| 0001H TO 0100H | File number | Designated when the file number is known. |
| FFFFH | File number unknown | Designated when the file number is retrieved by the Q series C24/E71.<br>(Random read requests from the Q series C24/E71 to the PLC CPU are delayed for one sequence scan time or longer.) |

a) Data communication in ASCII code
The file number shown above is converted to a 4-digit (hexadecimal) ASCII code and transmitted beginning from the most significant digit.
(Example)
When the file number is 1FH: It becomes "001F" and sequentially transmitted beginning from the first "0."

b) Data communication in binary code
The 2-byte value shown above is transmitted beginning from the low byte (L: bits 0 to 7).
(Example)
When the file number is 1FH: It becomes 001FH and 1FH is transmitted first and then 00H.

c) When file and step numbers are not designated (02H and 04H are used) in the monitor conditions, the file number becomes 0000H even if the monitor condition designated is selected by a subcommand.

d) When a monitor condition was selected not to be designated by a subcommand, it is not required to specify the file number.

e) The file number can be checked using the file search function described in Sections 3.8.6, 3.8.16, and 3.8.17.

6) Filename, extension, and attribute
These data items are for designating the target file of the sequence number in 9) below. They designate the filename, extension, and attribute when that file is registered (written) to the PLC CPU.
The initial attribute of a user file is 20H (disk file). This attribute may be changed by the user. (See Sections 3.8.15 and 3.8.24)

a) Data communication in ASCII code
The filename, extension, and attribute used during registration are sequentially transmitted beginning from the first character of each.
Blank space (code: 20H) is transmitted as the attribute.
When the filename is less than 8 characters, blank spaces (code: 20H) are added.
(Example)
The filename during registration was "ABCD12"
"ABCD12 ␣ ␣" is sequentially transmitted from the "A."

b) Data communication in binary code
For the filename and extension, each character code during registration is used as a binary value and is sequentially transmitted beginning from the first character.
The 1-byte value "20H " is transmitted as the attribute.
When the filename is less than 8 characters, "20H " is added.
(Example)
When the filename during registration was "ABCD12"
The filename becomes 41H, 42H, 43H, 44H, 31H, 32H, 20H, 20H and is sequentially transmitted beginning from 41H.

c) When file and step number are not designated (02H and 04H are used) in the monitor conditions, the filename and extension are replaced by a string of blanks or 20H codes even if the monitor condition designated is selected by a subcommand.
The attribute becomes a blank space or 20H code.

d) When a monitor condition was selected not to be designated by a subcommand, it is not required to specify the file name, file extension and attributes.

e) The attributes can be checked using the file search function described in Sections 3.8.5 and 3.8.16.

7) SFC pattern
This is one of the data that determine when to read data during the execution of the designated sequence step number of the MELSAP3 program (hereinafter abbreviated as SFC).

a) When a subcommand selected that monitor conditions should be designated, and the monitor condition of 3) designated are file and step number, the SFC pattern is designated by the data shown below.
Other patterns cannot be designated.

| Code | Write data | | Remarks |
|---|---|---|---|
| ASCII code | When SFC designated | "0003" | 4 characters are sequentially transmitted beginning from the first character ("0"). |
| | When SFC not designated | "0000" | |
| Binary mode | When SFC designated | 0003H | Transmits the 2-byte value shown, beginning from the low byte (L: bits 0 to 7). |
| | When SFC not designated | 0000H | |

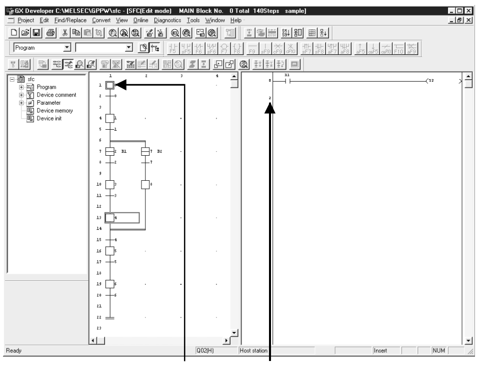
b) When file and step number was selected not to be designated as monitor conditions, the SFC pattern becomes "0000"/"0000H " even if the subcommand selected that monitor conditions should be designated.

c) When the SFC pattern has the value "0003" /0003H, the designated device memory data is read during the END processing of the PLC CPU when the sequence step number designated by 9) and the block number designated by 8) are executed.

d) When a monitor condition was selected not to be designated by a subcommand, it is not required to specify the SFC pattern.

POINT

(1) For a detailed description of MELSAP3 that supports the QCPU, refer to the Programming Manual and Operating Manual for MELSAP3
(2) The SFC step number designated in 8) and the sequence step number designated in 9) correspond to the following displayed parts when the MELSAP3 program is edited.



Step No. Sequence step No.

8) SFC block number and SFC step number

These data items designate the SFC block number and SFC step number containing the sequence steps that will determine when to read data (when the designated sequence step is being executed).
They can be designated when a subcommand selected that monitor conditions should be designated, the monitor condition control designated are file and step numbers, and the SFC pattern designation is SFC block.

a) Data communication in ASCII code
The following values are converted to a 4-digit (hexadecimal) ASCII code and sequentially transmitted beginning from the most significant digit ("0").
- SFC block No. : 0000H to 013FH (0 to 319)
- SFC step No. : 0000H to 01FFH (0 to 511)

b) The 2-byte values shown above are sequentially transmitted beginning from the low byte (L: bits 0 to 7).
(Example)
In case of 0005H: Transmitted from 05H and then 00H.

c) In the following cases, both the SFC block number and SFC step number become "0000H."
- When file and step number was selected not to be designated, even when monitor conditions were selected to be designated by the subcommand.
- When the SFC block number was selected not to be designated by the SFC pattern.

d) When a monitor condition was selected not to be designated by a subcommand, it is not required to specify the SFC block number and SFC step number.

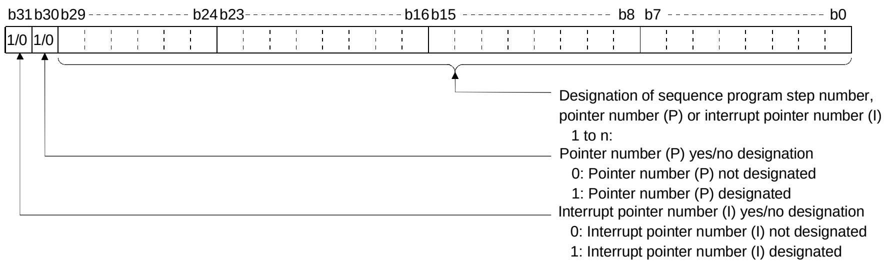
9) Sequence step number

This value designates the step number and pointer number (P) or interrupt pointer number (I) of the PLC CPU program that will determine when to read data (when the designated sequence step is being executed).
The sequence step number can be designated when a subcommand selected that monitor conditions should be designated, and the monitor conditions designated are file and step number.

a) Data communication in ASCII code
The values shown below that are within the range of values found in the objective file is converted to an 8-digit (hexadecimal) ASCII code and sequentially transmitted beginning from the most significant digit.

b) Data communication in binary code
The 4-byte values below that are within the range of values found in the objective file is sequentially transmitted beginning from the low byte (L: bits 0 to 7).
(Example)
In case of 00000005H: Sequentially transmitted from 05H.

c) The following is designated for the sequence step number.



(Example)

0000001C$_H$ when sequence program number 28 is designated.
8000001C$_H$ when the interrupt pointer I28 is designated.

d) When file and step numbers are not designated in the monitor conditions, the sequence step number becomes 00000000$_H$ even if the monitor condition designated is selected by a subcommand.

e) When a monitor condition was selected not to be designated by a subcommand, it is not required to specify the sequence step number.

10) Mask value and monitor condition value (for word)

These data items designate the value for word devices, etc. which will determine when to read data (when a word device is designated as a monitor condition).

- Mask value
  The mask value extracts values in an arbitrary range of bits from a word device used as a monitor condition.
  (Its function is identical to the "WAND" instruction of the sequence program.)
- Monitor condition value (for word)
  This data item designates the numerical value (the result of the logical operation between the monitor condition word device value and the mask value (identical to the "WAND" instruction)) to be used for the timing of reading data.
  (Example)
  The following values are designated when the timing of reading data is set to the condition where the result of extraction from bits 0 to 14 of D0 for monitor condition is 3E8$_H$ (1000).
  Mask value : 7FFF$_H$
  Monitor condition value (for word) : 03E8$_H$

A mask value and monitor condition value (for words) can be designated when monitor conditions are selected to be designated by a subcommand and the word device value is designated as monitor condition.

a) Data communication in ASCII code
The mask value and monitor condition value above are converted to a 4-digit (hexadecimal) ASCII code and sequentially transmitted beginning from the most significant digit.

b) Data communication in binary mode
Two-byte values representing the mask value and monitor condition value are sequentially transmitted beginning from the low byte (L: bits 0 to 7).
(Example)
0005$_H$: Sequentially transmitted from 05$_H$.

c) When the word device value is selected not to be designated as a monitor condition, the mask value and monitor condition value are replaced by blanks or "0000$_H$ " even if monitor conditions are selected to be designated by the subcommand.
In this case, the word device and device code for designating the word device value for a monitor condition designate an arbitrary word device memory and its device code.

d) When a monitor condition was selected not to be designated by a subcommand, it is not required to specify the mask value and monitor condition value.
(The word device and device code for designating the word device value for monitor condition do not have to be designated either.)

11) Monitor condition value (for bits)

The monitor condition value (for bits) is used for designating whether or not to read data (rising or falling).

It can be designated when a subcommand selected that monitor conditions should be designated and the monitor condition designated is a bit device.

a) Data communication in ASCII code

The value shown below is converted to a 2-digit (hexadecimal) ASCII code and sequentially transmitted beginning from the most significant digit ("0").

b) Data communication in binary mode

The 1-byte value shown below is transmitted.

c) The monitor condition value (for bits) can be designated as the following.

| Designated value | Timing to read |
|---|---|
| $02_H$ | During PLC CPU END processing when the designated bit device is raised |
| $04_H$ | During PLC CPU END processing when the designated bit device is dropped |

d) When a bit device not is selected to be designated as a monitor condition, the monitor condition value is "00" or "$00_H$" even if the subcommand selected that monitor conditions should be designated.

In this case, the bit device and device code for designating the monitor condition bit device are designated as an arbitrary bit device memory and its device code.

e) When a monitor condition was selected not to be designated by a subcommand, it is not required to specify the monitor conditions. (The bit device code for designating a bit device for monitor condition does not have to be designated either.)

## (2) Random read in word units (when monitor condition is not designated)

The examples shown in this section explain the control procedure of reading data by randomly designating bit device memories (16bit units) and word device memories (word units) without designating a monitor condition

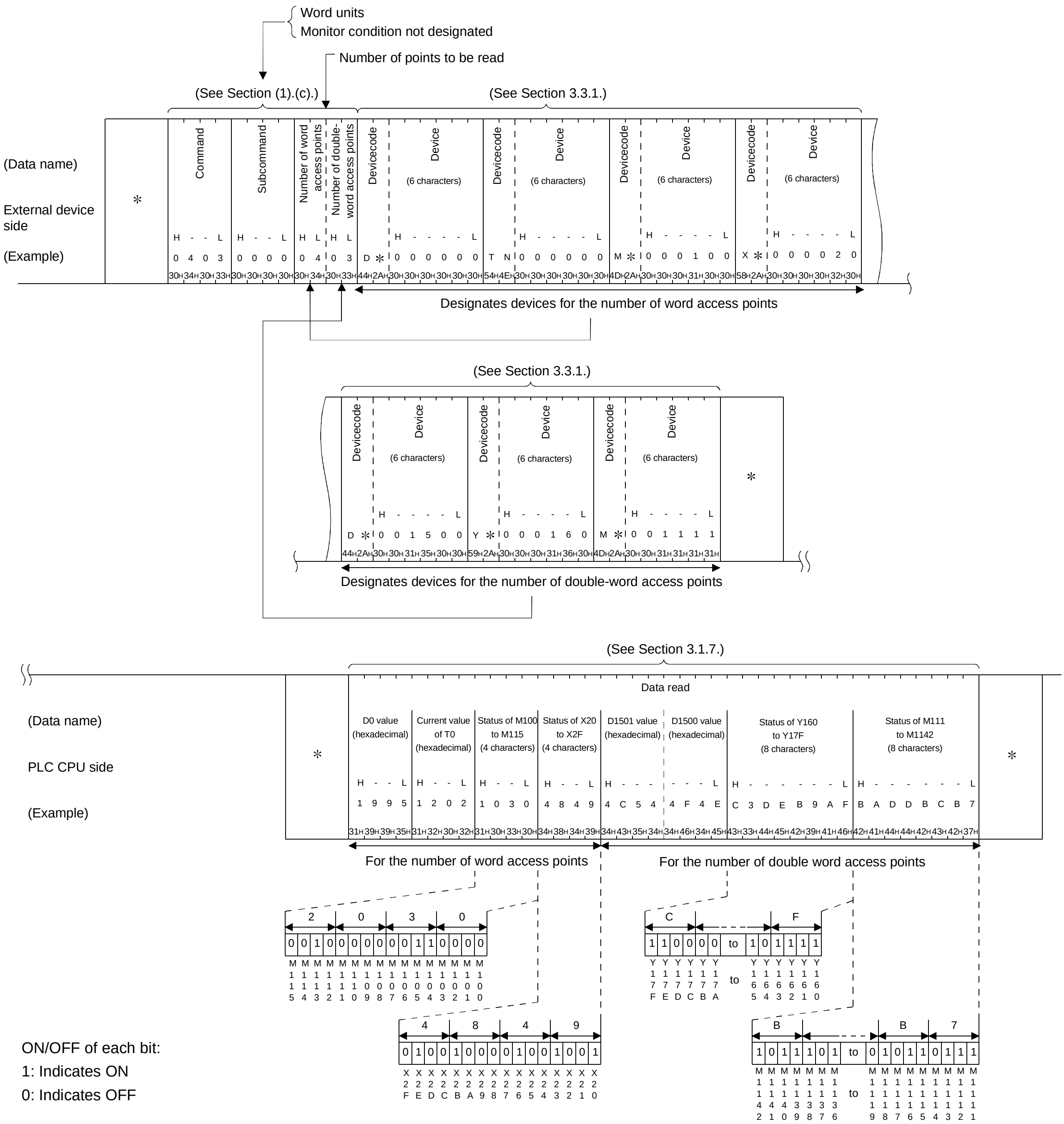
The order and content of data items of the areas marked by "*" shown in the control procedure diagram differ depending on the module used as well as the frame and format used for communication.

See the detailed information described in Section 3.1.

[Control procedure]

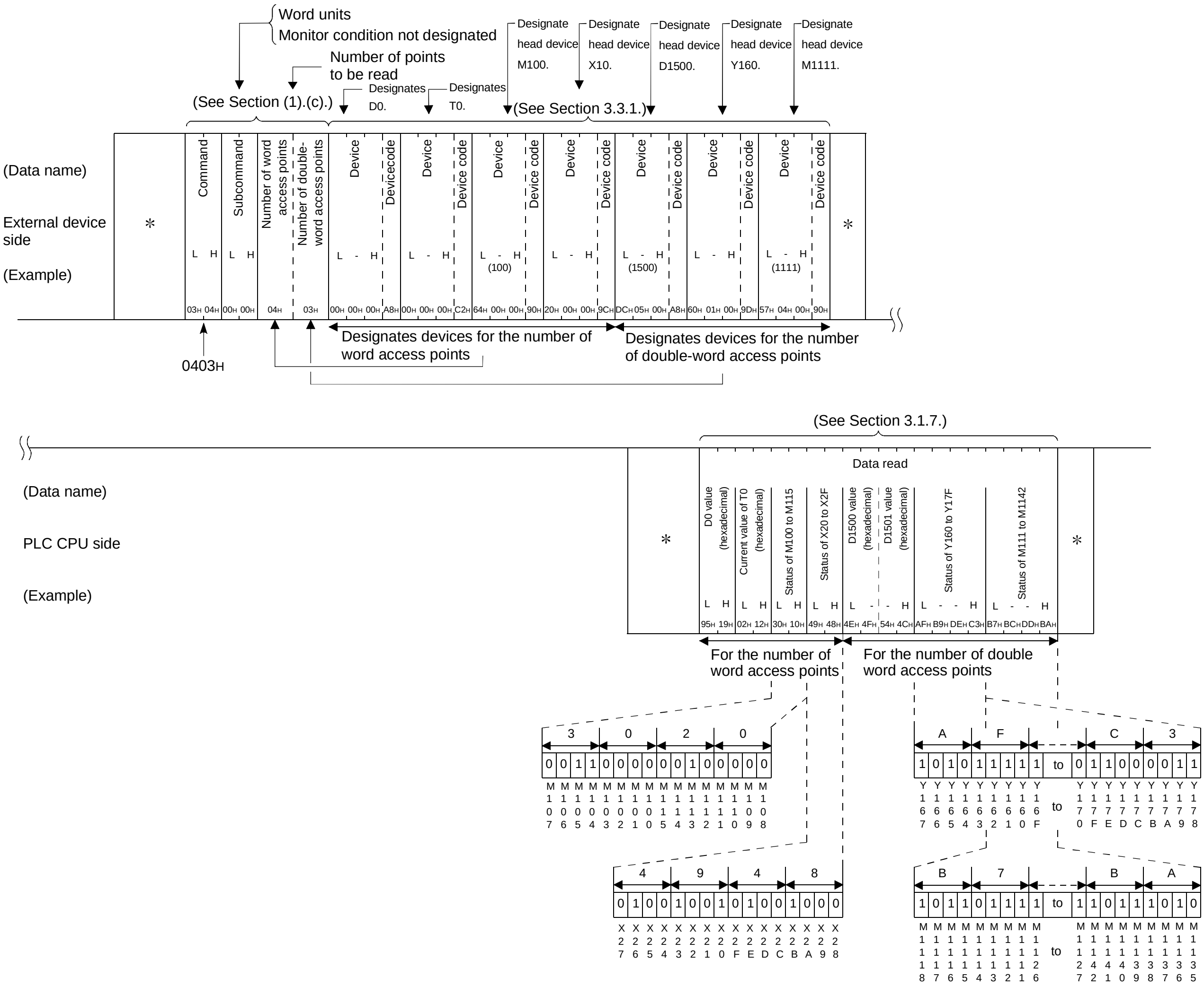
(a) Reading the following device memories while communicating in ASCII code

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



(b) Reading the following device memories while communicating in binary code

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



**POINT**

(1) Designate the number of access points within the following range.
For bit devices, one word access point is 16 bits and one double word access point is 32 bits.
For word devices, one word access point is one word and one double word access point is two words.
  (a) When accessing a QCPU on which the Q series C24/E71 is loaded (local station) or a QCPU via a network system (MELSECNET/H, MELSECNET/10, Ethernet) supporting the Q series
  Number of access points : $1 \leq$ (number of word access points + number of double word access points) $\leq 192$
  (b) When accessing a QnACPU (other station) or a Q/QnACPU (other station) via a network system (MELSECNET/10, Ethernet) supporting the QnA series
  Number of access points : $1 \leq$ (number of word access points + number of double word access points) $\leq 96$
  (c) When accessing PLC CPUs other than the above (other stations)
  Number of access points : $1 \leq$ number of word access points $\leq 10$

(2) When accessing a bit device of the PLC CPU of c), make sure that the device number is a multiple of 16 (0, 16 ... in case of decimal representation).

(3) Random read in word units (when monitor condition is designated)

The examples shown in this section explain the control procedure of reading data by randomly designating bit device memories (16-bit units) and word device memories (word units) with a monitor condition designated.
The order and content of data items of the areas marked by "∗" shown in the control procedure diagram differ depending on the module used as well as the frame and format used for communication.
See the detailed information described in Section 3.1.

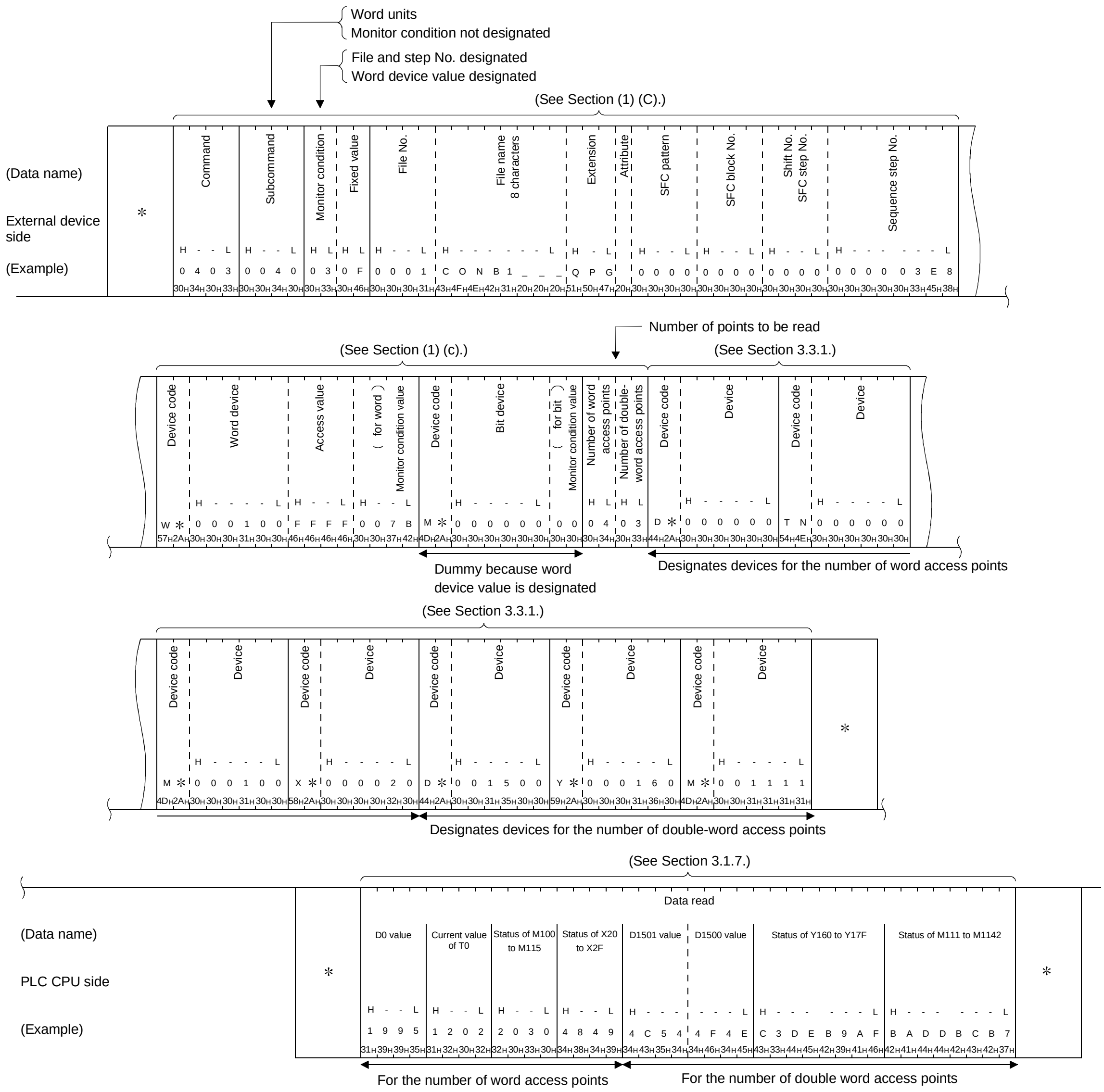
[Control procedure]

(a) Reading randomly while communicating in ASCII code under the following conditions

1) Monitor condition
When the value of link register W100 reaches "7BH" (123) during step number 1000 of the program file CONB1.QPG is executed.

2) Device memories to be read

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



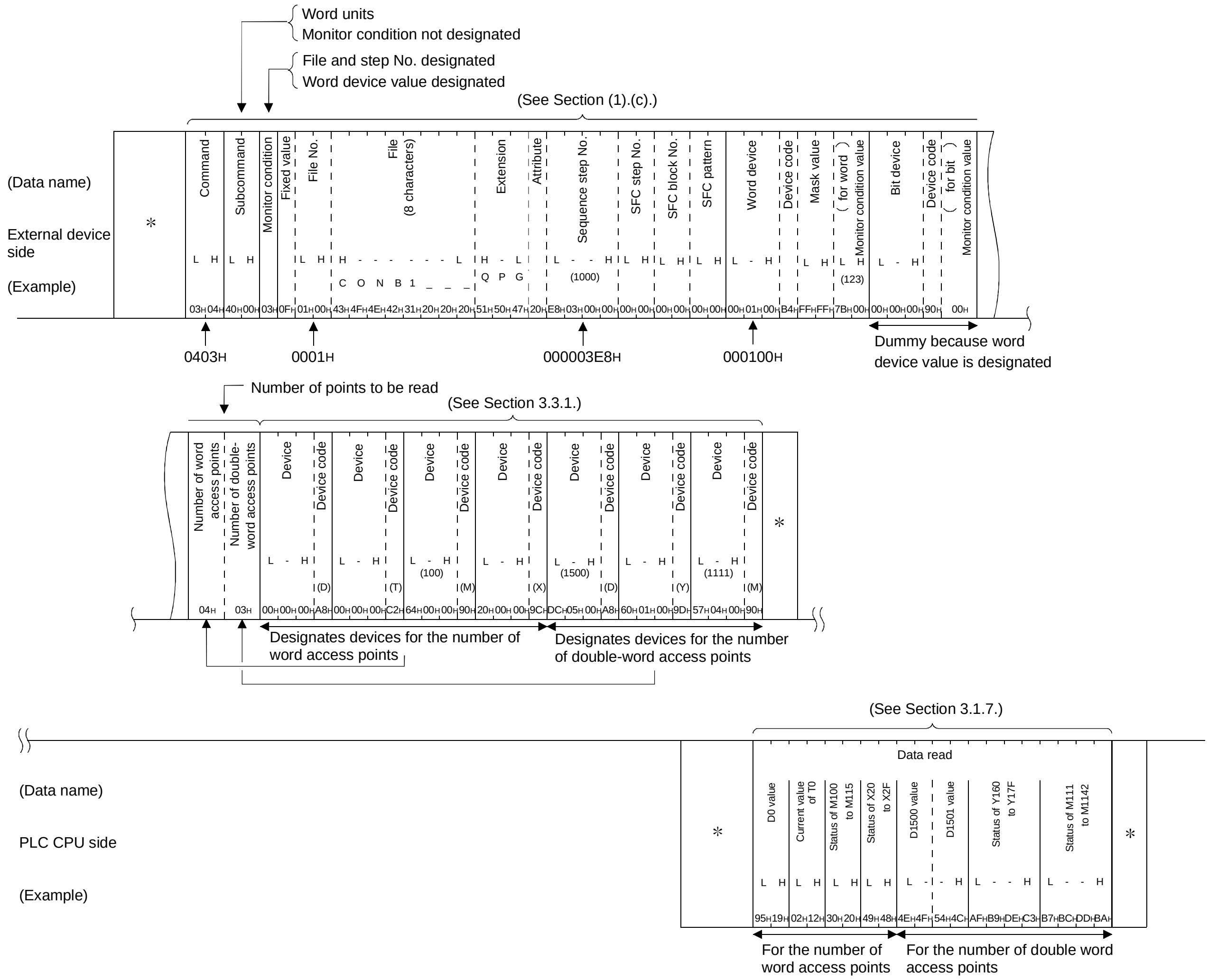
(b) Reading randomly while communicating in binary code under the following conditions

1) Monitor condition

When the value of link register W100 reaches "7BH" (123) during step number 1000 of the program file CONB1.QPG is executed.

2) Device memories to be read

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



**POINT**

(1) Designate the number of access points within the following range.
For bit devices, one word access point is 16 bits and one double word access point is 32 bits.
For word devices, one word access point is one word and one double word access point is two words.

(a) When accessing a QCPU on which the Q series C24/E71 is loaded (local station) or a QCPU via a network system (MELSECNET/H, MELSECNET/10, Ethernet) supporting the Q series

Number of access points : 1 ≤ (number of word access points + number of double word access points) ≤ 192

(b) When accessing a QnACPU (other station) or a Q/QnACPU (other station) via a network system (MELSECNET/10, Ethernet) supporting the QnA series

Number of access points : 1 ≤ (number of word access points + number of double word access points) ≤ 96

(c) When accessing PLC CPUs other than the above (other stations)

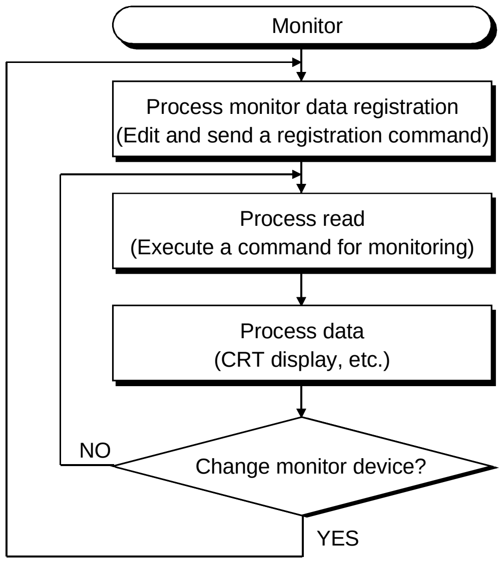
Number of access points : 1 ≤ number of word access points ≤ 10

(2) When accessing a bit device of the PLC CPU of (c), make sure that the device number is a multiple of 16 (0, 16 ... in case of decimal representation).

### 3.3.9 Monitoring device memory

The monitor data registration function registers the name and number of the devices on the external device to be monitored in the Q series C24/E71.
The monitor function reads the data content of the device registered to be monitored from the PLC CPU and processes it in the external device.
When the batch read function is used to read the devices, the device numbers must be consecutive. By using the monitor data registration function, on the other hand, devices can be monitored by designating the device numbers at random.
The control procedures for monitoring and registering devices and their numbers to be monitored in the Q series C24/E71 are explained below using examples.

(1) Monitoring procedure




Monitor data for one registration can be registered to the Q series C24/E71. (*1)
Command…0801

Command…0802

*1 The monitor registration should not be registered from multiple external devices.
During the execution, the last monitor data registration becomes valid.

POINT

(1) The monitor function for device memory can read data in the follow ways.
The designation method, contents of each data field in the character areas in the control procedure, and the timing to monitor (read) are the same as when the random read word units function is used. See Section 3.3.8 for more information.
   1) It is possible to designate bit device memory and word device memory together.
   2) Bit device memory can be read in 16/32 bit units and word device memory can be read in 1/2 word units.
   3) It is possible to designate a monitor condition that will determine when to monitor (read) data at the time of monitor data registration. (Multiple conditions can be combined and designated.)

(2) Intelligent function modules and GX Developer cannot monitor a device memory of a single Q/QnACPU by monitor condition designation at the same time.
When the external device sends the following command message to the Q series C24/E71, it will send an abnormal complete code to the external device if another intelligent functional module or GX Developer is monitoring the same Q/QnACPU by monitor condition designation. (When monitoring without designating monitor conditions, the Q series C24/E71 can perform monitoring with or without designating monitor conditions.)

| Command | Function | Reference section |
|---|---|---|
| 0403 | Random read word units | Section 3.3.8 |
| 0802 | Registered device memory monitoring | Item (4) in this section |

(3) When a monitoring is performed using the procedure above, the monitor data must always be registered. If a monitoring is performed without registering the monitor data, an error complete code will be returned.

(4) When a station on which the Q series C24/E71 is installed is rebooted, the registered monitor data is erased.
When monitor data is re-registered to the same station, the monitor data registered previously is erased.

(2) Monitor data registration (command: 0801) (when the monitor condition is not designated)

The examples shown in this section explain how to register monitor data in order to monitor the device memories by randomly designating bit device memories (16/32 bit units) and word device memories (1/2 word units) without a monitor condition (reading condition) designated.

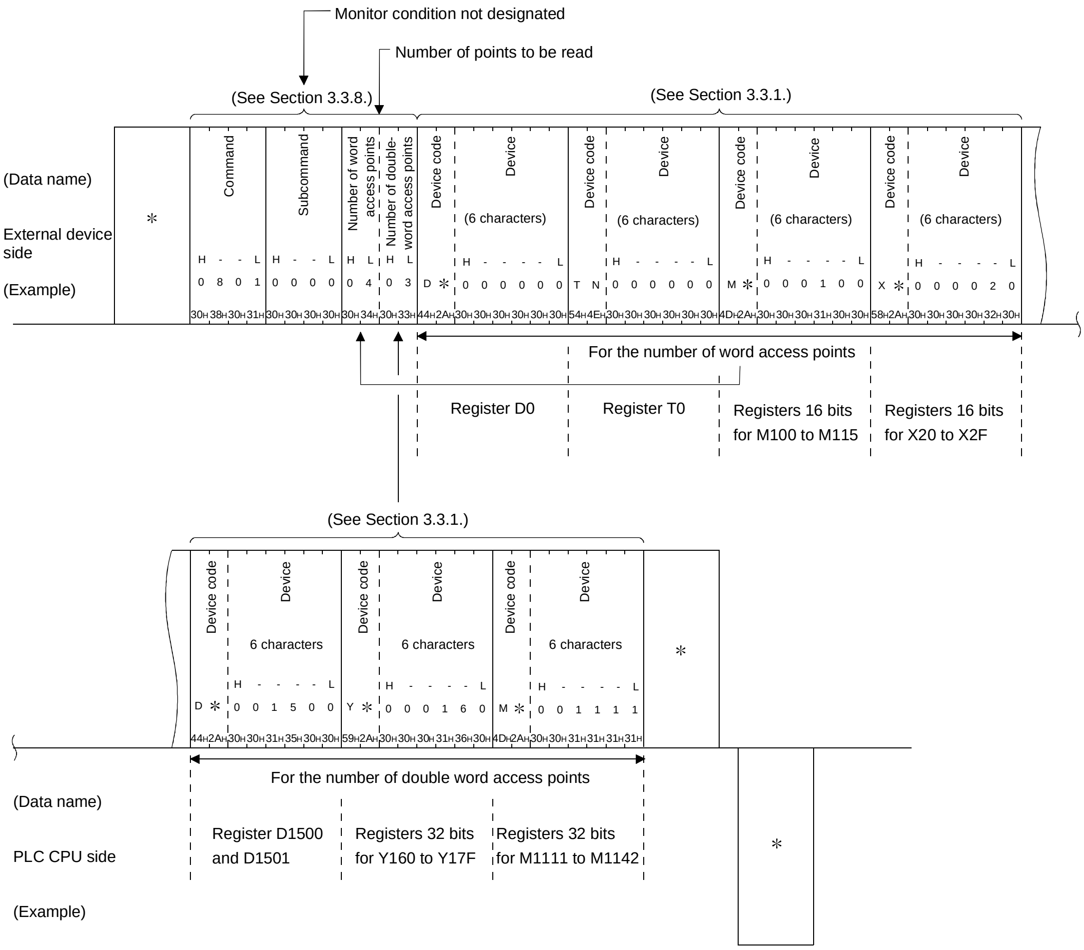
The order and content of data items of the areas marked by "✻" shown in the control procedure diagram differ depending on the module used as well as the frame and format used for communication.

See the detailed information described in Section 3.1.

[Control procedure]

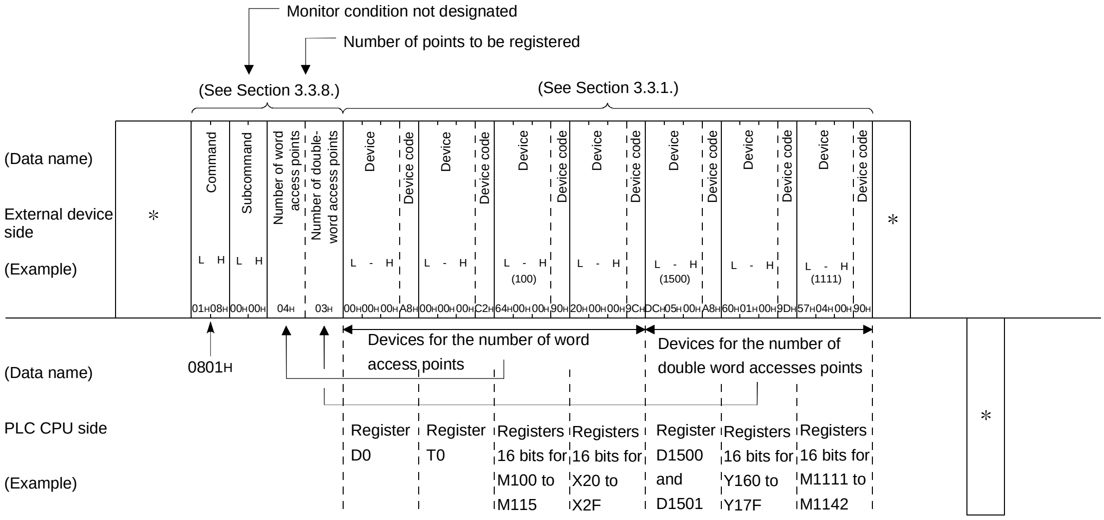
(a) Registering monitor data for the following device memories while communicating in ASCII code

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



(b) Registering monitor data for the following device memories while communicating in binary code

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




## POINT

Designate the number of access points within the following range.
For bit devices, one word access point is 16 bits and one double word access point is 32 bits.
For word devices, one word access point is one word and one double word access point is two words.

(1) When accessing a QCPU on which the Q series C24/E71 is loaded (local station) or a QCPU via a network system (MELSECNET/H, MELSECNET/10, Ethernet) supporting the Q series

Number of access points : 1 ≤ (number of word access points + number of double word access points) ≤ 192

(2) When accessing a QnACPU (other station) or a Q/QnACPU (other station) via a network system (MELSECNET/10, Ethernet) supporting the QnA series

Number of access points : 1 ≤ (number of word access points + number of double word access points) ≤ 96

(3) When accessing PLC CPUs other than the above (other stations)
Access is prohibited.

### (3) Monitor data registration (command: 0801) (When designating the monitor condition)

The examples shown in this section explain the control procedure for registering monitor data in order to monitor the device memories by randomly designating the bit device memories (16/32 bit units) and word device memories (1/2 word units) with the monitor condition (reading condition) designated.
The order and content of data items of the areas marked by "∗" shown in the control procedure diagram differ depending on the module used as well as the frame and format used for communication.
See the detailed information described in Section 3.1.

[Control procedure]

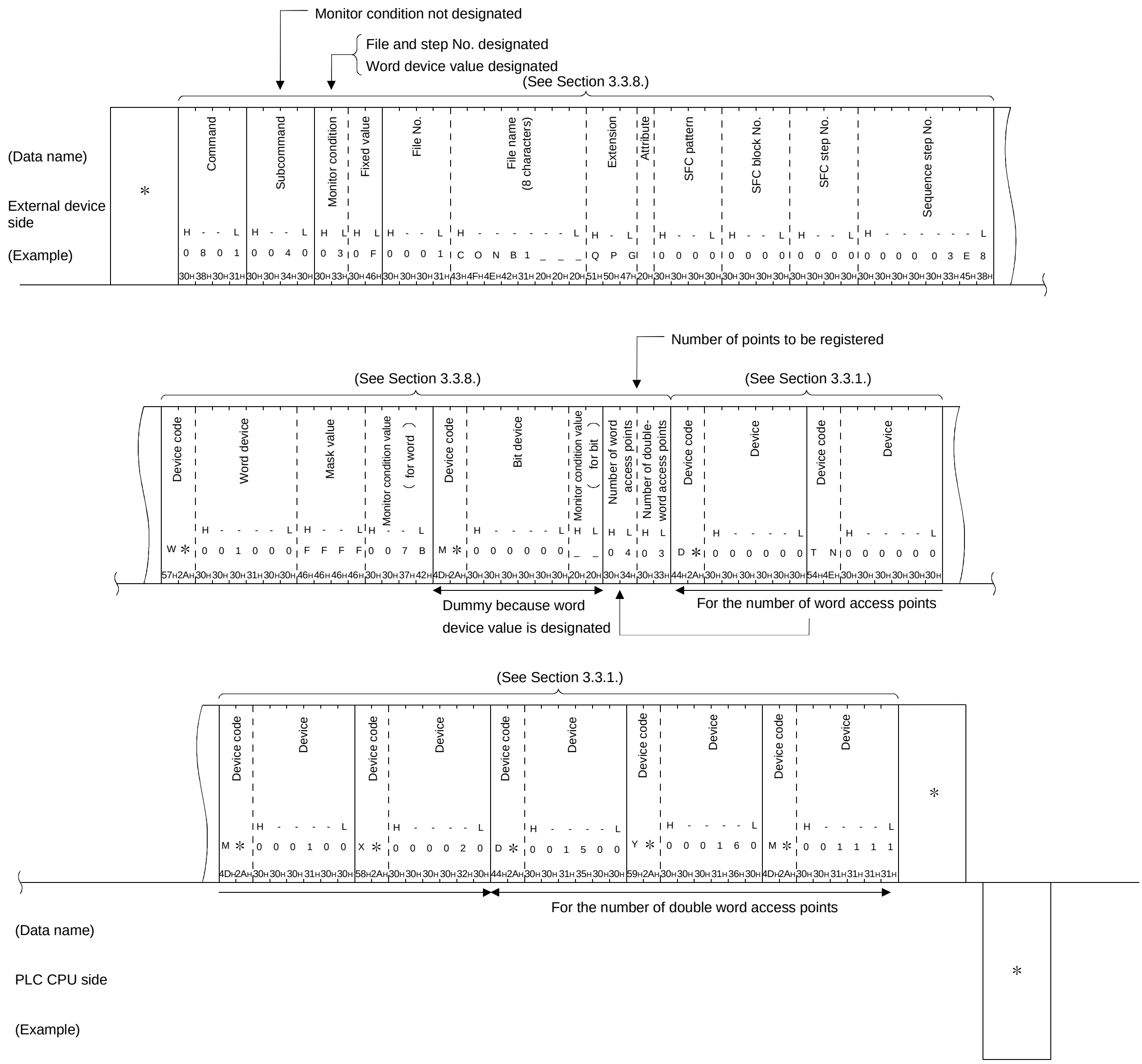
(a) Monitor data is registered in the following ways while communicating in ASCII code

1) Monitor condition

When the value of link register W100 reaches "$7B_H$" (123) while step number 1000 of the sequence program of the program file CONB1.QPG is being executed

2) Device memories to be monitored (read)

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


Monitor condition not designated
File and step No. designated
Word device value designated
(See Section 3.3.8.)
(Data name)
External device side
(Example)
Command
Subcommand
Monitor condition
Fixed value
File No.
File name (8 characters)
Extension
Attribute
SFC pattern
SFC block No.
SFC step No.
Sequence step No.
0 8 0 1
0 0 4 0
0 3
0 F
0 0 0 1
C O N B 1 _ _ _
Q P G
0 0 0 0
0 0 0 0
0 0 0 0
0 0 0 0 0 3 E 8
Number of points to be registered
(See Section 3.3.8.)
(See Section 3.3.1.)
Device code
Word device
Mask value
Monitor condition value (for word)
Bit device
Monitor condition value (for bit)
Number of word access points
Number of double-word access points
Device
W ✻ 0 0 1 0 0 0
F F F F
0 0 7 B
M ✻ 0 0 0 0 0 0
0 4
0 3
D ✻ 0 0 0 0 0 0
T N 0 0 0 0 0 0
Dummy because word device value is designated
For the number of word access points
(See Section 3.3.1.)
M ✻ 0 0 0 1 0 0
X ✻ 0 0 0 0 2 0
D ✻ 0 0 1 5 0 0
Y ✻ 0 0 0 1 6 0
M ✻ 0 0 1 1 1 1
For the number of double word access points
(Data name)
PLC CPU side
(Example)

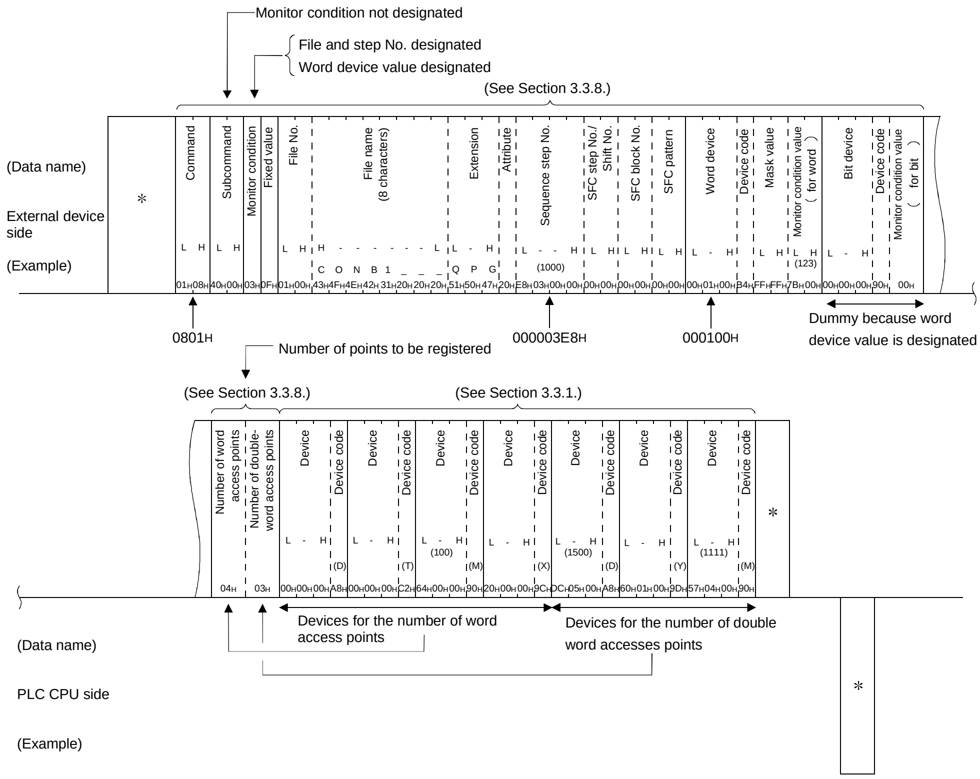
(b) Monitor data is registered in the following ways while communicating in binary code

1) Monitor condition

When the value of link register W100 reaches "7BH" (123) while step number 1000 of the program file CONB1.QPG is being executed

2) Device memories to be monitored (read)

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




## POINT

Designate the number of access points within the following range.
For bit devices, one word access point is 16 bits and one double word access point is 32 bits.
For word devices, one word access point is one word and one double word access point is two words.

(a) When accessing a QCPU on which the Q series C24/E71 is loaded (local station) or a QCPU via a network system (MELSECNET/H, MELSECNET/10, Ethernet) supporting the Q series

Number of access points : $1 \leq$ (number of word access points $\times$ 12 + number of double word access points x 14) $\leq$ 1920

(b) When accessing a QnACPU (other station) or a Q/QnACPU (other station) via a network system (MELSECNET/10, Ethernet) supporting the QnA series

Number of access points : $1 \leq$ (number of word access points $\times$ 12 + number of double word access points x 14) $\leq$ 960

(c) When accessing PLC CPUs other than the above (other stations)
Access is prohibited.

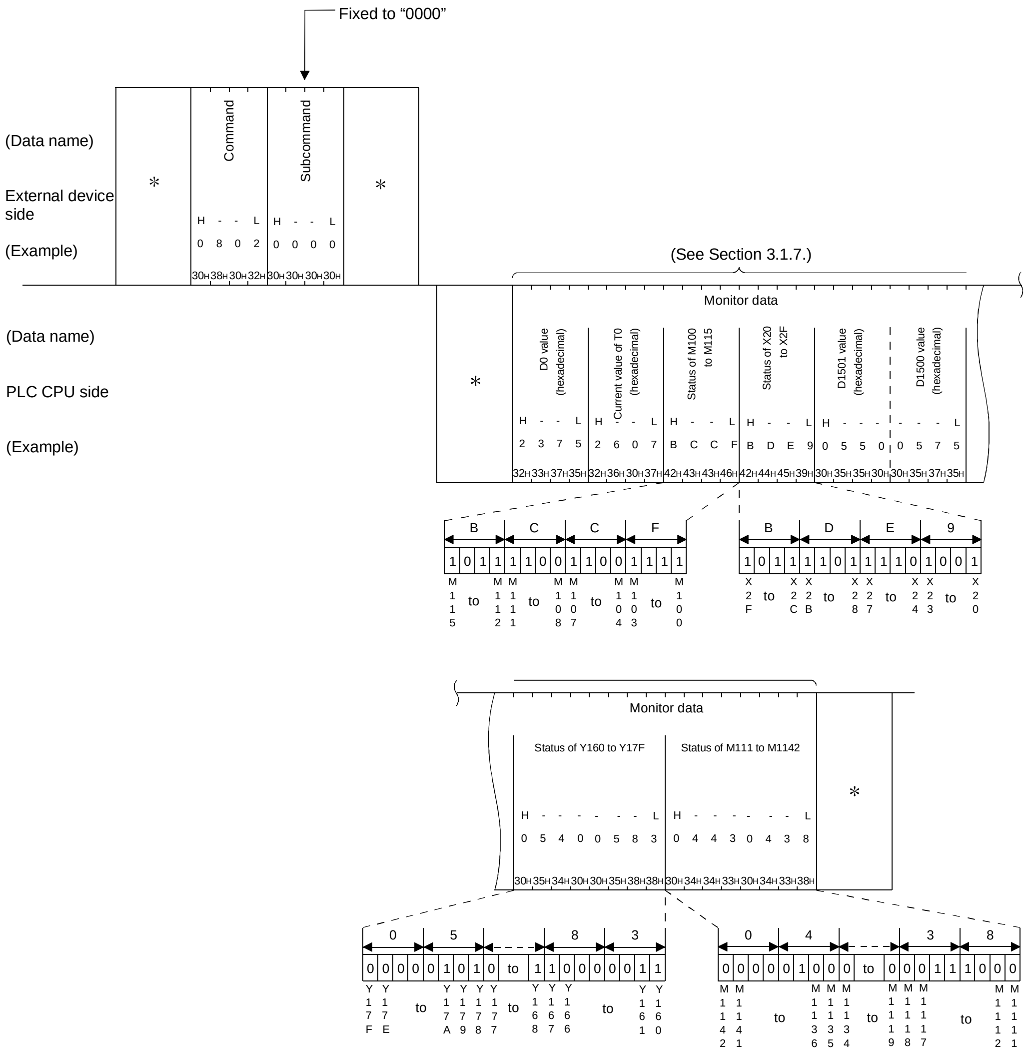
### (4) Monitoring registered device memories (command: 0802)

The following section explains the control procedure for monitoring a device memory registered using the monitor data registration (command: 0801).
The monitoring control procedure is the same regardless of whether or not a monitor condition is designated during monitor data registration.
The data arrangement and content of the areas marked with "∗" in the control procedure diagram differ depending on the module, communication frame and format used.
See the detailed information described in Section 3.1.

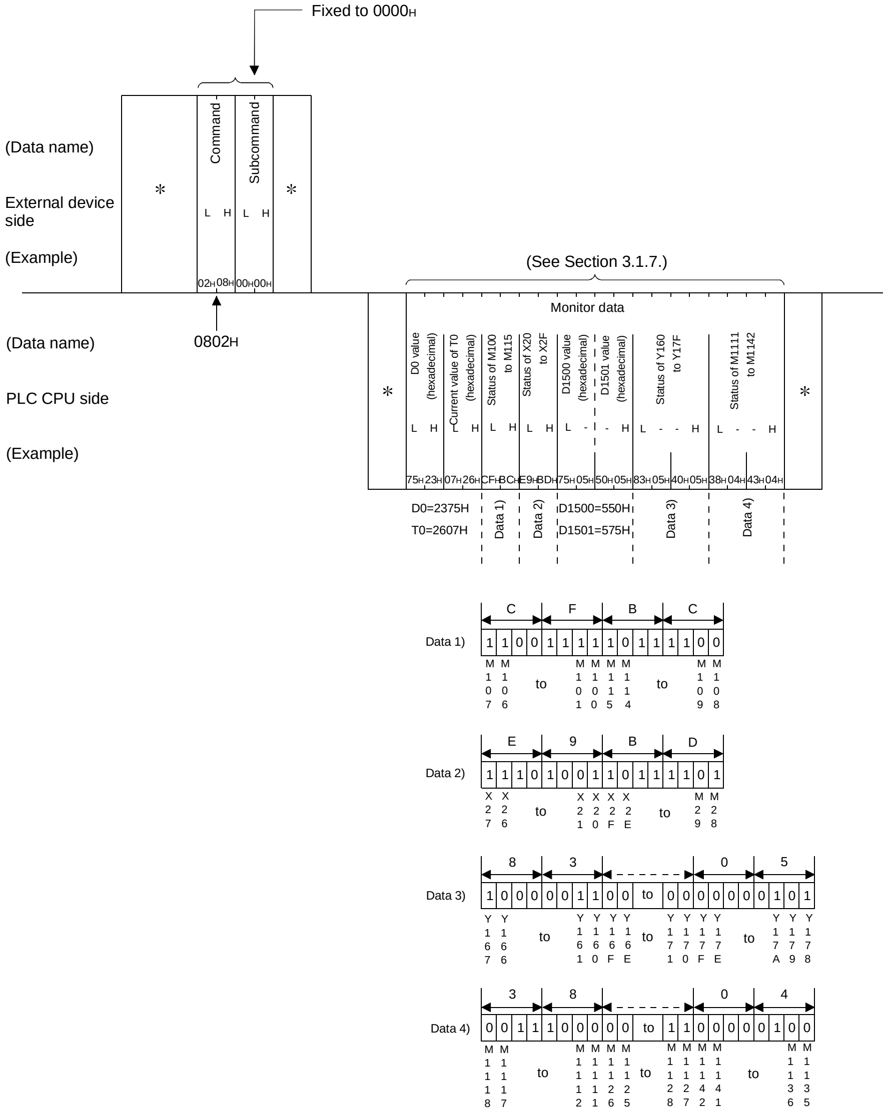
[Control procedure]

(a) When monitoring while communicating in ASCII code the following device memories with monitor data are registered
(Device memories with monitor data registered)

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



(b) When monitoring while communicating in binary code the following device memories with monitor data are registered
(Device memories with monitor data registered)
- Word access : D0, T0, M100 to M115, X20 to X2F
- Double word access : D1500 to D1501, Y160 to Y17F, M1111 to M1142



### 3.3.10 Multiple block batch read and batch write

The examples shown in this section explain the control procedure for reading and writing by randomly designating multiple blocks, where one block consists of n points of a bit device memory (1 point = 16 bits) and a word device memory (1 point = 1 word).

This function is used when accessing the Q/QnACPU.
The following modules are accessible:

- A QCPU on which the Q series C24/E71 is installed (local station) and QCPUs via a network system (MELSECNET/H, MELSECNET/10, Ethernet) supporting the Q series (other stations)
- A QnACPUs shown below (other stations) and QCPUs/QnACPUs shown below via a network system (MELSECNET/10H, Ethernet) supporting the QnA series (other stations)

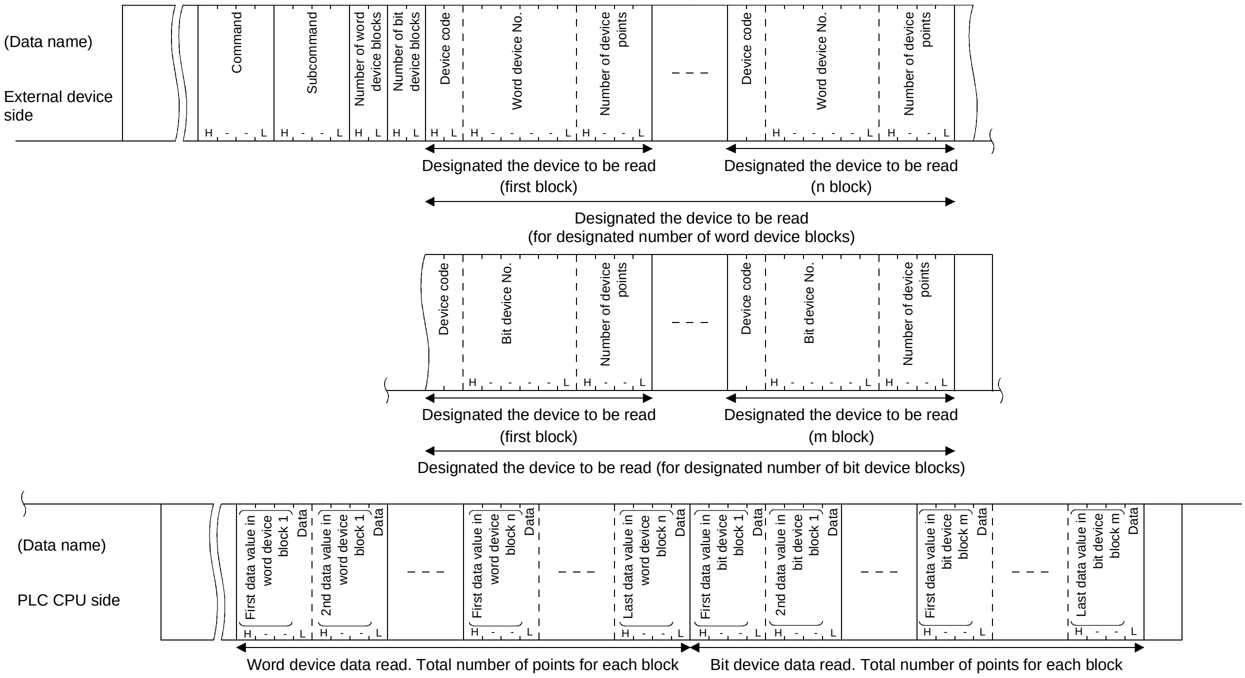
| Function | PLC CPU | | |
|---|---|---|---|
| | QnA | Q2AS (H) | Q4AR |
| Multiple block batch read/write | (Products 9707B or later) ✻ | | (All accessible) |

✻ This function is added to the products that have 9707 B or later on the package marking and DATE column of the rated plate.
(9707: Date of manufacture, B:Functional version (marked for version B or later))

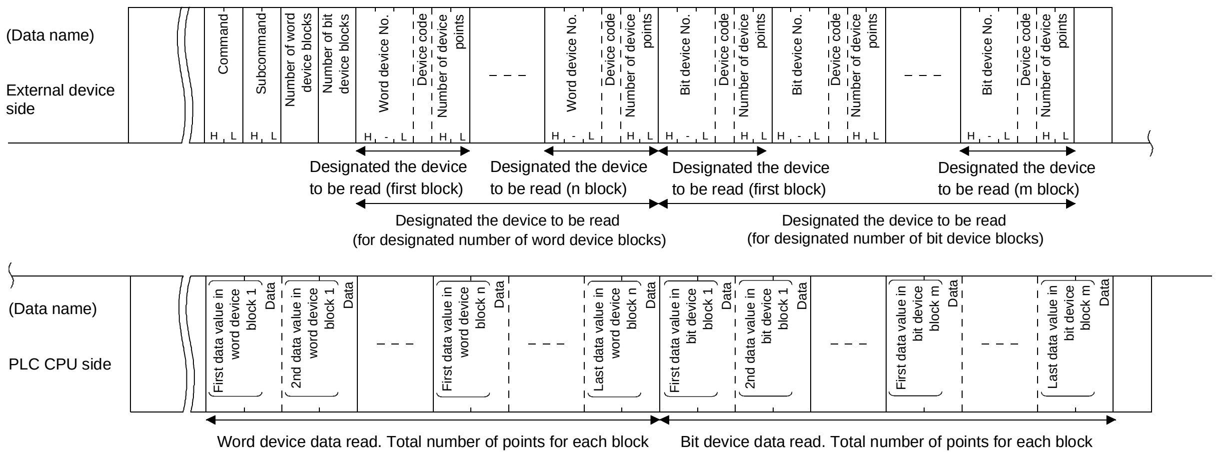
(1) Data order in the character area during the multiple block batch read

This section explains how data is ordered in the character areas during multiple block batch read.

(a) Data order when communicating in ASCII code



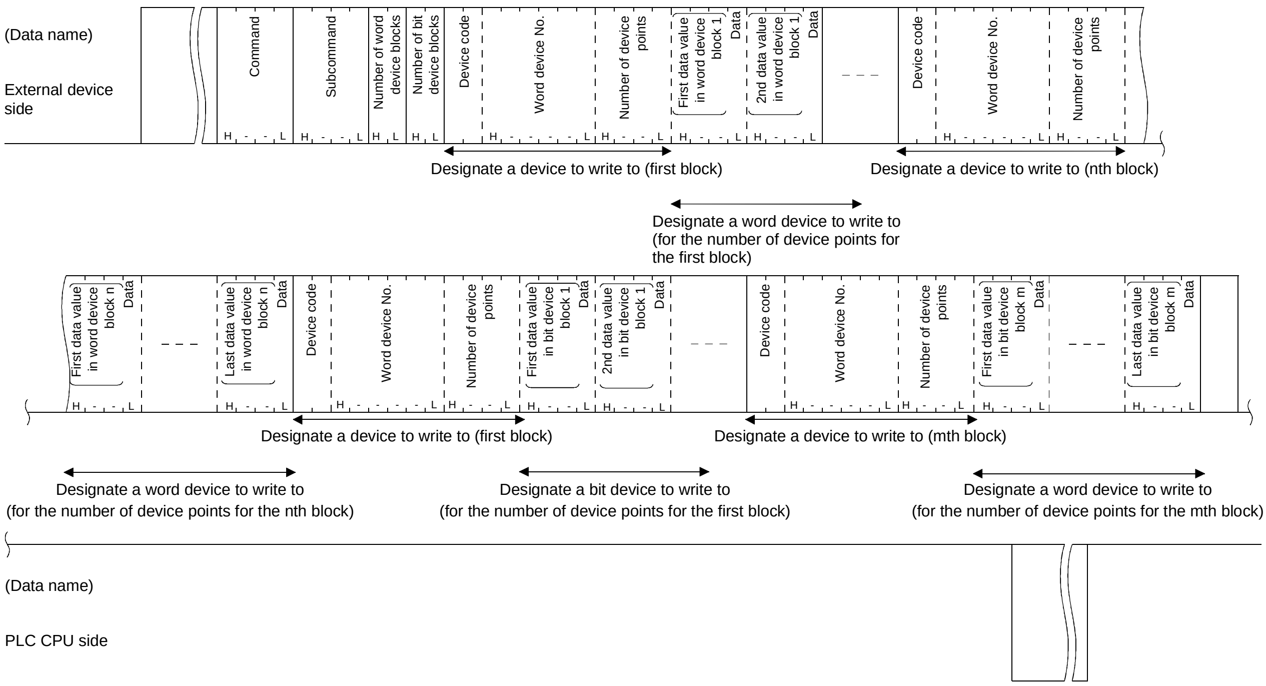

(b) Data order when communicating in binary code



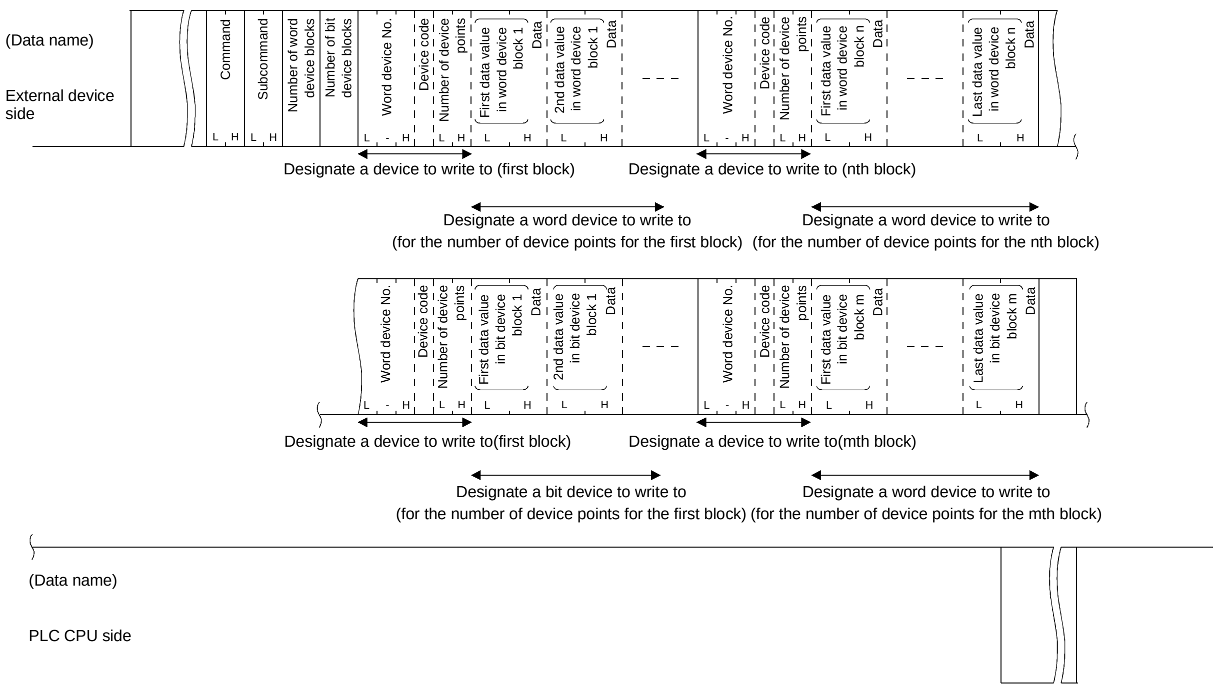
## (2) Data order in the character area during the multiple block batch write

This section explains how data is ordered in the character areas during multiple block batch write.

### (a) Data order when communicating in ASCII code




### (b) Data order when communicating in binary code



### (3) Contents of the character areas during multiple block batch read and batch write

This section explains what is contained in the character area when a multiple block batch read or batch write function is performed.
The contents are the same as when using the other commands, except for the data shown below.

(a) Number of word device blocks and number of bit device blocks
This value designates the number of word device blocks or bit device blocks to be sent directly after this data field in the batch read or batch write to the word device or bit device, respectively.

1) Data communication in ASCII code
Each number of blocks are converted to 2-digit ASCII code (hexadecimal) and transmitted.
(Example)
For 5 blocks : Converted to "05," and transmitted sequentially from "0."
In case of 20 blocks : Converted to "14," and transmitted sequentially from "1."

2) Data communication in binary code
1-byte numeric value indicating each number of blocks is transmitted.
(Example)
In case of 5 blocks : $05_H$ is transmitted.
In case of 20 blocks : $14_H$ is transmitted.

3) Designate each number of blocks so the following condition is satisfied:
120 ≥ number of word device blocks + number of bit device blocks

4) When setting either number of blocks to 0, the corresponding device number, device code, number of device points, and data designations are not necessary.

(b) Word device number and bit device number
This value designates the head word device or bit device for each block to which batch read or batch write is performed, where contiguous word/bit devices are considered one block.

1) Data communication in ASCII code
The head device number of each block is converted to 6-digit ASCII code and transmitted.
(Example)
Internal relay M1234 and link register W1234:
The internal relay M1234 is converted to "001234" or "␣␣1234" and the link register W1234 is converted to "001234" or "␣␣1234."
In both cases, the transmission starts from "0" or "␣."

2) Data communication in binary code
The head device number of each block is indicated in a 3-byte numeric value and transmitted.
(Example)
Internal relay M1234 and link register W1234:
The internal relay M1234 is converted to $0004D2_H$ and transmitted in the order of $D2_H$, $04_H$, and $00_H$.
The link register W1234 is converted to $001234_H$ and transmitted in the order of $34_H$, $12_H$, and $00_H$.

(c) Device code

This value identifies the head device memory for each block for which batch read or batch write is performed.
The device code for each device is shown in Section 3.3.1(3).

1) Data communication in ASCII code

Each device code is converted to 2-digit ASCII code (hexadecimal) and transmitted.
(Example)
Internal relay (M) and link register (W):
The internal relay (M) is converted to "M∗" and link register (W) is converted to "W∗," and transmitted from "M" and "W" respectively.

2) Data communication in binary mode

One-byte numeric value indicating each device code is transmitted.
(Example)
Internal relay
(M) and link register (W):
$90_H$ is transmitted for the internal relay (M) and $B4_H$ is transmitted for the link register (W).

(d) Number of device points

This value designates the number of points in the contiguous device range of each block for which batch read or batch write is performed (1 point = 16 bits for bit device memory and 1 point = 1 word for word device memory), where one block consists of contiguous word/bit devices.

1) Data communication in ASCII code

The number of points for each block is converted to a 4-digit ASCII code (hexadecimal) and transmitted.
(Examples)

| | | |
|---|---|---|
| In case of 5 points | : | Converted to "0005" and transmitted starting from "0." |
| In case of 20 points | : | Converted to "0014" and transmitted starting from "0." |

2) Data communication in binary code

2-byte numeric value indicating the number of points for each block is transmitted.
(Examples)

| | | |
|---|---|---|
| In case of 5 points | : | Converted to $0005_H$ and transmitted starting from $05_H$. |
| In case of 20 points | : | Converted to $0014_H$ and transmitted starting from $14_H$. |

3) Designate each number of blocks so that the appropriate condition is satisfied

- In case of multiple block batch read
  960 ≥ total number of points for all word device blocks + total number of points for all bit device blocks
- In case of multiple block batch write
  960 ≥ 4 × (number of word device blocks + number of bit device blocks) + total number of points for all word device blocks + total number of points for all bit device blocks

**POINT**

The extension designation is allowed for the device memory being read from or written to using the multiple block batch read or batch write functions.
See the explanation in the Appendix 1 for more information on designating the device memory extension.

(4) Multiple block batch read (command: 0406)

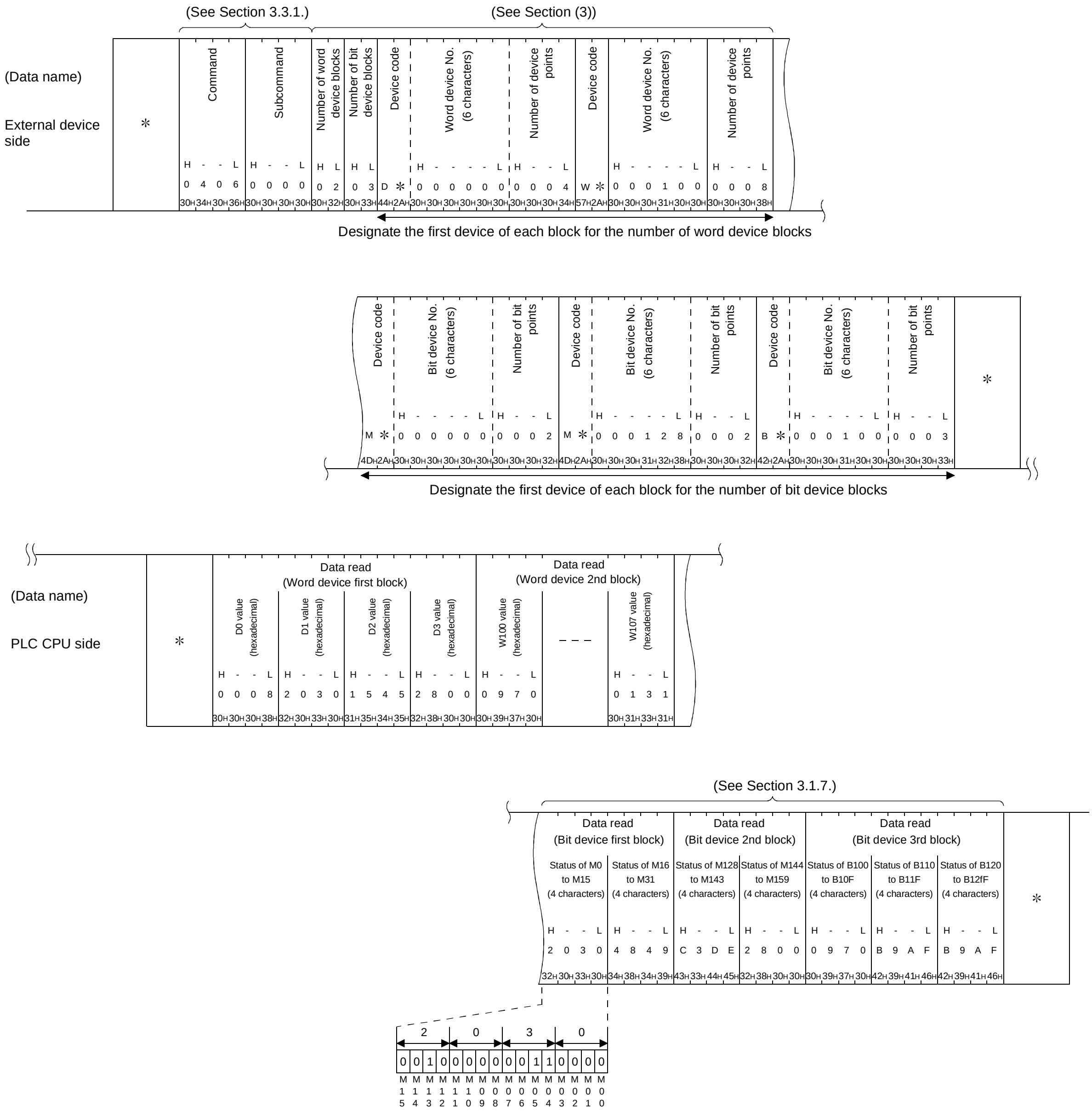
The examples shown in this section explain the control procedure for reading by randomly designating multiple blocks, where one block consists of n points of contiguous bit device memory (1 point = 16 bits) and word device memory (1 point = 1 word).

The data order and contents of the areas marked with "*" in the control procedure diagram differ depending on the module, communication frame and format used.

See the detailed information described in Section 3.1.

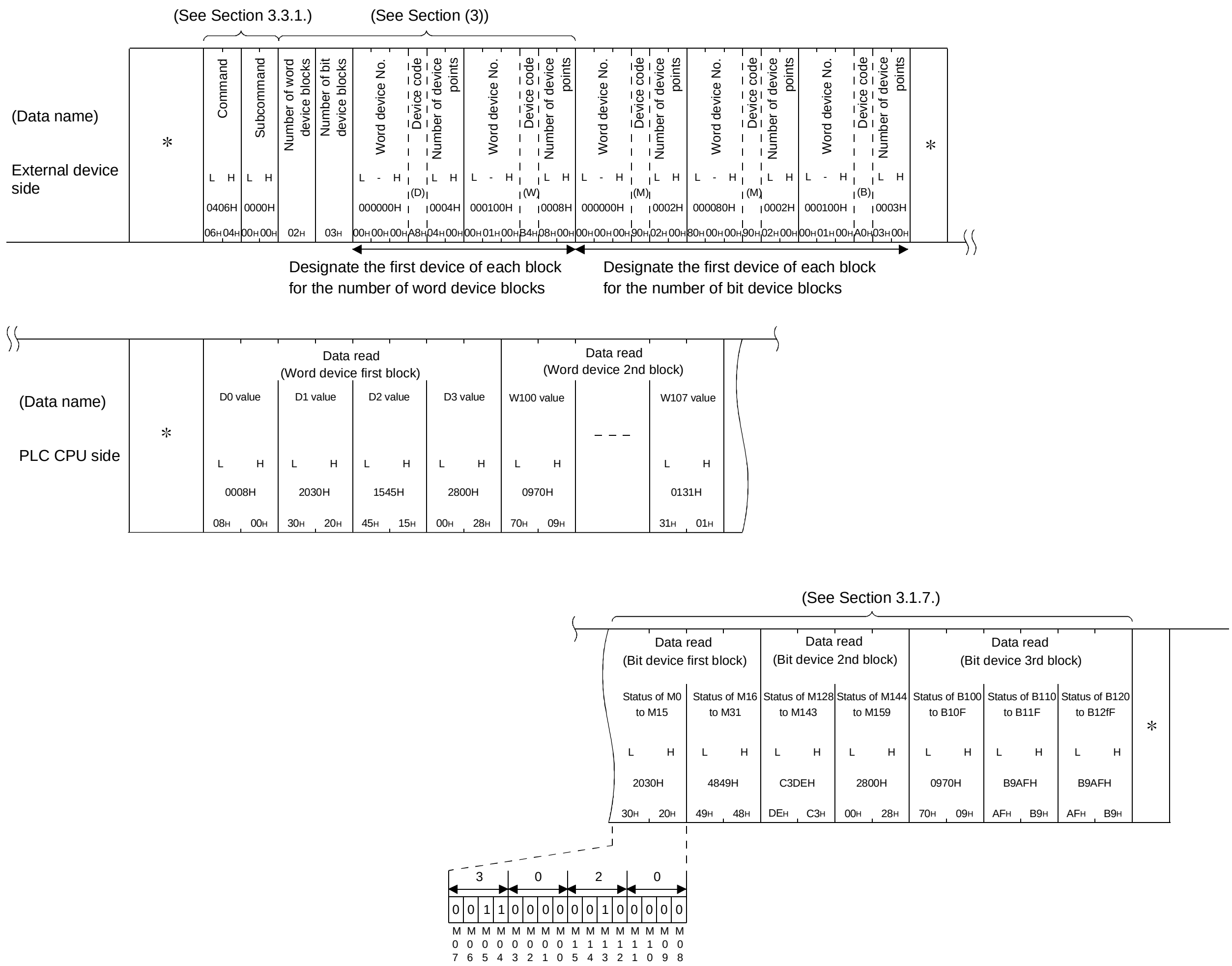
(a) The following device memories are read while communicating in ASCII code

- Word device memory : 2 blocks D0 to D3 (4 points), W100 to W107 (8 points)
- Bit device memory : 3 blocks M0 to M31 (2 points), M128 to M159 (2 points), B100 to B12F (3 points)



(b) The following device memories are read while communicating in binary code

- Word device memory : 2 blocks D0 to D3 (4 points),
  W100 to W107 (8 points)
- Bit device memory : 3 blocks M0 to M31 (2 points),
  M128 to M159 (2 points),
  B100 to B12F (3 points)




**POINT**

(1) Designate the number of blocks so that the following condition is satisfied:
120 ≥ number of word device blocks + number of bit device blocks

(2) Designate each number of device points so that the following condition is satisfied:
960 ≥ total number of points for all word device blocks + total number of points for all bit device blocks

(5) Multiple block batch write (command: 1406)

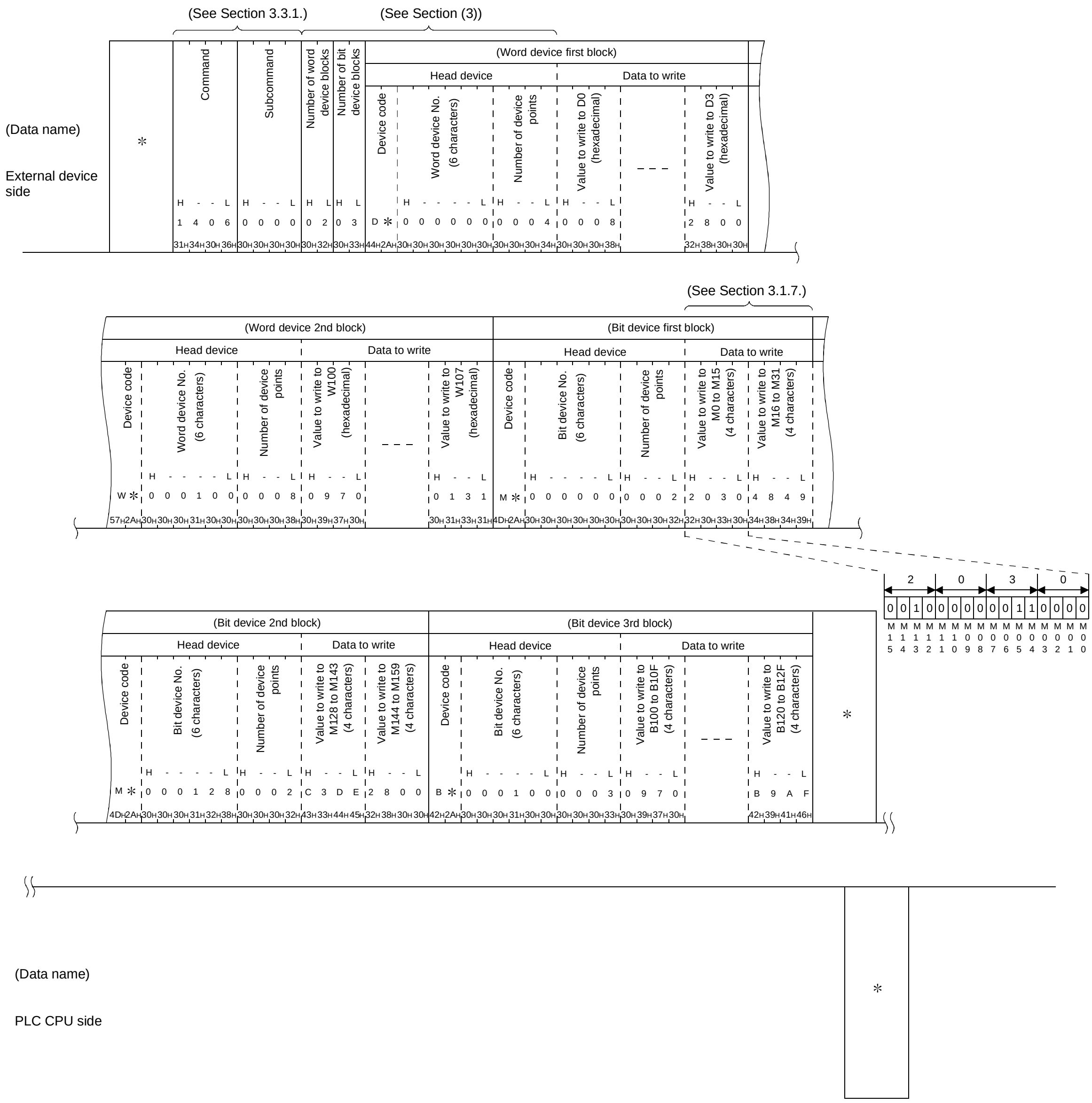
The examples shown in this section explain the control procedure for writing by randomly designating multiple blocks, where one block consists of n points of contiguous bit device memory (1 point = 16 bits) and word device memory (1 point = 1 word).

The data order and contents of the areas marked with "∗" in the control procedure diagram differ depending on the module, communication frame and format used.

See the detailed information described in Section 3.1.

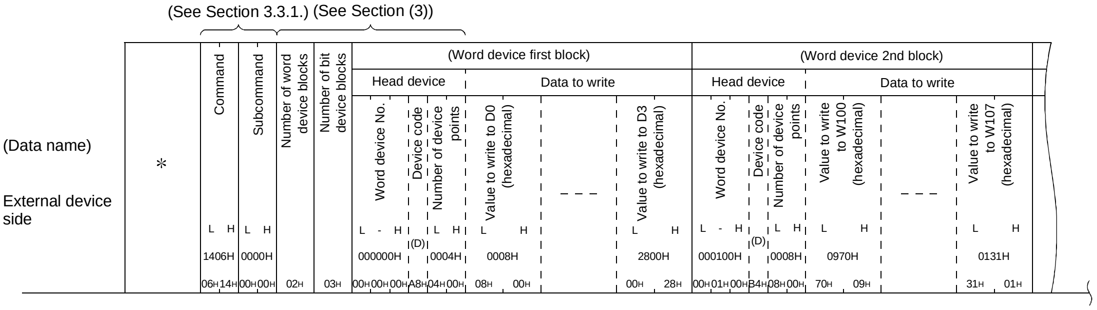
(a) The following device memories are written while communicating in ASCII code

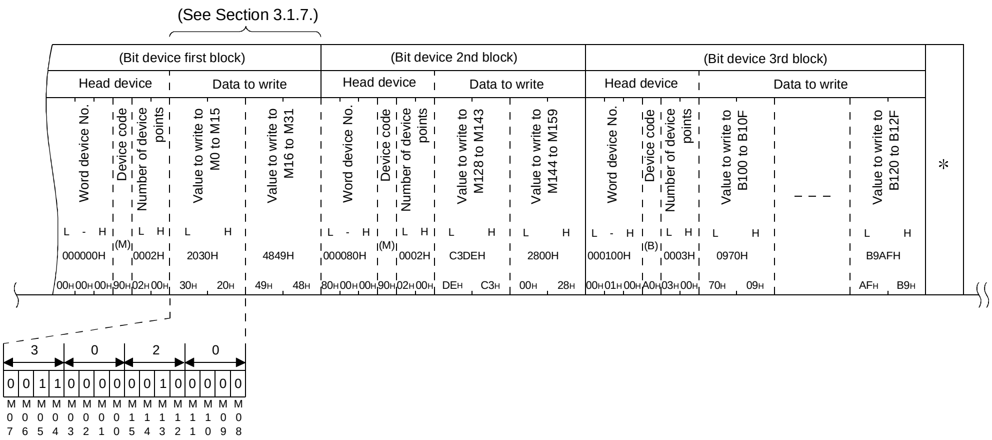
- Word device memory : 2 blocks D0 to D3 (4 points), W100 to W107 (8 points)
- Bit device memory : 3 blocks M0 to M31 (2 points), M128 to M159 (2 points), B100 to B12F (3 points)



(b) The following device memories are written while communicating in binary code

- Word device memory : 2 blocks D0 to D3 (4 points), W100 to W107 (8 points)
- Bit device memory : 3 blocks M0 to M31 (2 points), M128 to M159 (2 points), B100 to B12F (3 points)





**POINT**

Designate each number of device points so that the following condition is satisfied:
960 ≥ 4 × (number of word device blocks + number of bit device blocks) + total number of points for all word device blocks + total number of points for all bit device blocks

## 3.4 Buffer Memory Read/Write

This function is used to read from and write to the buffer memory of the Q series C24/E71 (including the use of multidrop connection) that is connected to the external device.

- Q series C24 (including multidrop connection station)
- Q series E71

Using this function, the communication between the external device and the Q series C24/E71 is performed immediately when the external device issues a read or write request, without waiting for the PLC CPU's END processing.
The PLC CPU uses the FROM or TO instructions to read or write the buffer memory data (data exchanged with the external device).
In this section, the control procedure of this function is described using examples.

### 3.4.1 Commands and buffer memory

This section explains the commands used and the buffer memory address designated by the control procedures when reading and writing in the Q series C24/E71 buffer memory.

(1) Commands

| Function | Command (subcommand) | Processing | Number of points processed per communication | PLC CPU status | | | Reference section |
|---|---|---|---|---|---|---|---|
| | | | | During STOP | During RUN | | |
| | | | | | Write allow setting | Write prohibit setting | |
| Batch read | 0613 (0000) | Reads from buffer memory. | 480 words (960 bytes) | ○ | ○ | ○ | Section 3.4.2 |
| Batch write | 1613 (0000) | Writes to buffer memory. | | ○ | ○ | ○ | Section 3.4.3 |

○ in the PLC CPU status column in the table above indicates that execution is possible.

(2) Buffer memory and access units
The buffer address designated by this function uses the addresses of each module in the buffer memory table shown in Chapter 3 of the User's Manual (Basics).
An address consists of one word (16 bits).
This function reads and writes data in word units.
- In case of the Q series C24, reading and writing are performed in word units regardless of the word/byte units designation.

POINT

- The usage of parts of the buffer memory is predetermined.
- The Q series C24/E71 will not function properly if data is written to areas in the memory whose usage is determined, ignoring the specifications.

(3) Contents of character area

The following explains the contents of the character area when the external device reads from or writes to the buffer memory of the Q series C24/E71.

(a) Head address

This value designates the head address of the range to which data is read (or written).

1) Data communication in ASCII code

Head addresses 0H to 2307H or 7FFFH are converted to 8-digit (hexadecimal) ASCII code and sequentially transmitted beginning from the most significant digit ("0").

(Example)

When the head address is 1E1H: It is converted to "000001E1" and sequentially transmitted beginning from the first "0."

2) Data communication in binary mode

A 4-byte value indicating head addresses 0H to 2307H or 7FFFH is transmitted sequentially beginning from the low byte (L: bits 0 to 7).

(Example)

When the head area address is 1E1H: It is converted to 000001E1H and sequentially transmitted beginning from E1H.

(b) Word length

This value designates the number of addresses (word count) of the range to which data is read (or written).

1) Data communication in ASCII code

The number of addresses in the range from 1H to 1E0H (1 to 480) are converted to 4-digit (hexadecimal) ASCII code and sequentially transmitted beginning from the most significant digit ("0").

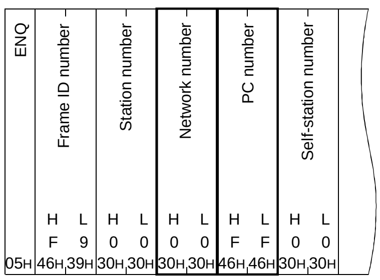
2) Data communication in binary code

A 2-byte value indicating the number of addresses 1H to 1E0H (1 to 480) is sequentially transmitted beginning from the low byte (L: bits 0 to 7).

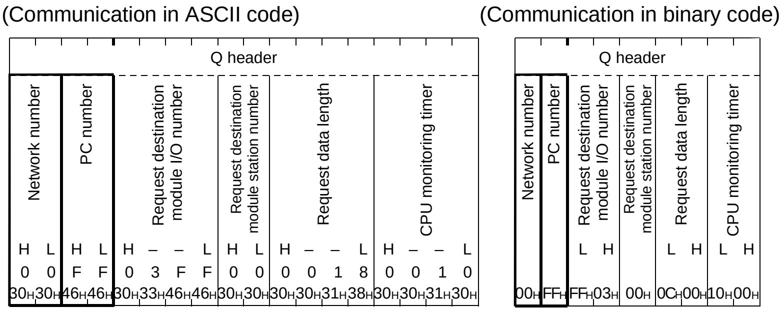
REMARK

Designate the following data for the local station for the network number and PC number data items in the message. (Network number: "00H" PC number: " FFH")

(For QnA compatible 3C frame format 1)




(For QnA compatible 3E frames)

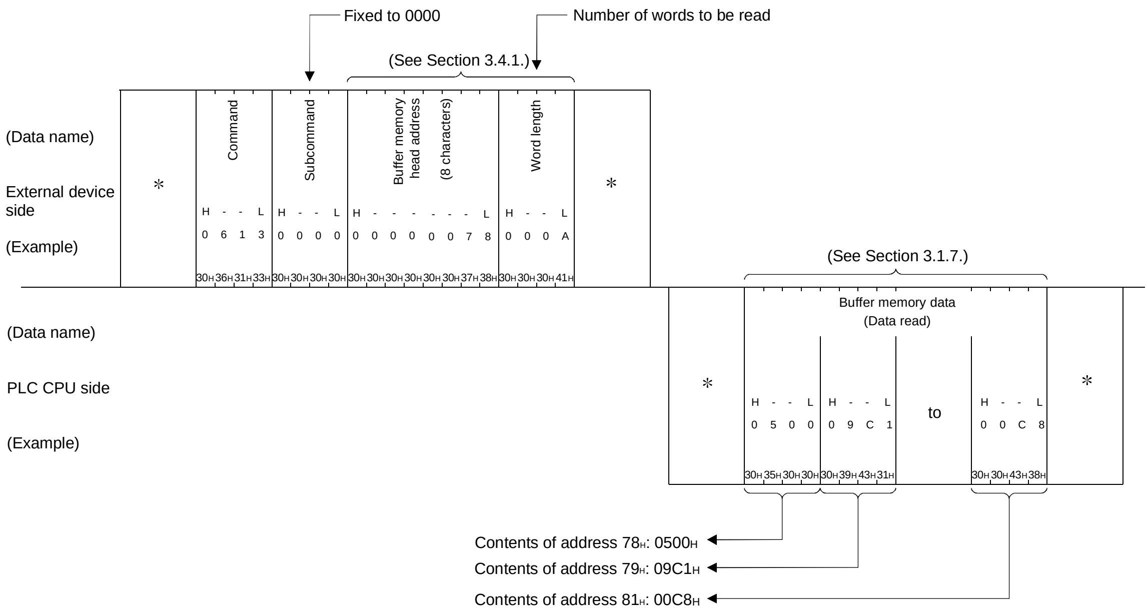


### 3.4.2 Reading buffer memory (command: 0613)

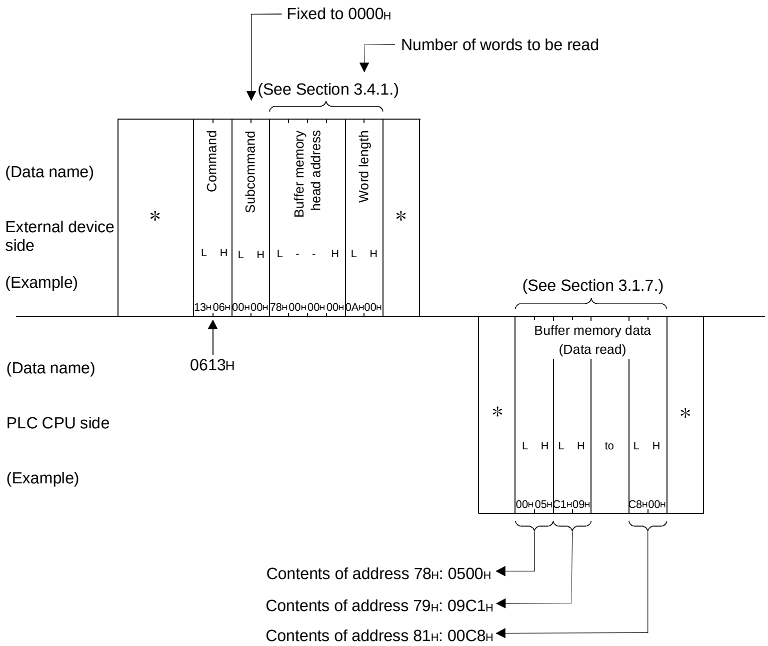
The examples shown in this section explain the control procedure for carrying out a batch read of the Q series C24/E71 buffer memory.
The data arrangement and contents of the areas marked with "∗" in the control procedure diagram differ depending on the module, communication frame and format used.
See the detailed information described in Section 3.1.

[Control procedure]

(1) Reading 10 words from the contents of the addresses 78H to 81H (120 to 129) of the buffer memory area while communicating in ASCII code



(2) Reading 10 words from the contents of the addresses 78H to 81H (120 to 129) of the buffer memory area while communicating in binary code




**POINT**

Designate the head address and word length within the following range.

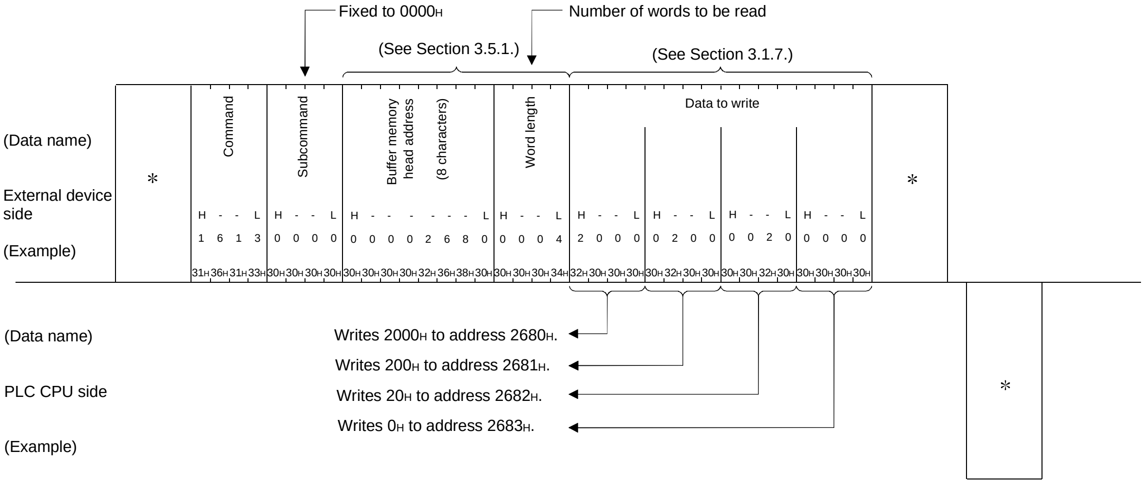
- Head address : 0H ≤ head address ≤ 2307H or 7FFFH
  - Q series C24 : 2307H
  - Q series E71 : 7FFFH
- Word length : 1H ≤ word length ≤ 1E0H (480)
- Access range: : (Head address + word length - 1) ≤ 2307H or 7FFFH

### 3.4.3 Writing to buffer memory (command: 1613)

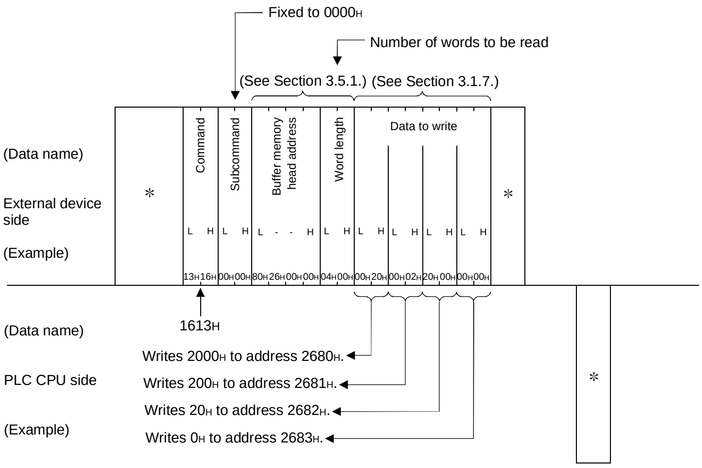
The examples shwon in this section explain the control procedure for carrying out a batch write of the Q series C24/E71 buffer memory.
The data arrangement and contents of the areas marked with "∗" in the control procedure diagram differ depending on the module, communication frame and format used.
See the detailed information described in Section 3.1.

[Control procedure]

(1) Writing four words to the addresses 2680H to 2683H (9856 to 9859) of the buffer memory while communicating in ASCII code.




(2) Writing four words to the addresses 2680H to 2683H (9856 to 9859) of the buffer memory while communicating in binary code.




**POINT**

Designate the head address and word length within the following range.

- Head address : 0H ≤ head address ≤ 2307H or 7FFFH
  Q series C24 : 2307H
  Q series E71 : 7FFFH
- Word length : 1H ≤ word length ≤ 1E0H (480)
- Access range: : (Head address + word length - 1) ≤ 2307H or 7FFFH

## 3.5 Reading from and Writing to the Buffer Memory of an Intelligent Function Module

The examples shown in this section explain the control procedure for reading/writing data in the buffer memory of an intelligent function module (∗1).
This command accesses the buffer memory of an intelligent function module using byte units.

∗1 The Intelligent function modules considered here also include the following modules connected to the external devices and the special function modules of the A/QnA series mentioned in Section 3.4.
- Q series C24 (including multidrop connection)
- Q series E71

### 3.5.1 Commands and buffer memory

This section explains the commands used and the buffer memory address designated by the control procedure when reading and writing in the intelligent function module.

(1) Commands

| Function | Command (subcommand) | Processing | Number of points processed per communication | PLC CPU status | | | Reference section |
|---|---|---|---|---|---|---|---|
| | | | | During STOP | During RUN | | |
| | | | | | Write allow setting | Write prohibit setting | |
| Batch read | 0601 (0000) | Reads from buffer memory. | 1920 words (960 bytes) | ○ | ○ | ○ | Section 3.5.3 |
| Batch write | 1601 (0000) | Writes to buffer memory. | | ○ | ○ | ○ | Section 3.5.4 |

○ in the PLC CPU status column in the table above indicates that execution is possible.

(2) Buffer memory and access unit
The buffer memory designated with this function should be specified by the method described in Section 3.5.2.
One address consists of one word (16 bits), but in this function reading and writing are performed in byte units.
∗ In case of the Q series C24, reading and writing are performed in byte units regardless of the word/byte units designation.

(3) Contents of the character area
Here the contents of the character area when the external device reads from and writes to the buffer memory of the intelligent function module, are explained.

(a) Head address
This value designates the head address of the area from which data is read (or written).
Part 3) below shows how to specify the head address.
The accessible modules and buffer memory head addresses are listed in Section 3.5.2.

1) Data communication in ASCII code
The address of the head area is converted to 8-digit (hexadecimal) ASCII code and sequentially transmitted beginning from the most significant digit ("0").
(Example)
When the head address is $1E1_H$: It is converted to "000001E1" and sequentially transmitted beginning from the first "0."

2) Data communication in binary mode
A 4-byte value indicating the head address is sequentially transmitted beginning from the low byte (L: bits 0 to 7).

(Example)

When the head address is 1E1H: It is converted to 000001E1H and sequentially transmitted beginning from E1H.

3) The head address is designated according to the following method when reading from and writing to the buffer memory of an intelligent function module.
The configuration of the buffer memory of an intelligent function module is 16 bits (one word) per one address. Reading and writing between the PLC CPU and an intelligent function module is performed using FROM/TO commands, etc.
However, one address equals to eight bits (one byte) when the external device reads to and writes from the buffer memory of an intelligent function module using the commands shown in Section 3.5.1. via the Q series C24/E71.
The address (hexadecimal) designated by the external device is specified by the address obtained by converting the address for the FROM/TO commands as follows.

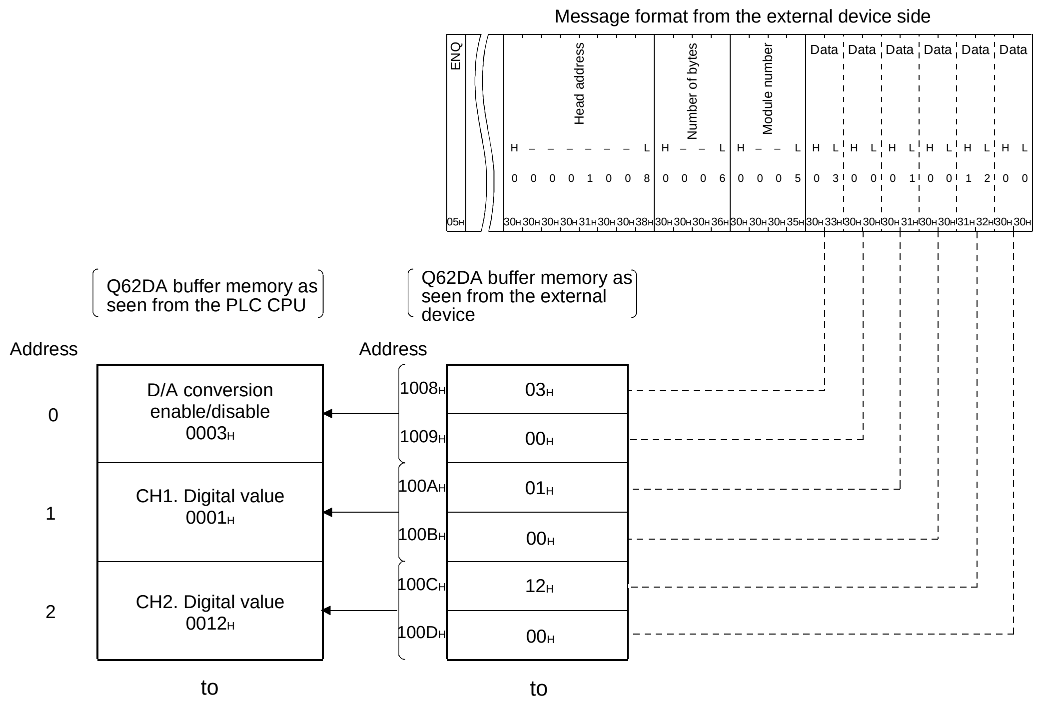
| |
|---|
| Head address (hexadecimal) = {convert (Address for FROM/TO commands $\times$ 2)} into hexadecimal and add buffer memory head address. |

The following is an example of data format for Q62DA when the intelligent function module buffer memory is accessed from the external device.

(Example)

Designating the FROM/TO command address 0 (D/A conversion enable/disable) for an Q62DA

Head address 1008H = FROM/TO instruction address 0 $\times$ 2 (0H) + buffer memory head address 1008H

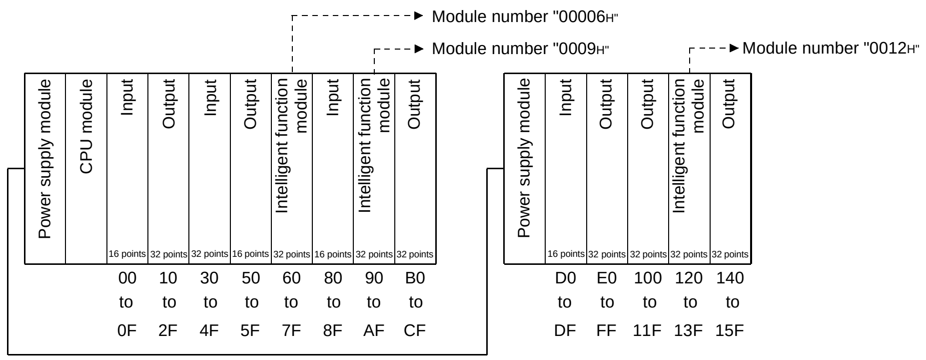



**POINT**

The intelligent function module buffer memory head addresses ("1008H" for the above Q62DA) when "head address" in the message is designated, is shown in Section 3.5.2.

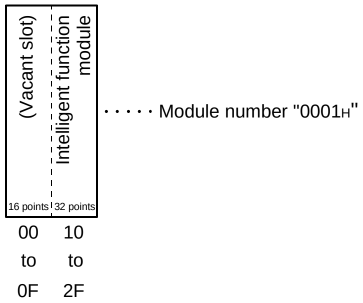
(b) Number of bytes
This value designates 2 bytes × the number of addresses of the area from which data is read (or written) and is designated by even byte.
1) Data communication in ASCII mode
The number of addresses × 2 (2 to 1920) is converted to 4-digit ASCII code (hexadecimal), used, and sequentially transmitted beginning from the most significant digit ("0").
2) Data communication in binary mode
A 2-byte value indicating the number of addresses × 2 (2 to 1920) is transmitted beginning from the low byte (L: bit 0 to 7).

(c) Module number
This is used to designate the intelligent function module that will read, or write, the data.
It is shown in 3) how to designate the module number.
The accessible modules, buffer memory head address, and modules when the accessible module is installed on to slot 0, are shown in Section 3.5.2.
1) Data communication in ASCII mode
The applicable intelligent function module input/output signal is expressed in four digits, and the first three digits are converted to 4-digit ASCII code (hexadecimal) and sequentially transmitted beginning from the first digit.
(Example)
The intelligent function module input/output signal is $0080_H$ to $009F_H$
The module No. is converted to "0008" and is sequentially transmitted beginning from "0."
2) Data communicaton in binary mode
The applicable intelligent function module input/output is expressed in four digits, and the 2-byte value of the first three digits are transmitted sequentially in the order low byte (L: bit 0 to 7) to high byte (H: bit 8 to 15).
(Example)
The intelligent function module input/output signal is $0080_H$ to $009F_H$
The module No. is converted to $0008_H$ and is transmitted in order from $08_H$ and $00_H$.
3) The module number should be designated in the following way when reading from or writing to the intelligent function module buffer memory.
• The module number should be designated as the head input/output signal allocated to the relevant intelligent function module in the installable station.
• For an intelligent function module occupying two slots, the value is designated as the head input/output signal of the slot on the intelligent function module side.
(When intelligent function modules occupy one slot)

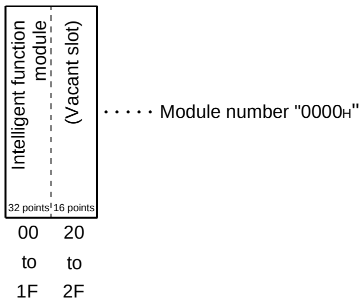


(When modules occupy two slots)
How many points an intelligent function module occupying two slots can occupy by each slot is predetermined for each module.
The module number is designated as the first three digits of the four digits that express the head address of the slot side allocated to a special function module. For information regarding the allocation of each module per slot, refer to the corresponding intelligent function module user's manual.

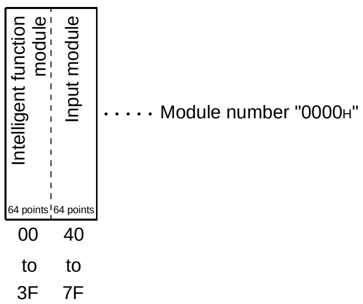
a) For modules that allocate the front-half slots as vacant slots. (AD72, A84AD, etc.)



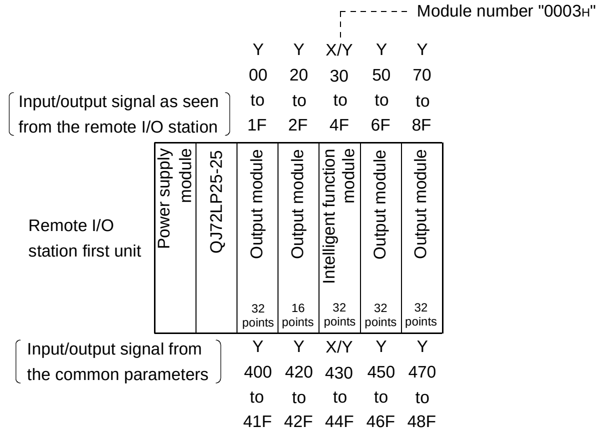

b) For modules that allocate the last-half slots as vacant slots. (A61LS, etc.)




c) For modules where both an intelligent functiol module and an input/output function are allocated.
(A81CPU, etc.)



(Intelligent function modules of a network system remote I/O station)
The module numbers of the MELSECNET/H, MELSECNET/10 remote I/O station special function modules all become the first three digits of the four digits that express the head number of the "input/output signal as seen from the remote I/O station".
Designate the value as the "input/output signal as seen from the remote I/O station" regardless of the contents of the common parameter set by the MELSECNET/H, MELSECNET/10 remote I/O network master station.




The module number for an intelligent module that occupies two slots should be designated using the method described in the previous page (When modules occupy two slots).

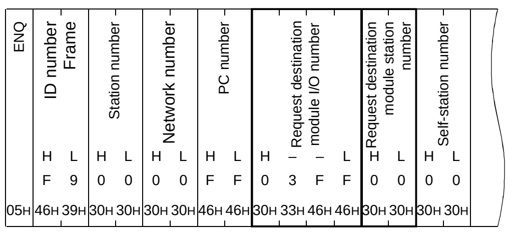
(d) Data read and written
This is the data read from or written to an intelligent function module and it has the order of data corresponding to the number of data shown in (b) above (a maximum of 1920 bytes).

1) Data communication in ASCII code
Each data code is converted to 2-digit (hexadecimal) ASCII code and sequentially transmitted beginning from the most significant digit.
(Example)
The data read/written is 12H
The data read/written is converted to "12" and transmitted from "1."

2) Data communication in binary code
Each data code is transmitted from the head address.

**POINT**

When using command "0601" and "1601," designate the following values as the request destination module I/O number and the module station number when data is communicated in a QnA compatible 4C frame.
Request destination module I/O number: "03FFH"
Request destination module station number: "00H"
(For QnA compatible 4C frame format 1)



### 3.5.2 Accessible intelligent function modules

The following table summarizes the intelligent function modules which can be read from/written to the buffer memory using commands "0601" and "1601", along with their "head address" and "module number" values.

REMARK

The "buffer memory head address" and "module number when loaded in slot 0" values shown in the table are used to designate the "head address" and "module number" in the command messages.

(1) Accessible intelligent function module mode names

| Module model name | Buffer memory head address (Hexadecimal) | Module number when loaded in slot 0 |
|---|---|---|
| Model Q62AD-DGH, Q64AD (-GH), Q68ADV/ADI Analog-Digital Conversion Module | $1008_H$ | $0000_H$ |
| Model Q62DA (-FG), Q64DA, Q68DAV/Q68DAI Digital-Analogue Conversion Module | $1008_H$ | |
| Model Q64TCTT/Q64TCRT Temperature Control module | $1000_H$ | |
| Model Q64TCTTBW/Q64TCRTBW Temperature Control module | $1000_H$ | $0010_H$ |
| Model Q64TD, Q64RD Thermocouple Input Module (Function version B) | $2000_H$ | $0000_H$ |
| Model Q64TD, Q64TDV-GH, Q64RD(-G) Thermocouple Input Module (Function version C) | $8000_H$ | |
| Model QD51 (-R24) Intelligent Communication module | $10000_H$ | |
| Model QD60P8-G Channel Isolated Pulse Input Module | $2000_H$ | |
| Model QD62, QD62E, QD62D High speed counter module | $3C_H$ | |
| Model QD70P4/P8 Positioning module | $5000_H$ | |
| Model QD75P1/P2/P4, QD75D1/D2/D4, QD75M1/M2/M4 Positioning module | $10000_H$ | |
| Model QJ61BT11 CC-Link System Master/Local Module | $10000_H$ | |
| Model QJ71C24N (-R2/R4), QJ71C24 (-R2) Serial Communication Module | $10000_H$ | |
| Model QJ71DN91 DeviceNet Master-Slave Module | $10000_H$ | |
| Model QJ71E71-100/-B5/-B2 Ethernet interface module | $10000_H$ | |
| Model QJ71FL71 (-B2) -F01 FL-net (OPCN-2) Interface Module | $10000_H$ | |
| Model QJ71PB92D PROFIBUS-DP Interface module | $10000_H$ | |
| Model AD61 (S1) High-speed Counter Module | $80_H$ | |
| Model A616AD Analog-Digital Conversion Module | $10_H$ | |
| Model A616DAI/DAV Digital-Analogue Conversion Module | $10_H$ | |
| Model A616TD Temperature-Digital Conversion Module | $10_H$ | |
| Model A62DA(S1) Digital-Analogue Conversion Module | $10_H$ | |
| Model A68AD(S2) Analog-Digital Conversion Module | $80_H$ | |
| Model A68ADN Analog-Digital Conversion Module | $80_H$ | |
| Model A68DAV/DAI Digital-Analogue Conversion Module | $10_H$ | |
| Model A68RD3/4 Temperature-Digital Conversion Module | $10_H$ | |
| Model A84AD Analog-Digital Conversion Module | $10_H$ | $0001_H$ |
| Model A81CPU PID Control Module | $200_H$ | $0000_H$ |
| Model A61LS Position Detection Module | $80_H$ | |
| Model A62LS (S5) Position Detection Module | $80_H$ | $0001_H$ |
| Model AJ71PT32 (S3) /AJ71T32-S3 MELSECNET/MINI (-S3) Master Module | $20_H$ | $0000_H$ |
| Model AJ61BT11 CC-Link System Master/Local Module | $2000_H$ | |
| Model AJ71C22 (S1) Multidrop Link Module | $1000_H$ | |
| Model AJ71C24 (S3/S6/S8) Computer Link Module | $1000_H$ | |
| Model AJ71UC24 Computer Link Module | $400_H$ | |
| Model AD51 (S3) Intelligent Communication Module | $800_H$ | $0001_H$ |
| Model AD51H (S3) Intelligent Communication Module | $800_H$ | |

| Module model name | Buffer memory head address (Hexadecimal) | Module number when loaded in slot 0 |
|---|---|---|
| Model AJ71C21 (S1) Terminal Interface Module | 400H | 0000H |
| Model AJ71B62 B/NET Interface Module | 20H | |
| Model AJ71P41 SUMINET Interface Module | 400H | |
| Model AJ71E71 (S3) Ethernet Interface Module | 400H | |
| Model AD51FD (S3) External Problem Diagnostic Module | 280H | 0001H |
| Model AD57G (S3) Graphic Controller Module | 280H | |
| Model AS25VS Vision Sensor Module | 100H | |
| Model AS50VS Vision Sensor Module | 100H | |
| Model AS50VS - GN Vision Sensor Module | 80H | |
| Model AD59 (S1) Memory Card Interface Module | 1800H (＊1) | 0000H |
| Model AD70 (D) (S2) Positioning Module | 80H | |
| Model AD71 (S1/S2/S7) Positioning Module | 200H | |
| Model AD72 Positioning Module | 200H | 0001H |
| Model AD75P1/P2/P3 (S3)/AD75M1/M2/M3 Positioning Module | 800H | 0000H |
| Model AJ61QBT11 CC-Link System Master/Local Module | 2000H | |
| Model AJ71QC24(N) (R2, R4) Serial Communication Module | 4000H | |
| Model AJ71QE71 (B5) Ethernet Interface Module | 4000H | |
| Model A1SD61/A1SD62 (E/D) High-Speed Counter Module | 10H | |
| Model A1S62DA Digital-Analog Conversion Module | 10H | |
| Model A1S62RD3/4 Temperature-Digital Conversion Module | 10H | |
| Model A1S64AD Analog-Digital Conversion Module | 10H | |
| Model A1SJ71 (U) C24-R2 Computer Link Module | 400H | |
| Model A1SJ71 (U) C24-PRF Computer Link Module | 400H | |
| Model A1SJ71 (U) C24-R4 Computer Link Module | 400H | |
| Model A1SJ71E71 (S3) Ethernet Interface Module | 400H | |
| Model A1SD70 Single Axis Positioning Module | 80H | |
| Model A1SD71-S2/S7 Positioning Module | 200H | 0001H |
| Model A1SD75P1/P2/P3 (S3)/A1SD75M1/M2/M3 Positioning Module | 800H | 0000H |
| Model A1S63ADA Analog I/O module | 10H | |
| Model A1S64TCTT (BW)-S1 Temperature Controller Module | 20H | |
| Model A1S64TCRT (BW)-S1 Temperature Controller Module | 20H | |
| Model A1S62TCTT (BW)-S2 Temperature Controller Module | 20H | |
| Model A1S62TCRT (BW)-S2 Temperature Controller Module | 20H | |
| Model A1SJ71PT32-S3 MELSECNET/MINI-S3 Master Module | 20H | |
| Model A1SJ61BT11 CC-Link System Master/Local Module | 2000H | |
| Model A1SJ71QC24(N) (R2) Serial Communication Module | 4000H | |
| Model A1SJ71QE71-B2/B5 Ethernet Interface Module | 4000H | |
| Model A1SJ61QBT11 CC-Link System Master/Local Module | 2000H | |

＊1 Changing the memory card bank using the I/O signals Y10 and Y11 between the PLC CPU and the AD59 (S1) makes it possible to read/write from the memory card access memory area only.

## (2) Example of head addresses of an intelligent function module designated by external device

The following table shows head addresses designated by external device when Q62DA is used.

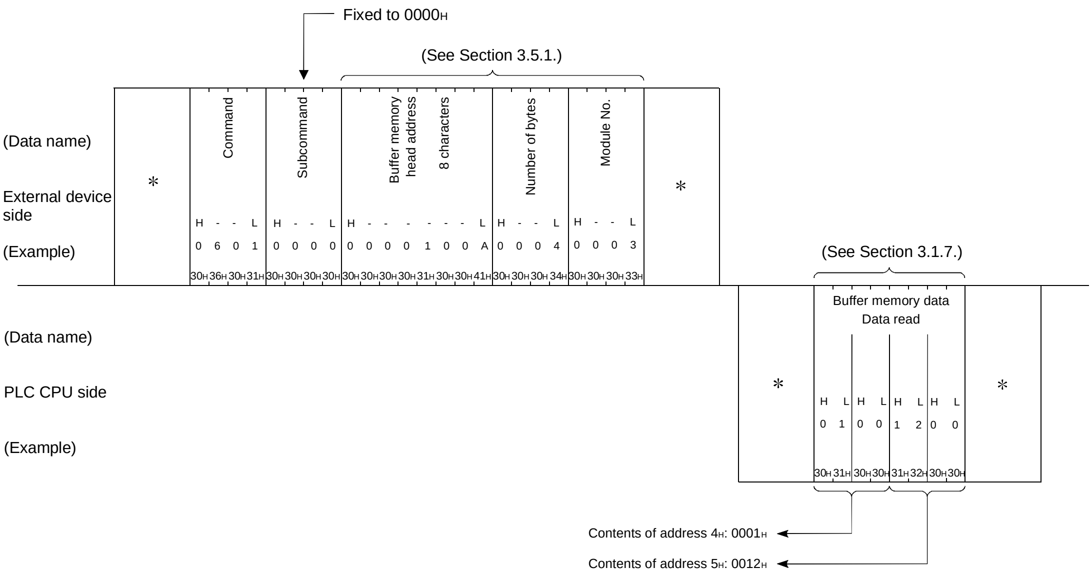
| Buffer memory | Head address | Address used in FROM/TO command |
|---|---|---|
| D/A conversion enable/disable | 1008H | 0H |
| | 1009H | |
| CH.1 Digital value | 100AH | 1H |
| | 100BH | |
| CH.2 Digital value | 100CH | 2H |
| | 100DH | |
| System area | | 3H to 10H |
| Offset / gain adjustment value specification | 102CH | 18H |

### 3.5.3 Reading the buffer memory of an intelligent function module (command: 0601)

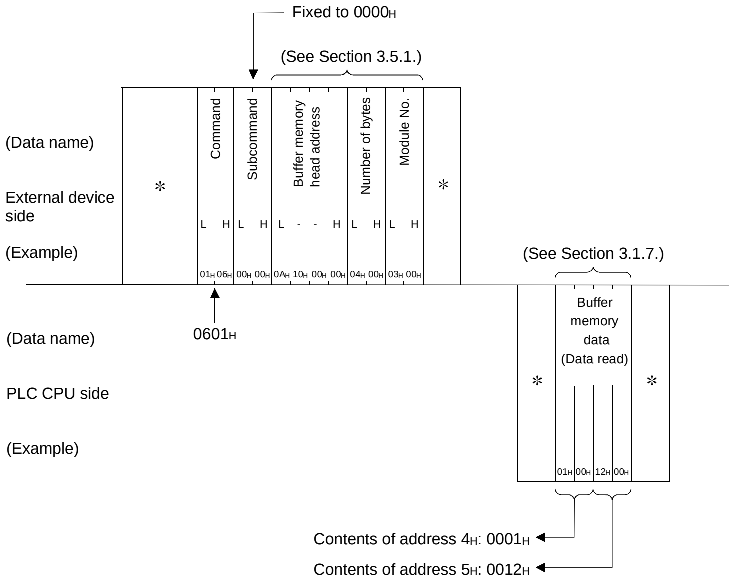
The examples shown in this section explain the control procedure for reading from the buffer memory of an intelligent function module.
The order and content of data items of the areas marked by "✻" shown in the control procedure diagram differ depending on the module used as well as the frame and format used for communication.
See the detailed information described in Section 3.1.

[Control procedure]

(1) Reading four bytes from the buffer memory addresses $1_H$ to $2_H$ of the Q62DA, whose input/output signal is $30_H$ to $4F_H$ (module number: $03_H$) while communicating in ASCII code.



(2) Reading four bytes from the buffer memory addresses $1_H$ to $2_H$ of the Q62DA, whose input/output signal is $30_H$ to $4F_H$ (module number: $03_H$) while communicating in binary code.




**POINT**

(1) Designate the head address and number of bytes within the following ranges.
- Head address : Address range of the target intelligent function module
- Number of bytes : 2 ($2_H$) ≤ Number of bytes ≤ 1920 ($780_H$)

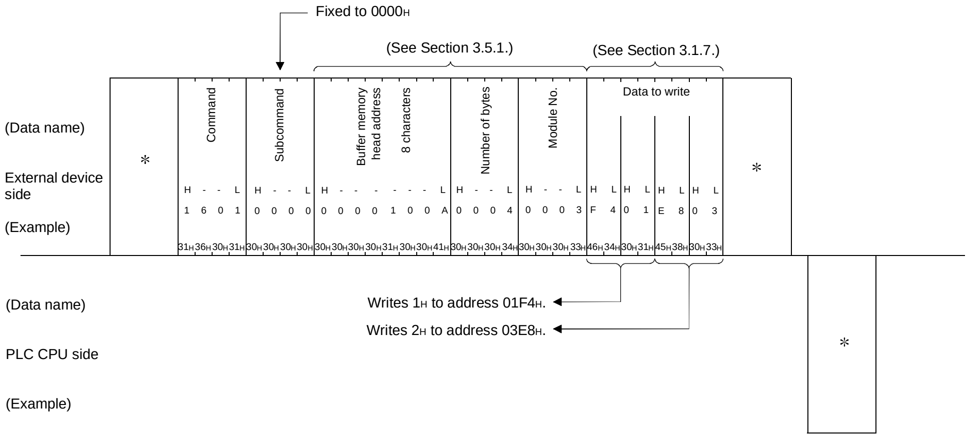
(2) One data value may extend over two or three bytes depending on the intelligent functional module; therefore, consult the manual of the target before designating the number of bytes and data to be written.

### 3.5.4 Writing to the buffer memory of an intelligent function module (command: 1601)

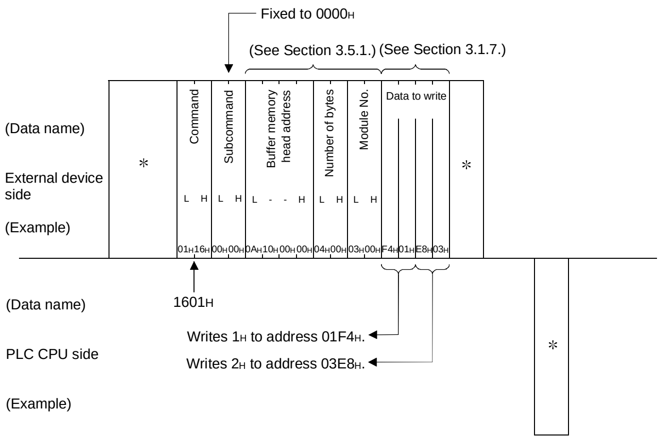
The example shown in this section explain the control procedure for writing to the buffer memory of an intelligent function module.
The order and content of data items of the areas marked by "＊" shown in the control procedure diagram differ depending on the module used as well as the frame and format used for communication.
See the detailed information described in Section 3.1.

[Control procedure]

(1) Writing four bytes to the buffer memory addresses $1_H$ to $2_H$ of the Q62DA, whose input/output signals are $30_H$ to $4F_H$ (module number: $03_H$) while communicating in ASCII code.




(2) Writing four bytes to the buffer memory addresses $1_H$ to $2_H$ of the Q62DA, whose input/output signals are $30_H$ to $4F_H$ (module number: $03_H$) while communicating in binary code.




**POINT**

(1) Designate the head address and number of bytes within the following ranges.
- Head address : Address range of the target intelligent function module
- Number of bytes : 2 ($2_H$) ≤ Number of bytes ≤ 1920 ($780_H$)

(2) One data value may extend over two or three bytes depending on the intelligent function module; therefore, consult the manual of the target before designating the number of bytes and data to be written.

## 3.6 PLC CPU Status Control

This function is used to remotely RUN/STOP/PAUSE/RESET a Q/QnACPU, clear the device memory of a QnACPU, and read the model name of a Q/QnACPU from an external device.

### 3.6.1 Commands, control contents, and character area contents

In this section the commands, control contents, and character area (data section in binary code communication) in the control procedure when controlling the status of the PLC CPU, are described.

(1) Commands

<table>
<tr><th rowspan="3">− Function</th><th rowspan="3">Command (subcommand)</th><th rowspan="3">Processing</th><th colspan="2">Number of points processed per communication</th><th colspan="3">PLC CPU status</th><th rowspan="3">Reference section</th></tr>
<tr><th rowspan="2">Q/QnA CPU</th><th rowspan="2">Other than Q/QnA CPU</th><th rowspan="2">During STOP</th><th colspan="2">During RUN</th></tr>
<tr><th>Write allow setting</th><th>Write prohibit setting</th></tr>
<tr><td>Remote RUN</td><td>1001 (0000)</td><td>Requests remote RUN (execute operation)</td><td rowspan="6">(For one station)</td><td rowspan="6">(Not allowed)</td><td rowspan="3">○</td><td rowspan="3">○</td><td rowspan="3">○</td><td>Section 3.6.2</td></tr>
<tr><td>Remote STOP</td><td>1002 (0000)</td><td>Requests remote STOP (stop operation)</td><td>Section 3.6.3</td></tr>
<tr><td>Remote PAUSE</td><td>1003 (0000)</td><td>Requests remote PAUSE (pause operation)<br>(Output status is kept)</td><td>Section 3.6.4</td></tr>
<tr><td>Remote latch clear</td><td>1005 (0000)</td><td>Request remote latch clear (clear a device memory) when the PLC CPU is in the STOP status.</td><td>○</td><td>×</td><td>×</td><td>Section 3.6.6</td></tr>
<tr><td>Remote RESET</td><td>1006 (0000)</td><td>Request remote RESET (start execution of an operation) when the PLC CPU is in the STOP status.</td><td>○</td><td>×</td><td>×</td><td>Section 3.6.5</td></tr>
<tr><td>CPU model name read</td><td>0101 (0000)</td><td>Request to read the model name of the PLC CPU.</td><td>○</td><td>○</td><td>○</td><td>Section 3.6.7</td></tr>
</table>

○ in the PLC CPU status column above indicates that execution is possible.

**POINT**

(1) If the QnACPU is powered off or reset after executing a remote RUN/STOP/PAUSE from an external device, the remote data is cleared.
(2) When there is a system protection on the Q/QnACPU, its status control cannot be made from the external device. An NAK message or response massage indicating abnormal completion is returned in response to each request.
(3) It is recommended to control the status of the PLC CPU of a local station using one of the following methods when the Q series E71 is loaded on the station.
    1) Control the status of the PLC CPU using an automatic open UDP port.
    2) Control the status of the PLC CPU using a passive open connection which is set to "Always wait for open" in the initial timing setting of the "Ethernet operation settings."

(2) Contents of the character area

Here, the contents of the character area when an external device controls the status of the Q/QnACPU, are described.

(a) Mode

This is used to force the Q/QnACPU to execute a remote RUN/remote PAUSE.

When a remote RUN/remote PAUSE could not be executed on the controlled Q/QnACPU due to problems in the Q series C24/E71 station or the external device that requested the remote STOP/PAUSE of the Q/QnACPU, a forced execution from another external device can be used to ensure that a remote RUN/remote PAUSE is executed.

1) Data communication in ASCII code
The designated values shown below are converted to a 4-digit (hexadecimal) ASCII code and sequentially transmitted beginning from the most significant digit ("0").

2) Data communication in binary code
The 2-byte value shown below is sequentially transmitted beginning from the low byte (L: bits 0 to 7).

3) The modes that can be designated are as follows.

| Designated value | Processing |
|---|---|
| $0001_H$ | Do not execute forcibly.<br>When another external device is issuing a remote STOP/PAUSE command, the remote RUN/remote PAUSE is not executed. |
| $0003_H$ | Force execution<br>Remote RUN/remote PAUSE is executed even if another device has issued a remote STOP/PAUSE command.<br>(Can only be designated in the remote RUN or remote PAUSE status.) |

4) When status control other than remote RUN and remote PAUSE is executed, "0001" or "$0001_H$" is transmitted.

(b) Clear mode

This value is used to designate a clearing (initialization) of the buffer memory of the Q/QnACPU when the Q/QnACPU operation is initiated by remote RUN.

After the designated buffer memory is cleared, the Q/QnACPU operates according to the parameter settings (PC file setting → device initial value).

1) Data communication in ASCII code
The designated value shown below is converted to a 2-digit (hexadecimal) ASCII code and sequentially transmitted beginning from the most significant digit ("0").

2) Data communication in binary mode
The 1-byte value shown below is transmitted.

3) The clear modes can be designated as follows.

| Designated value | Processing |
|---|---|
| $00_H$ | Do not clear device memory |
| $01_H$ | Clear the device memories outside the latch range |
| $02_H$ | Clear all device memories, including those in the latch range |

4) When status control other than remote RUN is executed, it is not necessary to designate a clear mode.

(c) Fixed values

1) When data is communicated in ASCII code, "00" is transmitted.

2) When data is communicated in binary mode, the 1-byte value "$00_H$" is transmitted.

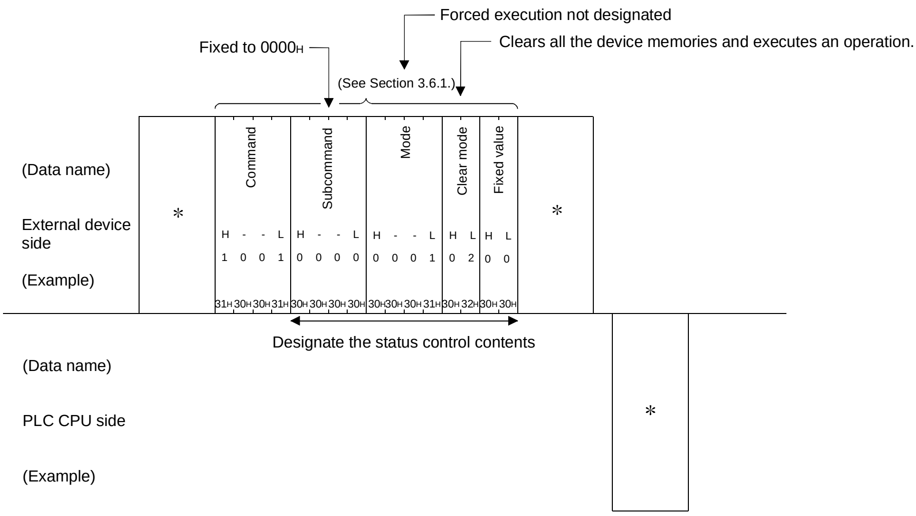
3) When status control other than remote RUN is executed, it is not necessary to designate a fixed value.

### 3.6.2 Remote RUN (command: 1001)

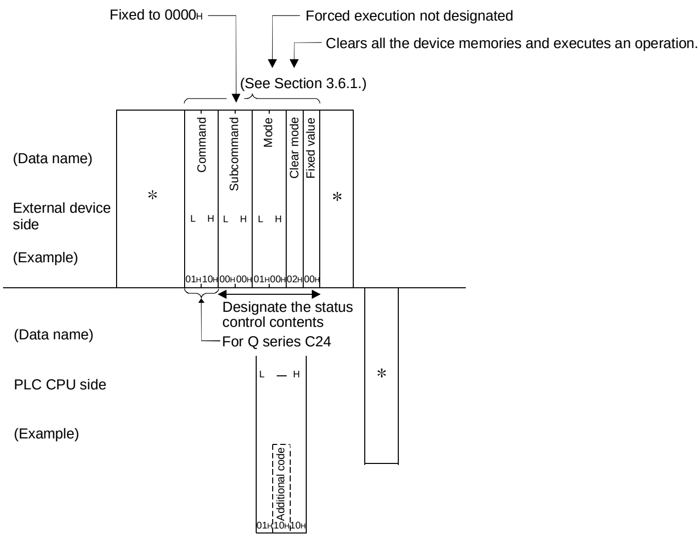
The examples shown in this section explain the control procedure of remote RUN.
The order and content of data items of the areas marked by "*" shown in the control procedure diagram differ depending on the module used as well as the frame and format used for communication.
See the detailed information described in Section 3.1.

[Control procedure]

(1) Executing remote RUN while communicating in ASCII code




(2) Executing remote RUN while communicating in binary code




**POINT**

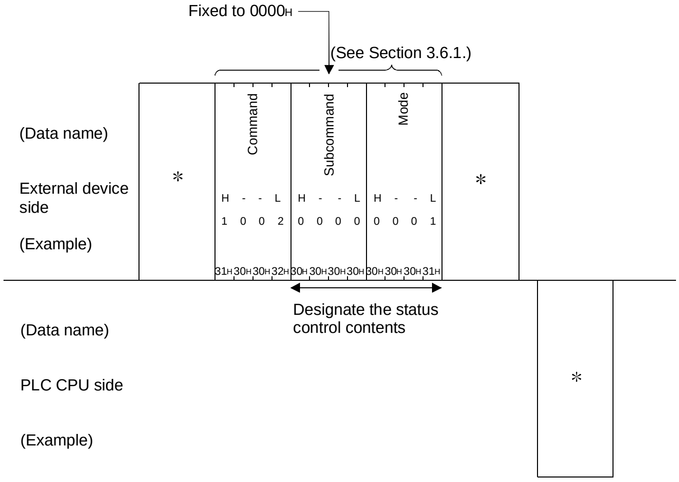
When "Mode" is not designated to force the execution, the target Q/QnACPU will not be placed in the RUN status if another external device has already issued a remote STOP/PAUSE command to it.

### 3.6.3 Remote STOP (command: 1002)

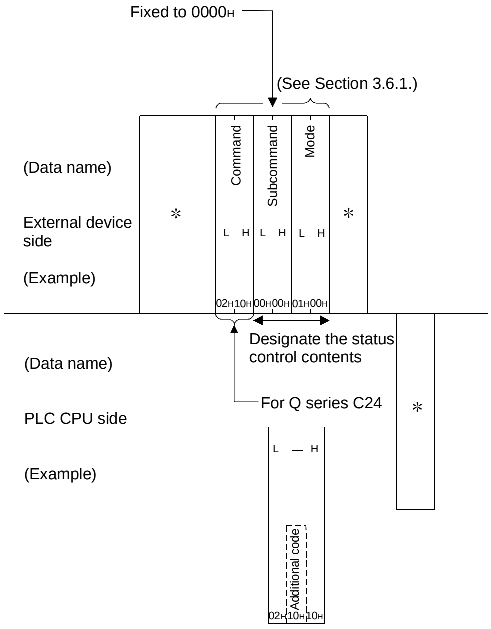
The examples shown in this section explain the control procedure of remote STOP.
The order and content of data items of the areas marked by "*" shown in the control procedure diagram differ depending on the module used as well as the frame and format used for communication.
See the detailed information described in Section 3.1.

[Control procedure]

(1) Executing remote STOP while communicating in ASCII code




(2) Executing remote STOP while communicating in binary code

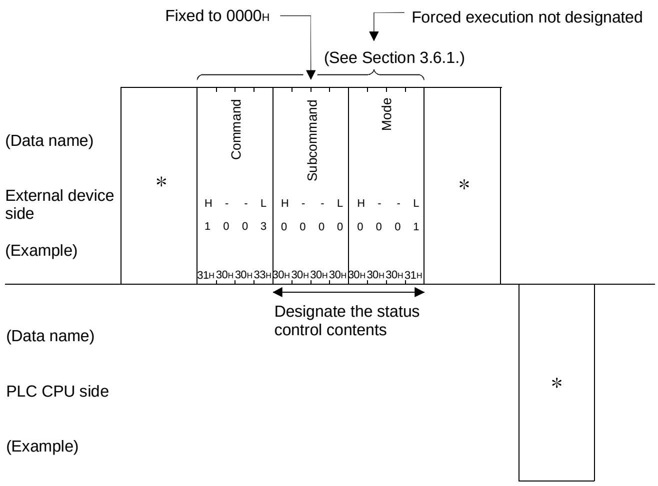


### 3.6.4 Remote PAUSE (command: 1003)

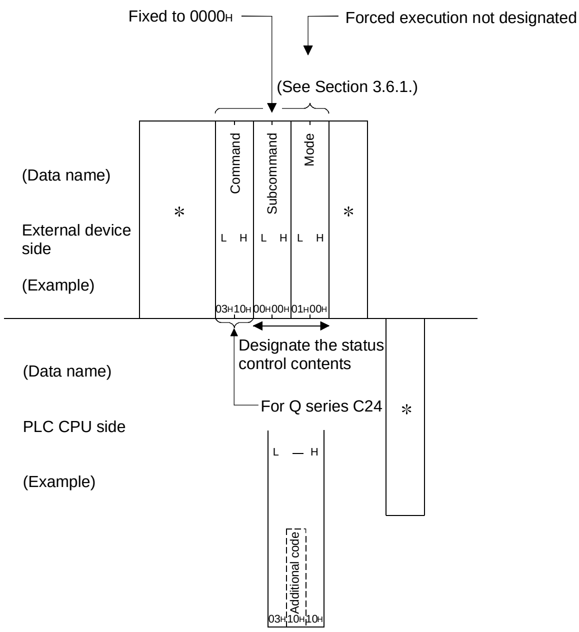
The examples shown in this section explain the control procedure of remote PAUSE.
The order and content of data items of the areas marked by "*" shown in the control procedure diagram differ depending on the module used as well as the frame and format used for communication.
See the detailed information described in Section 3.1.

[Control procedure]

(1) Executing remote PAUSE while communicating in ASCII code




(2) Executing remote PAUSE while communicating in binary code

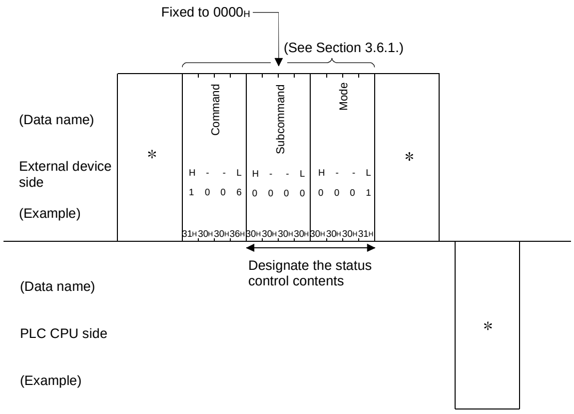


### 3.6.5 Remote RESET (command: 1006)

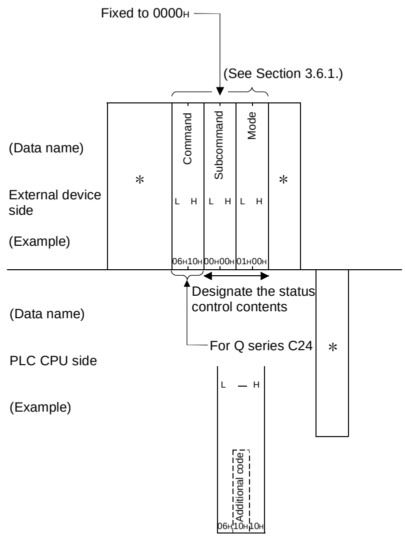
The examples shown in this section explain the control procedure of remote RESET.
The order and content of data items of the areas marked by "*" shown in the control procedure diagram differ depending on the module used as well as the frame and format used for communication.
See the detailed information described in Section 3.1.

[Control procedure]

#### (1) Executing remote RESET while communicating in ASCII code




#### (2) Executing remote RESET while communicating in binary code




**POINT**

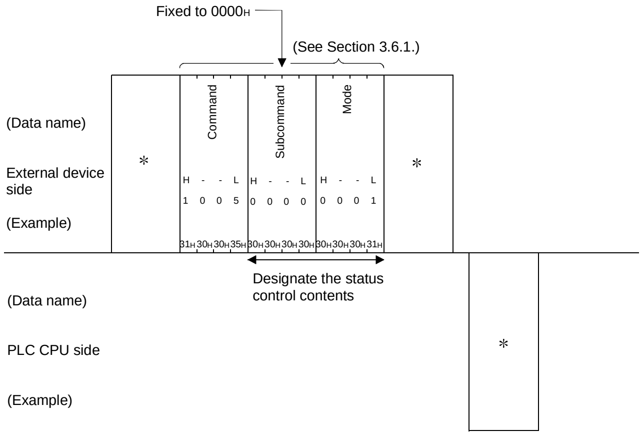
(1) Issue the remote RESET command to the target Q/QnACPU when it is in the STOP status due to an error.
(2) The remote RESET can also be executed when the Q/QnACPU is operating normally.
When the remote RESET is executed, the Q series C24/E71 is also reset and is reboosted in the same status as when the power is turned on.
(3) When the remote RESET is performed, be sure to set the remote reset to "Allow" with the PLC parameter setting of the GX Developer (PLC system setiting).

### 3.6.6 Remote latch clear (command: 1005)

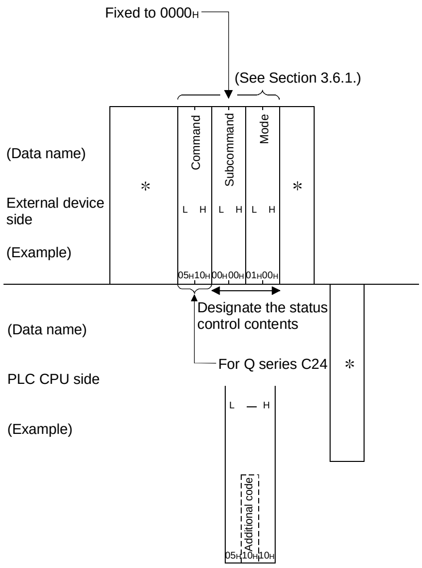
The examples shown in this section explain the control procedure of remote latch clear.
The order and content of data items of the areas marked by "∗" shown in the control procedure diagram differ depending on the module used as well as the frame and format used for communication.
See the detailed information described in Section 3.1.

[Control procedure]

(1) Executing remote latch clear while communicating in ASCII code



(2) Executing remote latch clear while communicating in binary code



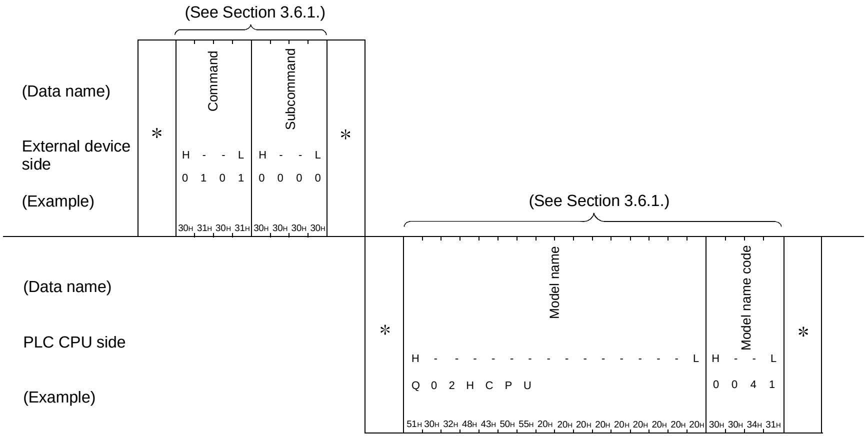
| POINT |
| --- |
| (1) Issue the remote latch clear command to the target Q/QnACPU after it is placed in the STOP status.<br>(2) Remote latch clear cannot be executed if the target Q/QnACPU is placed in the remote STOP/PAUSE status by a request from another external device; in this case, the command will not complete normally. |

### 3.6.7 CPU model name read (command: 0101)

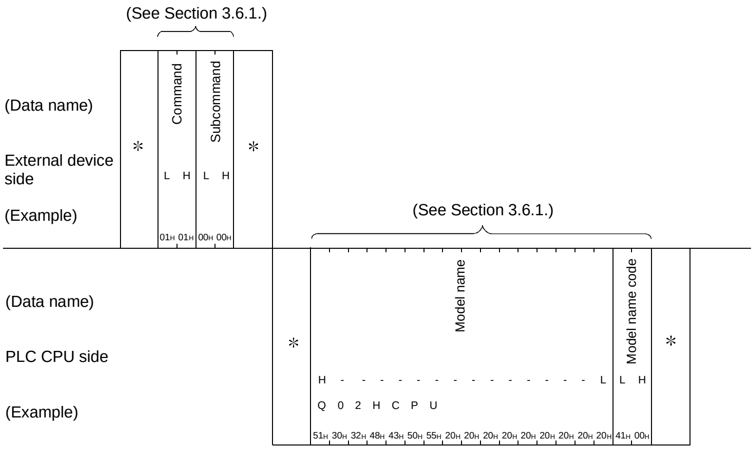
The examples shown in this section explain the control procedure for reading the model name of the PLC CPU.
The order and content of data items of the areas marked by "✻" shown in the control procedure diagram differ depending on the module used as well as the frame and format used for communication.
See the detailed information described in Section 3.1.

[Control procedure]

(1) Data communication in ASCII code




(2) Data communication in binary code



✻ This function reads the model name of the target PLC in the form of model name code and model name (for monitor).
The model name code and the model name can be read as follows.

| Model name of PLC | Content to be read | |
|---|---|---|
| | Model name code (hexadecimal) | Model name (for monitor) |
| Q00JCPU | $250_H$ | Q00JCPU |
| Q00CPU | $251_H$ | Q00CPU |
| Q01CPU | $252_H$ | Q01CPU |
| Q02CPU | $41_H$ | Q02CPU |
| Q02HCPU | $41_H$ | Q02HCPU |
| Q06HCPU | $42_H$ | Q06HCPU |
| Q12HCPU | $43_H$ | Q12HCPU |
| Q25HCPU | $44_H$ | Q25HCPU |
| Q12PHCPU | $43_H$ | Q12PHCPU |
| Q25PHCPU | $44_H$ | Q25PHCPU |
| Q12PRHCPU | $4B_H$ | Q12PRHCPU |
| Q25PRHCPU | $4C_H$ | Q25PRHCPU |

POINT

(1) Distinguish the PLC CPU model name by the model name code.
(2) When the model name read takes up less space than the number of designated bytes, the Q series C24/E71 adds blank spaces ($20_H$).

## 3.7 Drive Memory Defragmentation (for Other Station QnACPU)

An external device uses this function to perform the following operations on the QnACPU drive of another station that stores the program file containing the parameters and sequence programs.

### (1) Reading status of drive memory usage

Checks the status of the drive memory usage (cluster usage status) of the designated drive.

### (2) Drive memory defragmentation

Increases the contiguous free drive space by defragmenting the drive memory into cluster units when the memory areas containing valid data are scattered throughout the drive memory.

**POINT**

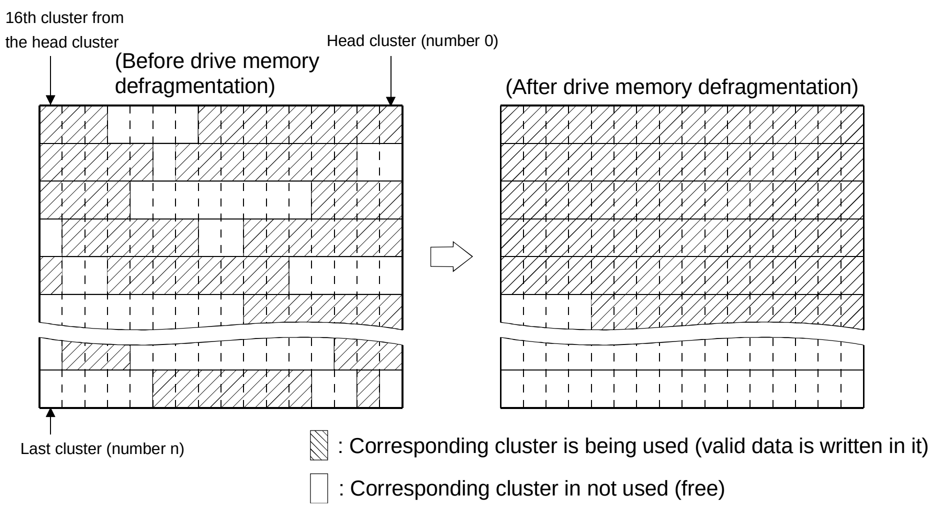
A cluster is the minimum unit when files are stored to drive memory (memory cards, etc.) and the memories containing data are managed by FAT(*1).
The size of one cluster of each QnACPU drive is shown below:

- Internal memory : 4096 bytes
- Other memory : 512 bytes

For example, if an amount of data less then 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.

*1 FAT (File Allocation Table)
Table used by the OS to manage the location of files in the drive memory.

(Illustration of drive memory defragmentation)



### 3.7.1 Commands and contents of character area

This section explains the commands used and the character area (data section in communication in binary code) in the control procedure used to defragment the drive memory.

(1) Commands

| Function | Command (subcommand) | Processing | Number of points processed per communication / Access station-2 (QnACPU described in Section 3.2 ＊8) | PLC CPU status: During STOP | During RUN: Write allow setting | During RUN: Write prohibit setting | Reference section |
|---|---|---|---|---|---|---|---|
| Memory usage status read | 0205 (0000) | Reads the cluster usage status of the driver | (For 256 clusters) | ○ | ○ | ○ | Section 3.7.2 |
| Memory defragmentation | 1207 (0000) | Increases the contiguous free area by defragmenting the drive memory. | (For one station) | ○ | × | × | Section 3.7.3 |

○ in the PLC CPU status column above indicates that execution is possible.

(2) Contents of character area

The following explains what is included in the character area when an external device instructs the QnACPU to defragment the drive memory.

(a) Keyword

Character string (maximum 6 characters) registered to the designated drive by the user. This data allows/prohibits access to that drive.
When a keyword has already registered, designate the same keyword.

1) Data communication in ASCII code

The keyword registered in the designated drive is transmitted as it is.

2) Data communication in binary code

The keyword registered in the designated drive is converted to a 3-byte binary code and sequentially transmitted beginning from the low byte (L: bits 0 to 7).

(Example)

| Registered keyword | Values converted to binary code | Transmission order | Remarks |
|---|---|---|---|
| "012345" | $01_H$, $23_H$, $45_H$ | $45_H$, $23_H$, $01_H$ | Sequentially transmitted beginning from $45_H$. |
| "012300" | $01_H$, $23_H$, $00_H$ | $00_H$, $23_H$, $01_H$ | Sequentially transmitted beginning from $00_H$. |

3) The key words of the character area are as follows when a keyword is not registered in the designated drive.

- In data communication in ASCII code "000000"
- In data communication in binary code $00_H$, $00_H$, $00_H$

(b) Setting flag

This flag indicates whether or not the keyword registered in the designated drive by the user matches the keyword of (a) above.

1) Data communication in ASCII code
The values shown below are converted to a 2-digit (hexadecimal) ASCII code and sequentially transmitted beginning from the most significant digit ("0").
2) Data communication in binary code
The 1-byte value shown below is transmitted.
3) The setting flag can be designated as follows.

| Designated value | Designation |
|---|---|
| $00_H$ | Keyword is invalid (designated by a dummy) |
| $01_H$ | Keyword is valid (the keyword registered in the designated drive is designated). |

(c) Drive name

This is used to read the status of the drive memory usage and designates the QnACPU drive to be defragmented.

1) Data communication in ASCII code
The values shown below, which indicate the drive to be accessed, are converted to a 4-digit (hexadecimal) ASCII code and sequentially transmitted beginning from the most significant digit ("0").
2) Data communication in binary code
The 2-byte values shown below, which indicate the drive to be accessed, are sequentially transmitted beginning from the low byte (L: bits 0 to 7).
3) The drive name can be designated as follows; other values cannot be designated.

| Designated value | Target drive |
|---|---|
| $0000_H$ | Internal memory (built-in RAM) |
| $0001_H$ | RAM area of memory card A |
| $0002_H$ | ROM area of memory card A |
| $0003_H$ | RAM area of memory card B |
| $0004_H$ | ROM area of memory card B |
| $000F_H$ | Drive storing the parameter files currently in use (designated with the QnACPU DIP switch). |

(d) Cluster number

This value designates the head cluster number of the range over which the status of the drive memory usage is to be read. It is designated in multiples of 16 (for hexadecimal, $00_H$, $10_H$, $20_H$...).

1) Data communication in ASCII code
Cluster No. $00_H$ or higher is converted to a 4-digit (hexadecimal) ASCII code and sequentially transmitted beginning from the most significant digit.
2) Data communication in binary code
The 2-byte value that indicates cluster No. $00_H$ or higher is sequentially transmitted beginning from the low byte (L: bits 0 to 7).
3) When the drive memory is defragmented, the cluster number does not have to be designated.

(e) Number of clusters to be read

This value designates the number of clusters within the drive memory range who are to be read. It is designated in multiples of 16 ($10_H$, $20_H$... in hexadecimal representation).

1) Data communication in ASCII code
   A number of clusters in the range from $10_H$ to $100_H$ (16 to 256) is converted to a 4-digit (hexadecimal) ASCII code and sequentially transmitted beginning from the most significant digit ("0").
2) Data communication in binary code
   The 2-byte value that indicates the number of clusters in the range from $10_H$ to $100_H$ (16 to 256) is sequentially transmitted beginning from the low byte (L: bits 0 to 7).
3) When the drive memory is defragmented, the number of clusters to read does not have to be designated.

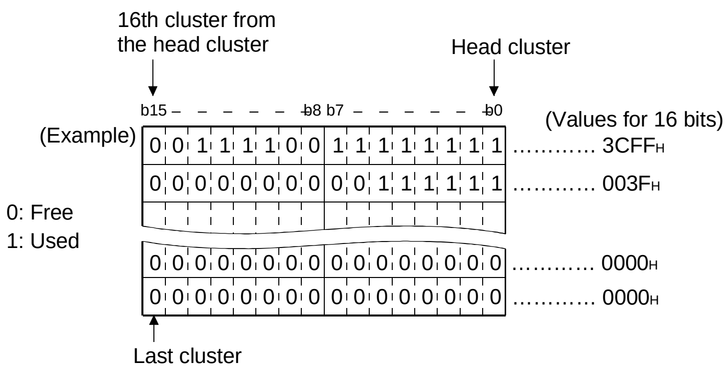
**POINT**

Designate the number of clusters to read within the range of the usable memory size after formatting the drive to be read from. (For the number of bytes in a cluster, see Section 3.7.)
Number of clusters = Usable memory size / Number of bytes in one cluster (4096 or 512)

(f) Free cluster table

This data section (indicating the status of cluster usage) is returned to the external device when the drive memory usage status is read.

1) Data communication in ASCII code
   The value shown below, which indicates the usage status, is converted to an n-digit (hexadecimal) ASCII code and transmitted to the external device side. (16 clusters/4 digits)
2) Data communication in binary code
   An m-byte value that indicates the usage status is transmitted to the external device side.
   (16 clusters/2 bytes)
3) The following items are included in the free cluster table.
   The usage status of each cluster is indicated by one cluster per bit.



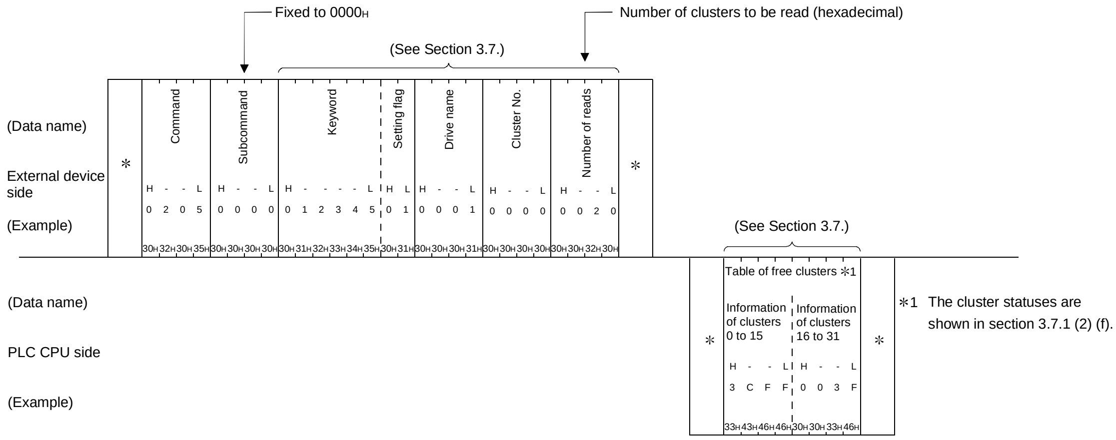

   In the usage status example shown above, the following free cluster table is returned to the external device.
   - 32 clusters are returned while communicating in ASCII code
     "3CFF003F" is returned and transmitted sequentially beginning from "3. "
   - 32 clusters are returned in the data communication in binary code
     $FF_H$, $3C_H$, $3F_H$, and $00_H$ are returned and transmitted sequentially beginning from $FF_H$.
4) When a drive memory is defragmented, a free cluster table is not returned.

### 3.7.2 Reading the status of the drive memory usage (command: 0205)

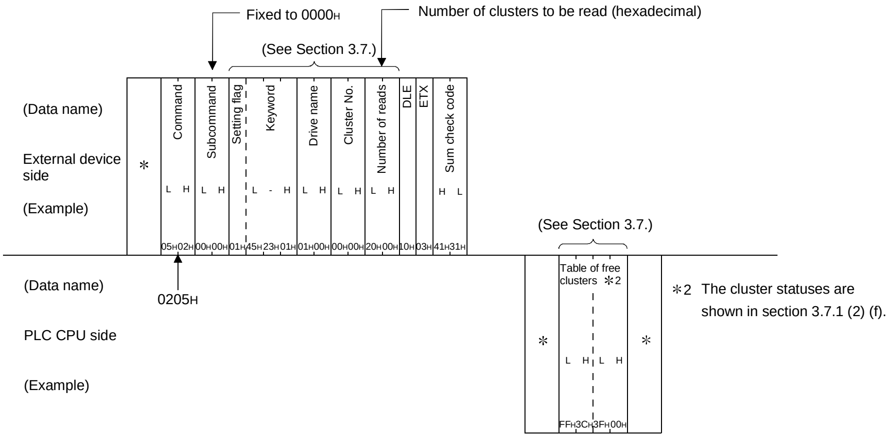
The examples shown in this section explain the control procedure to read the usage status of a drive memory.
The order and content of data items of the areas marked by "*" shown in the control procedure diagram differ depending on the module used as well as the frame and format used for communication.
See the detailed information described in Section 3.1.

[Control procedure]

(1) Reading the usage status of the drive memory of the RAM area in a memory card A (drive name: 01H) for 32 clusters while communicating in ASCII code format 1.




(2) Reading the usage status of the drive memory of the RAM area in a memory card A (drive name: 01H) for 32 clusters while communicating in binary code format 5.




**POINT**

(1) Designate the number of clusters to read in multiples of 16 (10H, 20H... in hexadecimal representation) within the range of 10H to 100H (16 to 256).

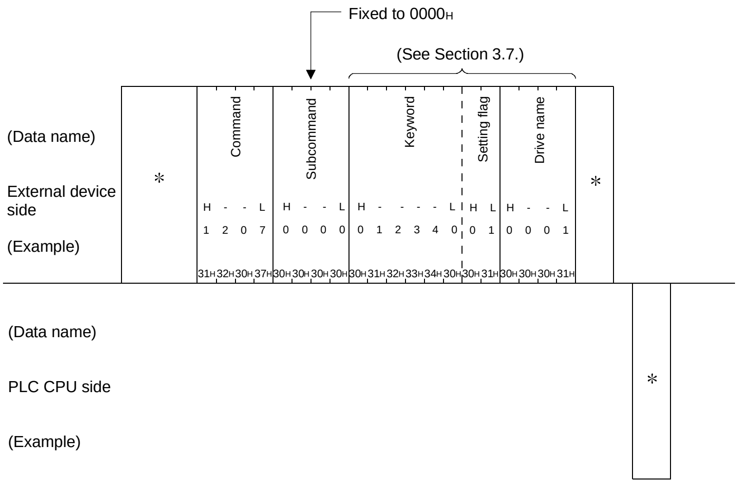
(2) When a new file is created (new registration), it is necessary to reserve a contiguous free area of the same size as the file to be created.
To find the size of the contiguous free area of the designated drive, check the number of consecutive free clusters (number of consecutive OFF bits) by reading the usage status of that drive memory.
Size of contiguous vacant area = Number of contiguous free clusters × 4096 or 512 (bytes)
If the contiguous free area is insufficient, defragment the memory according to the description in Section 3.7.3.

### 3.7.3 Drive memory defragmentation (command: 1207)

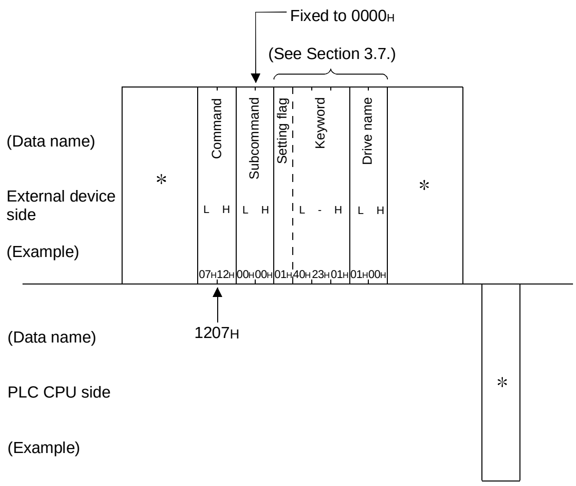
The examples shown in this section explain the control procedure for defragmenting the drive memory.
The order and content of data items of the areas marked by "✻" shown in the control procedure diagram differ depending on the module used as well as the frame and format used for communication.
See the detailed information described in Section 3.1.

[Control procedure]

(1) Defragmenting the drive memory of the RAM area in a memory card A (drive name: 01H) while communicating in ASCII code format 1.




(2) Defragmenting the drive memory of the RAM area in a memory card A (drive name: 01H) while communicating in binary code format 5.



**POINT**

(1) The drive memory should be defragtmented in the following cases.
- When the QnACPU of the designated station is in the STOP status
- When files cannot be stored because the drive memory usage status (checked by the command 0205) is scattered (fragmented).

(2) In the following cases, an error occurs and a NAK message is returned.
1) When there is system protection on the QnACPU
2) When the keyword designated in the instruction did not match the one registered in the designated drive.
3) When the drive memory is faulty (defective cluster, etc.).
4) When using an IC memory card read/writer for the following:
   - Creating subdirectories
   - Storing a file, and the file is not stored to one contiguous area in one location

## 3.8 File Control

This function reads the file registration status, registers new files, and deletes files, as well as reads and writes data in the designated drive memory of the Q/QnA.
It is used when an external device reads and stores parameters, sequence programs, etc. from the Q/QnACPU or writes parameters, programs, etc. to the Q/QnACPU depending on the control commands.

### 3.8.1 Commands and contents of character area

The following explains the file control procedure commands and what is included in the character area (data section in communication in binary code).

(1) Commands
The following two tables list the file control commands for the Q/QnACPU.

(a) File control commands for the QCPU

| Function | Command (subcommand) | Processing | Number of points processed per communication: Access station-1 (QCPU as described in Sec. 3.2 ＊7) | PLC CPU status during RUN: During STOP | PLC CPU status during RUN: During Run, Write allow setting | PLC CPU status during RUN: During Run, Write prohibit setting | Reference section |
|---|---|---|---|---|---|---|---|
| Read directory/file information | 1810 (0000) | Reads the file list data. | (for 36) | ○ | ○ | ○ | Section 3.8.5 |
| Search directory/file information | 1811 (0000) | Reads the file number of the designated file. | (for 16) | ○ | ○ | ○ | Section 3.8.6 |
| Create new file | 1820 (0000) | Reserves a storage area for the designated file. | (for 256) | ○ | ○ | × | Section 3.8.10 |
| Delete file | 1822 (0000) | Deletes a file. | (for 1) | ○ | ○ | × | Section 3.8.12 |
| Copy file | 1824 (0000) | Copies the designated file. | (for 1) | ○ | ○ | C24: ×<br>E71: ○ | Section 3.8.13 |
| Modify file attributes | 1825 (0000) | Changes the attributes of a file. | (for 1) | ○ | ○ | × | Section 3.8.15 |
| Modify date of file creation | 1826 (0000) | Changes the date when a file was created. | (for 1) | ○ | ○ | × | Section 3.8.14 |
| Open file | 1827 (0000) | Locks a file so that other devices cannot make changes to it. | (for 1) | ○ | ○ | ○ | Section 3.8.7 |
| Read file | 1828 (0000) | Reads data in a file. | 1920 bytes | ○ | ○ | ○ | Section 3.8.9 |
| Write to file | 1829 (0000) | Writes data to a file. | 1920 bytes | ○ | ○ | × | Section 3.8.11 |
| Close file | 182A (0000) | Unlocks the file locked in Open file. | (for 1) | ○ | ○ | ○ | Section 3.8.8 |

○ in the PLC CPU status column above indicates that execution is possible.

(b) File control commands for the QnACPU

| Function | | Command (subcommand) | Processing | Number of points processed per communication | PLC CPU status during RUN | | | Reference Section |
|---|---|---|---|---|---|---|---|---|
| | | | | Access station-2 (QCPU as described in Sec. 3.2 ＊8) | During STOP | During RUN | | |
| | | | | | | Write allow setting | Write prohibit setting | |
| Read file information | No header statement | 0201 (0000) | Reads a file list (file name, last update time and data, file size). | (for 36) | ○ | ○ | ○ | Section 3.8.16 (1) |
| | Header statement | 0202 (0000) | Reads a file list with a file heading. | (for 16) | | | | Section 3.8.16 (2) |
| | File No. usage status | 0204 (0000) | Reads the usage status of file number. | (for 256) | | | | Section 3.8.16 (3) |
| Modify file information | Modify last update | 1204 (0000) | Changes time and date when a file is last edited. | (for 1) | ○ | ○ | × | Section 3.8.24 (1) |
| | Modify file name/size | 1204 (0001) | Changes file name and file size. | | | | | Section 3.8.24 (2) |
| | Batch modification | 1204 (0002) | Changes file name, file size, and last update time and data. | | | | | Section 3.8.24 (3) |
| File Search (Reading file presence) | | 0203 (0000) | Reads file number and size of the designated file. | (for 1) | ○ | ○ | ○ | Section 3.8.17 |
| Read file contents (batch read) | | 0206 (0000) | Reads data in a file. | 960 bytes | ○ | ○ | ○ | Section 3.8.9 |
| New registration (register a file name) | | 1202 (0000) | Reserves area for the designated file. | (for 1) | ○ | ○ | × | Section 3.8.20 |
| Write file contents | Arbitrary data (batch write) | 1203 (0000) | Writes the designated data (for n bytes) to a file. | 960 bytes | ○ | ○ | × | Section 3.8.21 (1) |
| | Same data (FILL) | 1203 (0001) | Writes the designated data (1 word) to a file. | (for file size) | | | | Section 3.8.21 (2) |
| File Lock | Registration | 0808 (0001) | Locks a file so that other devices cannot make change to the file. Or cancels the lock. | (for 1) | ○ | ○ | ○ | Section 3.8.18 |
| | Clear | 0808 (0000) | | | | | | |
| File copy | | 1206 (0000) | Writes data in an existing file to a newly registered file, | 480 bytes | ○ | ○ | ○ | Section 3.8.23 |
| File delete | | 1205 (0000) | Deletes a file | (for 1) | ○ | ○ | × | Section 3.8.22 |

○ in the PLC CPU column in the table above indicates that execution is possible.

## (2) Contents of character area

The following explains the contents of the character area of the messages described in the sections beginning from 3.8.5 when an external device controls the Q/QnACPU files.

(a) Keyword (Password)

1) Keyword (maximum 4 characters) · · · · · QCPU
Character string registered in the subject memory file (program file, device comment file, device initial value file) by the user. This data allows/prohibits access to that memory file.

2) Keyword (maximum 6 characters) · · · · · QnACPU
Character string registered to the designated drive by the user. This data allows/prohibits access to that drive.

When a keyword is registered, the same keyword should be designated when accessing the drive.
See Section 3.7.1 (2) (a) for a description of the contents of the character area.

(b) Set flag

This flag indicates whether or not the keyword registered in the designated drive by the user matches the keyword of (a) above.
See Section 3.7.1 (2) (b) for a description of the contents of the character area.

(c) Drive name and number

This data section designates the drive of the Q/QnACPU whose files are to be controlled.

See Section 3.7.1 for how to designate drive names for QnACPUs and their corresponding drives.

How to designate drive names for QCPUs and their corresponding drives are described here.

1) Data communication in ASCII code

The following values indicating drives to be accessed are converted to 4-digit (hexadecimal) ASCII code and transmitted sequentially beginning from the most significant digit ("0").

2) Data communication in binary code

The following 2-byte values indicating drives to be accessed are sequentially transmitted beginning from the low byte (L: bits 0 to 7).

3) The following drive numbers and target drives can be designated; other designations are not allowed.

| Designated value | Objective Drive | Designated value | Target drive |
|---|---|---|---|
| $0000_H$ | QCPU built-in program memory | $0003_H$ | QCPU built-in standard RAM |
| $0001_H$ | Memory card (RAM)…SRAM card | $0004_H$ | QCPU built-in standard ROM |
| $0002_H$ | Memory card (ROM)…Flash card/ATA card | – | – |

(d) File number

This value designates the registration number when a file designated by filename and extension as described below was registered (written) to the PLC CPU, or the registration number when registering to the PLC CPU.

1) Data communication in ASCII code

The file numbers shown below are converted to a 4-digit (hexadecimal) ASCII code and sequentially transmitted beginning from the most significant digit.

(Example)

File number $1F_H$: It is converted to "001F" and sequentially transmitted beginning from the first "0."

2) Data communication in binary code

A 2-byte value indicating the file number shown below is sequentially transmitted from the low byte (L: bits 0 to 7).

(Example)

File number $1F_H$: The value 001F is transmitted in the order of $1F_H$ first, and then $00_H$.

3) The file numbers shown below can be designated.

| Designated value | Contents | Designation |
|---|---|---|
| $01_H$ to $100_H$ | File number | Designate when the file number is known. |
| $FFFF_H$ | File number unknown | Designate this when requesting the C24/E71 to search for file numbers. (Read/write request from the Q series C24/E71 to the PLC CPU will be delayed for more than one sequence scan time.) |

4) The file number of an already registered file can be checked using the functions described in Sections 3.8.6, 3.8.16, and 3.8.17.

On a QnACPU, when registering a new file, the unused file numbers can be checked using the read file number usage status function described in Section 3.8.16 (3). (This cannot be checked in a QCPU.)

(e) Number of file requests, total number of registered files, and number of file information

These data values indicate the number of files requested by the user, number of files registered in the designated drive, and number of files that return file information when reading the file information.

1) Data communication in ASCII code

The value given in the reference section relevant to the corresponding function is converted to a 4-digit (hexadecimal) ASCII code and sequentially transmitted beginning from the most significant digit ("0").

2) Data communication in binary code

The 2-byte value given in the reference section relevant to the corresponding function is sequentially transmitted from the low byte (L: bits 0 to 7).

(f) Number of characters of file name, filename, extension, and attribute

This is used to designate the file to be read, written, registered, etc.

1) When registering a new file or changing the file name, designate the filename (maximum 8 characters) and extension (maximum 3 characters) according to the file naming convention files of GX Developer.

2) The number of characters of file name, filename, extension, and attribute are handled in the following manner. The data order during data communication differs depending on the command used; however, it is the same whether communicating in ASCII code or in binary code.

(Pattern 1) For QCPU files

- Files to be accessed should be designated using the following data order.
  File name + "." + Extension
- The number of characters specified above should be designated as the number of characters in the file name.
- The following example shows how to designate the data when the file name is ABC.QPG.
  The number of characters of file name: 7
  File name: "ABC.QPG"

(Pattern 2) For mainly QnACPU files

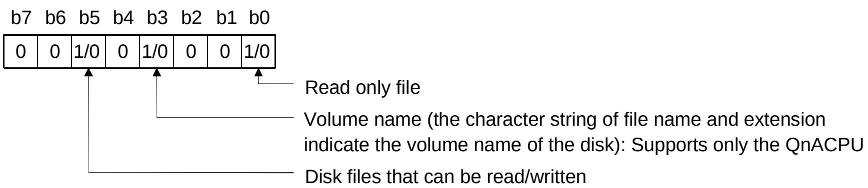
- Files to be accessed should be designated using the following data order.
  File name + Extension + Attribute
  When this file name is less than 8 characters, append a blank space (code: $20_H$) to the name to make it 8 characters.
- The following example shows how to designate the data when the file name is ABC.QPG.
  File name: "ABC␣␣␣␣␣QPG"
  ("␣" indicates blank)

3) Designate "20H" as the attribute value (disk files that can be read/written) for a file newly created by a user and when a dummy is designated.
The user can change this attribute (see Remark).
(Files for the QCPU: See Section 3.8.15; Files for the QnACPU: See Section 3.8.24)
＊ The attributes for existing files can be checked using the functions shown below.
Files for the QCPU: Functions for reading directory/file information (see Section 3.8.5)
Files for QnACPU: Functions for reading file information list (see Section 3.8.16)

4) For how to transmission data indicating the file name during data communication, see the reference sections for each command.

REMARK

The following is an outline of how to interpret the attributes of the files stored on each disk of the Q/QnACPU. Each bit of an attribute value has its own meaning. When each bit is turned on (1), the corresponding attribute is enabled.



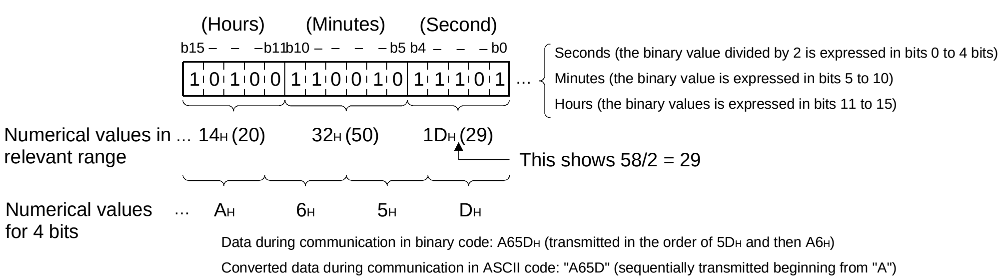

＊ The attribute of read/write enable disk file is used for user files by default.
The attribute of a file created by user can be changed between 01H (Read Only file) • 20H (disk files that can be read/written). (See Sections 3.8.15 and 3.8.24.)

(g) Time of last updating and date of last updating
This data section indicates the date and time when the current data was registered.

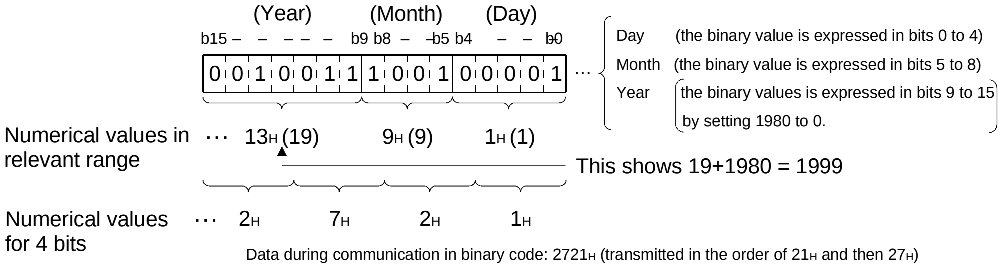
1) Data communication in ASCII code
The value shown below is converted to a 4-digit (hexadecimal) ASCII code and sequentially transmitted beginning from the most significant digit (time, year).
"0000" is transmitted when dummy data is designated.

2) Data communication in binary code
The 2-byte values shown below are sequentially transmitted beginning from the low byte (L: bits 0 to 7).
0000H is transmitted when dummy data is designated.

3) The contents of the values that indicate the time and date and the order in which they are transmitted are shown below.
• Time (hour, minute, second)
(Example)
20 hours 50 minutes 58 seconds



• Data (Year, month, day)
(Example)
September 1, 1999




Data during communication in binary code: 2721H (transmitted in the order of 21H and then 27H)

Converted data during communication in ASCII code: "2721" (sequentially transmitted beginning from "2")

(h) File size

This value indicates the size of the current file in number of bytes.

1) Data communication in ASCII code
The 2-word value is converted to an 8-digit (hexadecimal) ASCII code and sequentially transmitted beginning from the most significant digit.
(Example)
The file size is 7168 bytes
Converted to "00001C00" and sequentially transmitted beginning from the leading "0."
Head

2) Data communication in binary code
The 2-word value is sequentially transmitted from the low byte (L: bits 0 to 7).
(Example)
The file size is 7168 bytes
The file size becomes 00001C00H and is transmitted in the order of 00H, 1CH, 00H, and then 00H.

(i) Header statement

This is the header statement given to the designated file by the GX Developer supporting the Q/QnACPU (maximum 32 characters).

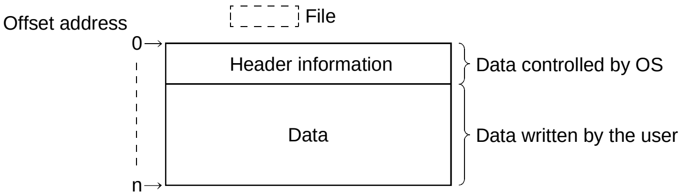
1) Data communication in ASCII code
Each header statement is sequentially transmitted from the first character.
When the header statement is less than 32 characters, append blanks (code: 20H).
(Example)
The header statement during the registration was "1 line-PC5"
The header statement becomes "1 line-PC5…" and is sequentially transmitted beginning from the "1."

2) Data communication in binary code
The character code of each character of the header statement is used as a binary value and is sequentially transmitted beginning from the code for the first character.
When the header statement is less than 32 characters, append 20H to make it 32 characters.
(Example)
The header statement during registration was "1 line-PC5"
The header statement becomes 31H, 6CH, 69H, 6EH, 65H, 2DH, 50H, 43H, 35H, 20H, 20H, ... and is sequentially transmitted beginning from 31H.

(j) Offset address

This data designates the head address of the file range for which data is read and written.

The address (one address/one byte) is counted from the head (offset address: $0_H$) of each file and is designated an even address.




1) Data communication in ASCII code

The address shown in the reference section of the relevant function is converted to an 8-digit (hexadecimal) ASCII code and sequentially transmitted beginning from the most significant digit.

2) Data communication in binary code

The 4-byte value that indicates the address given in the reference section of the relevant function is sequentially transmitted beginning from the low byte (L: bits 0 to 7).

3) To find out which offset addresses can be designated, check the file size (number of bytes) using the file information list read function described in Section 3.8.16 and obtain the offset addresses ($0_H$ to $n_H$) from this size.

(k) Number of bytes to read and number of bytes to write

These values designate the number of bytes of the file range for which data is read or written. It is designated as one address/one byte.

1) Data communication in ASCII code

The value shown in the reference section of the relevant function is converted to a 4-digit (hexadecimal) ASCII code and transmitted beginning from the most significant digit ("0").

2) Data communication in binary code

The 2-byte value shown in the reference section of the relevant function is sequentially transmitted beginning from the low byte (L: bit 0 to 7).

(l) Data read and data written (batch read and batch write functions)

These are the data read from or written to the Q/QnACPU file. They are arranged after the offset address.

1) Data communication in ASCII code

One byte (one address) is converted to a 2-digit (hexadecimal) ASCII code and the designated number of bytes are sequentially transmitted beginning from the most significant digit.

2) Data communication in binary code

The designated number of bytes is transmitted with one address equivalent to one byte.

3) When reading data from the Q/QnACPU, it is stored in the external device without changing the order of data read.

When writing data to the Q/QnACPU, it is designated without changing the order of data written.

(m) Data written (write same data function): For QnACPU files

This data section is for the write same data function, and used to write the same data to the existing QnACPU files.

1) Data communication in ASCII code

   The value for one word is converted to a 4-digit (hexadecimal) ASCII code and sequentially transmitted beginning from the most significant digit.

2) Data communication in binary code

   The value for one word is sequentially transmitted beginning from the low byte (L: bits 0 to 7).

(n) Size

This value is used to reserve the file area on the designated drive when registering a new file. It is designated in number of bytes.

1) Data communication in ASCII code

   The area size to be reserved for the designated file is expressed as two words and converted to an 8-digit (hexadecimal) ASCII code and sequentially transmitted beginning from the most significant digit.

2) Data communication in binary code

   The area size to be reserved for the designated file is expressed as two words and sequentially transmitted beginning from the low byte (L: bits 0 to 7).

3) An external device can register a new file with the same contents as the existing file.

   The size of the relevant existing file must be checked using the read file information list function (see Section 3.8.5, 3.8.16, and 3.8.17).

(o) Fixed values

1) When data is communicated in ASCII code, "0000" is transmitted.
2) When data is communicated in binary mode, the 2-byte value "$0000_H$" is transmitted.

(p) Modification pattern (for changing filename and file size): For QnACPU files
This data section designates what data is to be modified when the data (filename, size, date and time created) of the existing file is to be modified.

1) Data communication in ASCII code
The value shown below is converted to a 4-digit (hexadecimal) ASCII code and sequentially transmitted beginning from the most significant digit.
2) Data communication in binary code
The 2-byte value shown below is transmitted from the low byte (L: bits 0 to 7).
3) Values and data to be designated for the modification pattern are shown below.
• Modifying the date and time the file was created (command:1204, subcommand: 0000)

| b15 | – | – | – | – | – | – | – | – | – | – | b4 | b3 | b2 | – | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1/0 | 1/0 | 1/0 | 0 | 0 |

…… $04_H$, $08_H$, $0C_H$, $14_H$, $18_H$, or $1C_H$ can be designated.

b2: Last update time modification yes/no designation
0: Do not modify the last update time
1: Modify last the update time

b3: Last update date modification yes/no designation
0: Do not modify the last update date
1: Modify last the update date

b4: Modification mode for data and time designation
Designate which of the following data should be used when changing date and time.
0: Use date and time in a message
1: Use data and time of the QnACPU
(Designate dummy date and time for the message)

• Modifying the filename and file size (command: 1204, subcommand: 0001)

| b15 | – | – | – | – | – | – | – | – | – | b5 | – | – | – | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1/0 | 0 | 0 | 0 | 1/0 | 1/0 |

…… $01_H$, $02_H$, $03_H$, $20_H$, $21_H$, $22_H$ or $23_H$ can be designated.

b0: File name and extension modification yes/no designation
0: Do not modify the file name and extension
1: Modify the file name and extension

b1: File attribute modification yes/no designation
0: Do not modify the attribute
1: Modify the attribute

b5: File size modification yes/no designation
0: Do not modify the file size
1: Modify the file size

• Changing the file information in batch (command: 1204; subcommand: 0002)

| b15 | – | – | – | – | – | – | – | – | – | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1/0 | 1/0 | 1/0 | 1/0 | 1/0 | 1/0 |

…… Can be designated between $01_H$ to $3F_H$.

File name and extension modification yes/no designation
0: Do not modify the file name and extension
1: Modify the file name and extension

File attribute modification yes/no designation
0: Do not modify the attribute
1: Modify the attribute

Last update time modification yes/no designation
0: Do not modify the last update time
1: Modify last the update time

Last update date modification yes/no designation
0: Do not modify the last update date
1: Modify last the update date

Modification mode for data and time designation
Designate which of the following data should be used when changing date and time.
0: Use date and time in a message
1: Use data and time of the QnACPU
(Designate dummy date and time for the message)

File size modification yes/no designation
0: Do not modify the file size
1: Modify the file size

(q) File number usage status: For QnACPU files

This data section returns the usage status of the file numbers for 256 files to the external device when reading the usage status of a file number. The data is arranged from the first byte shown below.

1) Data communication in ASCII code

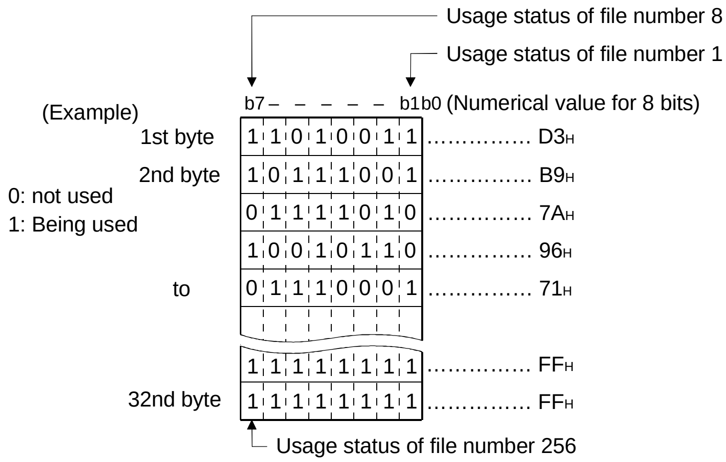
   The value shown below is converted to a 64-digit (hexadecimal) ASCII code and transmitted to the external device. (File numbers for 8 files: two digits)

2) Data communication in binary code

   The following 32-byte value indicating the usage status is sequentially transmitted to the external device beginning from the low byte (L: bits 0 to 7). (File numbers for 8 files: one byte)

3) How the usage status of file numbers is represented is shown below.

   The usage status of each file number is indicated as one file number per bit.




The following file number usage status value is returned to the external device for the above example.

- Data communication in ASCII code
  The value "D3B97A...FFFF" is returned, and are sequentially transmitted beginning from the "D."
- Data communication in the binary mode
  D3H, B9H, 7AH...FFH, FFH are returned, and are sequentially transmitted beginning from D3H.


3 - 159

(r) File lock mode: For QnACPU files

This flag designates whether or not a file lock is to be forcefully cleared in order to allow access to the designated file from another device.

1) Data communication in ASCII code

The value shown below is converted to a 4-digit (hexadecimal) ASCII code and sequentially transmitted beginning from the most significant digit "0."

2) Data communication in binary code

The 2-byte value shown below is transmitted from the low byte (L: bits 0 to 7).

3) The values and options that can be designated for the file lock mode are as follows; other values cannot be designated.

| Designated value | Designated contents |
|---|---|
| $0000_H$ | Normally execute file lock cancel. |
| $0002_H$ | Forcefully execute file lock cancel. |

4) When a file lock cancel instruction is issued for a designated file, the normal execution and forced execution differ as follows.

• Normal execution

The file lock cannot be canceled when another device has registered the file lock.
When the file lock cancel is requested, an error occurs and an NAK message is returned.

• Forced execution

The file lock is forcefully canceled even if another device registered the file lock.
Use the forced execution function when a file lock cannot be canceled because of problems in the device that registered the file lock.

(s) Copy mode: For QnACPU files

This flag designates whether or not to copy the last update time and data of the source file to the target file at the completion of copying.
When the date and time are not copied, the QnACPU management time when a new file was created, remains unchanged.

1) Data communication in ASCII code

The value shown below is converted to a 4-digit (hexadecimal) ASCII code and sequentially transmitted beginning from the most significant byte.

2) Data communication in binary code

The 2-byte value shown below is transmitted from the low byte (L: bits 0 to 7).

3) The values and options that can be designated for copy mode are as follows.

| Designated value | Designated contents |
|---|---|
| $0000_H$ | Do not copy the date and time the source file was last updated at the completion of copying. |
| $0001_H$ | Copy the date and time the source file was last updated at the completion of copying. |

(t) Number of directory characters: For QCPU files

This data is used to designate the storage area for files to be read, written, registered, etc.

1) Data communication in ASCII code

"0000" is transmitted.

2) Data communication in binary code

The 2-byte value "0000H" is transmitted.

(u) Reserved data n: For QnACPU files

1) Data communication in ASCII code

Null data (code: 00H) equivalent to four characters is transmitted.

2) Data communication in binary code

The 2-byte value "0000H" is transmitted.

(v) File pointer number: For QCPU files

This number is used to manage the file by the PLC CPU.

The data to be returned when the file is opened is used.

1) Data communication in ASCII code

Data equivalent to four characters is transmitted.

2) Data communication in binary code

A 2-byte value is transmitted

(w) Close type: For QCPU files

This is used to designate whether or nor to forcefully close a file to allow access to the designated file from other devices, etc.

1) Data communication in ASCII code

The value shown below is converted to a 4-digit (hexadecimal) ASCII code and sequentially transmitted beginning from the most significant digit "0."

2) Data communication in binary code

The 2-byte value shown below is transmitted from the low byte (L: bits 0 to 7).

3) The values and options that can be designated for the close type are as follows; other values cannot be designated.

| Designated value | Designated contents | Processing |
|---|---|---|
| 0000H | Normal close | Close only the target files. |
| 0001H | Forced close -1 | Forcefully close files, including other files that are opened by the modules/devices that opened the target file. |
| 0002H | Forced close -2 | Forcefully close all the open files. |

4) When a file close instruction is issued for a designated file, the normal closing and forced closing differ as follows.
- Normal close
  This option does not close files that are opened by other modules and devices.
  When a file close is requested, it is abnormally completed.
- Forced close -1
  This option closes all the files opened by other same module/device. (∗1)
- Forced close -2
  This option closes all the files currently open.(∗1)

∗1 Designate these options to close a file currently open from modules/devices other than the one that opened the file when the module/device that opened the file cannot close it because of problems, etc.

(x) Open mode: For QCPU files
This flag is used to designate whether the designated file is opened for reading or writing.

1) Data communication in ASCII code
The value shown below is converted to a 4-digit (hexadecimal) ASCII code and sequentially transmitted beginning from the most significant digit "0."

2) Data communication in binary code
The 2-byte value shown below is transmitted from the low byte (L: bits 0 to 7).

3) The values and options that can be designated for the open mode are as follows; other values cannot be designated.

| Designated value | Designated contents | Processing |
|---|---|---|
| $0000_H$ | Open for read | Open the target file for reading data. |
| $0100_H$ | Open for write | Open the target file for writing data. |

## 3.8.2 Precautions on file control

The following explains the notes when controlling files of the QCPU or QnACPU.

(1) The files read from the Q/QnACPU are used for storage on an external device side.
The external device cannot edit the contents of a file read from the Q/QnACPU.

(2) 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 | |
|---|---|---|
| | For QCPU | For QnACPU |
| Read file information list function | Section 3.8.5 | Section 3.8.16 |
| File search (Reading file presence) | Section 3.8.6 | Section 3.8.17 |

(3) If there is system protection on the Q/QnACPU when using the following functions, an error occurs and an abnormal complete message is returned.

| Function | Reference | |
|---|---|---|
| | For QCPU | For QnACPU |
| Create new file (filename registration) | Section 3.8.10 | Section 3.8.20 |
| Write to file | Section 3.8.11 | Section 3.8.21 |
| Delete file | Section 3.8.12 | Section 3.8.22 |
| Copy file | Section 3.8.13 | Section 3.8.23 |
| Modify file information (date created, attribute) | Section 3.8.14<br>Section 3.8.15 | Section 3.8.24 |

(4) When registering a keyword in the file, write down the keyword that was registered.
When accessing the following files, it is necessary to specify the registered keyword when opening and reading or writing to that file.
- Parameter file
- Program file

(5) The file attribute is valid only when the following functions are used.
For other commands, the attribute is treated as a dummy.

| Function | Reference | |
|---|---|---|
| | For QCPU | For QnACPU |
| Read directory/file information | Section 3.8.5 | Section 3.8.16 |
| New registration | Section 3.8.10 | Section 3.8.20 |
| Modify file information (date created, attribute) | Section 3.8.14<br>Section 3.8.15 | Section 3.8.24 |

(6) The following shows the procedure for creating the target files for the QCPU. The file types and file extensions of these files are listed in the table below.

1) Create a temporary file by specifying an arbitrary file extension excluding the file extensions listed below.
2) Open the newly created file. Write data to the file, and close the file.
3) Create a file with the target file extension using the copy function.
4) After the file copy processing is completed, delete the copy source file as needed.

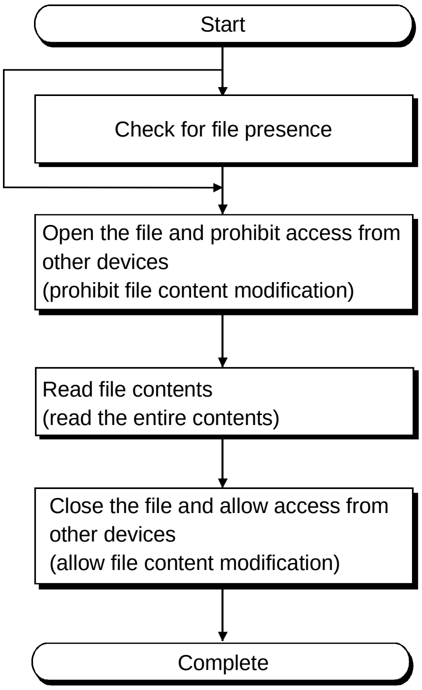
| File type | File extension |
|---|---|
| Header statement file | DAT |
| Sequence program file | QPG |
| Device comment file | QCD |
| Device initial value file | QDI |

(7) For notes other than mentioned above, see the corresponding function explanation section.

### 3.8.3 File control execution procedure for the QCPU

The following flow charts show the procedures for executing file control on the QCPU.

(1) Procedure to read the contents of a file




| (Function name) | (Command) | (Reference section) |
|---|---|---|
| Read directory/file information or File search (checking file presence) | 1810 | Section 3.8.5 |
| | 1811 | Section 3.8.6 |
| Open file | 1827 | Section 3.8.7 |
| Read data in file (batch read) | 1828 | Section 3.8.9 |
| Close file | 182A | Section 3.8.8 |

**POINT**

Store the following file information of the file read to an external device.
- File number
- Filename and file attribute
- File size

### (2) Procedure for creating a new file and writing data to it

| | (Function name) | (Command) | (Reference section) |
|---|---|---|---|
| Start | | | |
| Check for file presence | Read directory/file information or File search (checking file presence) | 1810<br>1811 | Section 3.8.5<br>Section 3.8.6 |
| Register file name and reserve size in memory | Create new file | 1820 | Section 3.8.10 |
| Prohibit access from other devices (prohibit file content modification) | Open file | 1827 | Section 3.8.7 |
| Write data | Write to file | 1829 | Section 3.8.11 |
| Allow access from other devices (allow file content modification) | Close file | 182A | Section 3.8.8 |
| Modify date and time of file creation | Modify file information | 1826 | Section 3.8.14 |
| Check the file number of the created file only when creating a new file | File search (checking file presence) | 1811 | Section 3.8.6 |
| Complete | | | |

**POINT**

(1) Each file to be stored in the QCPU is stored in a contiguous area in the program area (memory).
In order to avoid a situation where there is insufficient free space when an external device creates a new file, reserve a contiguous area larger than the necessary size by taking either of the following actions using GX Developer before operating the QCPU.
1) Delete any unnecessary files.
2) Defragment the PC memory.

(2) See Section 3.8.2 for creating a new file of QCPU.

(3) Procedure for copying a file

| | (Function name) | (Command) | (Reference section) |
|---|---|---|---|
| Start | | | |
| Check for file presence | Read directory/file information or File search (checking file presence) | 1810<br>1811 | Section 3.8.5<br>Section 3.8.6 |
| Copy the file | Copy file | 1824 | Section 3.8.13 |
| Modify date and time of file creation | Modify file information | 1826 | Section 3.8.14 |
| Check the file number of the created file only when creating a new file | File search (checking file presence) | 1811 | Section 3.8.6 |
| Complete | | | |

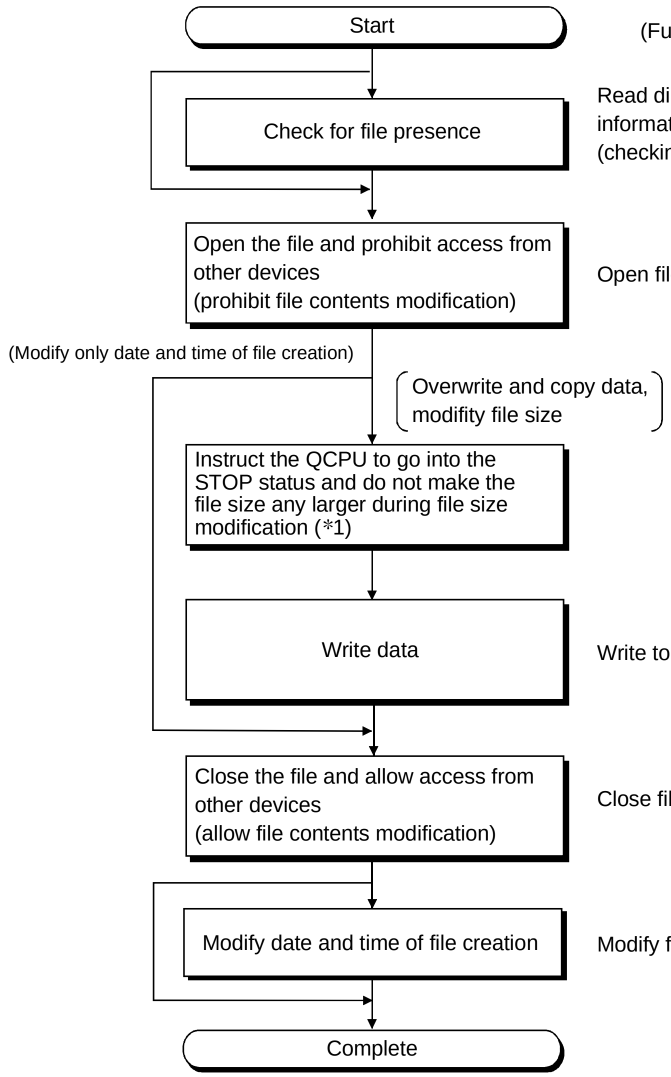
**POINT**

Each file to be stored in the QCPU is stored in a contiguous area in the program area (memory).
In order to avoid a situation where there is insufficient free space when an external device copies a file, reserve a contiguous area larger than the necessary size by taking either of the following actions using GX Developer before operating the QCPU.

1) Delete any unnecessary files.
2) Defragment the PC memory.

(4) Procedure for overwriting data in an existing file

Procedure to modify the file information

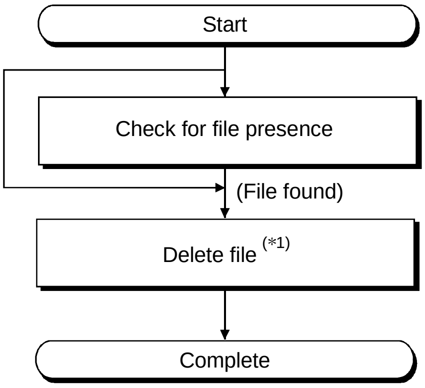



| (Function name) | (Command) | (Reference section) |
|---|---|---|
| Read directory/file information or File search (checking file presence) | 1810 | Section 3.8.5 |
| | 1811 | Section 3.8.6 |
| Open file | 1827 | Section 3.8.7 |
| Write to file | 1829 | Section 3.8.11 |
| Close file | 182A | Section 3.8.8 |
| Modify file information | 1826 | Section 3.8.14 |

*1 The sizes of files that already exist on the QCPU cannot be changed. When it is necessary to change any file size, follow the procedure below to create the file again.

1) Read all the data in the target file following the procedure shown in (1) of this section.
2) Delete the target following the procedure shown in (5) of this section.
3) Create a new file and write all the data to it following the procedure shown in (2) of this section.

(5) Procedure for deleting a file



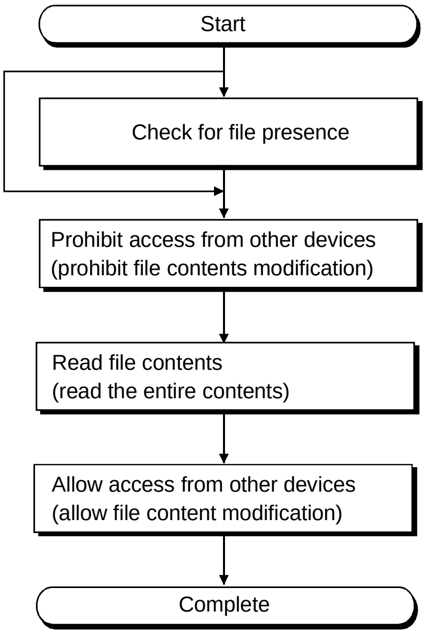

| (Function name) | (Command) | (Reference section) |
| --- | --- | --- |
| Read directory/file information or File search (checking file presence) | 1810 | Section 3.8.5 |
| | 1811 | Section 3.8.6 |
| File deletion | 1822 | Section 3.8.12 |

＊1 Determine the file deletion timing for the entire system, including the QCPU and related devices.

### 3.8.4 File control execution procedure for the QnACPU

The following flow charts illustrate the procedure for executing file control on the QnACPU.

(1) Procedure for reading the contents of a file




| (Function name) | (Command) | (Reference section) |
|---|---|---|
| File presence read | 0203 | Section 3.8.17 |
| or<br>File information read | 0201<br>0202 | Section 3.8.16 |
| File lock registration | 0808 | Section 3.8.18 |
| Read data in file (batch read) | 0206 | Section 3.8.19 |
| File lock cancel | 0808 | Section 3.8.18 |

**POINT**

Store the following file information of the file read to an external device.
- File number
- Filename and file attribute
- File size

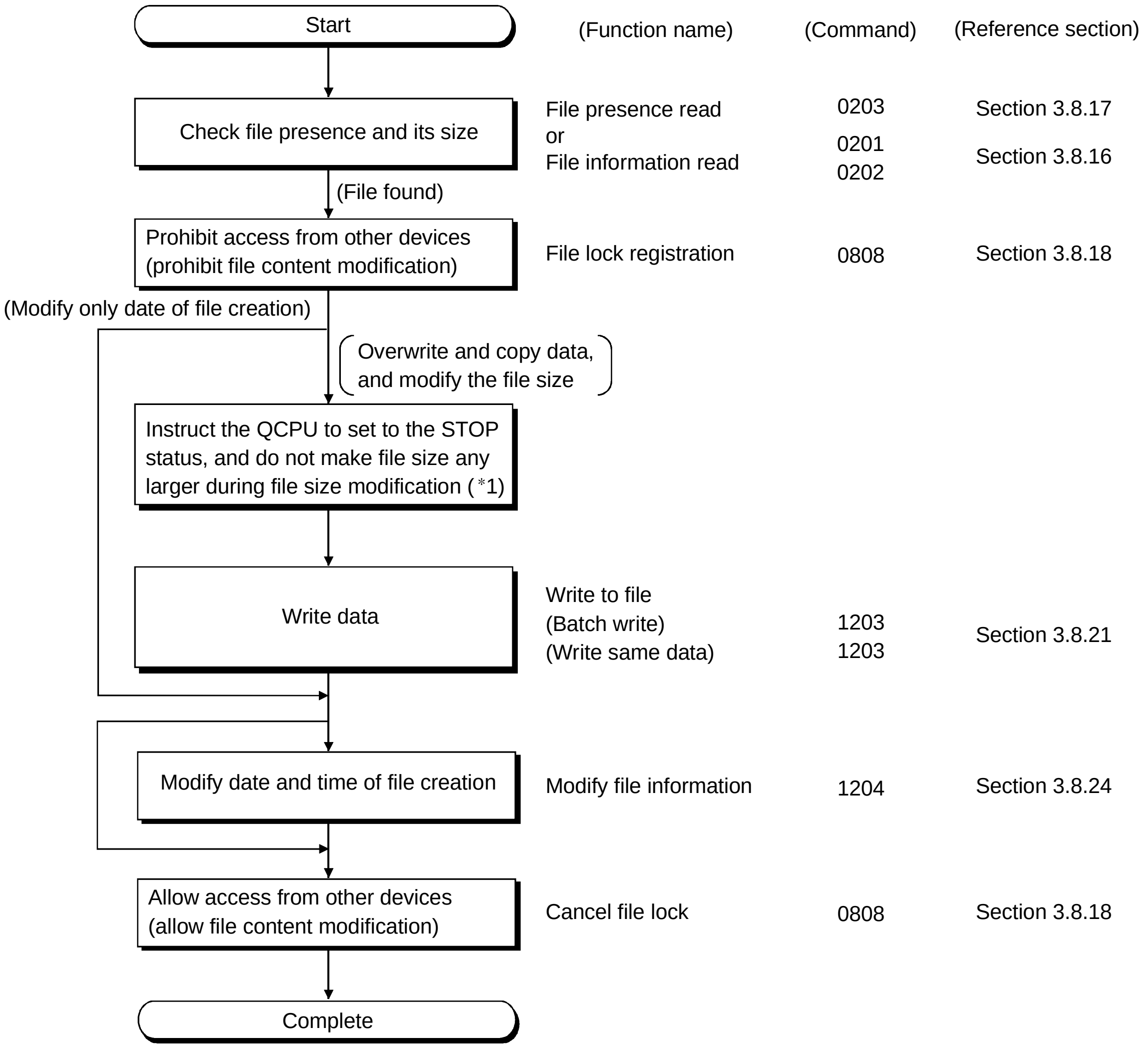
(2) Procedure for creating a new file and write data to it
Procedure for copying the data of an existing file to a new file

| Step | (Function name) | (Command) | (Reference section) |
|---|---|---|---|
| Start | | | |
| Check for file presence | File presence read<br>or<br>File information read | 0203<br>0201<br>0202 | Section 3.8.17<br>Section 3.8.16 |
| (File not found) Check unused file numbers only when copying a file | File number usage status read | 0204 | Section 3.8.16 |
| Check size of contiguous free memory space | Memory usage status read | 0205 | Section 3.7.2 |
| (Sufficient free space) → skip to "Register file name registration and reserve size in memory"; (Insufficient free space) → next step | | | |
| Delete unnecessary files,<br>or<br>increase contiguous free space *1 | File deletion<br>Memory defragmentation | <br>1207 | <br>Section 3.7.3 |
| Register file name registration and reserve size in memory | Create new file | 1202 | Section 3.8.20 |
| Prohibit access from other devices (prohibit file content modification) | File lock registration | 0808 | Section 3.8.18 |
| Write data | Write to file<br>(Batch write)<br>(Write same data)<br>(Copy file) | <br>1203<br>1203<br>1206 | Section 3.8.21 |
| Modify date and time of file creation | Modify file information | 1204 | Section 3.8.24 |
| Allow access from other devices (allow file contents modification) | File lock cancel | 0808 | Section 3.8.18 |
| Check the file number of the created file only when creating a new file | File presence read | 0203 | Section 3.8.17 |
| Complete | | | |

*1 Use the remote STOP (command: 1002) function described in Section 3.6.3 to instruct the QnACPU to go into the STOP status before defragmenting the memory.
After completion of the processing of this section, the remote RUN (command: 1001) function described in Section 3.6.2 can be used to set the QnACPU to the RUN status.

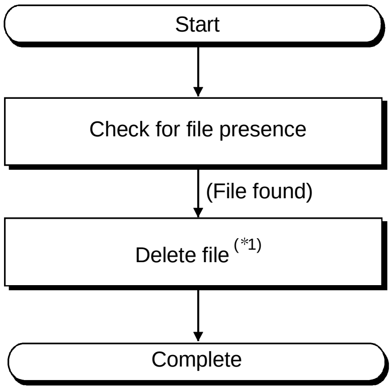
### (3) Procedure for overwriting data in an existing file

Procedure for modifying the file information




*1 The file size can be changed using the modify file information function (command: 1204) described in Section 3.8.24 only when the file size is made smaller.
When the file size must be made larger, use the procedure described in (2) of this section to create a new file and then write data.

(4) Procedure for deleting a file




| (Function name) | (Command) | (Reference section) |
|---|---|---|
| File presence read | 0203 | Section 3.8.17 |
| or | | |
| File information read | 0201<br>0202 | Section 3.8.16 |
| File deletion | 1205 | Section 3.8.22 |

*1 Determine the file deletion timing for the entire system, including the QnACPU and related devices.

### 3.8.5 Read directory/file information (command: 1810): for the QCPU

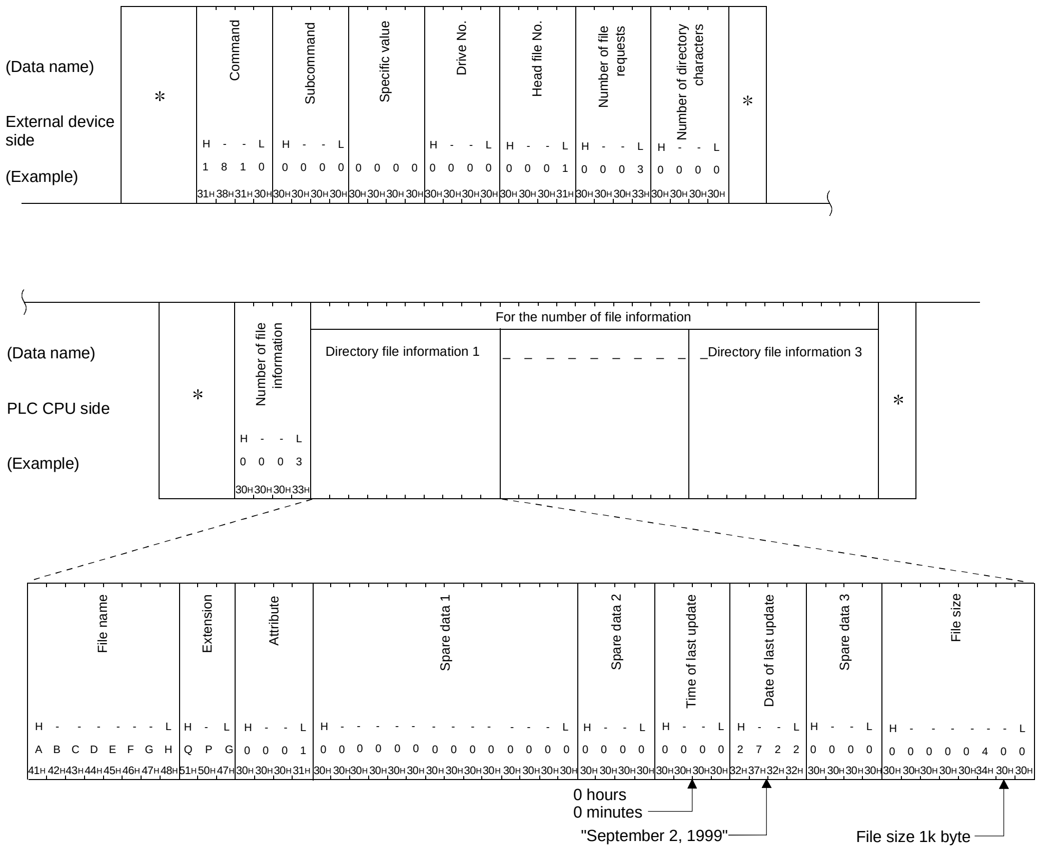
The examples shown in this section explain the control procedure for reading the directory and file information.
The order and content of data items of the areas marked by "∗" shown in the control procedure diagram differ depending on the module used as well as the frame and format used for communication.
See the detailed information described in Section 3.1.

[Control procedure]

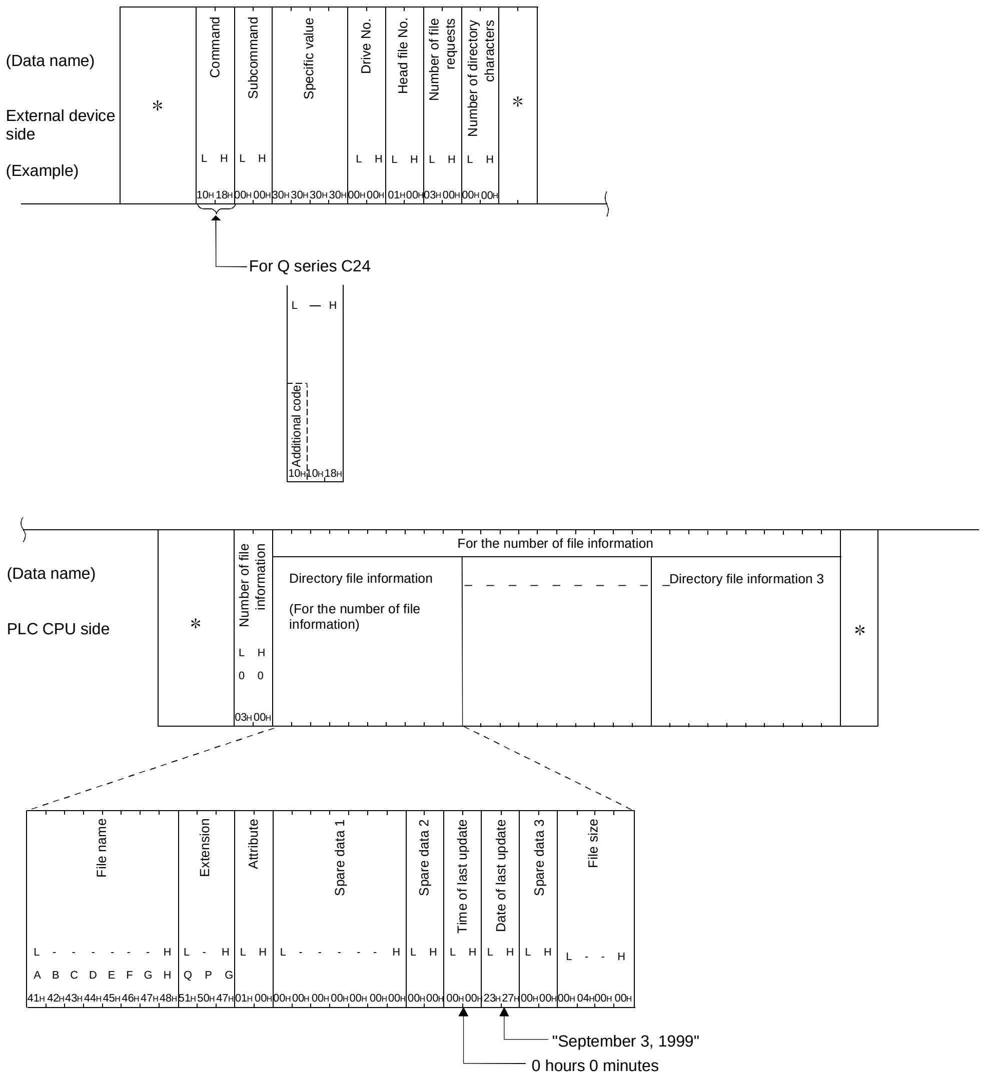
The following conditions apply to the example of reading a directory/file below.

| | |
|---|---|
| Drive No. | 0 |
| Head file No. | 1 |
| Number of file requests | 3 |

#### (1) Communication in ASCII code



(2) Communication in binary code




## POINT

(1) Designate the following items within the following ranges indicated below. The values for these items will also be returned within the following ranges.

- Head file number : $1 \leq$ file number $\leq 256$
- Number of requested files : $1 \leq$ file number $\leq 36$
- Number of file information : $0 \leq$ No. of files $\leq$ number of requested files (0: No registered files after the designated head file number)

(2) When all files are not registered within the designated file number range, the number of file information becomes the number of files registered in the designated range (number of file information returned).

### 3.8.6 Search directory/file information (command: 1811): for the QCPU

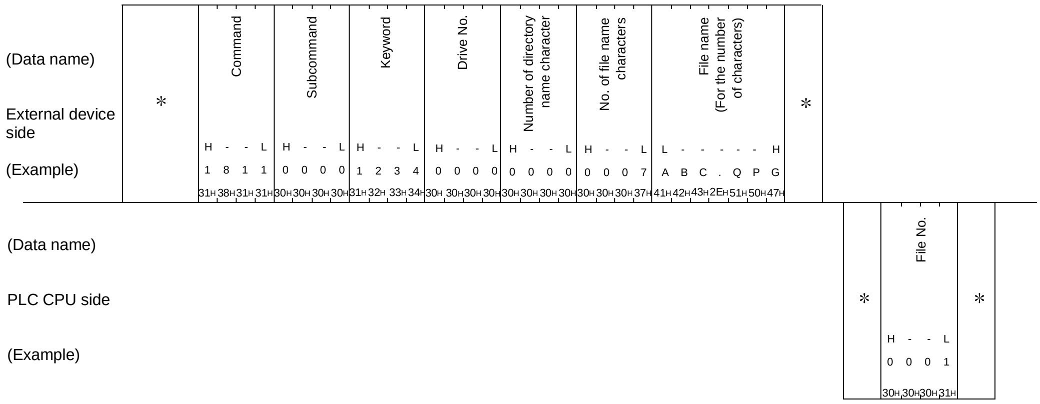
The examples shown in this section explain the control procedure for searching the directory and file information.
The order and content of data items of the areas marked by "*" shown in the control procedure diagram differ depending on the module used as well as the frame and format used for communication.
See the detailed information described in Section 3.1.

[Control procedure]

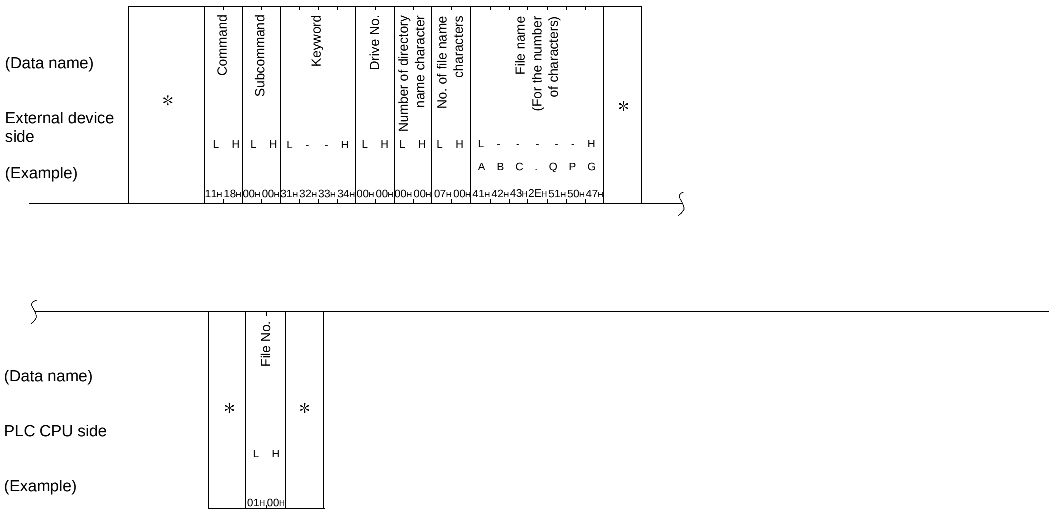
The following conditions apply to the example of searching a directory/file below.

| | |
|---|---|
| Keyword | 1234 |
| Drive No. | 0 |
| Filename | ABC.QPG |

(1) Communication in ASCII code




(2) Communication in binary code




**POINT**

When the designated file does not exist, an error occurs and an abnormal completion code will be returned.

### 3.8.7 Open file (command: 1827): for the QCPU

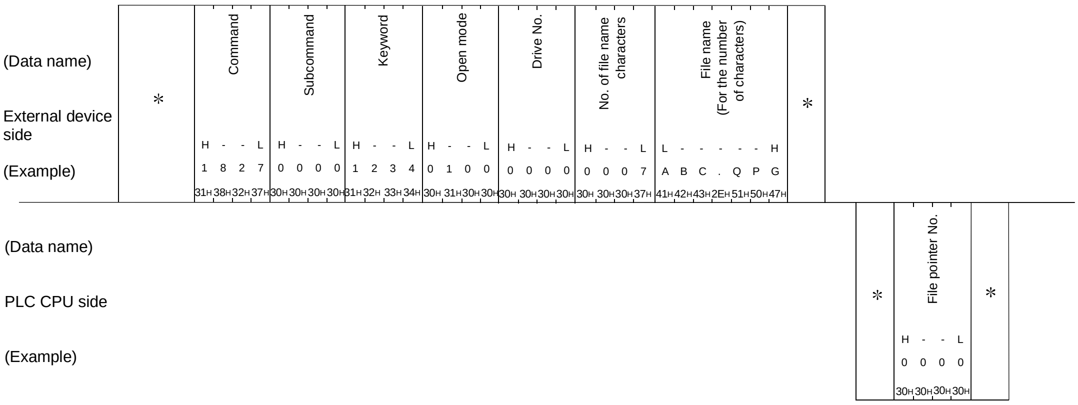
The examples shown in this section explain the control procedure for opening a file.
The order and content of data items of the areas marked by "✻" shown in the control procedure diagram differ depending on the module used as well as the frame and format used for communication.
See the detailed information described in Section 3.1.

[Control procedure]

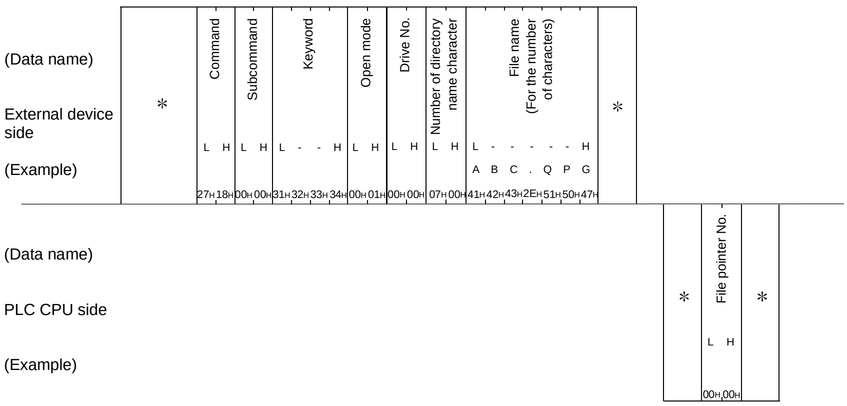
The following conditions apply to the example of opening a file below.

| | |
|---|---|
| Keyword | 1234 |
| Drive No. | 0 |
| Filename | ABC.QPG |
| Open mode | For read |

(1) Communication in ASCII code




(2) Communication in binary code




**POINT**

When a file is opened, it will be set to file CLOSE status by rebooting the QCPU (resetting the CPU), etc.

### 3.8.8 Close file (command: 182A): for the QCPU

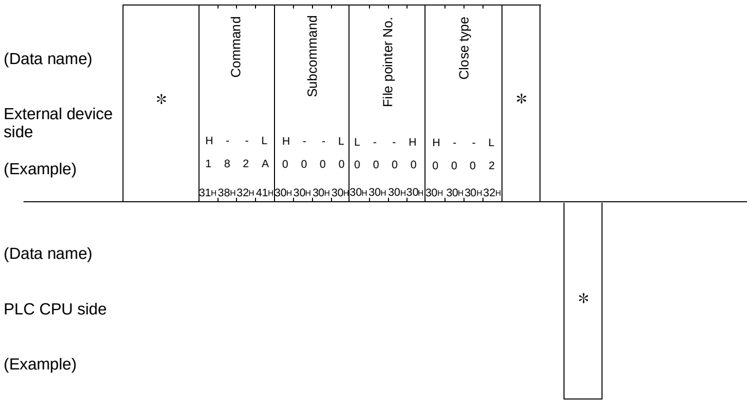
The examples shown in this section explain the control procedure for closing a file.
The order and content of data items of the areas marked by "*" shown in the control procedure diagram differ depending on the module used as well as the frame and format used for communication.
See the detailed information described in Section 3.1.

[Control procedure]

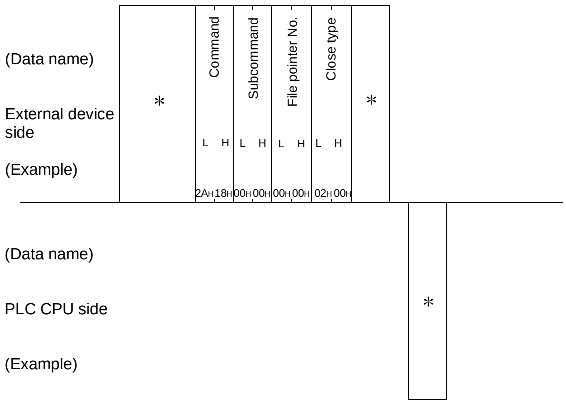
The following conditions apply to the example of closing file below.

File pointer number 0
Close type 2

(1) Communication in ASCII code




(2) Communication in ASCII code




**POINT**

(1) When a file is opened, it will be set to file CLOSE status by rebooting the QCPU (resetting the CPU), etc.

## 3.8.9 Read file (command: 1828): for the QCPU

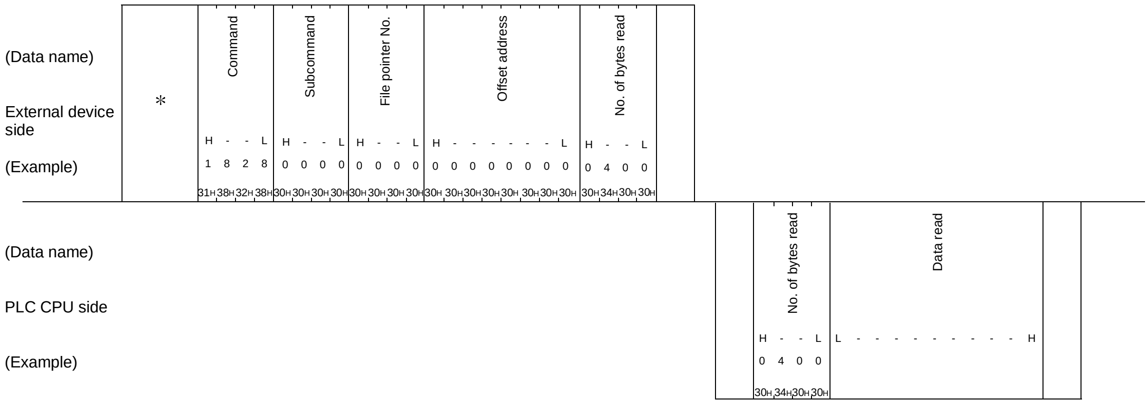
The examples shown in this section explain the control procedure for reading a file.
The order and content of data items of the areas marked by "✻" shown in the control procedure diagram differ depending on the module used as well as the frame and format used for communication.
See the detailed information described in Section 3.1.

[Control procedure]

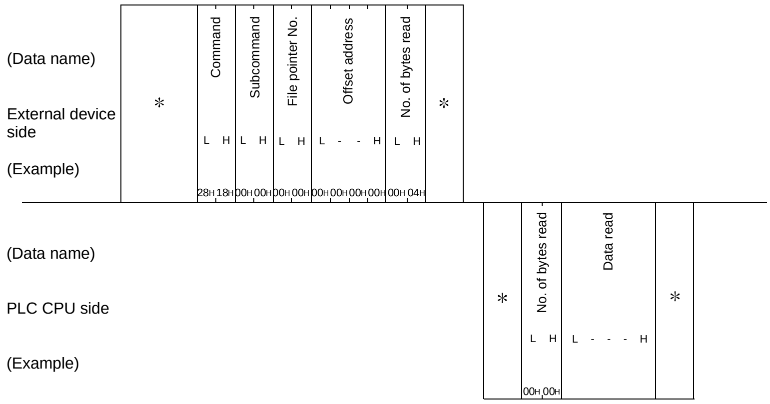
The following conditions apply to the example of reading a file below.

| | |
|---|---|
| File pointer number | 0 |
| Number of bytes read | 1K bytes |

### (1) Communication in ASCII code



(2) Communication in binary code




**POINT**

(1) The maximum number of bytes per data read operation when reading data is fixed.
Read all the data written to the designated file by dividing it into several parts by adjusting the offset address and number of bytes read.
Store the data read to the external device as is.
The file size can be checked using the following functions.

- Read file information list read function : See Section 3.8.5.
- File search function: : See Section 3.8.6.

(2) Designate the relevant values within the following range.

- Offset address : Designate an even number address within the following range.
  $0 \leq$ address $\leq$ (file size - 1)
- Number of bytes read : $0 \leq$ number of bytes $\leq 1920$

### 3.8.10 Create new file (command: 1820): for the QCPU

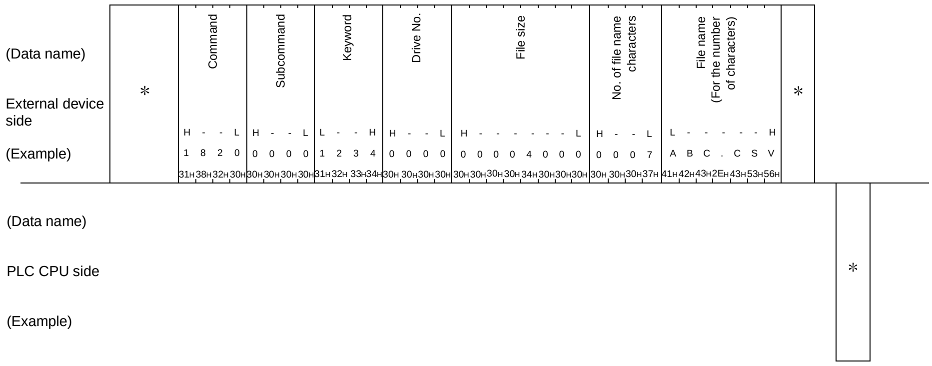
The examples shown in this section explain the control procedure for creating a new file. The order and content of data items of the areas marked by "✻" shown in the control procedure diagram differ depending on the module used as well as the frame and format used for communication.
See the detailed information described in Section 3.1.

[Control procedure]

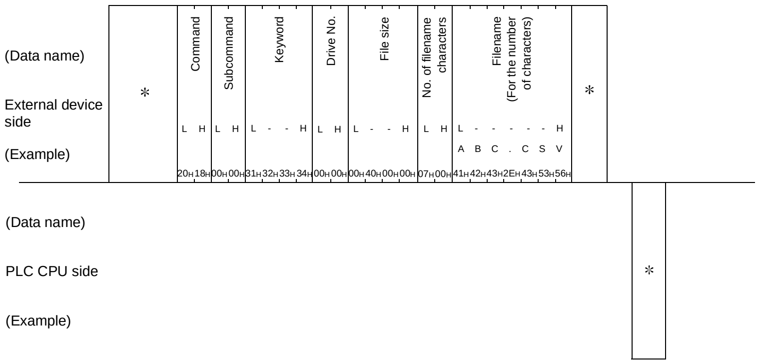
The following conditions apply to the example of creating a new file below:

| | |
|---|---|
| Keyword | 1234 |
| Drive No. | 0 |
| Filename | ABC.CSV |
| File size | 1K bytes |

(1) Communication in ASCII code




(2) Communication in binary code




**POINT**

(1) A new file can be created within the size of unused contiguous clusters on the designated drive memory.
When creating a new file, it is recommended to take data to be added later into account when deciding the file size.

(2) Designate "20H" as the attribute of the new file (disk files that can be read/written).

(3) The QCPU management time is registered as the date and time of last update for a new file created using this function.

### 3.8.11 Write to file (command: 1829): for the QCPU

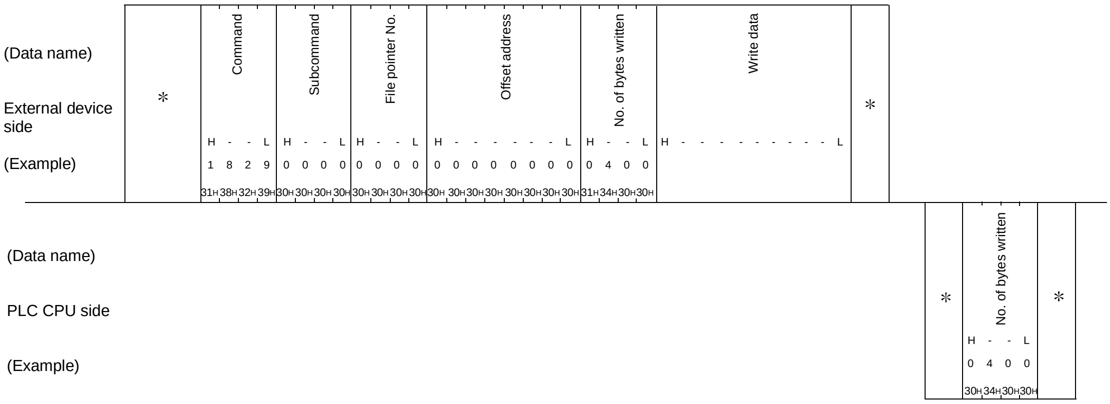
The examples shown in this section explain the control procedure for writing to a file.
The order and content of data items of the areas marked by "∗" shown in the control procedure diagram differ depending on the module used as well as the frame and format used for communication.
See the detailed information described in Section 3.1.

[Control procedure]

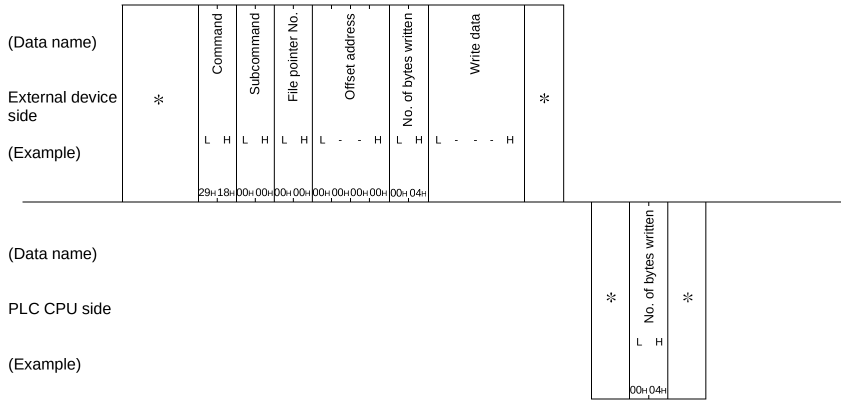
The following conditions apply to the example of writing to a file below.

| | |
|---|---|
| File pointer number | 0 |
| Offset address | 0 |
| Number of write bytes | 1K bytes |

#### (1) Communication in ASCII code



(2) Communication in binary code




POINT

(1) The maximum number of bytes per write operation when writing data is fixed. Write all the data read from the QCPU and stored in the external device to the designated file by dividing it into several parts by adjusting the offset address and number of bytes written.

(2) When the following files are designated while the QCPU is in operation, an error occurs and an abnormal completion code will be returned.
- Parameter file
- Files in the program memory (drive name: 00H) of the QCPU that is currently in operation.

(3) Designate the relevant values within the following range.
- Offset address : 0 ≤ address ≤ (file size - 1)
  When writing data to a file in the drive named "00H " (program memory), designate the address in multiples of 4 (0, 4,8,... in decimal notation) within the range above.
  When writing data to a file in a drive other than "00H," designate the addresses in even-numbered addresses (0, 2, 4, 6, 8,... in decimal notation) within the range above
- Number of bytes written : 0 ≤ number of bytes ≤ 1920

(4) When writing data to a file, the volume of the data must be within the file size reserved in the new file creation.
Writing data exceeding the file size may result in the write error (error code: 402BH).

### 3.8.12 Delete files (command: 1822): for the QCPU

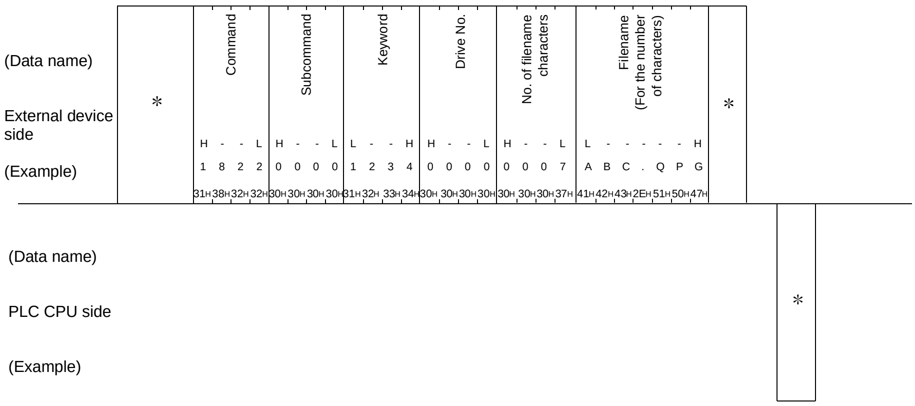
The examples shown in this section explain the control procedure for deleting a file.
The order and content of data items of the areas marked by "✻" shown in the control procedure diagram differ depending on the module used as well as the frame and format used for communication.
See the detailed information described in Section 3.1.

[Control procedure]

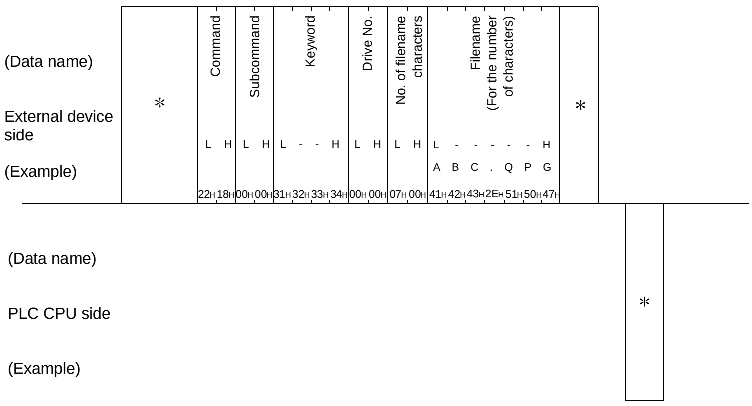
The following conditions apply to the example of deleting a file below.

| | |
|---|---|
| Keyword | 1234 |
| Drive No. | 0 |
| Delete filename | ABC.QPG |

(1) Communication in ASCII code




(2) Communication in binary code




**POINT**

(1) Determine the file deletion timing for the entire system, including the QCPU and related devices.

(2) Files on which the file open function has been executed cannot be deleted.

(3) When the QCPU is in the RUN status, the following files cannot be deleted.
- Program file (□. QPG)
- Parameter file (□. QPA)
- Boot setting file (□. QBT)

### 3.8.13 Copy files (command: 1824): for the QCPU

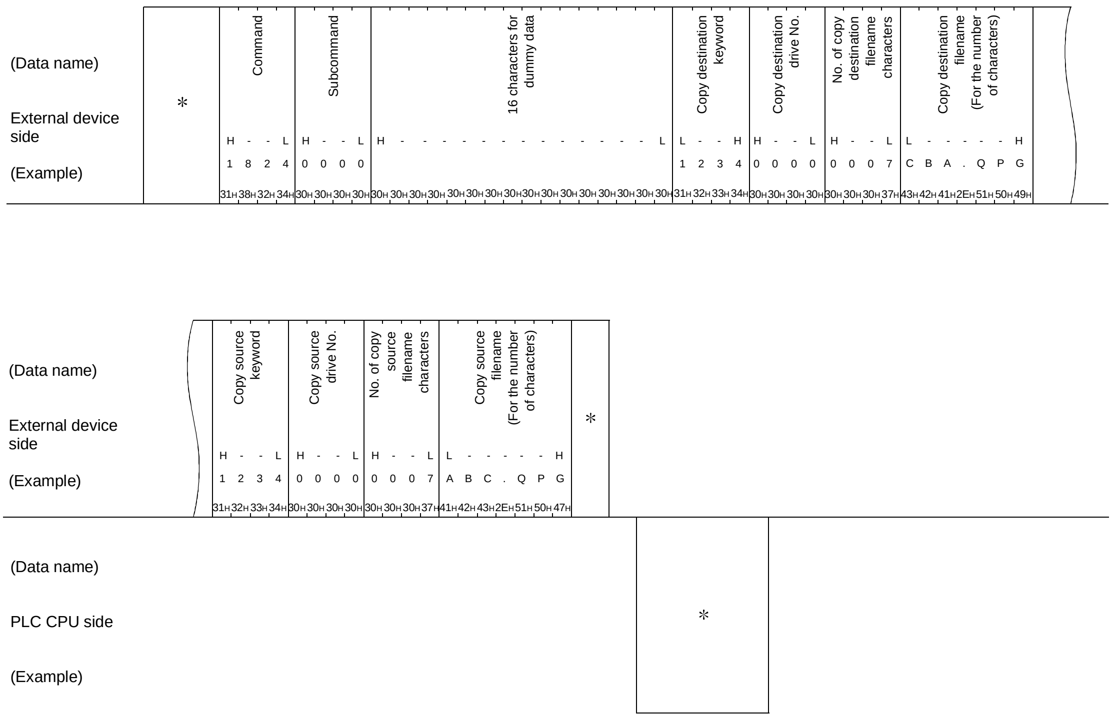
The examples shown in this section explain the control procedure for copying a file.
The order and content of data items of the areas marked by "✻" shown in the control procedure diagram differ depending on the module used as well as the frame and format used for communication.
See the detailed information described in Section 3.1.

[Control procedure]

The following conditions apply to the example of copying a file below.

| | |
|---|---|
| Keyword copy source/destination | 1234 |
| Drive number of copy source | 0 |
| Drive number of copy destination | 1 |
| Filename of copy source | ABC.QPG |
| Filename of copy destination | CBA.QPG |

(1) Communication in ASCII code



## (2) Communication in binary code

External device side

| (Data name) | * | Command | Subcommand | 16 characters for dummy data | Copy destination keyword | Copy destination drive No. | No. of copy destination filename characters | Copy destination filename (For the number of characters) | Copy source keyword | Copy source drive No. | No. of copy source filename characters | Copy source filename (For the number of characters) | * |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| (Example) | | L H | L H | L - - - - - - H | L - - H | L H | L H | L - - - - - H | H - - L | H - | - L | L - - - - - H | |
| | | | | | | | | A B C . Q P G | | | | A B C . Q P G | |
| | | 24H 18H | 00H 00H | 00H 00H 00H 00H 00H 00H 00H 00H | 31H 32H 33H 34H | 00H 00H | 07H 00H | 41H 42H 43H 2EH 51H 50H 47H | 31H 32H 33H 34H | 00H 00H | 07H 00H | 41H 42H 43H 2EH 51H 50H 47H | |

PLC CPU side

| (Data name) | * |
|---|---|
| (Example) | |

**POINT**

(1) The maximum number of bytes per copy operation when copying is fixed. Write all the data written to the existing file to the newly registered file by dividing it into several parts by adjusting the offset address and number of bytes written.

(2) When the following files are designated while the QCPU is in operation, an error occurs and an abnormal completion code will be returned.
- Parameter file
- Files in the program memory (drive name: 00H) of the QCPU that is currently in operation.

(3) Designate the relevant values within the following range.
- Offset address : 0 ≤ address ≤ (file size - 1)
  When writing data to a file in the drive named "00H " (program memory), designate the address in multiples of 4 (0, 4,8,... in decimal notation) within the range above.
  When writing data to a file in a drive other than "00H," designate the addresses in even-numbered addresses (0,2, 4, 6, 8,... in decimal notation) within the range above.

### 3.8.14 Change date of file creation (command: 1826): for the QCPU

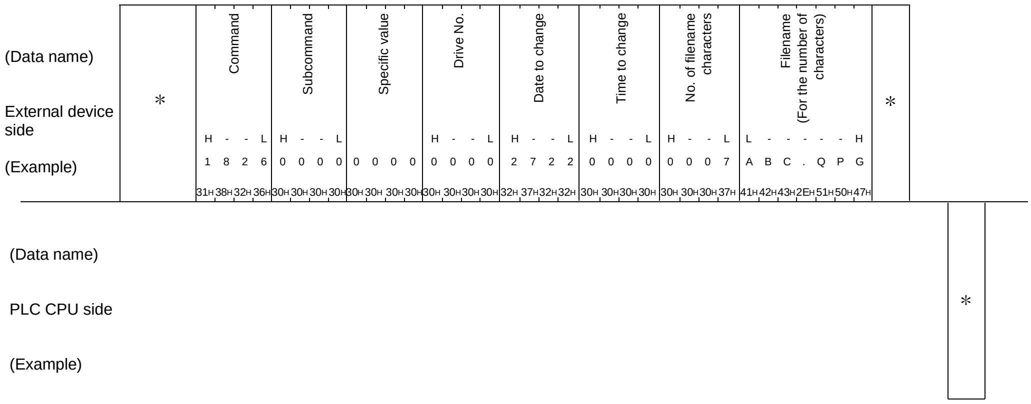
The examples shown in this section explain the control procedure for changing the date of file creation.
The order and content of data items of the areas marked by "∗" shown in the control procedure diagram differ depending on the module used as well as the frame and format used for communication.
See the detailed information described in Section 3.1.

[Control procedure]

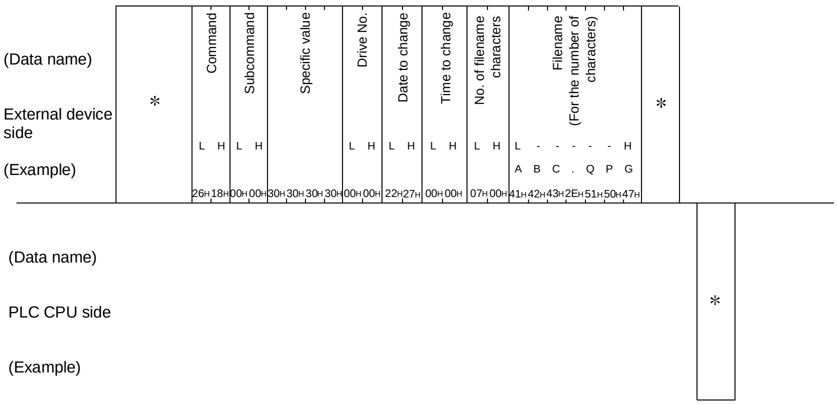
The following conditions apply to the example of changing the date of file creation below.

| | |
|---|---|
| Drive No. | 0 |
| Data to change | 1999/09/02 |
| Time to change | 0 a.m. |
| Filename | ABC.QPG |

(1) Communication in ASCII code




(2) Communication in binary code




**POINT**

When the following files are designated while the QCPU is in operation, an error occurs and an abnormal completion code will be returned.
- Parameter file
- Files in the program memory (drive name: 00H) of the QnACPU that is currently in operation.

### 3.8.15 Change file attributes (command: 1825): for the QCPU

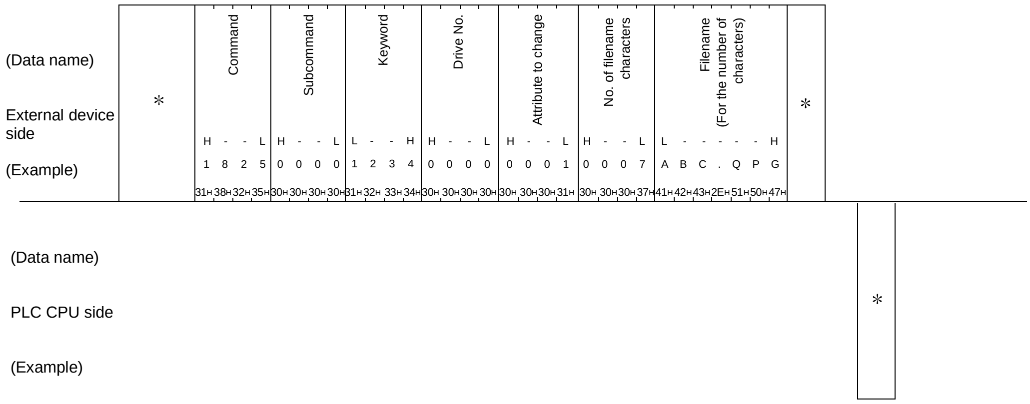
The examples shown in this section explain the control procedure for changing file attributes.
The order and content of data items of the areas marked by "✻" shown in the control procedure diagram differ depending on the module used as well as the frame and format used for communication.
See the detailed information described in Section 3.1.

[Control procedure]

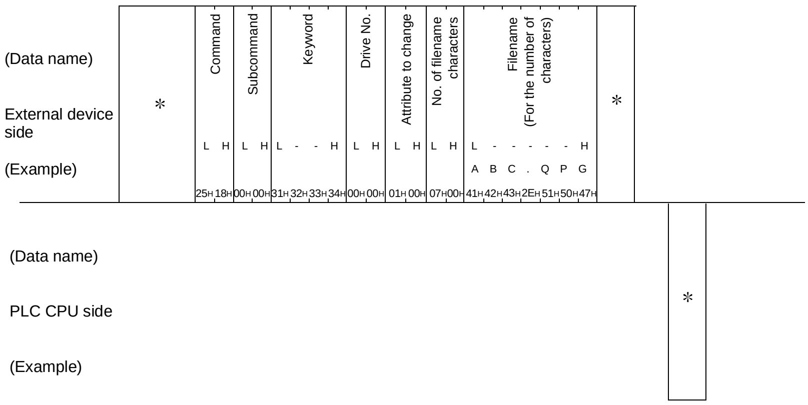
The following conditions apply to the example of changing file attributes below.

| | |
|---|---|
| Keyword | 1234 |
| Drive number | 0 |
| Filenames whose attributes are to be changed | ABC.QPG |
| Attributes to change | Read only |

(1) Communication in ASCII code




(2) Communication in binary code




**POINT**

(1) When the following files are designated while the QCPU is in operation, an error occurs and an abnormal completion code will be returned.
- Parameter file
- Files in the program memory (drive name: 00H) of the QCPU that is currently in operation.

(2) Attributes can be changed only between 01H (read only file) ↔ 20H (files that can be read/written to).

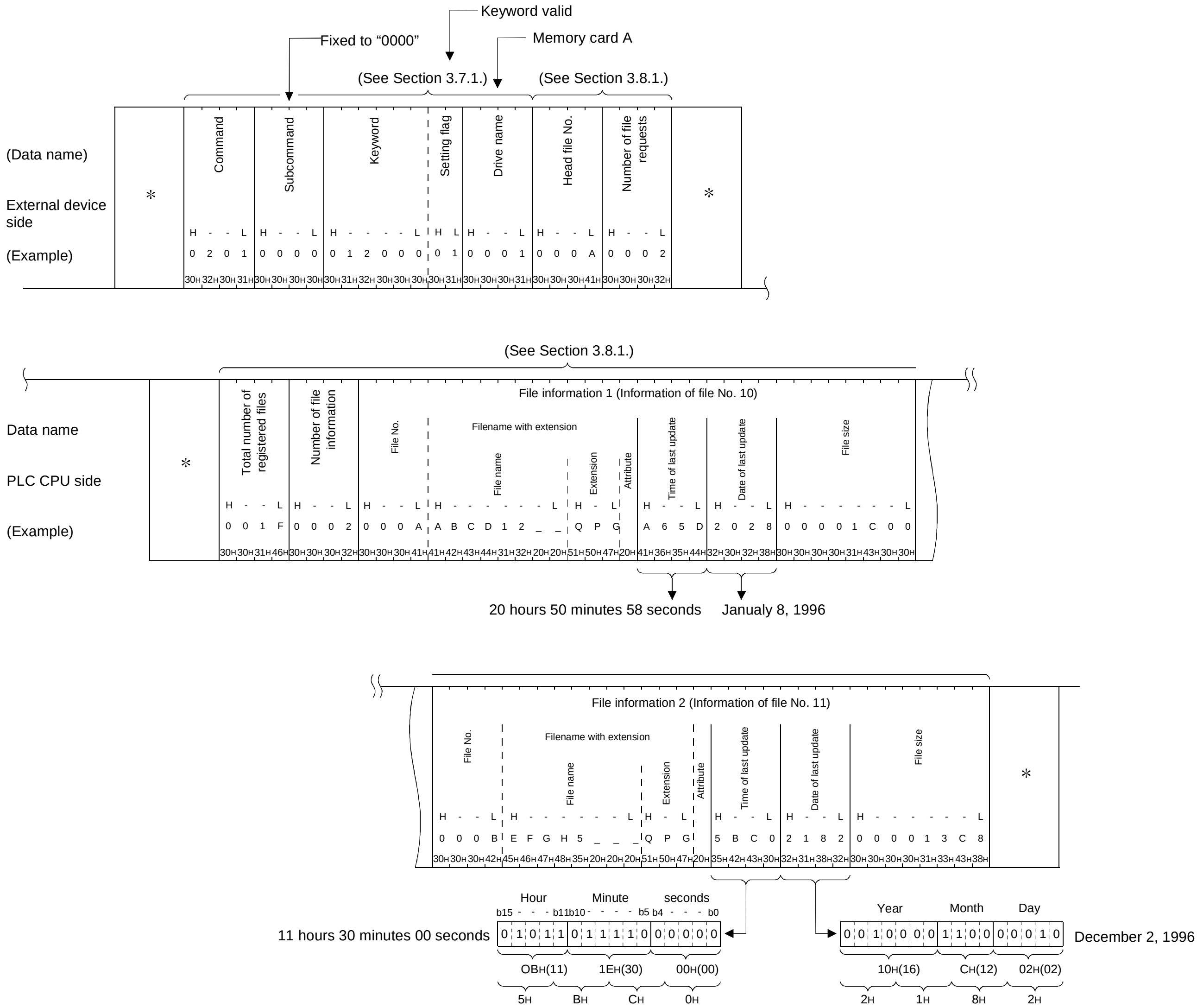
### 3.8.16 Read file information list: for the QnACPU

(1) Reading the file information list without header statement (command: 0201): For QnACPU

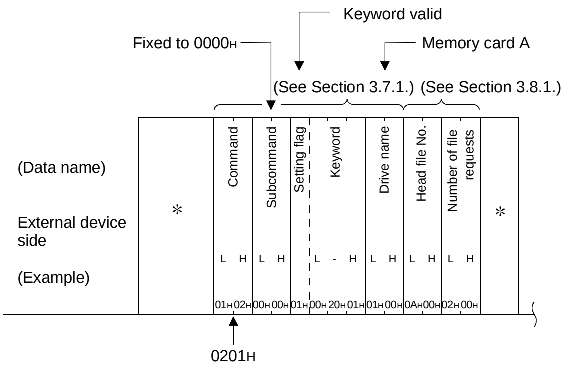
The examples shown in this section explain the control procedure for reading file information within the range of designated file numbers.
The order and content of data items of the areas marked by "*" shown in the control procedure diagram differ depending on the module used as well as the frame and format used for communication.
See the detailed information described in Section 3.1.

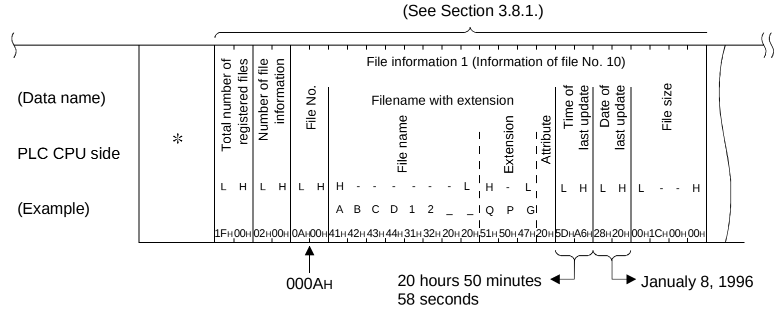
[Control procedure]

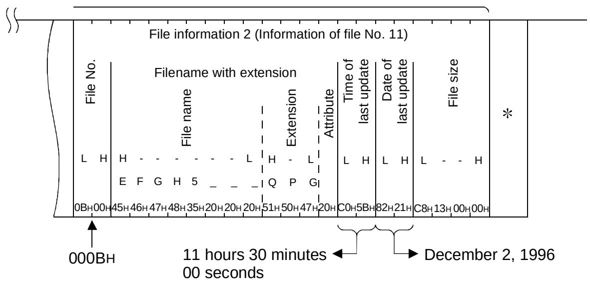
(a) Reading file information for two files from memory card A (RAM area, drive name: 01H) file number 10 (AH) while communicating in ASCII code.



(b) Reading file information for two files from memory card A (RAM area, drive name: 01H) file number 10 (AH) while communicating in binary code.




**POINT**

(1) Designate the following items within the following ranges indicated below. The values for these items will also be returned within the following ranges.

- Head file number : 1 ≤ file number ≤ 256
- Number of file requested : 1 ≤ file number ≤ 36
- Total number of registered files : 1 ≤ number of files ≤ 256
- Number of file information : 0 ≤ file count ≤ number of file requested (0: No registered files after the designated head file number)

(2) The total number of registered files is the current total number of files registered in the designated drive.

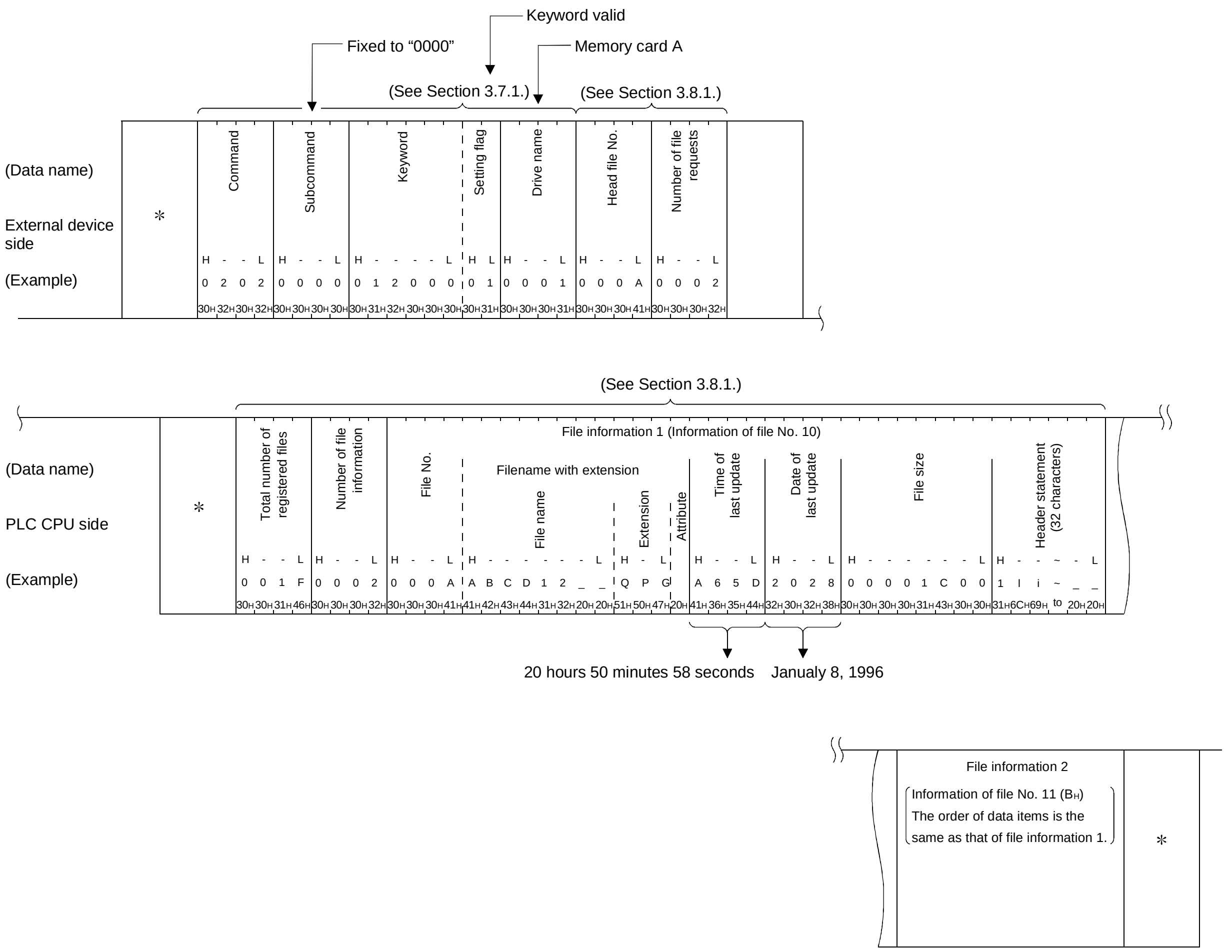
(3) When all the files are not registered within the designated file number range, the number of file information becomes the number of files registered in the designated range (Return number of file information).

## (2) Read file information list with header statement (command: 0202): for QnACPU

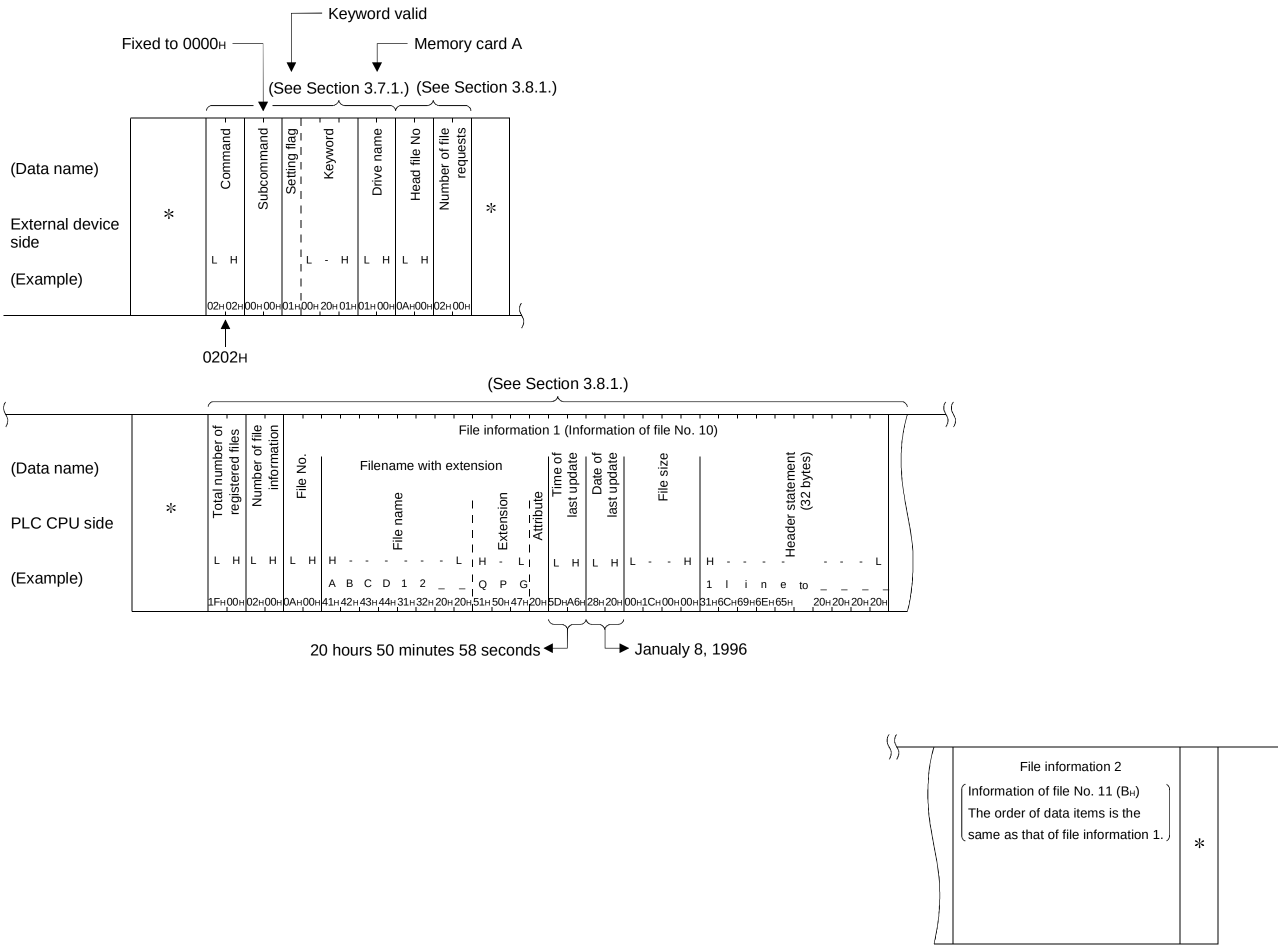
The examples shown in this section explain the control procedure for reading file information within the range of designated file numbers with header.
The order and content of data items of the areas marked by "∗" shown in the control procedure diagram differ depending on the module used as well as the frame and format used for communication.
See the detailed information described in Section 3.1.

[Control procedure]

(a) Reading file information for two files from memory card A (RAM area, drive name: $01_H$) file number 10 ($A_H$) while communicating in ASCII code.



(b) Reading file information for two files from memory card A (RAM area, drive name: 01H) file number 10 (AH) while communicating in binary code.




## POINT

(1) Designate the following items within the following ranges indicated below. The values for these items will also be returned within the following ranges.

- Head file number : 1 ≤ file number ≤ 256
- Number of file requested : 1 ≤ file number ≤ 16
- Total number of registered files : 1 ≤ number of files ≤ 256
- Number of file information : 0 ≤ file count ≤ number of file requested (0: No registered files after the designated head file number)

(2) The total number of registered files is the current total number of files registered in the designated drive.

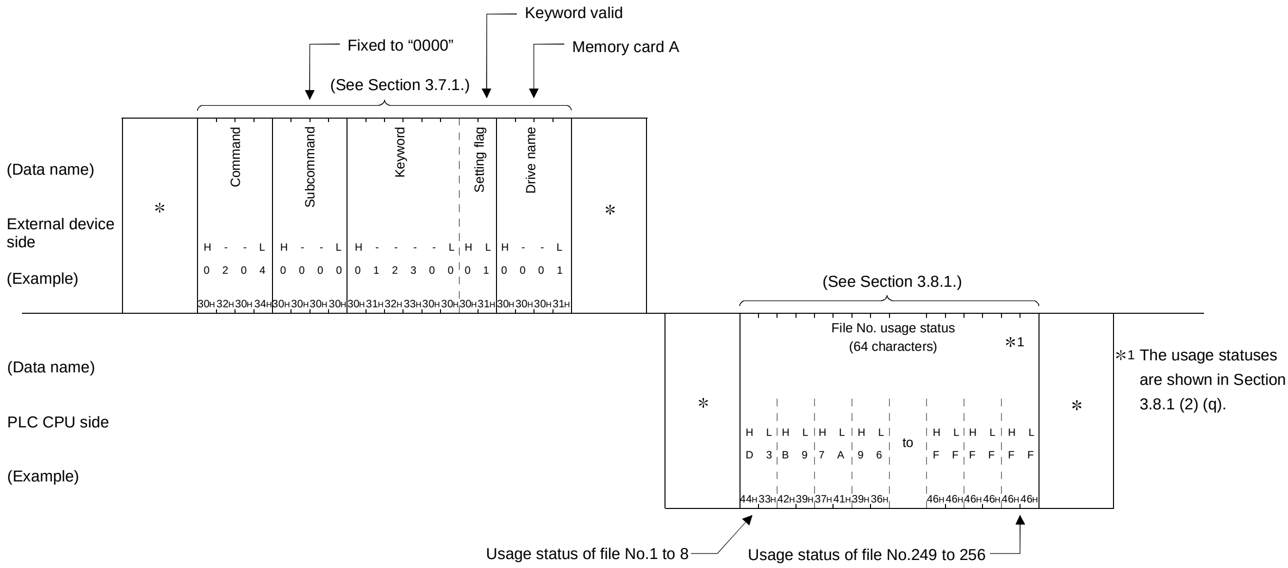
(3) When all the files are not registered within the designated file number range, the number of file information becomes the number of files registered in the designated range (Return number of file information).

(3) Read file number usage status (command: 0204): For QnACPU

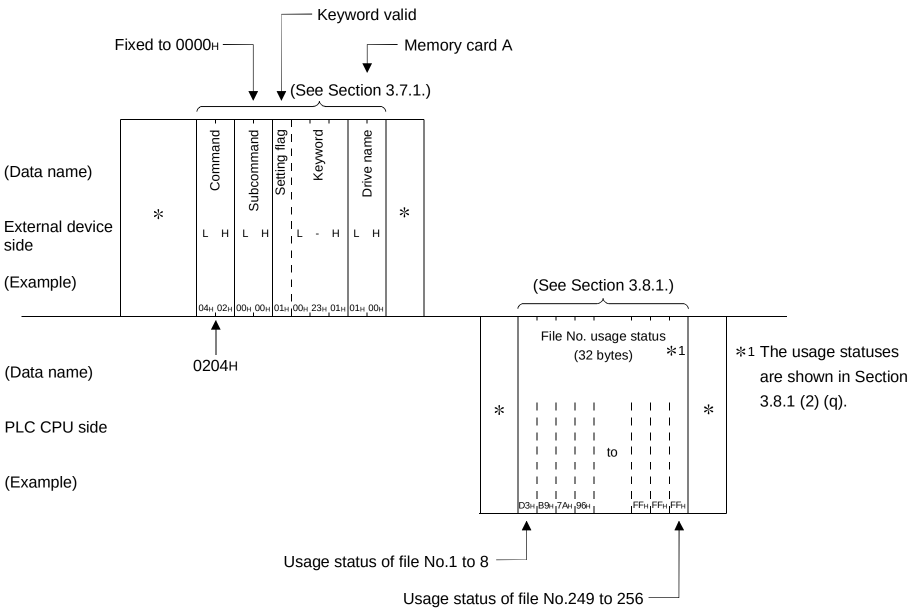
The examples shown in this section explain the control procedure for reading the usage status of a file number.
The order and content of data items of the areas marked by "∗" shown in the control procedure diagram differ depending on the module used as well as the frame and format used for communication.
See the detailed information described in Section 3.1.

[Control procedure]

(a) Reading the usage status of the memory card A (RAM area, drive name: 01H) file number usage status while communicating in ASCII code.



(b) Reading the usage status of the memory card A (RAM area, drive name: 01H) file number usage status while communicating in binary code.




**POINT**

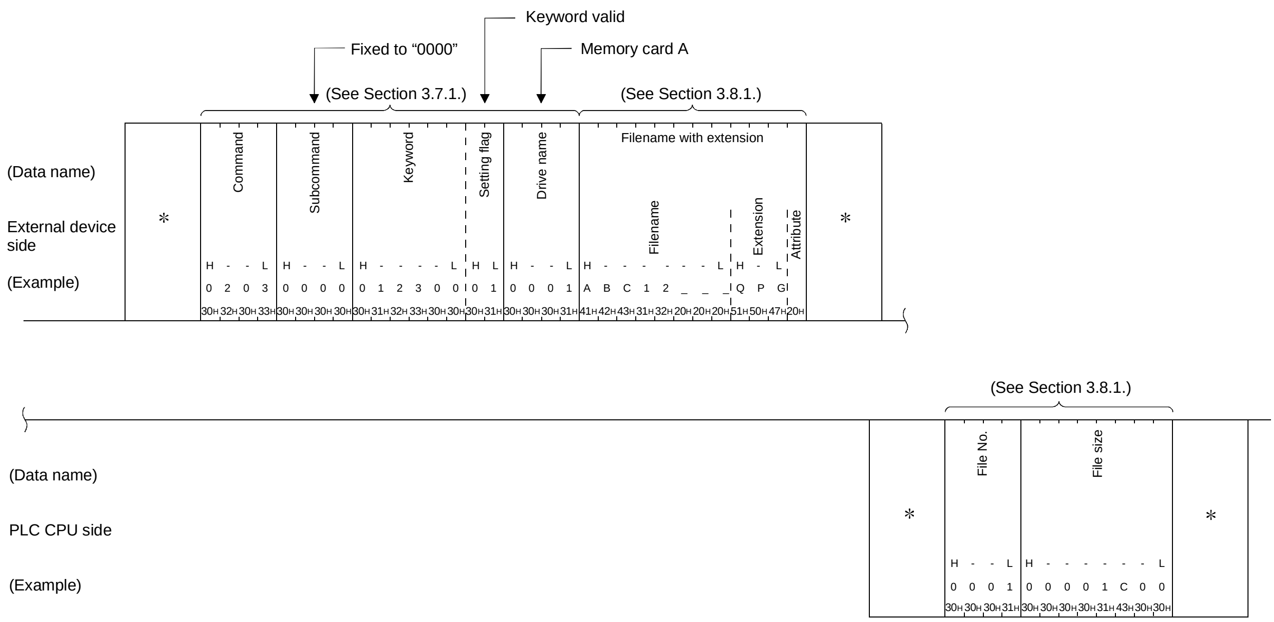
When a drive memory that cannot store more than 256 files is designated, the file numbers that correspond to the files which cannot be stored (exceeding 256 files) becomes "in use" (bit: 1).

### 3.8.17 Reading file presence information (file search) (command: 0203): for the QnACPU

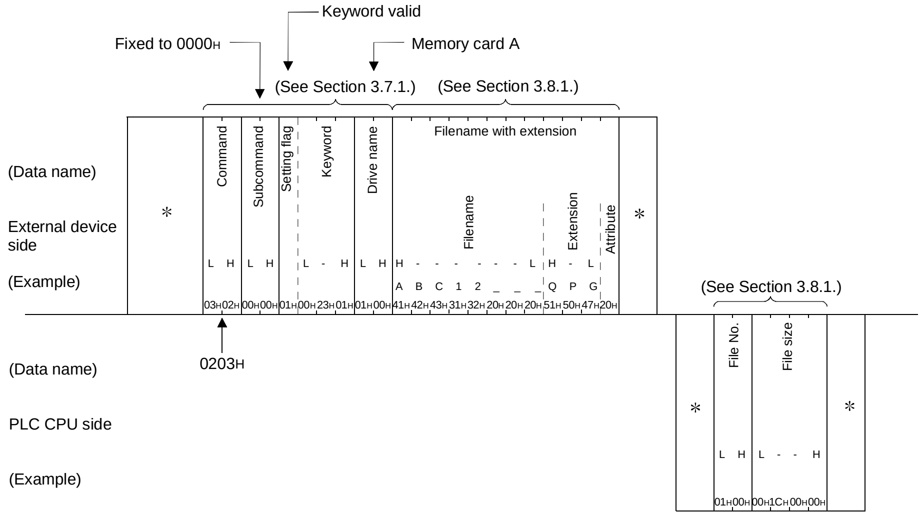
The examples shown in this section explain the control procedure for searching for the designated file, and reading the file No. and file size if the file is found.
The order and contents of data items of the areas marked by "*" shown in the control procedure diagram differ depending on the module used as well as the frame and format used for communication.
See the detailed information described in Section 3.1.

[Control procedure]

(1) When searching for the memory card A (RAM area, drive name: $01_H$) filename "ABC12.QPG" by communication in ASCII code



(2) When searching for the memory card A (RAM area, drive name: 01H) filename "ABC12.QPG" by communication in binary code




| POINT |
|---|
| (1) The file attribute to be read is handled as dummy data. |
| (2) When the designated file does not exist, an error occurs and the complete code at the occurrence of error is returned. |

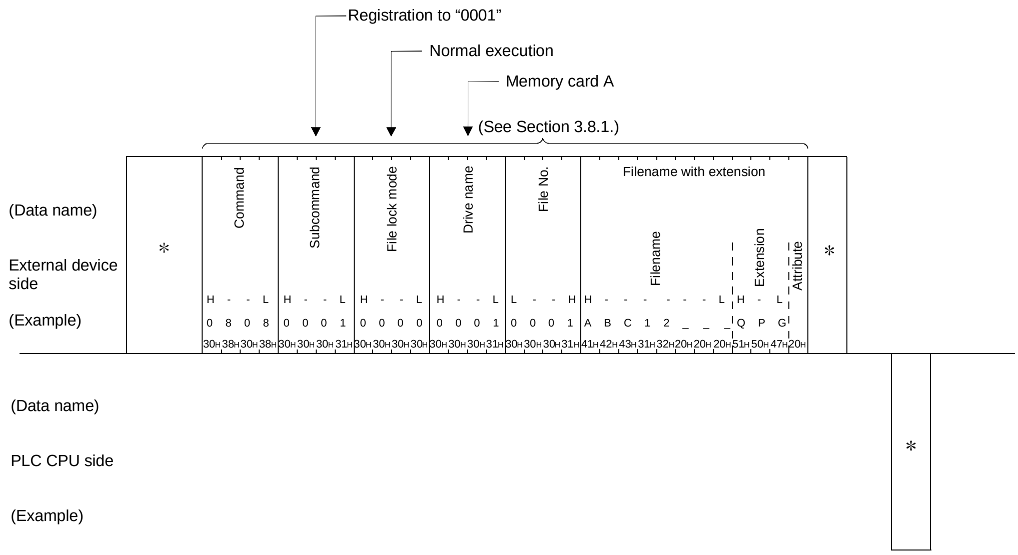
### 3.8.18 Registering and clearing file locks (command: 0808): for the QnACPU

The examples shown in this section explain the control procedure for registering and clearing the following file locks:

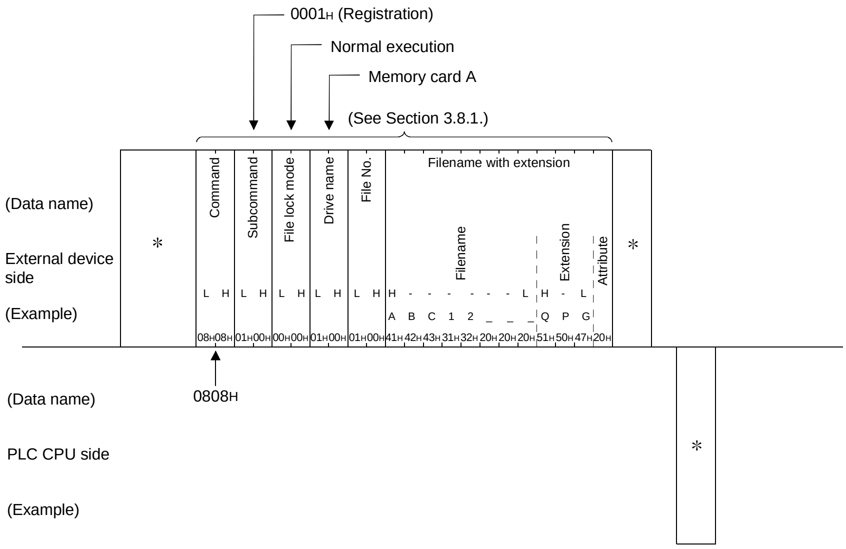
- File lock registration (Disable access from other devices)
  Registers file lock so that the contents of the file cannot be changed from another device and the same file cannot be accessed from another device while the designated file is being accessed.
- File lock clear (Enable access from other devices)
  Clears file lock of a file that was locked so that the same file can be accessed from other devices.
  The order and contents of data items of the area marked by "*" shown in the control procedure diagram differ depending on the module used as well as the frame and format used for communication.
  See the detailed information described in Section 3.1.

[Control procedure]

(1) When performing file lock registration in normal execution mode for the memory card A (RAM area, drive name: 01H) file No. 1 with filename "ABC12.QPG" by communication in ASCII code



(2) When performing file lock registration in normal execution mode for the memory card A (RAM area, drive name: 01H) file No. 1 with filename "ABC12.QPG" by communication in binary code



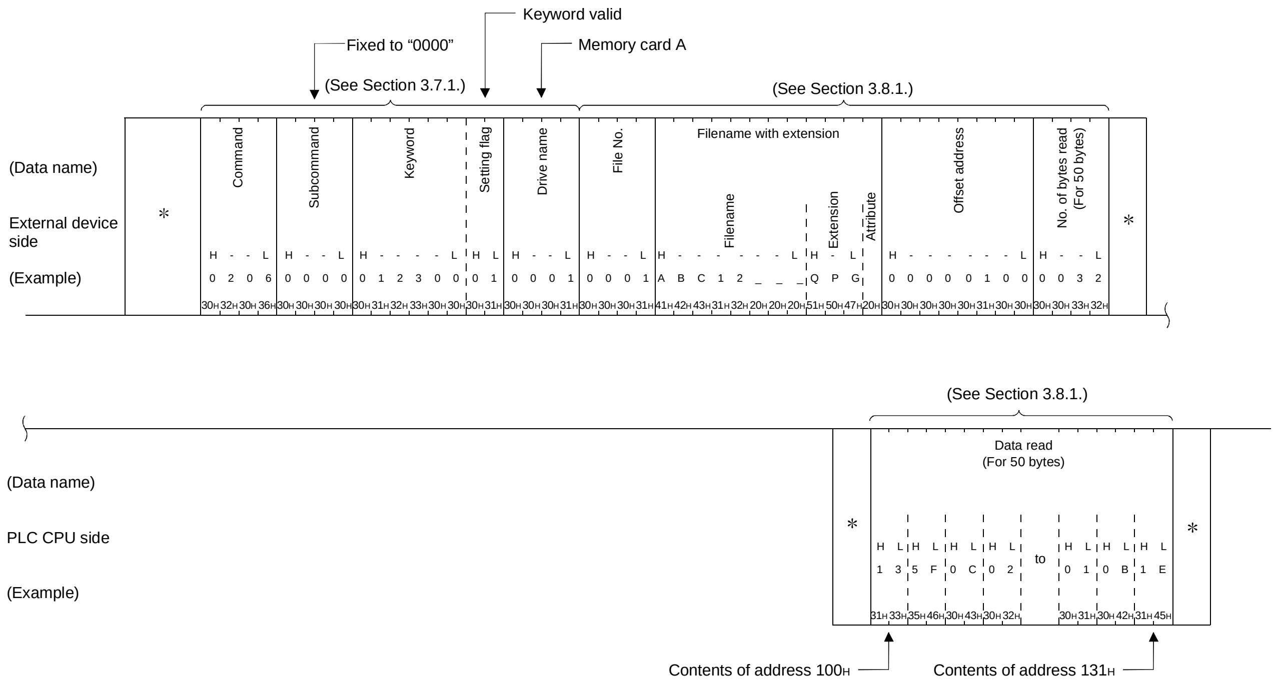

| POINT |
| --- |
| (1) The attribute designated during file creation, etc. is valid as the attribute of the file that will be locked and cleared. When registering and clearing file lock, treat the file attribute as dummy data.<br>(2) When file lock is being registered, the file lock registration will be cleared when the QnACPU is restarted (CPU reset, etc.). |

### 3.8.19 Reading the contents of a file (command: 0206): for the QnACPU

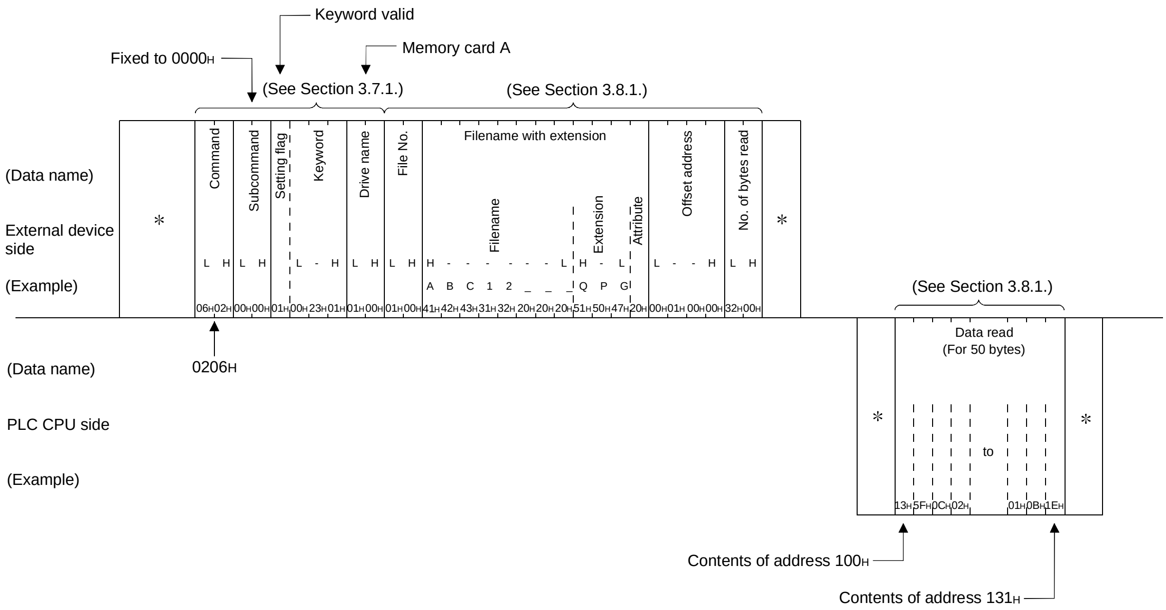
The examples shown in this section explain the control procedure for reading the data written in the designated file.
The order and contents of data items of the areas marked by "*" shown in the control procedure diagram differ depending on the module used as well as the frame and format used for communication.
See the detailed information described in Section 3.1.

[Control procedure]

(1) When reading the 50-byte data beginning from offset address 100H of the memory card A (RAM area, drive name: 01H) file No. 1 with filename "ABC12.QPG" by communication in ASCII code



(2) When reading the 50-byte data beginning from offset address 100H of the memory card A (RAM area, drive name: 01H) file No. 1 with filename "ABC12.QPG" by communication in binary code




**POINT**

(1) The maximum number of bytes per data read is fixed.
Read all the data written to the designated file by dividing it into several parts by adjusting the offset address and number of bytes read.
Store the data read to the external device as is.
The file size can be checked using the following functions.

- Read file information list read function : See Section 3.8.16.
- File search function : See Section 3.8.17.

(2) The file attribute to be read is handled as dummy data.

(3) Designate the relevant values within the following range.

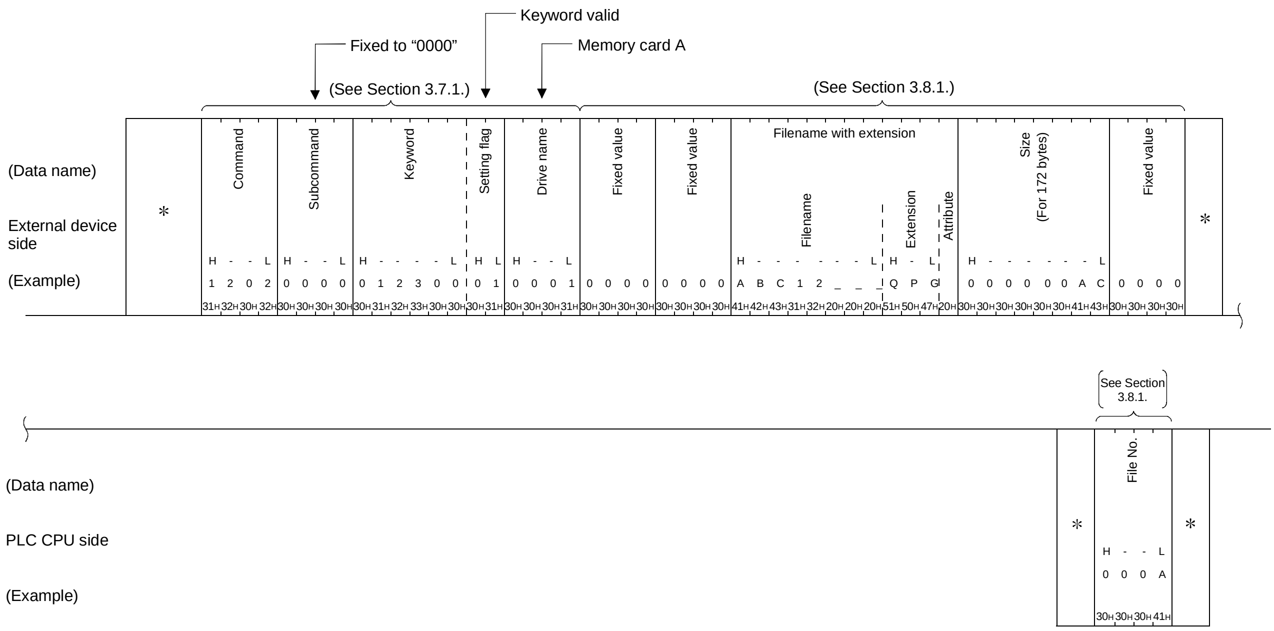
- File number : 0 ≤ file No. ≤ 256
- Offset address : Designate an even number address within the following range.
0 ≤ address ≤ (file size - 1)
- Number of bytes read : 0 ≤ number of bytes ≤ 960

## 3.8.20 Creating a new file (filename registration) (command: 1202): for the QnACPU

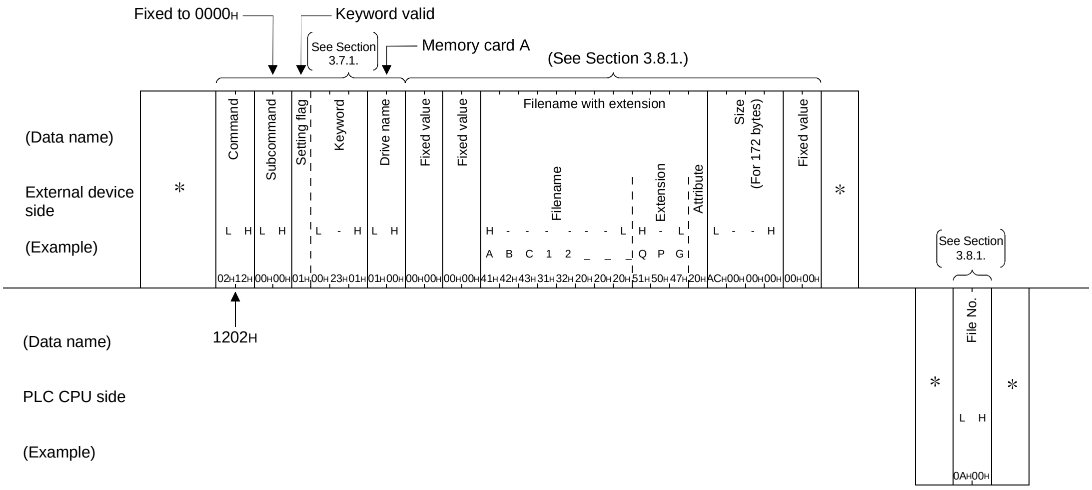
The examples shown in this section explain the control procedure for registering a new file to and reserving a file area on the designated disk.
The order and contents of data items of the areas marked by "*" shown in the control procedure diagram differ depending on the module used as well as the frameand format used for communication.
See the detailed information described in Section 3.1.

[Control procedure]

(1) When registering a new 172-byte file to and reserve a file area on memory card A (RAM area, drive name: 01H) with filename "ABC12.QPG" by communication in ASCII code



(2) When registering a new 172-byte file to and reserve a file area on memory card A (RAM area, drive name: 01H) with filename "ABC12.QPG" by communication in binary code




| POINT |
|---|

(1) A new file can be created within a size that links unused clusters (see Section 6.5) on the designated drive memory.
When creating a new file, it is recommended that the future data addition be taken into account when deciding the file size.

(2) Designate "20H" (readable/writable disk file) as the attribute of the new file.

(3) Use the write to file function (command: 1203) described in Section 6.6.9 to write data to a new file created using this function.
The contents of a file in which data has not been written cannot be read.

(4) The QnACPU management time is registered as the date and time of last update to a new file created using this function.

### 3.8.21 Writing to a file (command: 1203): for the QnACPU

#### (1) Batch write

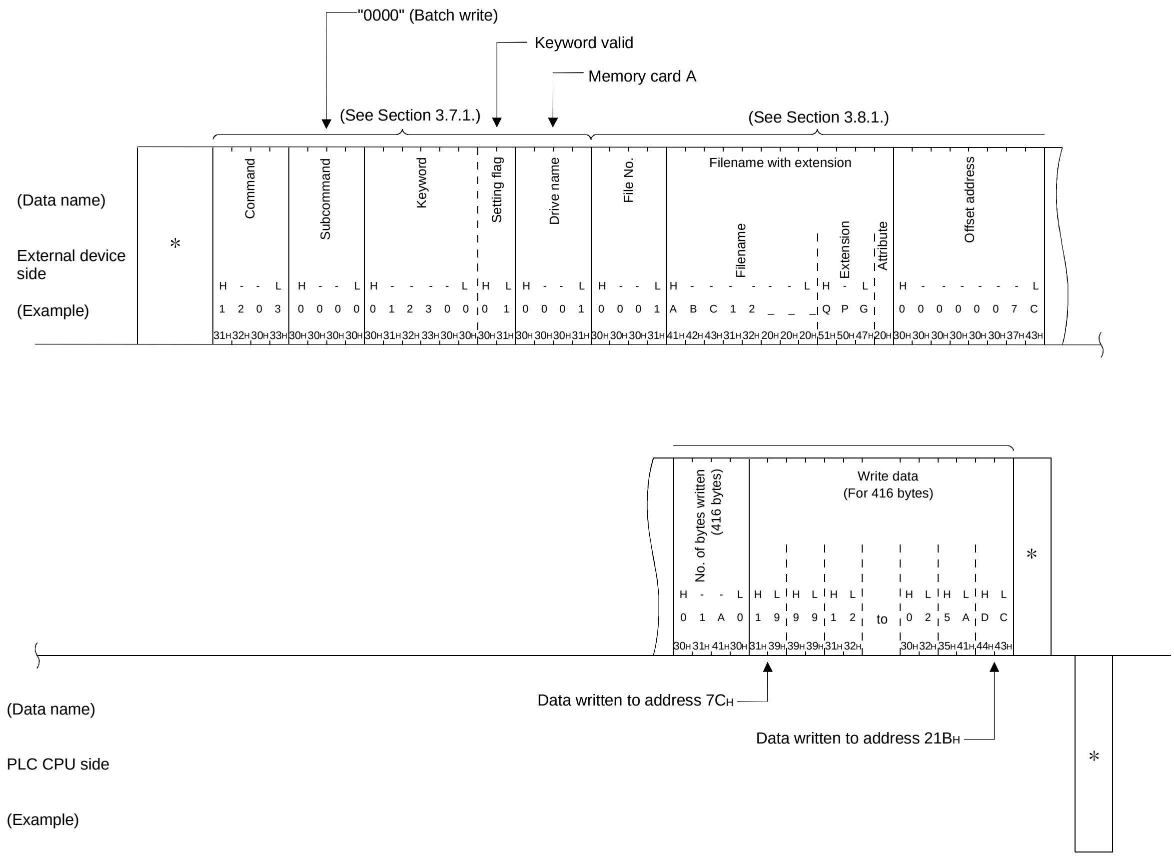
The examples shown in this section explain the control procedure for writing the data of the file that has been read from the QnACPU and stored in the external device to the designated file.

The order and contents of data items of the areas marked by "∗" shown in the control procedure diagram differ depending on the module used as well as the frame and format used for communication.

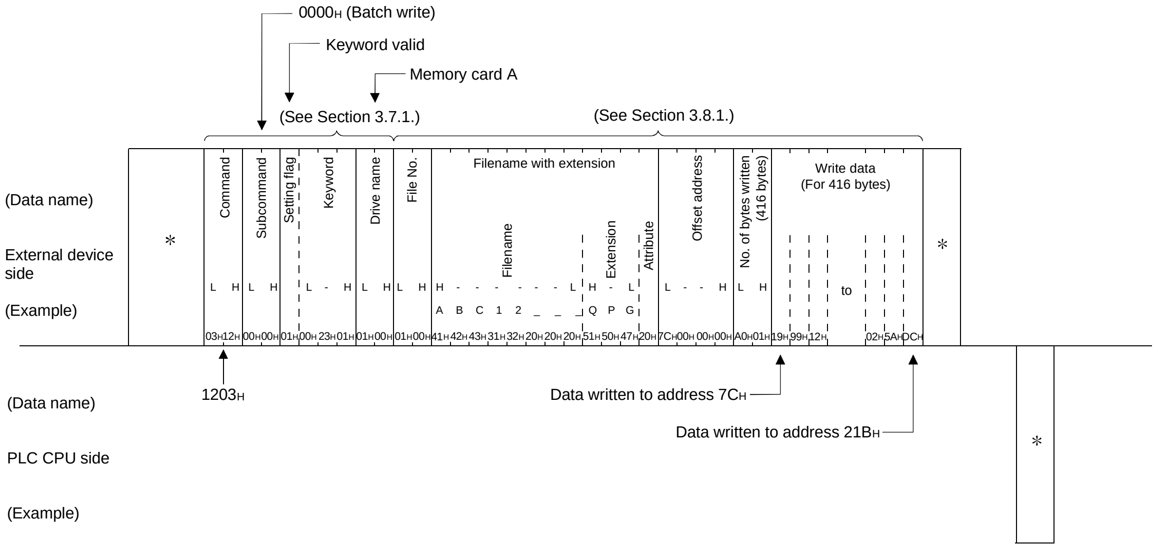
See the detailed information described in Section 3.1.

[Control procedure]

(a) When writing 416-byte data, beginning from the offset address 7CH, to the memory card A (RAM area, drive name: 01H) file No. 1 with filename "ABC12.QPG" by communication in ASCII code



(b) When writing 416-byte data, beginning from the offset address 7CH, to the memory card A (RAM area, drive name: 01H) file No. 1 with filename "ABC12.QPG" by communication in binary code




| POINT |
|---|

(1) The maximum number of bytes per write operation when writing data is fixed. Write all the data read from the QnACPU and stored in the external device to the designated file by dividing it into several parts by adjusting the offset address and number of bytes written.

(2) The attribute designated during file creation, etc. is valid as the attribute of the file to which data is written.
When writing data, treat the attribute as dummy data.

(3) When the following files are designated while the QnACPU is in operation, an error occurs and an abnormal completion code will be returned.
- Parameter file
- Currently executing file in the built-in RAM (drive name: 00H)

(4) Designate the relevant values within the following range.
- Offset address: : 0 ≤ address ≤ (file size - 1)
When writing data to a file in the drive named "00H" (built-in RAM), designate the address in multiples of 4 (0, 4, 8,... in decimal notation) within the range above.
When writing data to a file in a drive other than "00H", designate the addresses in even-numbered addresses (0, 2, 4, 6, 8,... in decimal notation) within the range above.
- Number of bytes written : 0 ≤ number of bytes ≤ 960

(2) Writing the same data (FILL)

The examples shown in this section explain the control procedure for writing n-bytes of arbitrary one-word data to the designated file.

The order and contents of data items of the areas marked by "∗" shown in the control procedure diagram differ depending on the module used as well as the frame and format used for communication.

See the detailed information described in Section 3.1.

[Control procedure]

(a) When writing 416 bytes of "FFFFH", beginning from offset address 7CH, to the memory card A (RAM area, drive name: 01H) file No. 1 with filename "ABC12.QPG" by communication in ASCII code

"0001" (Writing the same data)

Keyword valid

Memory card A

(See Section 3.7.1.) (See Section 3.8.1.)

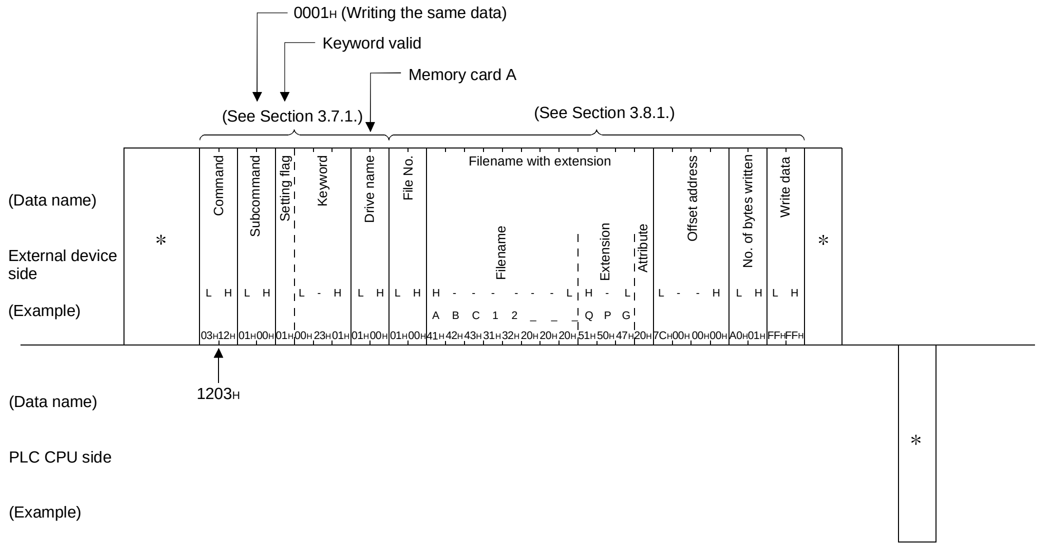
External device side

| (Data name) | ∗ | Command | Subcommand | Keyword | Setting flag | Drive name | File No. | Filename with extension: Filename | Filename with extension: Extension | Filename with extension: Attribute | Offset address | No. of bytes written (416 bytes) | Write data | ∗ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | H - - L | H - - L | H - - - - L | H L | H - - L | H - - L | H - - - - - - L | H - L | | H - - - - - - L | H - - L | H - - L | |
| (Example) | | 1 2 0 3 | 0 0 0 1 | 0 1 2 3 0 0 | 0 1 | 0 0 0 1 | 0 0 0 1 | A B C 1 2 _ _ _ | Q P G | | 0 0 0 0 0 0 7 C | 0 1 A 0 | F F F F | |
| | | 31H 32H 30H 33H | 30H 30H 30H 31H | 30H 31H 32H 33H 30H 30H | 30H 31H | 30H 30H 30H 31H | 30H 30H 30H 31H | 41H 42H 43H 31H 32H 20H 20H 20H | 51H 50H 47H | 20H | 30H 30H 30H 30H 30H 30H 37H 43H | 30H 31H 41H 30H | 46H 46H 46H 46H | |

PLC CPU side

| (Data name) | ∗ |
|---|---|
| (Example) | |

(b) When writing 416 bytes of "FFFFH", beginning from offset address 7CH, to the memory card A (RAM area, drive name: 01H) file No. 1 with filename "ABC12.QPG" by communication in binary code




## POINT

(1) The maximum number of bytes that can be written by one write operation is fixed. Adjust the offset address and the number of write bytes, and write arbitrary one-word data within the size of the designated file in more than one write operation. Since data is written in byte units, when the size of the remaining data is one byte, the value of the higher byte (bits 8 to 15) of one-word data will not be written.

(2) The attribute designated during file creation, etc. is valid as the attribute of the file to which data is written.
When writing data, treat the attribute as dummy data.

(3) When the following files are designated while the QnACPU is in operation, an error occurs and an abnormal completion code will be returned.
- Parameter file
- Currently executing file in the built-in RAM (drive name: 00H)

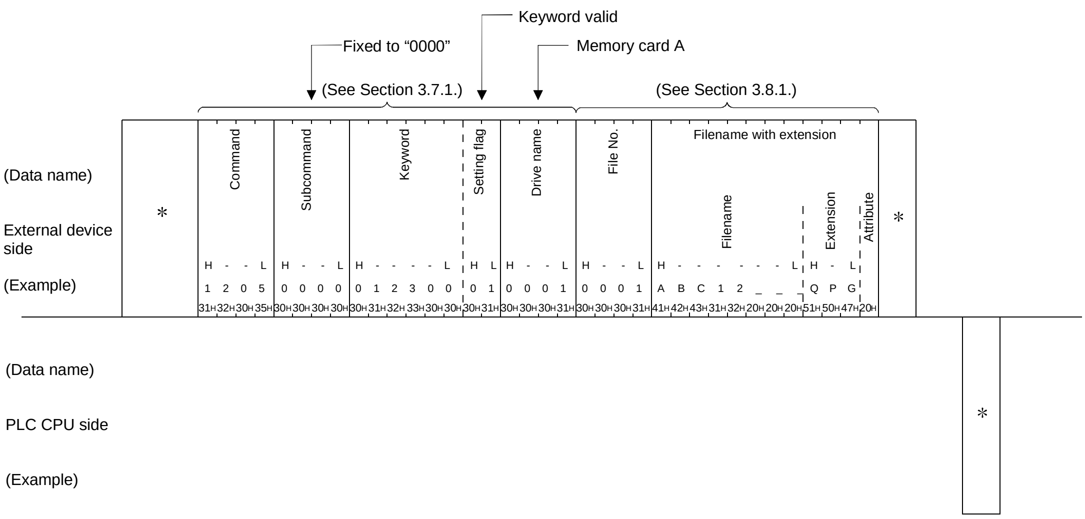
(4) Designate the relevant values within the following range.
- Offset address : $0 \leq$ address $\leq$ (file size - 1)
  When writing data to a file in the drive named "00H " (built-in RAM), designate the address in multiples of 4 (0, 4, 8,... in decimal notation) within the range above.
  When writing data to a file in a drive other than "00H", designate the addresses in even-numbered addresses (0, 2, 4, 6, 8,... in decimal notation) within the range above.
- Number of bytes written : $0 \leq$ number of bytes $\leq 960$

### 3.8.22 Deleting files (command: 1205): for the QnACPU

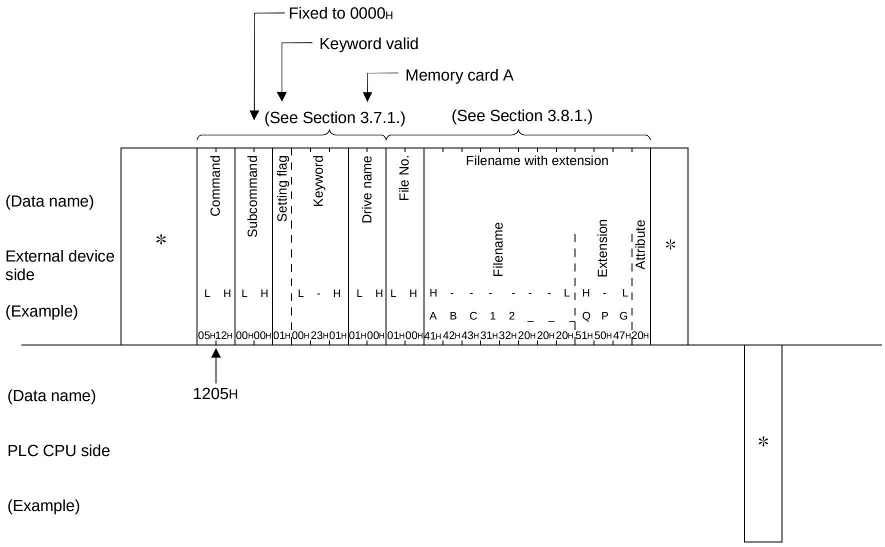
The examples shown in this section explain the control procedure for deleting an existing file.
The order and contents of data items of the areas marked by "∗" shown in the control procedure diagram differ depending on the module used as well as the frame and format used for communication.
See the detailed information described in Section 3.1.

[Control procedure]

(1) When deleting existing memory card A (RAM area, drive name: 01H) with filename "ABC12.QPG" by communication in ASCII code



(2) When deleting existing memory card A (RAM area, drive name: 01H) with filename "ABC12.QPG" by communication in binary code




**POINT**

(1) The attribute designated during file creation, etc. is valid as the attribute of the file to be deleted.
When deleting a file, treat the attribute as dummy data.

(2) Determine the file deletion timing for the entire system, including the QnACPU and related devices.

(3) Locked files cannot be deleted.

(4) When the QnACPU is in the RUN status, the following files cannot be deleted.
- Program file ( □. QPG)
- Parameter file ( □. QPA)
- Boot setting file ( □. QBT)

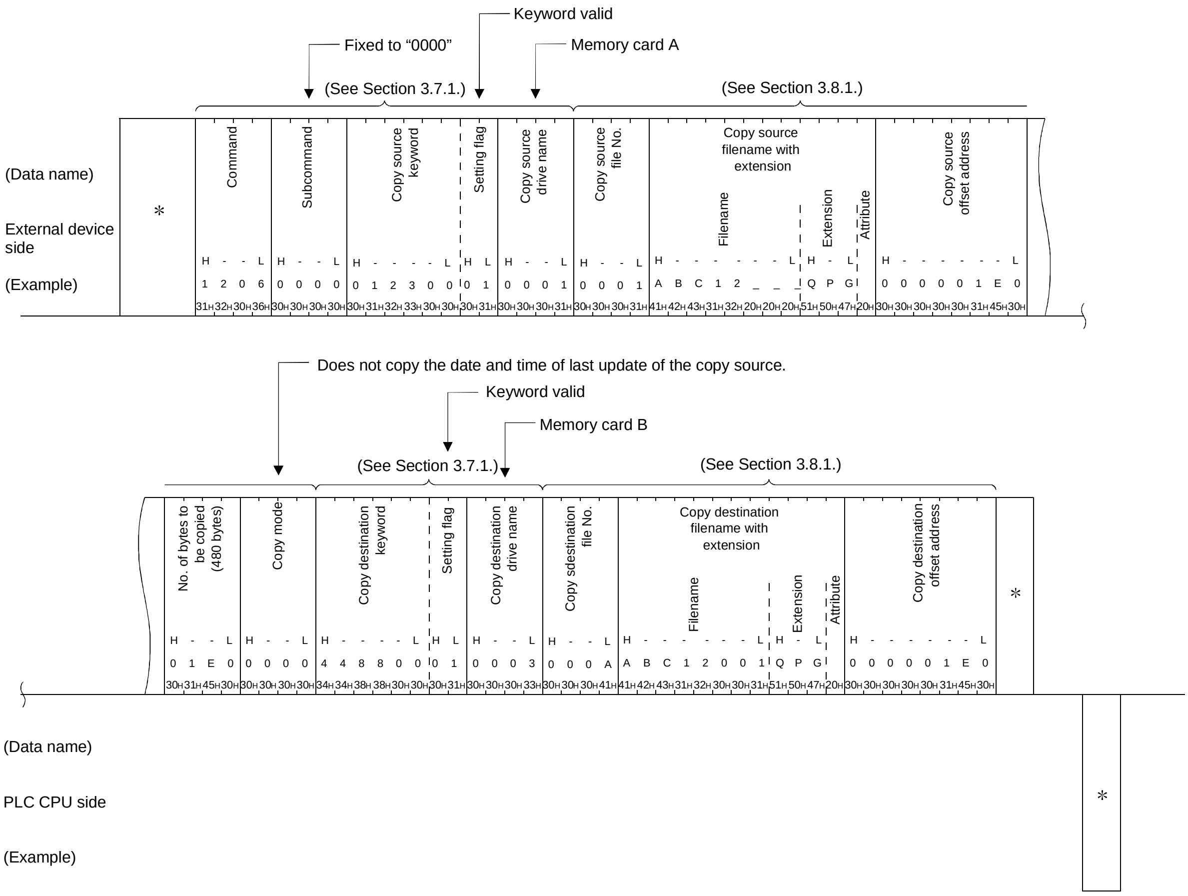
### 3.8.23 Copying files (command: 1206): for the QnACPU

The examples shown in this section explain the control procedure for writing (overwriting) the data written in an existing file to a new file.
The order and contents of data items of the areas marked by "∗" shown in the control procedure diagram differ depending on the module used as well as the frameand format used for communication.
See the detailed information described in Section 3.1.

[Control procedure]

(1) When copying the contents of an existing file in memory card A (RAM area, drive name: 01H) to a new file by communication in ASCII code

Designate offset address 1E0H for both files and copy 480 bytes of data.
Existing file File No. : 0001H (1), filename : "ABC12.QPG"
New file File No. : 000AH (10), filename : "ABC12001.QPG"
(Does not copy the date and time of last update of the copy source.)

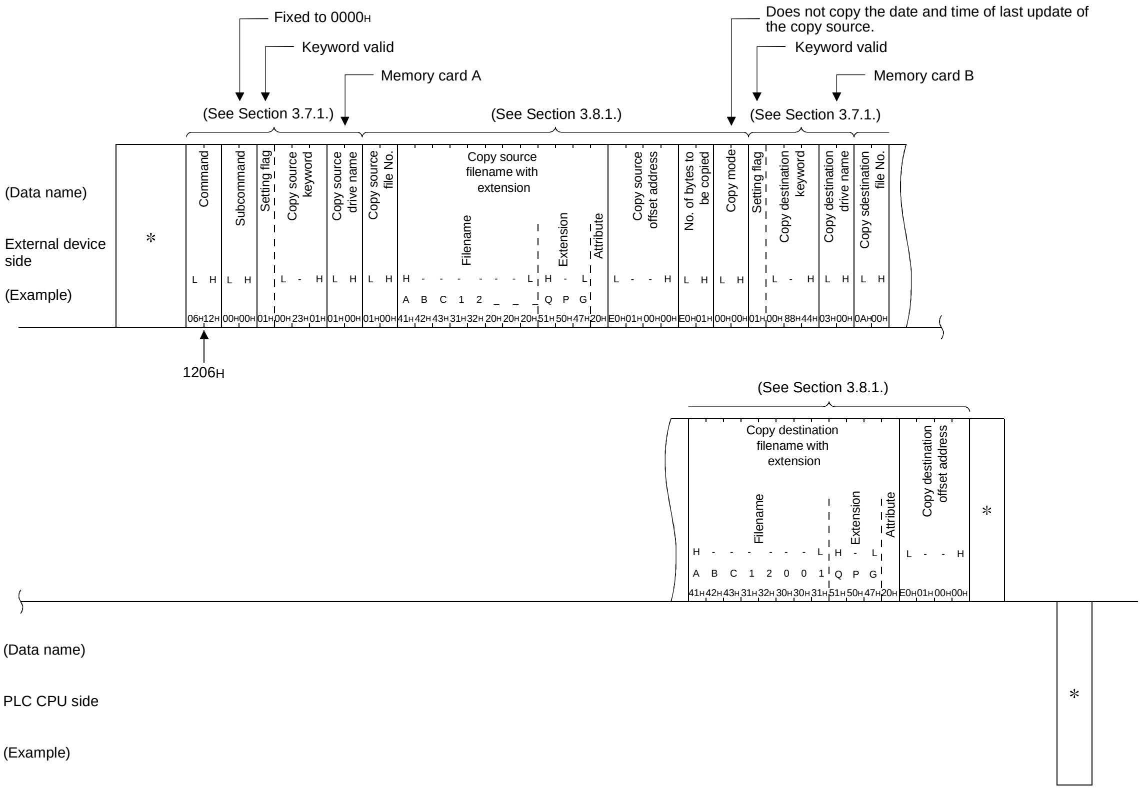


(2) When copying the contents of an existing file in memory card A (RAM area, drive name: 01H) to a new file by communication in binary code

Designate offset address 1E0H for both files and copy 480 bytes of data.

Existing file File No. : 0001H (1), filename : "ABC12.QPG"

New file File No. : 000AH (10), filename : "ABC12001.QPG"




**POINT**

(1) The maximum number of bytes per copy operation when copying is fixed.
Write all the data written to the existing file to the newly registered file by dividing it into several parts by adjusting the offset address and number of bytes written.

(2) The attribute designated during file creation, etc. is valid as the attribute of the source and destination files.
When copying a file, treat the attribute as dummy data.

(3) When the following files are designated while the QnACPU is in operation, an error occurs and an abnormal completion code will be returned.
- Parameter file
- Currently executing file in the built-in RAM (drive name: 00H)

(4) Designate the relevant values within the following range.
- Offset address : $0 \leq$ address $\leq$ (file size - 1)
When writing data to a file in the drive named "00H" (built-in RAM), designate the address in multiples of 4 (0, 4, 8,... in decimal notation) within the range above.
When writing data to a file in a drive other than "00H", designate the addresses in even-numbered addresses (0, 2, 4, 6, 8,... in decimal notation) within the range above.
- Number of copy bytes : $0 \leq$ Number of bytes $\leq 480$

### 3.8.24 Changing the file information (command: 1204): for the QnACPU

(1) Changing the file creation date (command: 1204, subcommand: 0000)

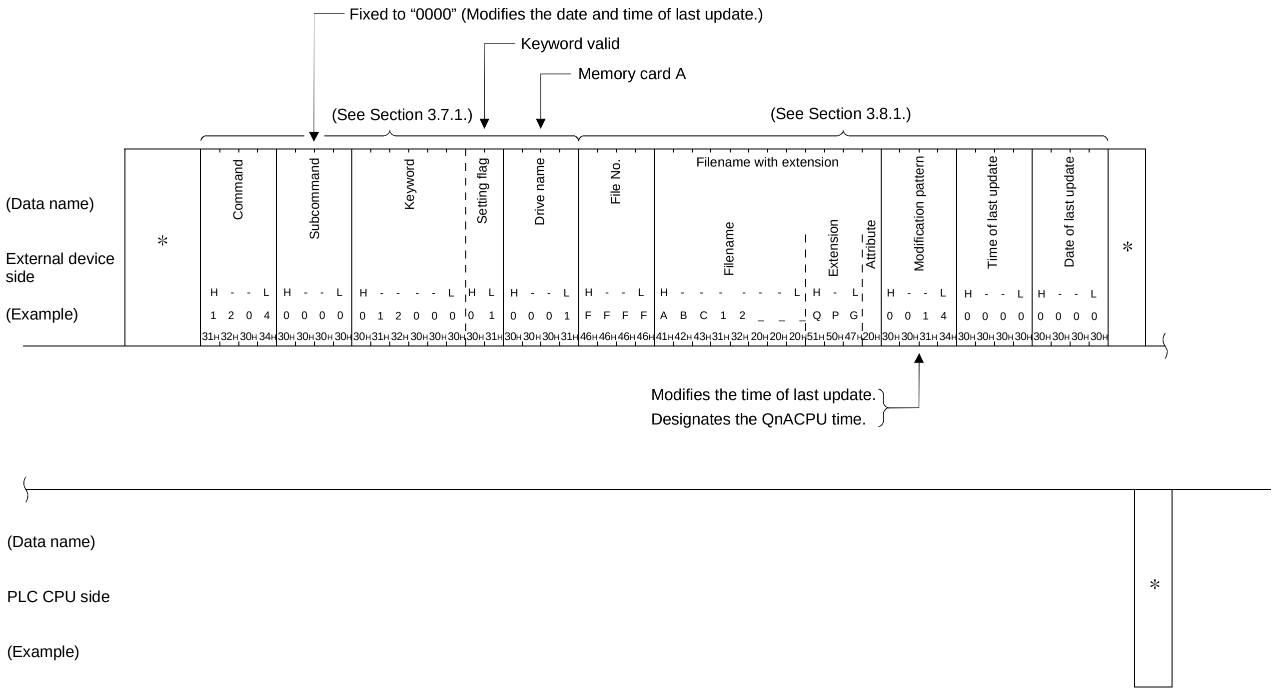
The examples shown in this section explain the control procedure for changing the date and time when the designated file was last updated.

The order and contents of data items of the areas marked by "∗" shown in the control procedure diagram differ depending on the module used as well as the frame and format used for communication.

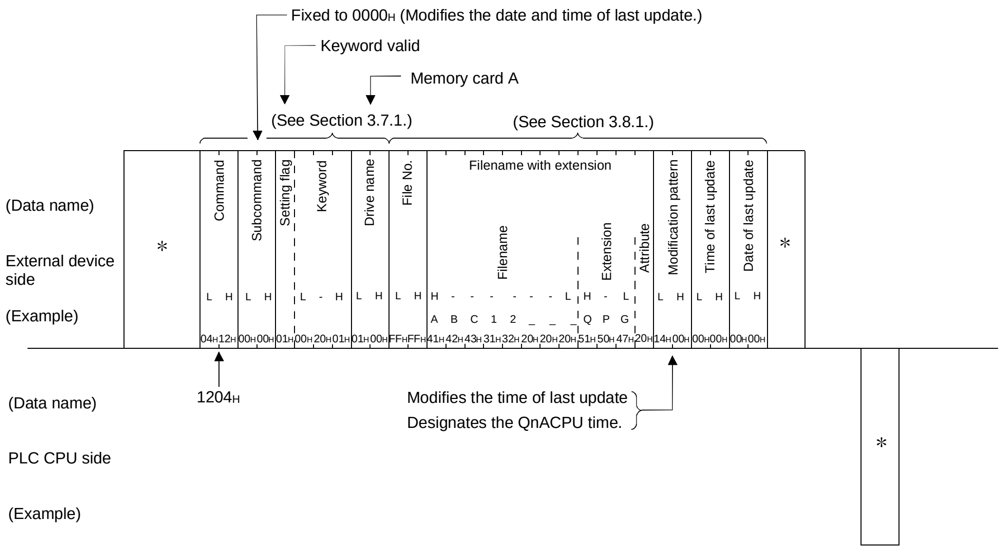
See the detailed information described in Section 3.1.

[Control procedures]

(a) When changing the time when memory card A (RAM area, drive name: 01H) filename "ABC12.QPG" was last updated by communication in ASCII code (The file number is unknown and the time of last update is the QnACPU time.)



(b) When changing the time when memory card A (RAM area, drive name: 01H) filename "ABC12.QPG" was last updated by communication in binary code (The file No. is unknown and the time of last update is the QnACPU time.)




**POINT**

(1) Designate or return the designated values within the following range.

- File number : 1 ≤ file number ≤ 256
- Modification pattern : 04H, 08H, 0CH, 14H, 18H or 1CH

(2) When the following files are designated while the QnACPU is in operation, an error occurs and an abnormal completion code will be returned.

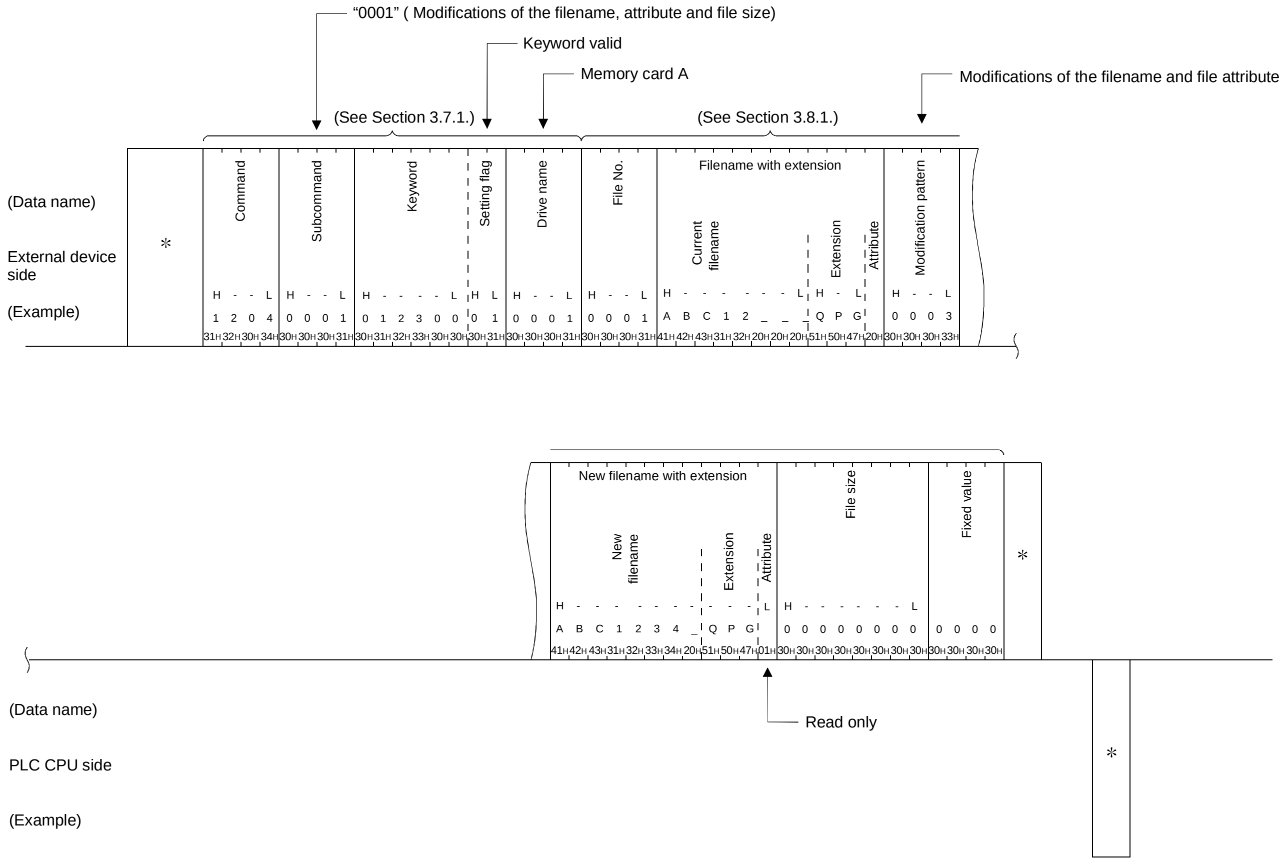
- Parameter file
- Currently executing file in the built-in RAM (drive name: 00H)

### (2) Changing the filename, attribute, and file size (command: 1204, subcommand: 0001)

The examples shown in this section explain the control procedure for changing the filename, attribute, and file size of the designated file.
The order and contents of data items of the areas marked by "*" shown in the control procedure diagram differ depending on the module used as well as the frame and format used for communication.
See the detailed information described in Section 3.1.

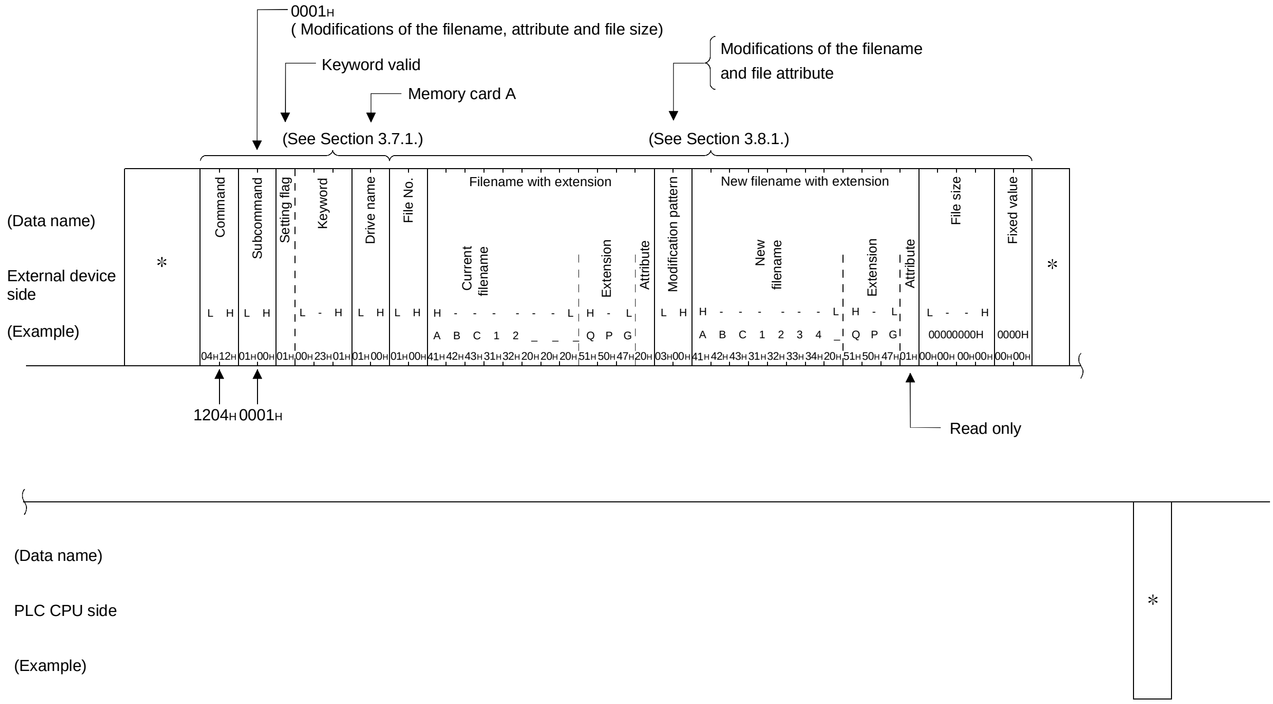
[Control procedure]

(a) When changing the filename and attribute of memory card A (RAM area, device name: 01H) file No.1 with filename "ABC12.QPG" by communication in ASCII code
(Changes the filename to "ABC1234.QPG" and the attribute to read only.)



(b) When changing the filename and attribute of memory card A (RAM area, device name: 01H) file No.1 with filename "ABC12.QPG" by communication in binary code

(Changes the filename to "ABC1234.QPG" and the attribute to read only.)




POINT

(1) Designate or return the designated values within the following range.
- File number : 1 ≤ file number ≤ 256
- Modification pattern : 01H, 02H, 03H, 20H, 21H 22H or 23H

(2) When the following files are designated while the QnACPU is in operation, an error occurs and an abnormal completion code will be returned.
- Parameter file
- Currently executing file in the built-in RAM (drive name: 00H)

(3) The attribute can only be modified between 01H (read only file) ↔ 20H (read/write enabled file)

(4) The size can be modified only while the QnACPU is in the STOP status. A contiguous free area of the designated size is necessary on the designated drive.
The vacant area can be checked using the memory usage status read function described in Section 3.7.2.

### (3) Batch modification of file information (command: 1204, subcommand: 0002)

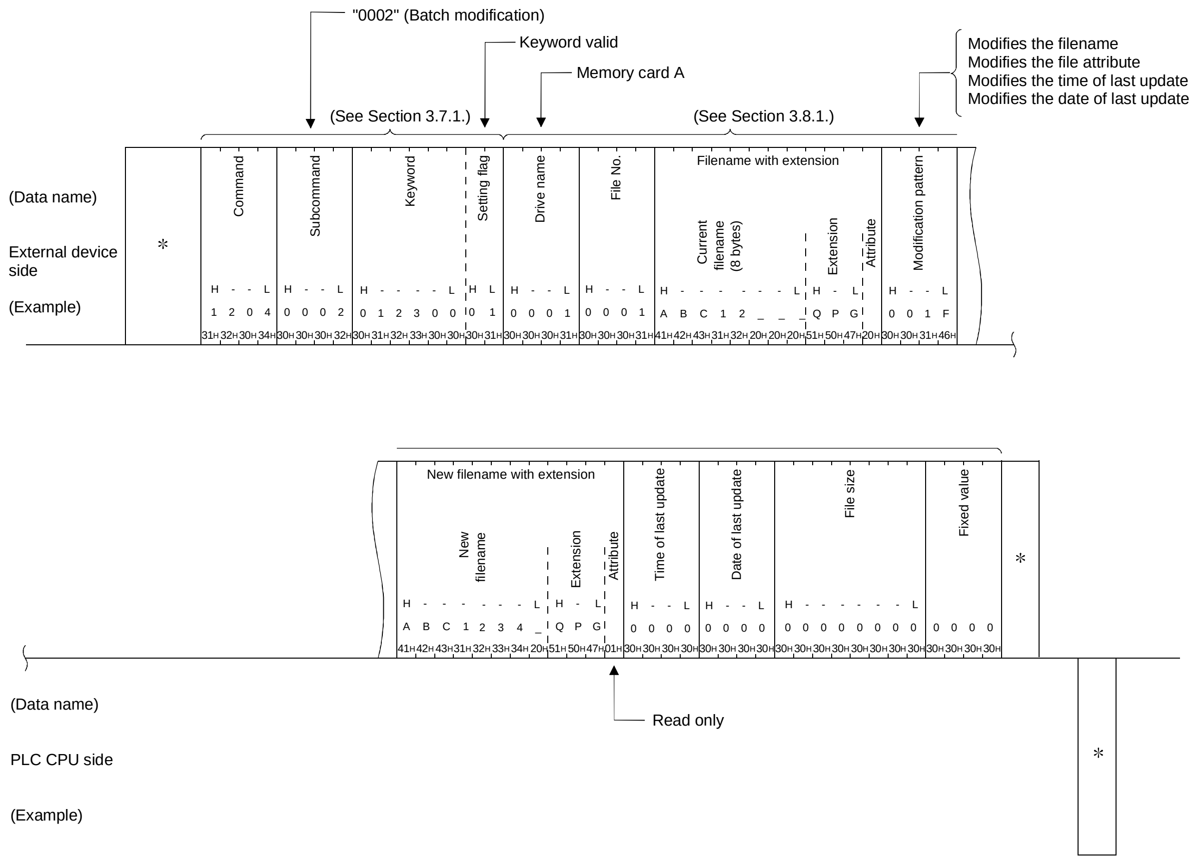
The examples shown in this section explain the control procedure for changing file information in batches.

The order and content of data items of the areas marked by "✻" shown in the control procedure diagram differ depending on the module used as well as the frame and format used for communication.

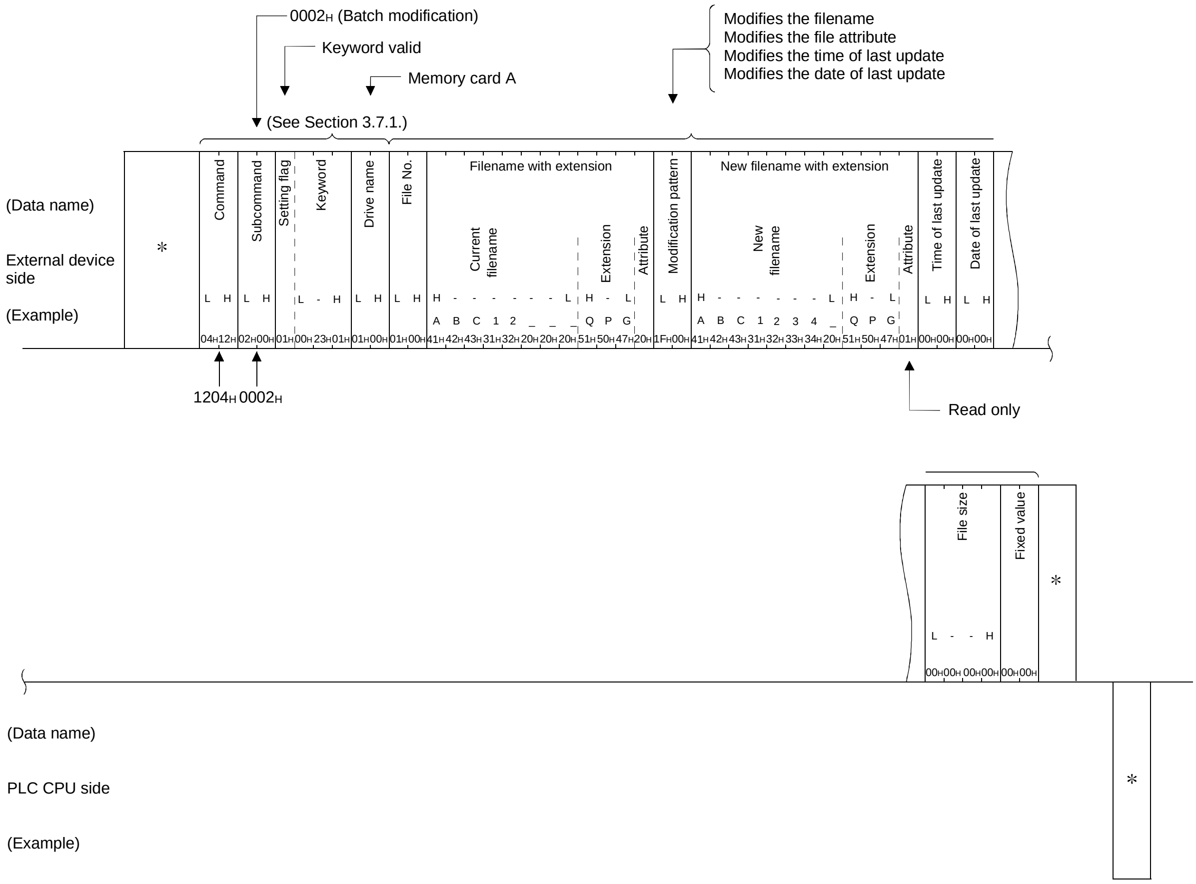
See the detailed information described in Section 3.1.

[Control procedure]

(a) When changing the filename, attribute, and date and time of last update of memory card A (RAM area, drive name: 01H) file number 1/filename "ABC12.QPG" by communication in ASCII code.
(Use the date and time of the QnACPU for the date and time of last updating. The file attribute is changed to read only.)



(b) When changing the filename, attribute, and date and time of last update of memory card A (RAM area, drive name: 01H) file number 1/filename "ABC12.QPG" by communication in binary code.
(Use the date and time of the QnACPU for the date and time of last updating. The file attribute is changed to read only.)




**POINT**

(1) Designate or return the designated values within the following range.
- File number : 1 ≤ file number ≤ 256
- Modification pattern : 01H ≤ modification pattern ≤ 3FH

(2) When the following files are designated while the QnACPU is in operation, an error occurs and an abnormal completion code will be returned.
- Parameter file
- Currently executing file in the built-in RAM (drive name: 00H)

(3) The attribute can only be modified between 01H (read only file) ↔ 20H (read/write enabled file)

(4) The size can be modified only while the QnACPU is in the STOP status. A contiguous free area of the designated size is necessary on the designated drive.
The vacant area can be checked using the memory usage status read function described in Section 3.7.2.

## 3.9 Registering, Deleting and Reading User Frames: for Serial Communication Modules

User frames are used to communicate data by making it possible for the user to select the data format for the beginning and end of the messages exchanged between and external device and QCPU.

- Transmission of on-demand data using the MC protocol
- Data transmission and reception using the non procedure protocol

This section explains how to issue the commands to register, delete, and read user frames to the Q series C24 from an external device .
For transmission and reception of data in a user frame, see Chapters 10 and 11 of the User's Manual (Application) of the Q series C24.

**POINT**

(1) See Chapter 9 of the User's Manual (Application) of the Q series C24 for a description of the user frame.

(2) This function can only be used with the Q series C24 connected to an external device (including multidrop link stations).
It cannot be used with another station's Q series C24 connected via a network system.

(3) When an external device issues a register, delete, or read request, the data communication described in this section is performed without waiting for the PLC CPU END processing.

### 3.9.1 Commands and contents of character area

The following explains the commands and the contents of the character area (data section for communication in binary code) in the control procedure when an external device registers a user frame, etc. to the Q series C24.

(1) Commands

| Function | | Command (subcommand) | Processing | Number of points processed per communication | PLC CPU status | | | Reference section |
|---|---|---|---|---|---|---|---|---|
| | | | | | During STOP | During RUN | | |
| | | | | | | Write allow setting | Write prohibit setting | |
| User frame | Data registration | 1610 (0000) | Registers the user frame (data order). | 80 bytes | ○ | ○ | ○ | Section 3.9.2 |
| | Delete Registered data | 1610 (0001) | Deletes the user frame of the designated frame number. | (for 1) | | | | |
| | Read registered data | 0610 (0000) | Reads the registered frame of the designated frame number. | 80 bytes | | | | Section 3.9.3 |

○ in the PCL CPU status field in the table above indicates that execution is possible

(2) Contents of the character area

The following explains the contents of the character area when an external device registers, deletes, or reads a user frame from the Q series C24.

(a) Frame number

This value specifies the user frame number to be registered, deleted, or read.

1) Data communication in ASCII code

The value shown below is converted to a 4-digit (hexadecimal) ASCII code and sequentially transmitted beginning from the most significant digit.

(Example)

The user frame number is $3E8_H$ (frame registered by the user), it is converted to "03E8" and sequentially transmitted beginning from "0".

2) Data communication in binary code

The 2-byte value shown below is sequentially transmitted beginning from the low byte (L: bits 0 to 7).

3) The frame number designated value and possible designation contents are shown below.

Values other than these cannot be designated.

| Designated value | Designated contents | Registration destination |
|---|---|---|
| $1_H$ to $3E7_H$ | Default registration frame | ROM for the Q series 24 OS (read only) |
| $3E8_H$ to $4AF_H$ | User frame | Flash ROM of the Q series C24 (can be read, written, and deleted) |
| $8001_H$ to $801F_H$ | User frame | The buffer memory of the Q series C24 (addresses $1B00_H$ to $1FF6_H$)<br>(Can be read, written, and deleted) |

(b) Registration data byte count

This value specifies the number of contiguous bytes of the registration data. When the user frame contains variable data (control data for replacing part of the user frame with a sum check code, etc.), this value is different from the frame byte count described below.

Refer to the User's Manual (Application) for the data order and byte count during registration as well as the frame byte count during communication.

1) Data communication in ASCII code

The value $0_H$ (the value when designated during deletion) or $1_H$ to $50_H$ (1 to 80) is converted to a 4-digit (hexadecimal) ASCII code and sequentially transmitted beginning from the most significant digit.

(Example)

The byte count is 10

The value 10 is converted to "000A" and sequentially transmitted from the first "0."

2) Data communication in binary code

A 2-byte value indicating the byte count $0_H$ (the value when designated during deletion) or $1_H$ to $50_H$ (1 to 80) is sequentially transmitted beginning from the low byte (L: bits 0 to 7).

(c) Frame byte count

This value designates the number of bytes in the frame to be registered or read.

The variable data part is calculated with the 2-bytes of $FF_H$ + $\square_H$ as one byte.

1) Data communication in ASCII code

The value $0_H$ (the value when designated during deletion) or $1_H$ to $50_H$ (1 to 80) is converted to a 4-digit (hexadecimal) ASCII code and sequentially transmitted beginning from the most significant digit.

2) Data communication in binary code

A 2-byte value indicating byte count $0_H$ (the value when designated during deletion) or $1_H$ to $50_H$ (1 to 80) is sequentially transmitted beginning from the low byte (L: bits 0 to 7).

(d) Registration data

This value designates the data order of the frame to be registered to the Q series C24. It must be a continuous set of data (maximum of 80 bytes) corresponding to the read/write byte counts described in item (b) above. When deleting a registered user frame, it is not necessary to designate the registration data.

1) Data communication in ASCII code

The data codes making up the frame are converted to 2-digit (hexadecimal) ASCII codes and transmitted beginning from the most significant digit.

(Example)

Designating a frame to transmit/receive ENQ + module station number + blank (space)

This code is converted to "05FF0120" and sequentially transmitted from the first "0" (head data).

2) Data communication in binary mode

The data codes making up the frame are sequentially transmitted beginning from the head data.

(Example)

Designating a frame to transmit/receive END + module station number + blank (space)

The registration data becomes $05_H$, $FF_H$, $01_H$ and $20_H$ and is sequentially transmitted from $05_H$.

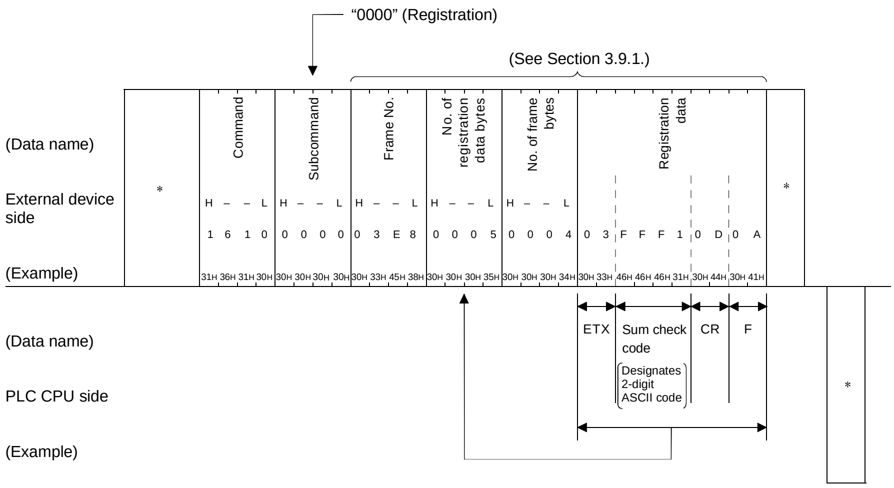
### 3.9.2 Registering and deleting user frames (command: 1610)

The examples shown in this section explain the control procedure for registering user frames to the Q series C24.
The order and content of data items of the areas marked by "∗" shown in the control procedure diagram differ depending on the module used as well as the frame and format used for communication.
See the detailed information described in Section 3.1.

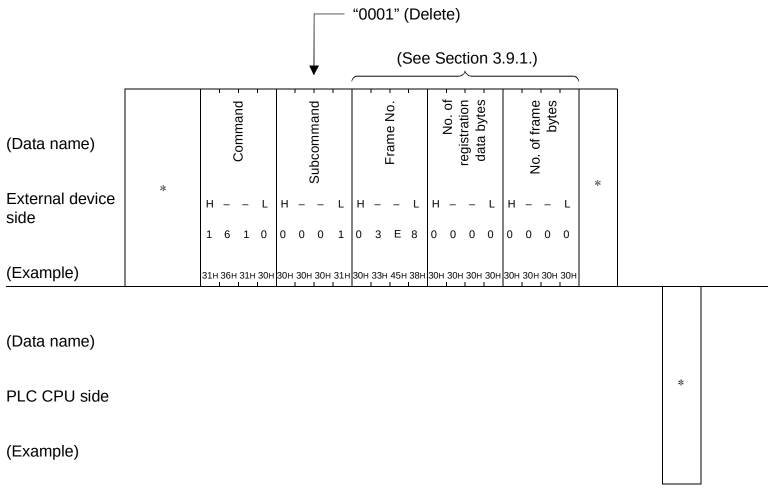
[Control procedure]

(1) Registering and deleting by communication in ASCII code

(a) Registering a frame with frame number 3E8H for transmitting and receiving ETX + sum check code + CR + LF (codes and order after registration: 03H, FFH, F1H, 0DH, 0AH)
In this example, the lower byte (8 bits) of the binary code obtained by summing the data section except the head frame is converted into 2-digit ASCII code (hexadecimal), and this character string is designated as the sum check code.



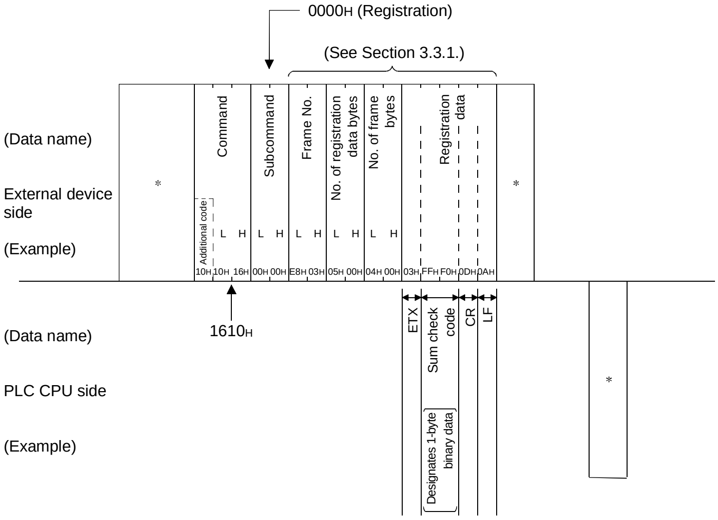

(b) Deleting a user frame already registered as number 3E8H



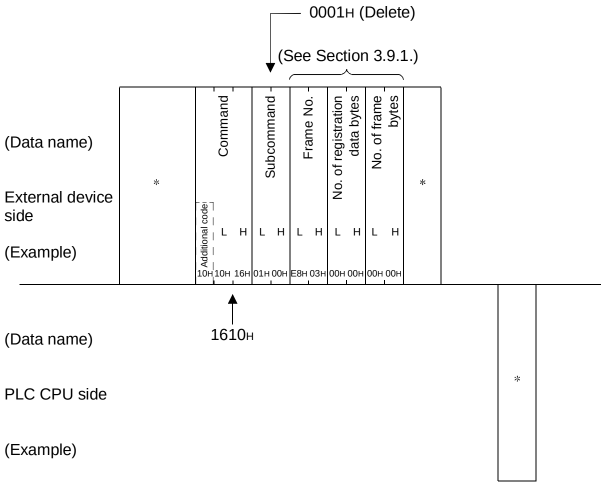
(2) Registering and deleting by communication in binary code

(a) Registering a frame with frame number 3E8H for transmitting and receiving ETX + sum check code + CR + LF (codes and order after registration: 03H, FFH, F0H, 0DH, 0AH)

In this example, the lower byte (8 bits) of the binary code obtained by summing the data section except the head frame is designated as the sum check code as is.




(b) Deleting a user frame already registered as number 3E8H




**POINT**

(1) When registering a user frame with the same frame number again, it should be deleted first and then registered again.
If it is attempted to register a user frame by designating an already registered frame number, an error occurs and a NAK message is returned.

(2) If it is attempted to delete a user frame by designating a frame number that has not yet been registered, an error occurs and a NAK message is returned.

(3) Frame numbers should be designated within the following range.

- When registering to the Flash ROM of the Q series C24 : 3E8H ≤ frame number ≤ 4AFH
- When registering to the buffer memory of the Q series C24 : 8001H ≤ frame number ≤ 801FH

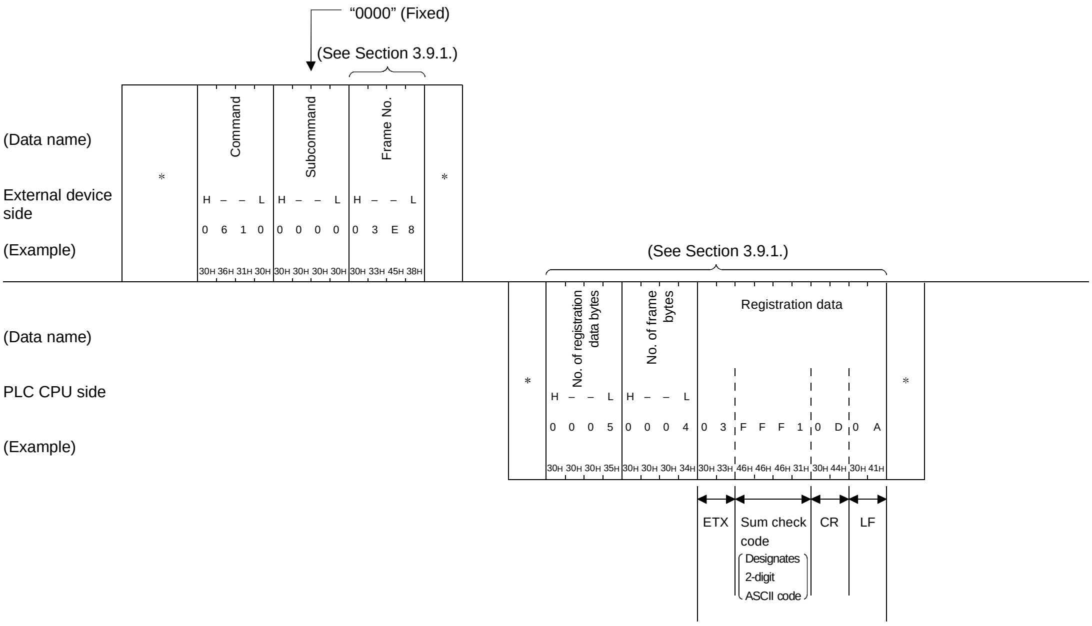
### 3.9.3 Reading user frames (command: 0610)

The examples shown in this section explain the control procedure for reading the registered contents (order of registered data) of user frames from the Q series C24.
The order and content of data items of the areas marked by "*" shown in the control procedure diagram differ depending on the module used as well as the frame and format used for communication.
See the detailed information described in Section 3.1.

[Control procedure]

(1) Reading the contents registered in frame number 3E8H by communication in ASCII code

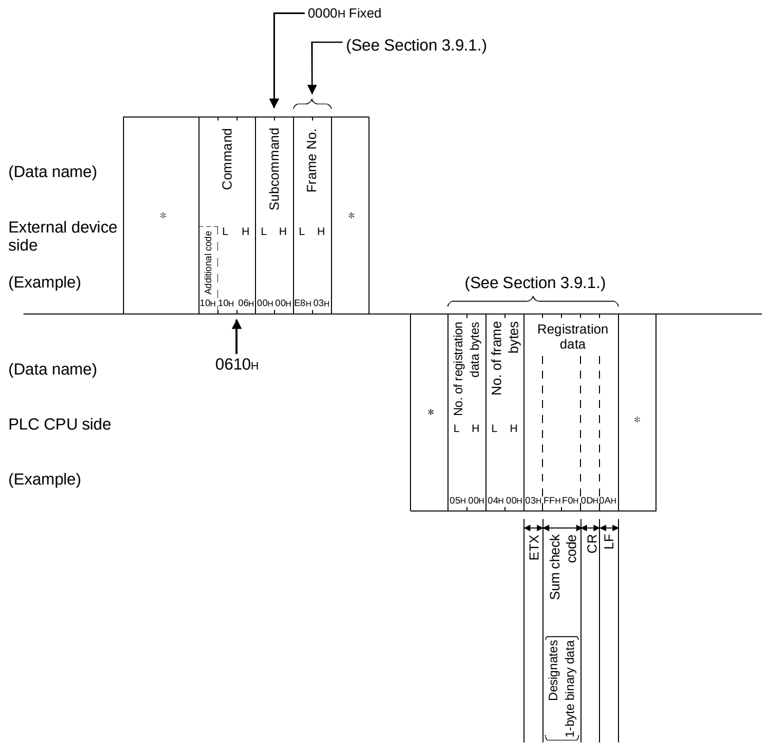
The contents registered are frames for transmitting and receiving ETX + sum check code + CR + LF.
In this example, the lower byte (8 bits) of the binary code obtained by summing the data in the fields except the head frame is converted into 2-digit ASCII code (hexadecimal), and this character string is designated as the sum check code.



(2) Reading the contents registered in frame number 3E8H by communication in binary code

The contents registered are frames for transmitting and receiving ETX + sum check code + CR + LF.

In this example, the lower byte (8 bits) of the binary code obtained by summing the data section except the head frame is designated as the sum check code as is.




**POINT**

(1) If it is attempted to designate a frame number for which no user frame has been registered, an error occurs and a NAK message is returned.

(2) Frame numbers should be designated within the following ranges.

- Default registration frame registered in the ROM of the OS of the Q series C24:
  1H ≤ frame No. ≤ 3E7H
- User frame registered in the Flash ROM of the Q series C24:
  3E8H ≤ frame No. ≤ 4AFH
- User frame registered in the buffer memory of the Q series C24:
  8001H ≤ frame No. ≤ 801FH

## 3.10 Global Function: for Serial Communication Modules

The global function turns global signals (X1A/X1B) of the PLC CPU of the Q series C24 loaded station connected to an external device over a multidrop link on/off.
This function is used to set for emergency instructions, simultaneous startup, and data communication enable/disable interlock signals issued to the PLC CPU.
The following examples describe the control procedure for using the global function.

**POINT**

(1) When an external device issues the GW command by an A compatible 1C frame to a computer link module on a station connected by multidrop link, the global signal (input signal) X1A or X1B is turned on/off on that interface on the serial communication module of the Q/QnACPU station.
(Example)
When the GW command was received from the Q series C24 CH1 interface, the Q series C24 turns X1A on/off.

(2) When the global function is issued to an ACPU + C24 station connected over a multidrop link, the ACPU X2 signal is turned on/off.

(3) When the PLC CPU is restarted after the global signal is turned either on or off, the global signal is turned off.

(4) The function can only be used with the Q/QnACPU + Q series C24 stations (including stations connected by multidrop) connected to an external device. Other ACPU + Q series 24 stations connected over a network system cannot use this function.

### 3.10.1 Commands and contents of the character area

The following explains the commands and contents of the character area (data section during communication in binary code) in the control procedure when an external device turns on/off the global signals of the Q series C24.

(1) Commands

| Function | — | Command (subcommand) | Processing | PLC CPU status: During STOP | PLC CPU status: During RUN: Write allow setting | PLC CPU status: During RUN: Write prohibit setting | Reference section |
|---|---|---|---|---|---|---|---|
| Global | Global signal OFF | 1618 (0000) | Turns off the global signal. | ○ | ○ | ○ | Section 3.10.2 |
| | Global signal ON | 1618 (0001) | Turns on the global signal. | | | | |

○ in the PLC CPU status field in the table above indicates that execution is possible.

## (2) Contents of the character area

This section explains data in the character area that designates the global signals when an external device turns the global signal on/off.

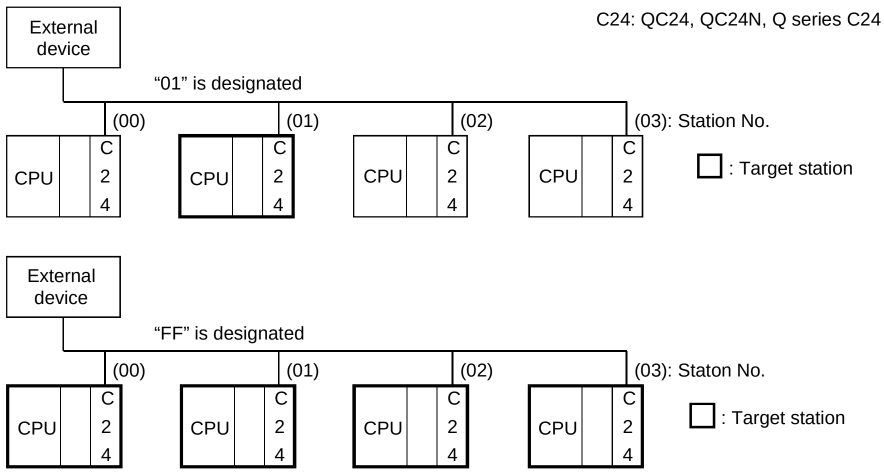
### (a) Station number

This value designates the station number whose global signal is instructed by the opposite station to be turned on/off.

Designate one of the following.

| Designated value | | Processing by global function |
|---|---|---|
| In ASCII code | In binary code | |
| "00" to "1F" | $00_H$ to $1F_H$ | Turns on/off the global signal of only the designated station |
| "FF" | $FF_H$ | Turns on/off the global signal of all stations connected by a multidrop link |

(Example)




### (b) Global signal designation

This value is for designating which global signal on the Q/QnACPU to turn on/off.

1) Data communication in ASCII code
   The value shown below is converted to a 4-digit (hexadecimal) ASCII code and sequentially transmitted beginning from the most significant byte "0."
2) Data communication in binary code
   The 2-byte value shown below is transmitted from the low byte (L: bits 0 to 7).
3) The designated values and the corresponding processing are summarized below; other values cannot be designated.

| Designated value | Designated contents |
|---|---|
| $0000_H$ | Turns on/off the global signal of the interface that received the global function command<br>• When the command was received through the CH1 interface, X1A is turned on/off<br>• When the command was received through the CH2 interface, X1B is turned on/off |
| $0001_H$ | Turns X1A on/off regardless of the interface that received the global function command. |
| $0002_H$ | Turns X1B on/off regardless of the interface that received the global function command. |

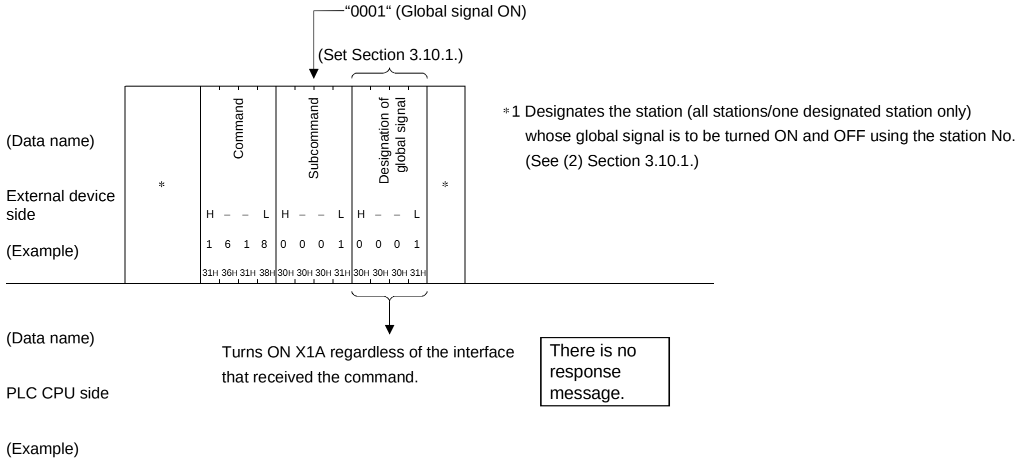
## 3.10.2 Control procedure of the global function (command: 1618)

The examples shown in this section explain the control procedure for turning on/off the global signals to the PLC CPU from an external device.
The order and contents of data items of the areas marked by "*" shown in the control procedure diagram differ depending on the module used as well as the frame and format used for communication.
See the detailed information described in Section 3.1.

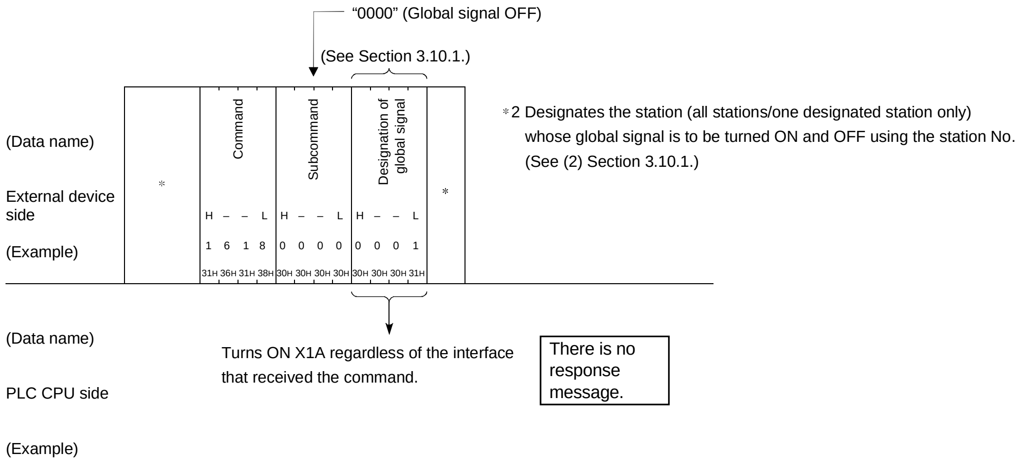
[Control procedure]

### (1) Turning on/off the global signals using ASCII code format 1.

(a) Turning on the global signal X1A of all stations in a multidrop link.

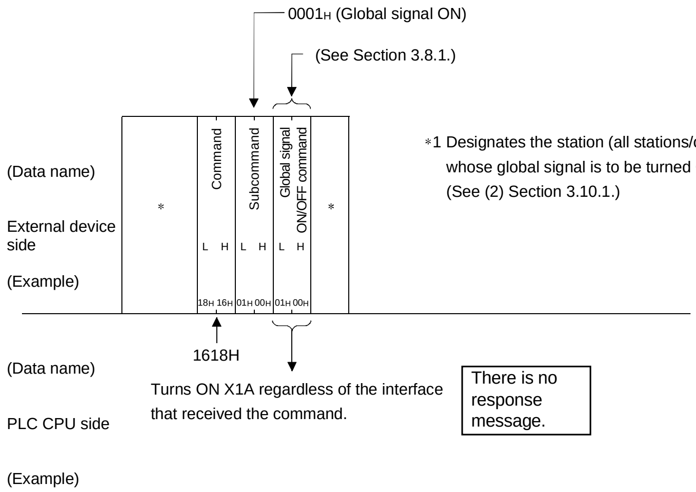



(b) Turning off the global signal X1A of all stations in a multidrop link.



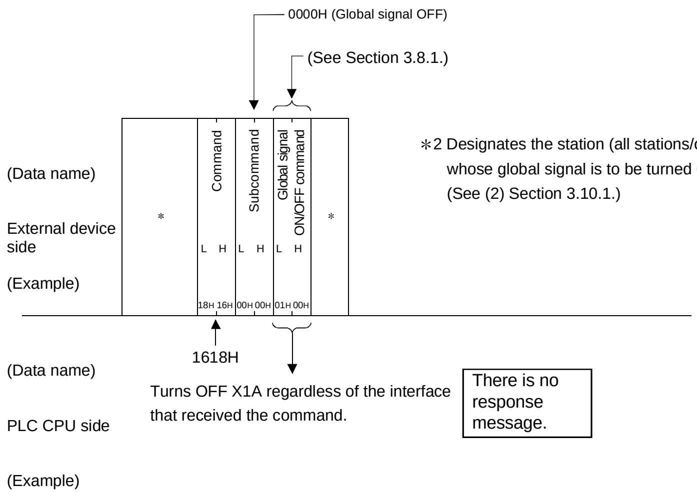
(2) Turning on/off the global signals using binary code format 5.

(a) Turning on the global signal X1A of all stations in a multidrop link.




*1 Designates the station (all stations/one designated station only) whose global signal is to be turned ON and OFF using the station No. (See (2) Section 3.10.1.)

(b) Turning off the global signal X1A of all stations in a multidrop link.

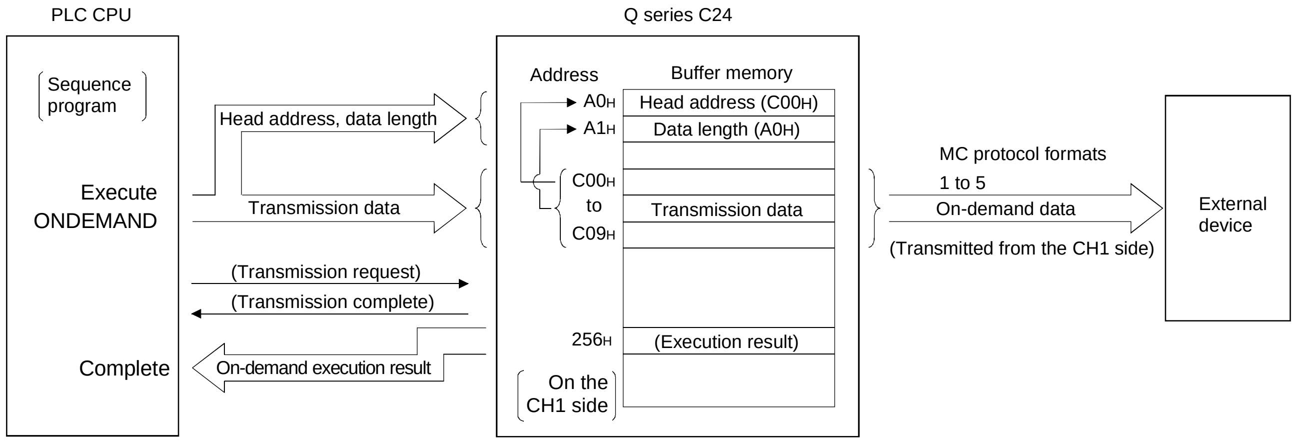



*2 Designates the station (all stations/one designated station only) whose global signal is to be turned ON and OFF using the station No. (See (2) Section 3.10.1.)

## 3.11 Data Transmission to an External device (On-Demand Function): for Serial Communication Modules

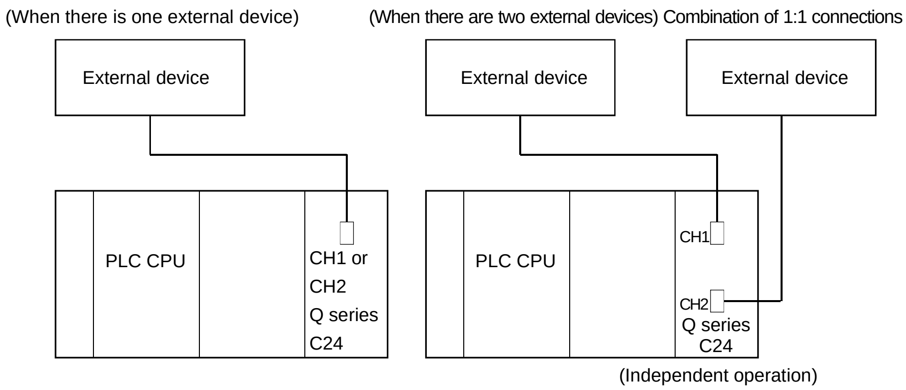
Functions that request data transmission from the PLC CPU to an external device when communicating data in the MC protocol, are referred to as on-demand function.
Data communication using the MC protocol is normally performed based on the instructions from the external devices.
When the PLC CPU has emergency data that requires transmission to an external device, the on-demand function is used to transmit that data as on-demand data to the external device, starting the transmission from the PLC CPU.

(When transmitting from an interface on CH1 side)




**POINT**

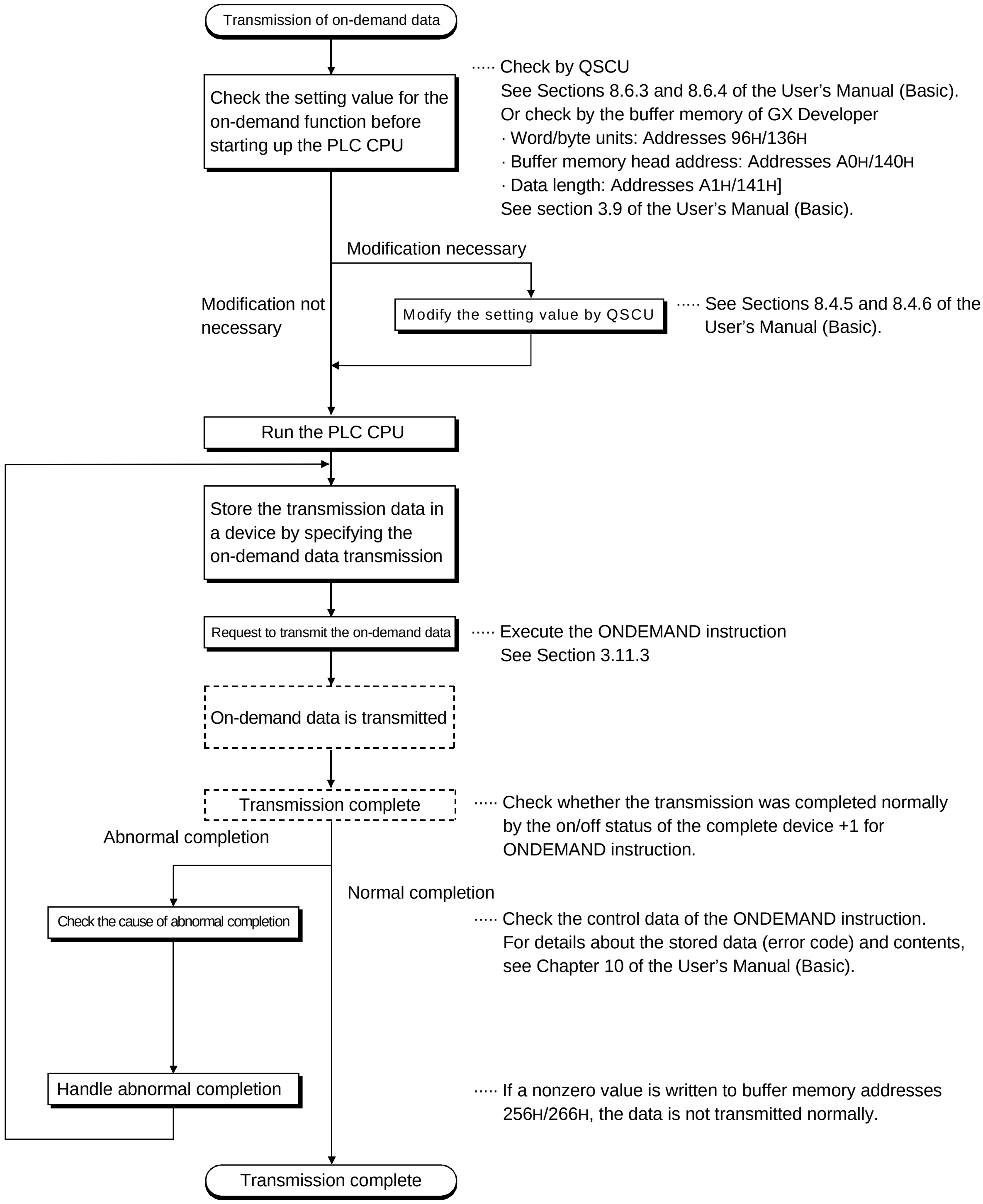
The on-demand function can be used when the external device/PLC CPU system configuration is 1:1.



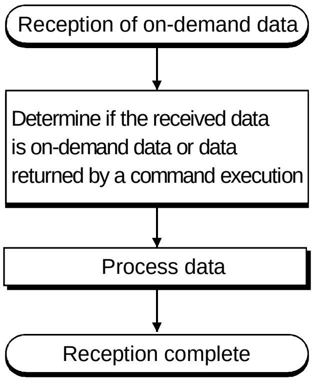

Do not use the on-demand function when the system configuration is not 1:1.
If the on-demand function is used when the external device and PLC CPU are connected by a 1:n or m:n multidrop link, the communication data of control procedure formats 1 to 5 and the on-demand data will be destroyed and data may not be transmitted correctly.

### 3.11.1 Execution procedure of the on-demand function

(1) Execution procedure on the PLC CPU side



(2) Execution procedure on the external device side



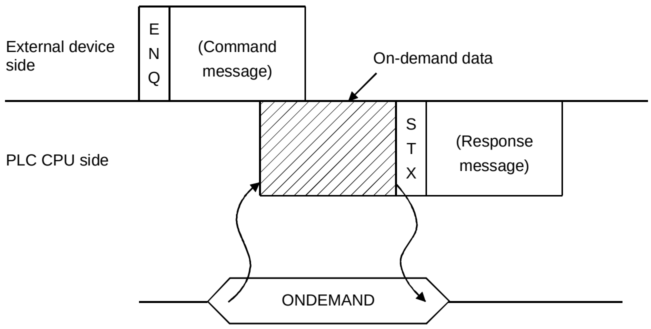

····· When on-demand data is transmitted, the Q series C24 sends “FE”/FEH as the PC number.
· “FE” is transmitted in ASCII code
· FEH is transmitted in binary code
The external device treats the data received as on-demand data only when the PC number is “FE”/FEH.

(3) The following describes the timing charts when an on-demand transmission request was issued.

(a) In case of full-duplex communication

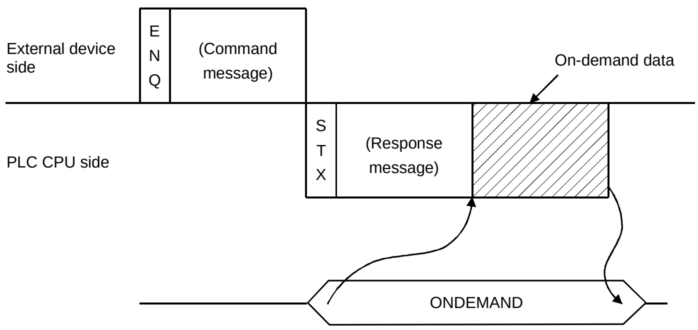
When the Q series C24 is receiving data:

The Q series C24 does not transmit any on-demand data is completed before transmitting a response message (STX to) in reply to the command message (ENQ to).




When the Q series C24 is transmitting data:

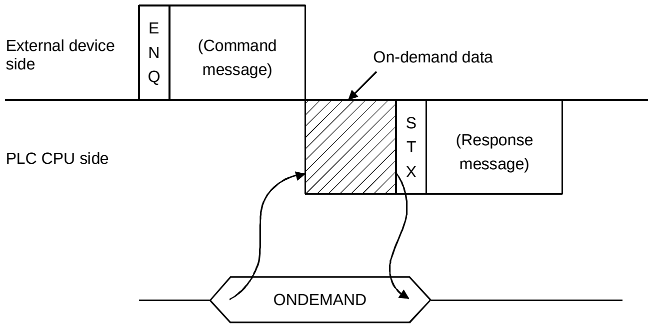
The Q series C24 does not transmit any on-demand data until the transmission of a response message (STX to) in reply to a command message (ENQ to) from the external device is completed.



(b) In case of half-duplex communication: See Chapter 8 of the User's Manual (Application).

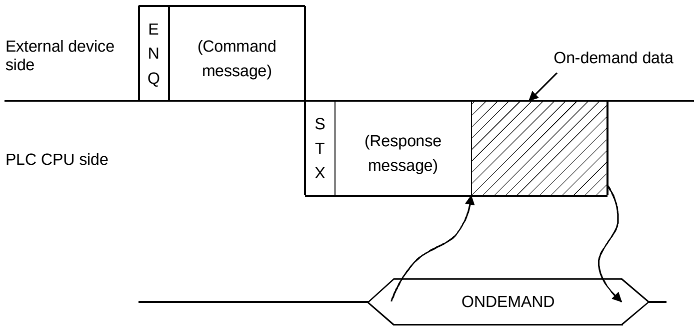
When the Q series C24 is receiving data:

The Q series C24 does not transmit any on-demand data until the reception of a command message (ENQ to) from the external device is completed.




When the Q series C24 is transmitting data:

The Q series C24 does not transmit any on-demand data until the transmission of a response message (STX to) in reply to a command message (ENQ to) from the external device is completed.




REMARK

While transmitting the on-demand data and response data, the timeout of each is checked by the transmission monitoring time (timer 2), which is described in Chapter 6 of the User's Manual (Application).
If a timeout error was generated, reset the transmission monitoring time so that the transmission is normally completed within the set time.

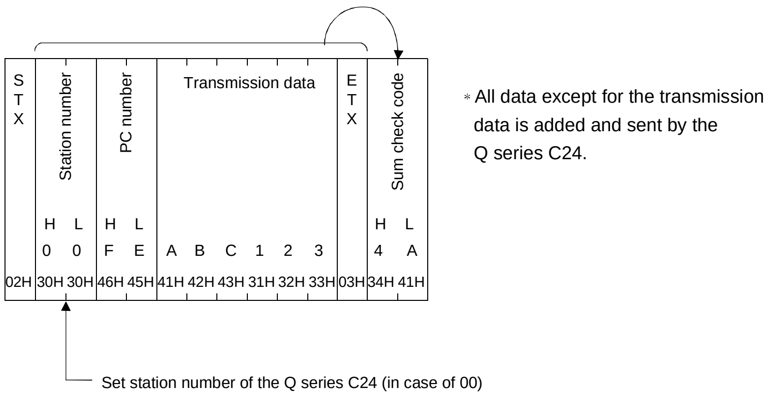
## 3.11.2 Data transmission format of the on-demand function

The format number of the communication protocol setting by the GX Developer of the Q series C24 determines the arrangement of the on-demand data transmitted by the on-demand function. The on-demand data is transmitted using the same order as the setting format for the following frames.

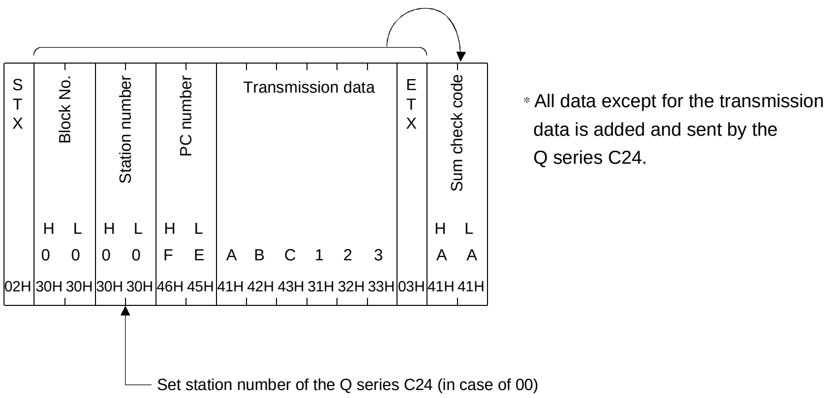
- When the setting format is "1" to "4" : The arrangement of "format 1" to "format 4" of A compatible 1C frames
- When the setting format is "5" : The arrangement of "format 5" of QnA compatible 4C frames

The following example shows the arrangement and contents of transmitted on-demand data.
Each data section included in the on-demand data, except for the transmission data, number of data bytes, and sum check code, is transmitted as data of ASCII code or binary code as shown below.
(The station number value indicates the set station number (00H to 1FH) of the Q series C24.)

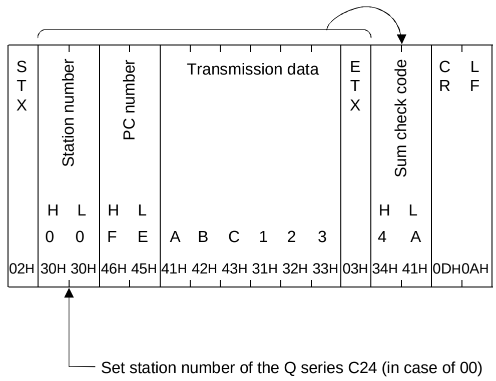
### (1) On-demand data transmission format in A compatible 1C frame formats 1 and 3




### (2) On-demand data transmission format in A compatible 1C frame format 2

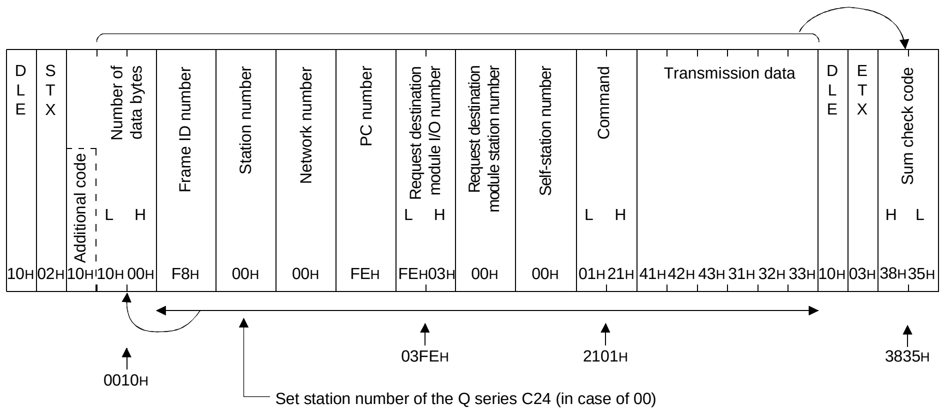


(3) On-demand data transmission format in A compatible 1C frame format 4




* All data except for the transmission data is added and sent by the Q series C24.

(4) On-demand data transmission format in QnA compatible 4C frame format 5




**POINT**

When the data is to be transmitted in the arrangement of a user-selected format of QnA compatible 3C/4C frames other than the above, use the data communication function by the user frames described in Chapters 9 and 10 of the User's Manual (Application).

### 3.11.3 Control procedure of the on-demand function (command: 2101)

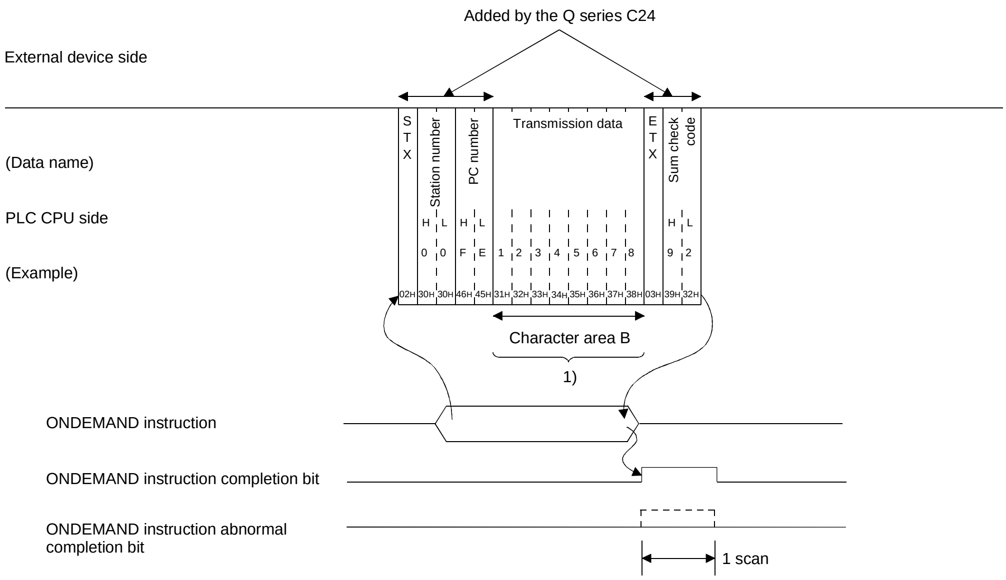
The examples in this section explain the control procedure for sending on-demand data to an external device with the on-demand function.
The arrangement and contents of the data items shown in the control procedure diagram differ depending on the frame and format used in communication.
When the communication protocol setting in GX Developer is "1 to 4," the on-demand data is transmitted in A compatible 1C frames. For the arrangement and contents of data items in each format of the A compatible 1C frame, see Section 5.1
When the communication protocol setting in GX Developer is "5," the on-demand data is transmitted in format 5 of the QnA compatible 4C frame. For the arrangement and contents of data items in each format of the A compatible 4C frame, see Section 3.1

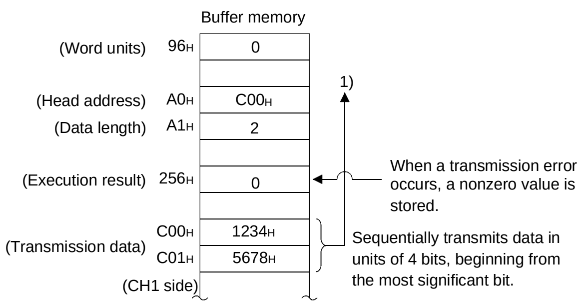
[Control procedure]

The following examples show the control procedure in format 1 of the A compatible frame and the format 5 of QnA compatible 4C frame.

(1) Format 1

(a) When the "Word/byte units designation" in GX Configurator-SC is set to "0" (word units)
The following diagram shows the control procedure when two words are designated as transmission data



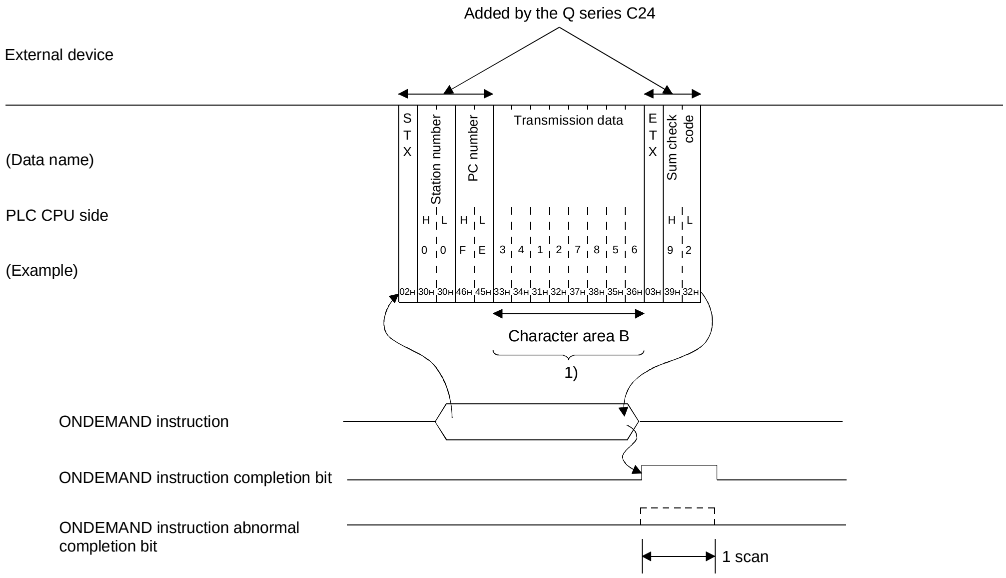



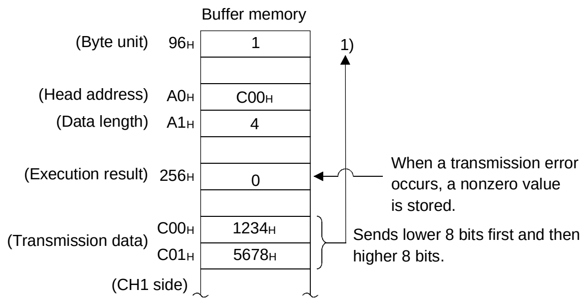
**POINT**

(1) When control procedure format 2 is used, the block number should be "00".

(2) The number of characters in the transmission data section of the on-demand data is the data length × 4.
[Four characters are used for one byte of data.
Thus, one byte of data is expressed in 4-digit (hexadecimal).]

(b) When the "Word/byte units designation" in GX Configurator-SC is set to "1" (byte units)
The following diagram shows the control procedure when two words (four bytes) are designated as transmission data




**POINT**

(1) When control procedure format 2 is used, the block number should be "00."

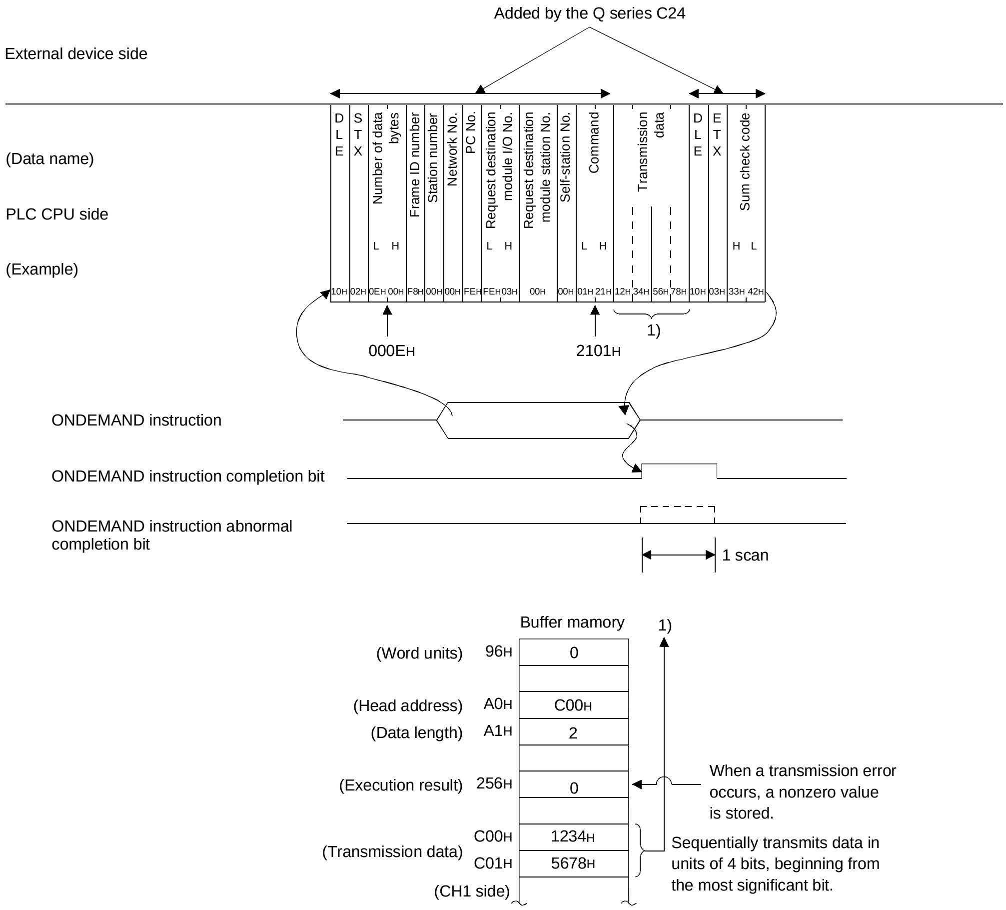
(2) The number of characters in the transmission data section of the on-demand data is the data length × 2.
[Two characters are used for one byte of data.
Thus, one byte of data is expressed in 2-digit (hexadecimal).]

(3) When the data length is an odd number, the data of the lower byte (bits 0 to 7) of the last data storage device is transmitted.

(2) Format 5

(a) When the "Word/byte units designation" in GX Configurator-SC is set to "0" (word units)

The following diagram shows the control procedure when two words are designated as transmission data

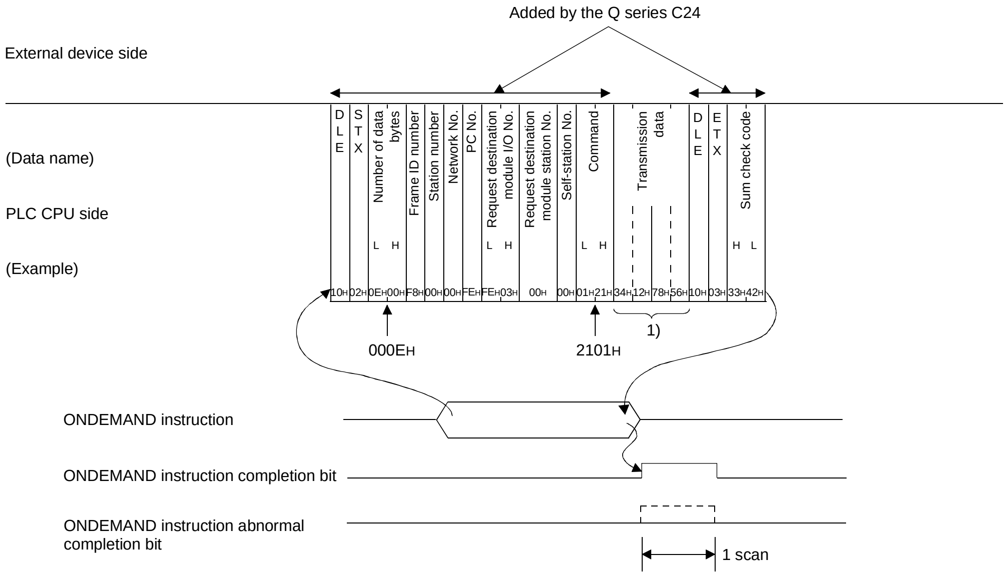


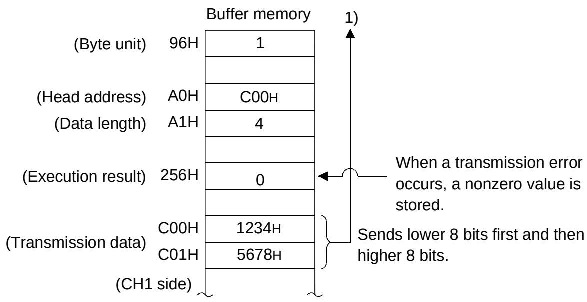

**POINT**

The number of bytes in the transmission data section of the on-demand data is the data length $\times$ 2.
(Two bytes are used for one word of data.)

(b) When the "Word/byte units designation" in GX Configurator-SC is set to "1" (byte units)

The following diagram shows the control procedure when two words (four bytes) are designated as transmission data



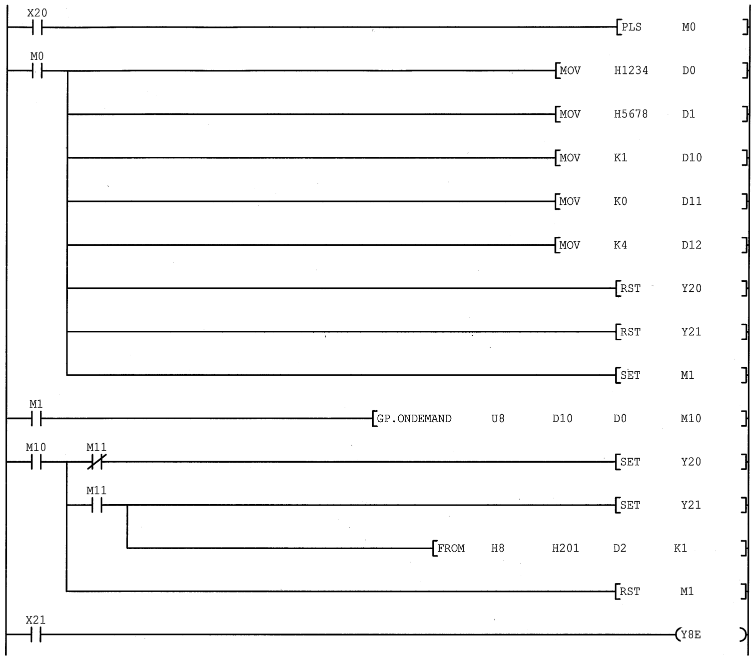

**POINT**

(1) The number of bytes in the transmission data section of the on-demand data is the same as the data length.
(One byte is used for one byte of data.)

(2) When the data length is an odd number, the data of the lower byte (bits 0 to 7) of the last data storage device is transmitted.

(3) Sample sequence program using the on-demand function (the ONDEMAND instruction)

The following is a sample program demonstrating the use of the on-demand function, using the controls described in (1) and (2) above as an example.
(Q series C24 I/O signals are X/Y80 to X/Y9F, transmission from CH1)
See Chapter 9 of the User's Manual (Basic) for more information about the ONDEMAND instruction.




Convert the transmission command to PLS

Set the transmission data

Designate the number of the interface (CH□ ) that will be used to transmit the data.
Clear the device that stores the result of the transmission to 0.
When the transmission data units are word units, this should be changed to [MOV K2 D12].

Set the transmission ready complete flag.

Start the on-demand function (request transmission)

Normal completion
Abnormal completion
Output signal for operator confirmation

Determine the cause of the abnormal completion by reading the result of the on-demand execution in D11 and the on/off status of the LED in D2.

D11 — 0 : Transmitted normally
Non-zero : Data was not transmitted correctly due to an error.

Request to turn off Error LED

Reset the stored value of the on-demand executed result.

**POINT**

(1) The status of communications by a dedicated command can be read using the SPBUSY instruction.

(2) See Chapter 9 of the User's Manual (Basic) for a detailed description of Point (1).

(3) Designate the transmission data storage size (stored in D0 and D1 in the sample program above) and data length (stored in D12 in the example program above) so that they do not exceed the buffer memory range allocated to the on-demand function by the user.

## 3.12 Initializing the Transmission Sequence: for Serial Communication Modules

This function initializes the data communication transmission sequence by the QnA compatible 4C frame format 5 of the MC protocol, and instructs the Q series C24 to wait for commands from an external device.
To initialize the Q series C24 transmission sequence from the external device during data communication in format 5, transmit the command message described in this section to the Q series C24.
The function described in this section is equivalent to the EOT and CL functions in data communication in ASCII code.
See the descriptions of the EOT and the CL in Section 3.1.6 (1) (b) for the following items:
- When the external device initializes the Q series C24 transmission sequence
- Processing and operation when the Q series C24 initializes the transmission sequence
- How to initialize the transmission sequence during data communication in ASCII code

**POINT**

This function can be used only for the Q series C24 connected to an external device (including a multidrop link station).
It cannot be used for the Q series C24 of other stations connected over a network system.

### 3.12.1 Commands

The following explains the commands used when an external device initializes the Q series C24 transmission sequence.

| — / Function | Command (subcommand) | Processing | PLC CPU status | | |
|---|---|---|---|---|---|
| | | | During STOP | During RUN | |
| | | | | Write allow setting | Write prohibit setting |
| Transmission sequence initialization | 1615 (0000) | Terminates the current processing request and instructs the Q series C24 to set to the command wait status. | ○ | ○ | ○ |

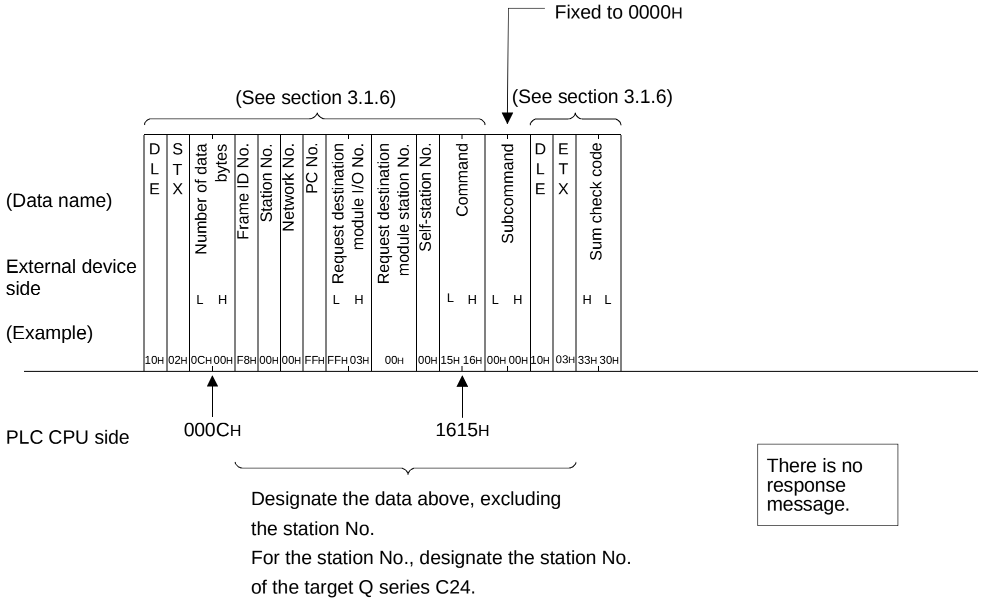
○ in the PLC CPU Status column in the table above indicates that execution is possible.

### 3.12.2 Transmission sequence initialization (command: 1615)

The following example explains the control procedure for initializing the transmission sequence.

[Control procedure]

The QnA compatible 4C frame format 5 is used to describe the control procedure in this example.



## 3.13 Mode Switching: for Serial Communication Module

This function forcefully switches the current communication protocol (operation mode) and transmission specifications of the designated interface from an external device after the Q series C24 is started.
This function should be used when it is desired to continue communication without rebooting the QCPU after changing the communication protocol and transmission specifications of the designated interface.

This section deals only with how to use the command for switching the Q series C24 mode from an external device.

**POINT**

(1) Read Chapter 15 of the User's Manual (Application) before switching the mode. Chapter 15 also explains how to switch the Q series C24 mode from the PLC CPU.

(2) The mode of the Q series C24 can only be switched from an external device (including stations connected by mulitdrop) connected to the Q series C24. Other stations of the Q series C24 connected over a network system cannot be used.

(3) The Q series C24 mode switching starts when the mode switching request is issued.
When the mode switching request is issued, the Q series C24 terminates any processing it has been performing.

### 3.13.1 Commands and contents of the character area

The following explains the commands and the contents of the character area (data section in communication in binary code) when the Q series C24 mode is switched from an external device.

#### (1) Commands

| Function \ — | Command (subcommand) | Processing | Number of processing performed per communication | PLC CPU status | | | Reference section |
|---|---|---|---|---|---|---|---|
| | | | Access station-1 (See Section 3.2 ＊7) | During STOP | During RUN: Write allow setting | During RUN: Write prohibit setting | |
| Mode switching | 1612 (0000) | Switches the operation mode and transmission specifications of the designated interface. | (One station) | ○ | ○ | ○ | Section 3.13.2 |

○ in the PLC CPU status column in the table above indicates that execution is possible.

#### (2) Contents of the character area

The following explains the contents of the character area when the Q series C24 mode is switched from an external device.
For the contents and constraints for each setting value, see Section 4.5.2 of the User's Manual (Basic).

(a) Channel number
This value designates the interface (CH□) whose mode is to be switched.

1) Data communication in ASCII code

The value shown in 3) is converted to a 2-digit (hexadecimal) ASCII code and sequentially transmitted beginning from the most significant digit ("0").

2) Data communication in binary code

Transmit using the 1-byte value shown in 3).

3) The channel number values designated and corresponding target interfaces are shown below.

| Designated value | Target interface |
|---|---|
| $1_H$ | CH1 interface of the Q series C24 |
| $2_H$ | CH2 interface of the Q series C24 |

(b) Switching instruction

This data designates which mode switching will be performed.

1) Data communication in ASCII code

The value shown below is converted to a 2-digit (hexadecimal) ASCII code and sequentially transmitted beginning from the most significant digit.

2) Data communication in binary code

The 1-byte value shown below is transmitted.

3) The designated values and mode switching to be performed in the switching instruction are shown below.

When all bits are 0 (OFF), the mode switching is performed according to each switching setting in GX Developer.

| b7 | – | – | – | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 1/0 | 1/0 | 1/0 |

..... $01_H$ to $07_H$ can be designated.

b0: Mode number (operation mode) switching instrucion
- 0 : Switch to the number set in the communication protocol setting in GX Developer
- 1 : Switch to the number designated in the mode number item (c)

b1: Transmission specification switching instrucion
- 0 : Switch to the specifications set in the transmission setting in GX Developer
- 1 : Switch to the specification designated in the transmission setting item (d)

b2: Communication rate switching instrucion
- 0 : Switch to the rate set in the communication rate setting in GX Developer
- 1 : Switch to the rate designated by the communication rate setting item (d)

(c) Mode number

This data if for designating the communication protocol setting after the mode is switched.

1) Data communication in ASCII code

The 2-digit (hexadecimal) ASCII codes are specified in the table below. Transmit from the most significant digit.

(Example: if the code is "01", transmit from "0".)

| Designated value | | Mode number (operation mode) |
|---|---|---|
| ASCII | Binary | |
| 01 | $01_H$ | MC protocol (format 1) |
| 02 | $02_H$ | MC protocol (format 2) |
| 03 | $03_H$ | MC protocol (format 3) |
| 04 | $04_H$ | MC protocol (format 4) |
| 05 | $05_H$ | MC protocol (format 5) |
| 06 | $06_H$ | Non procedure protocol |
| 07 | $07_H$ | Bidirectional protocol |
| FF | $FF_H$ | GX Developer connection ＊1 |

＊1 If designating the GX Developer connection mode set the communication protocol to "$00_H$". (See section 4.5.2 of the User's Manual (Basic).)

2) Data communication in binary code
Transmit using the 1-byte value ($01_H$ to $07_H$, $FF_H$) shown in 1).

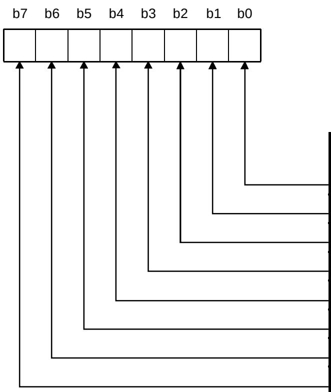
3) When the "mode number (operation mode) switching instruction" of indicated in (b) above is "0", designate a dummy value of "01" to "07" and "FF" (ASCII code) in this section. Do not designate "0".

(d) Transmission setting and communication speed setting
This data is for designating the transmission specifications after the mode is switched.

1) Data communication in ASCII code
The value shown below is converted to a 2-digit (hexadecimal) ASCII code and sequentially transmitted from the most significant digit.

2) Data communication in binary code
Transmits using the 1-byte value shown below.

3) The following shows the designated values and contents of the transmission and communication speed settings.
[Transmission setting] ∗1




0/1 (OFF/ON)

| Bit | Contents | 0 (OFF) | 1 (ON) | Remarks |
|---|---|---|---|---|
| b0 | Operation setting | Independent | Linked | Always set to off for CH1 |
| b1 | Data bit | 7 bits | 8 bits | Does not include parity bit |
| b2 | Parity bit | No | Yes | Vertical parity |
| b3 | Even/odd parity | Odd | Even | Valid only when parity bit is set to Yes |
| b4 | Stop bit | 1 bit | 2 bits | — |
| b5 | Sum check code | No | Yes | — |
| b6 | Write during RUN | Prohibit | Allowed | — |
| b7 | Setting modification | Prohibit | Allowed | — |

∗1 All items in the table should be set to OFF for the interfaces which "GX Developer connection" is set in the communication protocol setting.

[Communication speed setting] ∗1 ∗2 ∗3

| Communication speed (Unit: bps) | Setting value | Communication speed (Unit: bps) | Setting value |
|---|---|---|---|
| 50 | $0F_H$ | 14400 | $06_H$ |
| 300 | $00_H$ | 19200 | $07_H$ |
| 600 | $01_H$ | 28800 | $08_H$ |
| 1200 | $02_H$ | 38400 | $09_H$ |
| 2400 | $03_H$ | 57600 | $0A_H$ |
| 4800 | $04_H$ | 115200 | $0B_H$ |
| 9600 | $05_H$ | 230400 | $0C_H$ |

∗1 Transmission speed of 230400 bps is available for only CH1 of the QJ71C24N (-R2/R4).
∗2 When connecting external devices to both of two interfaces, the total of the communication speed should be 115200 bps of less (230400 bps or less if using QJ71C24N (-R2/R4)). When connecting an external device to either of two interfaces, the maximum of 115200 bps is available for the interface (a maximum of 230400 bps if using QJ71C24N (-R2/R4)). In this case, set 300 bps for the other interface to which no external device is connected.
∗3 Set "$00_H$" to the interface for which "GX Developer connection" is set in the communication protocol setting. Serial communication module will operate at the communication speed set on the GX Developer.

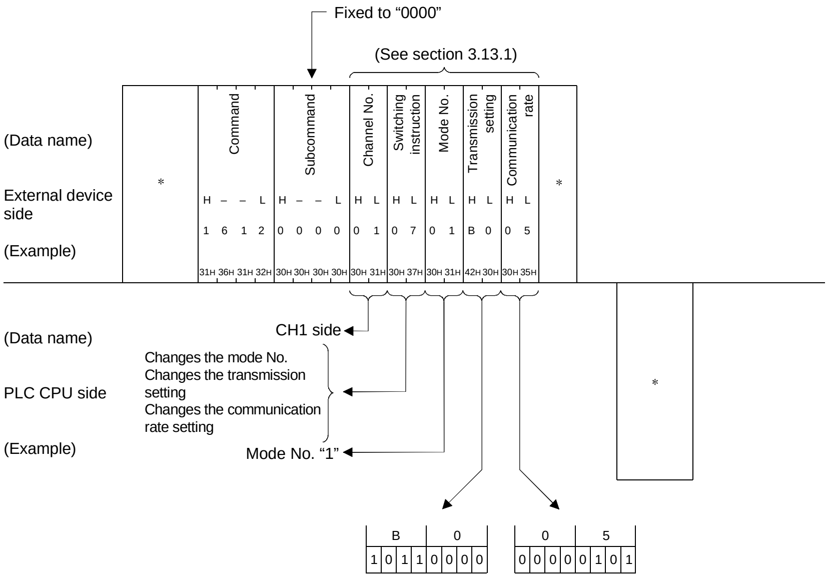
4) When the "transmission specifications switching instruction" and "communication speed switching instruction" indicated in (b) above is "0", designate a dummy value of "00" (ASCII code) in this section.

### 3.13.2 Mode switching (command: 1612)

The examples shown in this section explain the control procedure for switching the Q series C24 mode from an external device.
The order and contents of data items of the area marked by "✻" shown in the control procedure diagram differ depending on the module used as well as the frame and format used for communication.
See the detailed information described in Sections 3.1.4 through 3.1.6.

[Control procedure]

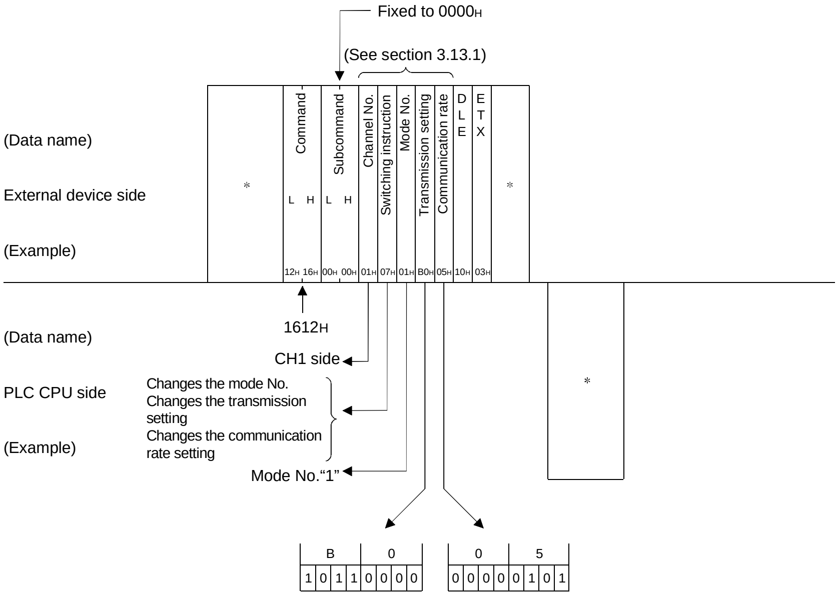
(1) Switching the mode of the interface on the CH1 side while communicating in ASCII code




✻ The following is designated in the diagram above.

| Designation item | Designation contens | Corresponding switch (set value) | |
|---|---|---|---|
| Operation mode | MC protocol format 1 | Communication protocol setting (1) | |
| Operation setting | Independent operation | Transmission setting | Bit 0: off |
| Data bits setting | 7 bits | | Bit 1: off |
| Parity bit setting | No | | Bit 2: off |
| Even parity/odd parity setting | Odd | | Bit 3: off |
| Stop bit setting | 2 bits | | Bit 4: on |
| Sum check setting | Yes | | Bit 5: on |
| Write during RUN allow/prohibit setting | Prohibit Write at RUN time | | Bit 6: off |
| Setting modification allow/prohibit setting | Enable modification | | Bit 7: on |
| Communication rate | 9600 bps | Communication rate setting ($5_H$) | |

(2) Switching the mode of the interface on the CH1 side while communicating in binary code




∗ The mode switching settings for the diagram above are shown in (1).

## 3.14 Turning Off Displayed LEDs and Initializing Communication Error Information and Error Code: for Serial Communication Module

This function is used by the external device to turn off the ERR LEDs of the Q series C24 LED display, or to initialize the communication error information and error code stored in the buffer memory.
Use this function to initialize the current error information caused by a NAK response to a command message, etc. and restore it to the normal status, or to initialize the error code storage area in the buffer memory.
The Q series C24 processes this function in the same way as the processing by the output signals and buffer memory shown below.
The processing corresponds to the on/off status of bits 0 to 3 of the value indicated by subcommands described in Section 3.14.1 (2) (a).
Please read Chapter 10 of the User's Manual (Basic) before using this function.
This section only explains how to use the commands.

| Status of memory bit that indicates the designated value by a subcommand | Output signal and buffer memory that perform the same processing | Processing | Reference section |
|---|---|---|---|
| When bit 0 is 1 (ON) | Error information initialization request signal (YE) | Turns off the ERR LED that is lit due to errors on the CH1 side, turns off input signal XE, and initializes the error code of the CH1 side in the buffer memory. | Chapter 10, User's Manual (Basic) |
| When bit 1 is 1 (ON) | Error information initialization request signal (YF) | Turns off the ERR LED that is lit due to errors on the CH2 side, turns off input signal XF, and initializes the error code of the CH2 side in the buffer memory. | |
| When bit 2 is 1 (ON) | Communication error information initialization request area on the CH1 side (buffer memory address $0_H$) | Initializes communication error information. | |
| When bit 3 is 1 (ON) | Communication error information initialization request area on the CH2 side (buffer memory address $1_H$) | Initializes communication error information. | |

**POINT**

(1) Initialize the target communication error information and error code after checking what processing the Q series C24 is performing when the bits 0 to 3 indicated by subcommands are on/off.
It is recommended to initialize each interface, or both interfaces simultaneously.
(The subcommand should be designated with a value of $0005_H$, $000A_H$, or $000F_H$.)

(2) This function can only be used for the Q series C24 that are connected to an external device (including multidrop link stations).
It cannot be used for the Q series C24 of other stations that are connected over a network system.

### 3.14.1 Commands and contents of the character area

The following explains the commands and the contents of the character area (data section in communication in binary code) when turning off displayed LEDs of the Q series C24 and initializing the communication error information from an external device.

(1) Commands

| Function | Command (subcommand) | Processing | PLC CPU status: During STOP | PLC CPU status: During RUN, Write allow setting | PLC CPU status: During RUN, Write prohibit setting | Reference section |
|---|---|---|---|---|---|---|
| Turning off displayed LED and initializing error code | 1617 (000□) | Turns off the displayed LED, initializes error code, etc. | ○ | ○ | ○ | Section 3.14.2 |

○ in the PLC CPU status column in the table above indicates that execution is possible.

(2) Contents of the character area

The following explains the contents of the character area when initializing communication error information and error codes of the Q series C24 from an external device.

(a) Subcommand

This data is for designating which processing is to take place on the Q series C24 side.

1) Data communication in ASCII code

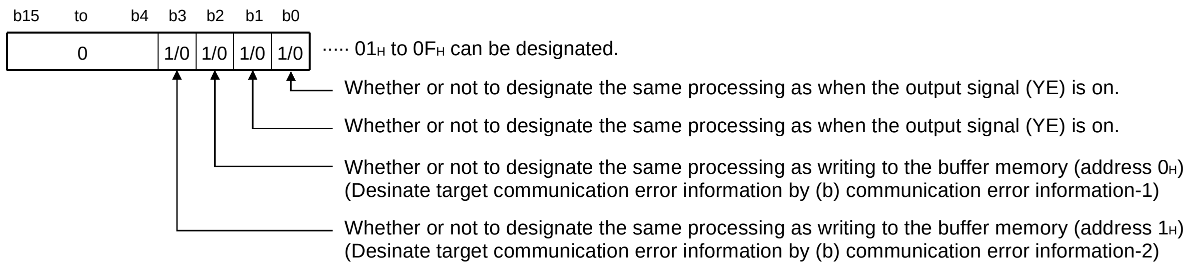
The value shown below is converted to a 2-digit (hexadecimal) ASCII code and sequentially transmitted beginning from the most significant digit ("0").

2) Data communication in binary code

The 1-byte value shown below is transmitted.

3) The designated values in the subcommand and corresponding processing are as follows.

In order to designate that the relevant processing should take place, set the corresponding bit to 1(ON).



(b) Communication error information-1 and communication error information-2

These values indicate the communication error information to be initialized.

- Communication error information-1 (mainly for designation for the CH1 side)
- Communication error information-2 (mainly for designation for the CH2 side)

1) Data communication in ASCII code

The value shown below is converted to 4-digit (hexadecimal) ASCII code and transmitted sequentially beginning from the most significant digit.

(Example)

When all the communication error information on the CH2 side is initialized, the value is converted to "C0FF" and transmitted sequentially beginning from the "C."

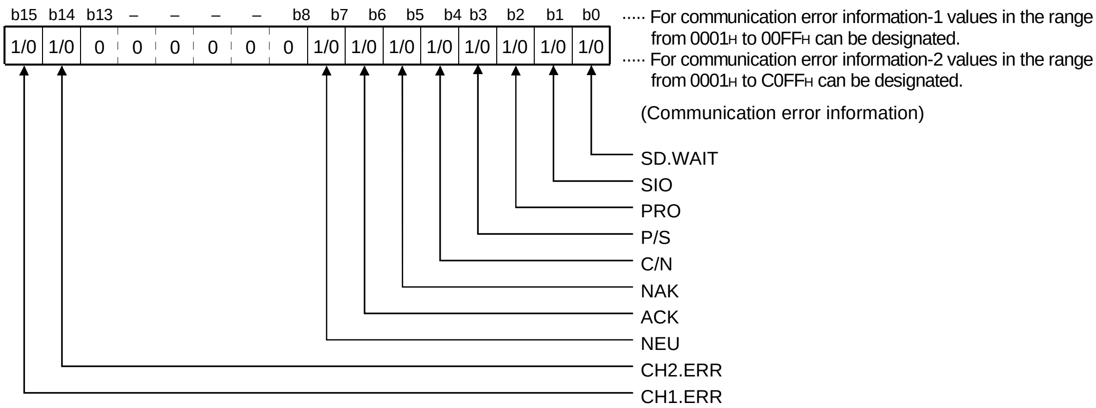
2) Data communications in binary code

The 2-byte value shown below is sequentially transmitted beginning from the low byte (L: bits 0 to 7)

3) The values and settings that can be designated for communication error information-1 and communication error information-2 are as follows.

The communication error information to be initialized can be designated by setting the corresponding bit to 1(ON).

| Subcommand | Communication error information-1 | Communication error information-2 | Remark |
|---|---|---|---|
| 0001$_H$ to 0003$_H$ | 00$_H$ | 00$_H$ | Values other than 00$_H$ should be designated according to the diagram below. |
| 0004$_H$ to 0007$_H$ | 0001$_H$ to 00FF$_H$ | 00$_H$ | |
| 0008$_H$ to 000B$_H$ | 00$_H$ | 0001$_H$ to C0FF$_H$ | |
| 000C$_H$ to 000F$_H$ | 0001$_H$ to 00FF$_H$ | 0001$_H$ to C0FF$_H$ | |

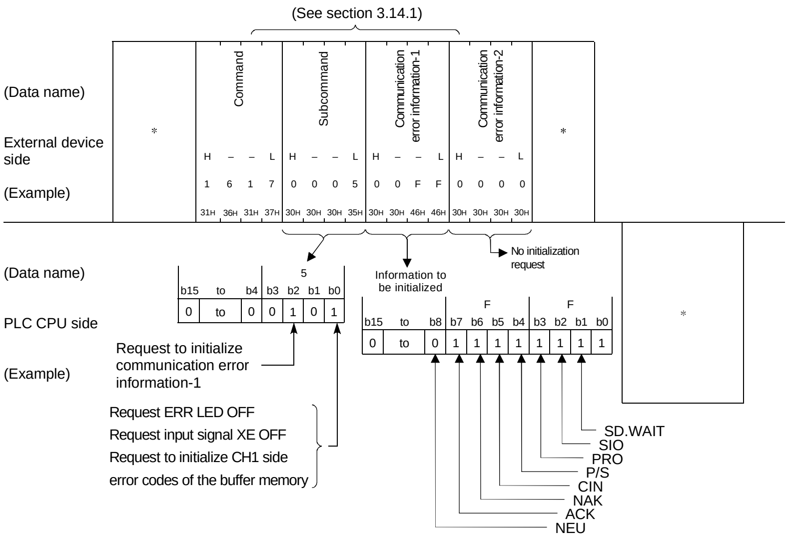


### 3.14.2 Turning off displayed LEDs and initializing communication error information and error code (command: 1617)

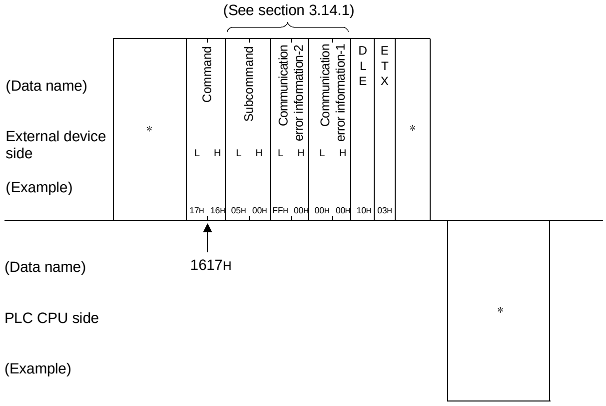
The examples shown in this section explain the control procedure for turning off the displayed LEDs of the Q series C24 and for initializing the communication error information and error code from an external device.
The order and contents of data items of the area marked by "＊" shown in the control procedure diagram differ depending on the module used as well as the frame and format used for communication.
See the detailed information described in Sections 3.1.4 through 3.1.6.

[Control procedure]

(1) Turning off both the ERR LEDs of the interfaces on the CH1 side and the input signal XE, initializing the error code on the CH1 side in the buffer memory, and initializing the error information designated by the communication error information-1 item during communication in ASCII code.




(2) Turning off both the ERR LEDs of the interfaces on the CH1 side and the input signal EX, initializing the error code on the CH1 side in the buffer memory, and initializing the error information designated by the communication error information-1 item during communication in binary code.



## 3.15 Turning Off the COM.ERR LED: for Ethernet Modules

This function is for turning off the COM.ERR LED on the front of the Q series E71 from an external device.

**POINT**

This function can be used only for the Q series E71 in the same Ethernet as the external device that requests to turn off the COM.ERR LED.
This function cannot be used for the Q series E71 of other stations that are connected via a network system.

### 3.15.1 Commands and contents of the character area

The following explains the commands and character area in the control procedure when turning off the displayed LED of the Q series E71 from an external device.

(1) Commands

| Function | Command (subcommand) | Processing | PLC CPU status | | | Reference section |
|---|---|---|---|---|---|---|
| | | | During STOP | During RUN | | |
| | | | | Write allow setting | Write prohibit setting | |
| Turn LED off | 1617 (0000) | Turns off the displayed LED. | ○ | ○ | ○ | Section 3.15.2 |

○in the PLC CPU status column in the table above indicates that execution is possible.

(2) Contents of the character area
Only the subcommand is transmitted according to the communication code.

(a) Subcommand

1) Data communication in ASCII code
"0000" is converted to 2-digit (hexadecimal) ASCII code and transmitted sequentially beginning from the most significant digit.

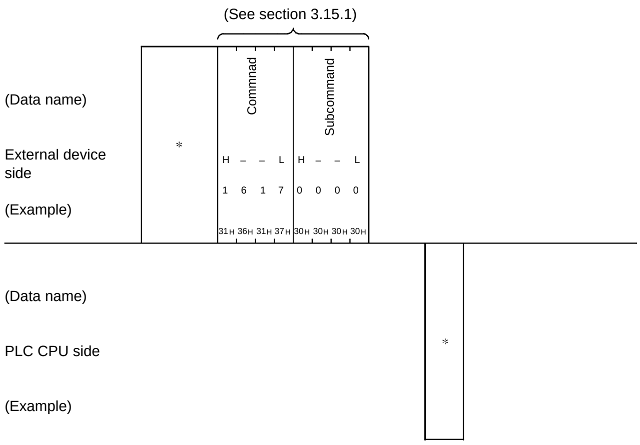
2) Data communication in binary code
$0000_H$ is transmitted.

## 3.15.2 Turning off the COM.ERR LED (command: 1617)

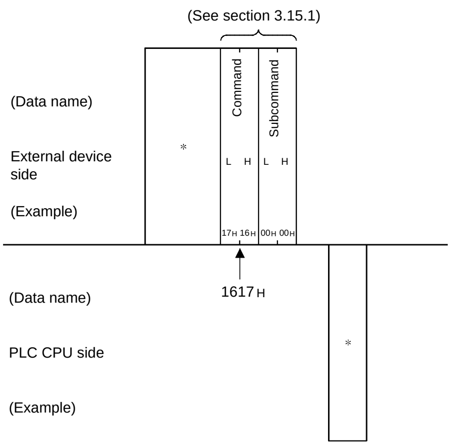
The example shown in this section explain the control procedure for turning off the displayed LED on the front of the Q series E71 from an external device.
The order and contents of data items of the area marked by "✻" shown in the control procedure diagram, see a detailed information descibed in Sections 3.1.2 and 3.1.3.

[Control procedure]

### (1) Turning off the COM.ERR LED while communicating in ASCII code




### (2) Turning off the COM.ERR LED while communicating in binary code



## 3.16 Loopback Test

A loopback test checks whether or not the communication function between an external device and the Q series C24/E71 module operates normally. The examples show the control procedure using this function.

**POINT**

(1) A loopback test allows to check whether or not the connection between the external device and the Q series C24/E71 is correct and that the data communication function operates properly when the Q series C24/E71 is started up or a problem occurs.
(2) This function can be used only for the Q series C24/E71 connected to an external device (including a multidrop link station).
It cannot be used for the Q series C24/E7 of other stations connected over a network system.

### 3.16.1 Commands and contents of character area

This section explains the commands and character area in the control procedure (data section in communication in binary code) when an external device performs a loopback test with the Q series C24/E71.

(1) Commands

| Function | Command (subcommand) | Processing | PLC CPU status | | | Reference section |
|---|---|---|---|---|---|---|
| | | | During STOP | During RUN | | |
| | | | | Write allow setting | Write prohibit setting | |
| Loopback test | 0619 (0000) | Check whether or not data communication is performed normally. | ○ | ○ | ○ | Section 3.16.2 |

○ in the PLC CPU status column in the table above indicates that execution is possible.

(2) Contents of the character area

This section explains the contents of the character area when an external device performs a loopback test with the Q series C24/E71.

(a) Number of loopback data
This value indicates the number of bytes in the loopback data area.
1) Data communication in ASCII code
The number of bytes is converted to 4-digit (hexadecimal) ASCII code and transmitted sequentially beginning from the most significant digit ("0").
2) Data communication in binary code
A 2-byte value that indicates the number of bytes is sequentially transmitted beginning from the low byte (L: bits 0 to 7).

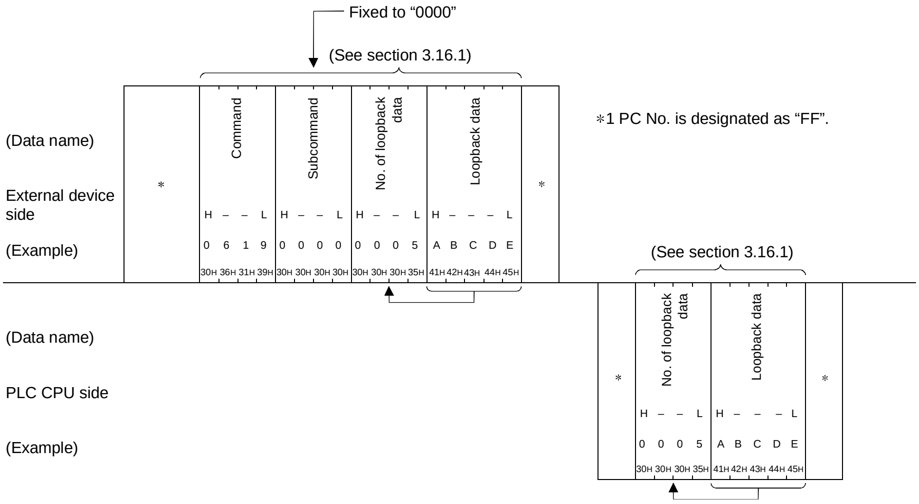
(b) Loopback data
This data is for designating the user data of a message whose data is communicated in a loopback test.
1) Data communication in ASCII code
A string consisting of a maximum of 960 characters (containing "0" to "9" and "A" to "F" characters) is transmitted in order from the first character.
2) Data communication in binary code
Each character code of the sequence of characters ("0" to "9" and "A" to "F") is converted to 1-byte values and transmitted from the first character code for a maximum of 960 bytes.

### 3.16.2 Loopback test (command: 0619)

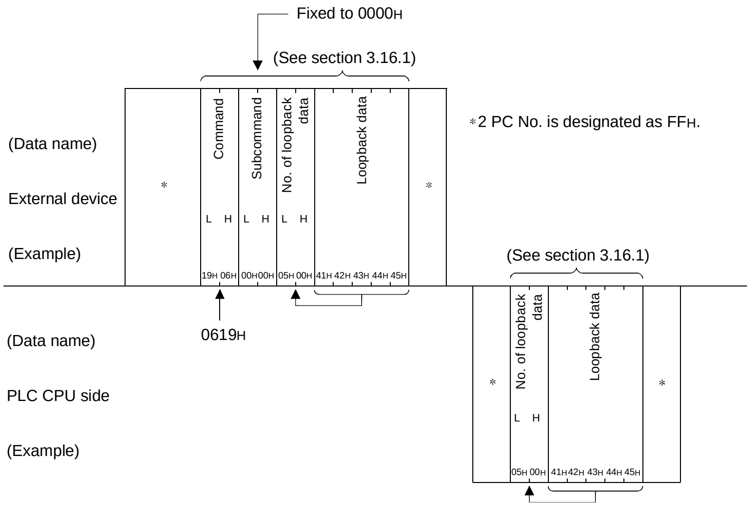
The examples shown in this section explain the control procedure for an external device performing a loopback test with the Q series C24/E71.
The order and content of data items of the areas marked by "∗" shown in the control procedure diagram differ depending on the module used as well as the frame and format used for communication.
See the detailed information described in Section 3.1.

[Control procedure]

(1) Performing a loopback test while communicating in ASCII code (The loopback data is "ABCDE")




(2) Performing a loopback test while communicating in binary code (The loopback data is "ABCDE")




**POINT**

The number of loopback data and the loopback data transmitted by an external device are returned to the external device as they are.

## 3.17 Registering or Canceling PLC CPU Monitoring: for Serial Communication Modules

These functions are for registering (including the start instruction) or canceling PLC CPU monitoring from an external device or the PLC CPU in order to utilize the PLC CPU monitoring function of the Q series C24.
By using the PLC CPU monitoring function, the external device can receive the previously registered device data and abnormal CPU information or message at a timing designated by a user.
This section explains the overview of the PLC CPU monitoring function as well as how to use the commands to register/cancel PLC CPU monitoring that are used by an external device communicating using the MC protocol in order to perform the PLC CPU monitoring function.
Please read Chapter 2 of the User's Manual (Application) before registering/canceling the PLC CPU monitoring described in this section using the PLC CPU monitoring function.
The User's Manual (Application) provides the details of the PLC CPU monitoring function, its execution procedure, the message format for transmitting information to an external device, how to register/cancel PLC CPU monitoring from the PLC CPU, etc.

### (1) Overview of the PLC CPU monitoring function

(a) The Q series C24 monitors the following monitor information on the PLC CPU, which should be registered by the user in advance.
- The numerical values stored in a word device
- The on/off status of a bit device
- The status of the PLC CPU of the local station (whether any abnormality occurs)

(b) The Q series C24 transmits the following user designated monitor information to an external device when the monitor information satisfies the information transmission condition designated by the user, or at a constant time interval.
(It is not necessary to process the transmission by the sequence program.)
- Multiple device data registered by a user or abnormal CPU information (It is possible to transmit monitoring information by combined use with the modem function)
- A notification message in combination with the modem function of the Q series C24 (notification is not possible using the MC protocol)

＊ If the information transmission to an external device by the PLC CPU monitoring function occurs at the same time as the following operations, it is performed at the same timing as the data transmission by the on-demand function described in Section 3.11 (see Section 3.11.1 (3)).
  - While transmitting a response message during communication using the MC protocol.
  - While receiving data from an external device in the half-duplex communication.

(c) The Q series C24 starts monitoring the PLC CPU when an external device or the PLC CPU registers a PLC CPU monitoring. The monitoring ends when an external device or the PLC CPU cancels the PLC CPU monitoring.

### (2) Interfaces to which the PLC CPU monitoring function transmits information

The information is transmitted to the interfaces for which the MC protocol or non-procedure protocol is set.
＊ The explanation of the PLC CPU monitoring function using the non-procedure protocol is provided in the User's Manual (Application).
This manual does not cover the PLC CPU monitoring function using the non-procedure protocol.

### (3) Message format of information transmitted by the PLC CPU monitoring function

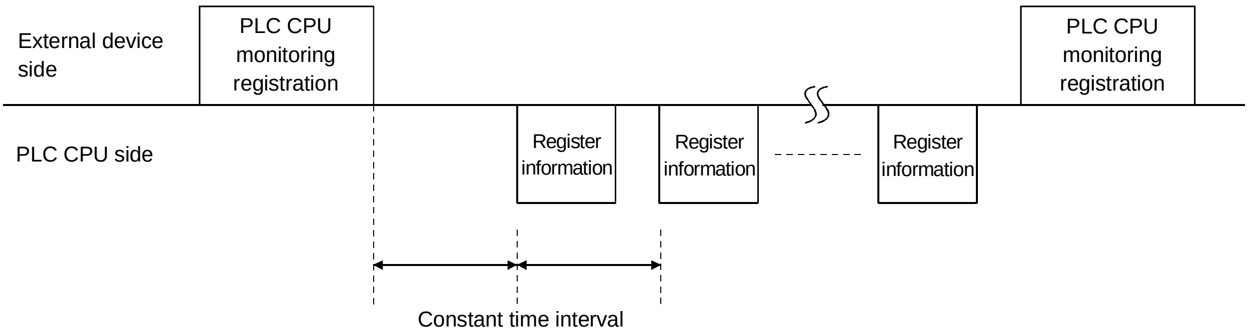
The message format of the information transmitted by the PLC CPU monitoring function is equivalent to the format of the message transmitted by the on-demand function described in Section 3.11 (see Section 3.11.2).

∗ The data for this function is the transmission data section of the on-demand data.

### (4) Control procedure

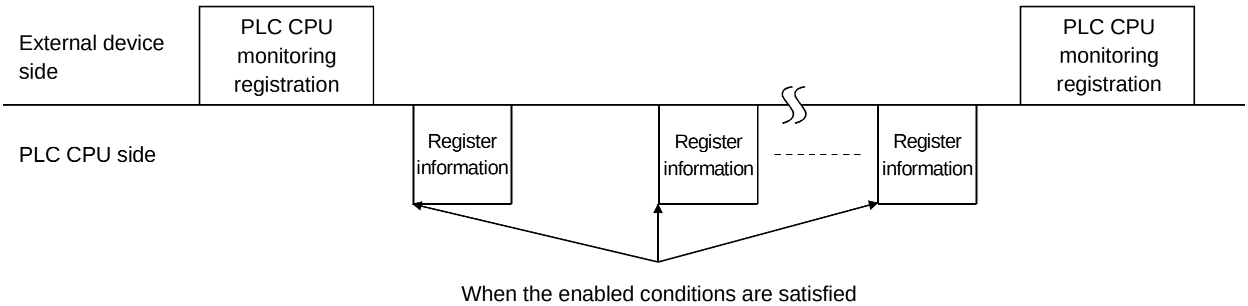
(Example)

Constant cycle transmission (One block is registered)




(Example)

Condition agreement transmission (One block is registered)




∗ Register information can be transmitted in the same way as the transmission in user frames for the on-demand function.
For further information, see the User's Manual (Application).

**POINT**

(1) The PLC CPU monitoring registration can be performed through the PLC CPU monitoring function, either from an external device, the PLC CPU or GX Configurator-SC.

(2) The external device or PLC CPU cannot register PLC CPU monitoring to the same Q series C24 twice in order to register a monitoring device, etc.
When PLC CPU monitoring has already been registered, cancel it first and then register a new monitoring.

(3) This function can be used only for the Q series C24 connected to an external device (including a multidrop link station).
It cannot be used for the Q series C24 of other stations connected over a network system.

## 3.17.1 Commands and contents of the character area

The following explains the commands and the contents of the character area (data section in communication in binary code) in the control procedure when the PLC CPU monitoring function is used from an external device.

### (1) Commands

| Function | — | Command (subcommand) | Processing | PLC CPU status: During STOP | PLC CPU status: During RUN, Write allow setting | PLC CPU status: During RUN, Write prohibit setting | Reference section |
|---|---|---|---|---|---|---|---|
| PLC CPU monitoring | Registration | 0630 (0000) | Registers the monitoring of a device memory and the CPU status, and starts the PLC CPU monitoring. | ○ | ○ | ○ | Section 3.17.2 |
| | Cancel | 0631 (0000) | Ends the PLC CPU monitoring. | | | | Section 3.17.3 |

○ in the PLC CPU status column in the table above indicates that execution is possible.

### (2) Contents of the character area

The following describes the contents of the character area when the PLC CPU monitoring function is used from an external device.

(a) Cycle time unit

This data designates the unit for the cycle time (b).

1) Data communication in ASCII code

The value shown below is converted to a 2-digit (hexadecimal) ASCII code and sequentially transmitted beginning from the most significant digit ("0").

2) Data communication in binary code

The 1-byte value is transmitted.

3) The following table summarizes the values that can be designated and their corresponding time units.

| Designated value | Time unit |
|---|---|
| $00_H$ | 100 ms |
| $01_H$ | 1 s |
| $02_H$ | 1 min |

4) The time designated by the cycle time unit and the cycle time (b) determines the time interval at which the Q series C24 reads the monitor information from the PLC CPU and is also used as transmission time interval for the constant cycle transmission.

(b) Cycle time

This value designates the time interval (one cycle time) at which the Q series C24 reads the monitor information from the PLC CPU.

1) Data communication in ASCII code

The cycle time (1 to 65535) is converted to 4-digit (hexadecimal: "0001" to "FFFF") ASCII code and transmitted sequentially beginning from the most significant digit ("0" in case of "0001".)

2) Data communication in binary code

A 2-byte value ($0001_H$ to $FFFF_H$) that indicates the cycle time (1 to 65535) is sequentially transmitted beginning from the low byte (L: bits 0 to 7).

(c) PLC CPU monitoring function (cycle monitoring function)

This value designates the transmission timing (constant cycle transmission or condition agreement transmission) when transmitting the resulting monitoring information of the PLC CPU (device information or CPU status information) to an external device.

1) Data communication in ASCII code
The value shown below is converted to a 2-digit (hexadecimal) ASCII code and sequentially transmitted beginning from the most significant digit ("0").
2) Data communication in binary code
The 1-byte value is transmitted.
3) The following table summarizes the values that can be designated and the corresponding transmission method employed by the PLC CPU monitoring function.

| Designated value | Transmission method used by the PLC CPU monitoring function | Information transmission timing |
|---|---|---|
| $01_H$ | Constant cycle transmission | Transmits information in the cycle time interval |
| $02_H$ | Condition agreement transmission | Transmits information when the information matches with the designated condition |

4) The time interval to read the information from the PLC CPU designated by the data items (a) and (b) above is also used as the transmission interval for constant cycle transmission.
5) The transmission timing for condition agreement transmission is designated by the data item (h) below.
The conditions for condition agreement transmission designated by the data items (i) and (j) below.
6) When a device monitoring result is sent by a condition agreement transmission, the head device of each block designated by (g) below is monitored.

(d) PLC CPU monitoring transmission measure

This value designates the method for transmission measure the results of PLC CPU monitoring to an external device.
In the MC protocol, only the data given below for "data transmission" can be designated (data transmission: transmission of device data and CPU status data).

1) Data communication in ASCII code
"00" is transmitted.
2) Data communication in binary code
The 1-byte value ($0_H$) is transmitted.

(e) Number of registered word blocks and number of registered bit blocks

These values designate the number of blocks of a word device (number of registered word blocks) and the number of blocks of a bit device (number of registered bit blocks) to be registered to the Q series C, when monitoring device data and transmitting the results while monitoring.

1) Data communication in ASCII code
The number of blocks is converted to a 2-digit (hexadecimal) ASCII code and transmitted sequentially beginning from the most significant digit ("0").
2) Data communication in binary code
A 1-byte value that indicates the number of blocks is transmitted.

3) Designate the device registration to the Q series C24 within the following ranges

10 blocks ≥ (total number of word blocks + total number of bit blocks)
960 points ≥ (total number of device points of all word blocks + total number of device points of all bit blocks)

4) The range for devices that can be registered is designated by item (g) below for each block.

(f) CPU abnormal monitoring

This flag designates whether or not the PLC CPU monitoring should include a monitoring of abnormalities in the PLC CPU of the local station.

1) Data communication in ASCII code
The value shown below is converted to a 2-digit (hexadecimal) ASCII code and sequentially transmitted beginning from the most significant digit ("0").
2) Data communication in binary code
The 1-byte value is transmitted.
3) The following table summarizes the values that can be designated and the corresponding CPU abnormal monitoring designation.

| Designated values | CPU abnormal monitoring designation |
|---|---|
| $00_H$ | Do not monitor abnormalities in the PLC CPU of the local station. |
| $01_H$ | Monitor abnormalities in the PLC CPU of the local station. |

(g) Device code, monitoring head device, and number of registered points (number of read points)

These values designate the ranges of the devices for each block for the number of blocks designated by item (e), the number of registered word blocks and number of registered bit blocks, when monitoring device data and transmitting the results while monitoring the PLC CPU.

＊ When using the PLC CPU monitoring function to monitor devices, the head device of each block is monitored.

Blocks designated for a word device : Head word device (for one word)
Blocks designated for a bit device : Head bit device (for one bit)

1) The device code is used for identifying the target devices in the corresponding block.
2) The monitoring head device is used for designating the head of the target device range in the corresponding block.
3) The number of registered points (number of read points) is used for designating the number of points from the head in the target device range in the corresponding device.
For a bit device, the number of points is designated by word units (one point = 16 bits).
4) The designation method for each data is the same as when reading/writing device memory described in Section 3.3.
Follow the designation method described in the explanation in (2) (c) 2) to 4) of Section 3.3.1.

(h) Transmission method in condition agreement transmission

This flag designates which method should be used to transmit the results of monitoring the PLC CPU when condition agreement transmission is designated in PLC CPU monitoring function (c).

1) Data communication in ASCII code
The value shown below is converted to a 2-digit (hexadecimal) ASCII code and sequentially transmitted beginning from the most significant digit ("0").
2) Data communication in binary code
The 1-byte value is transmitted.
3) Designate "00" or $00_H$ when constant cycle transmission is designated in PLC CPU monitoring function (c).
4) How the designated values correspond to transmission method of condition agreement transmission is shown below.

| Designated value | Transmission method in condition agreement transmission | Transmission method |
|---|---|---|
| $00_H$ | Edge trigger transmission | Transmits monitoring result only when the monitor conditions and data read match. |
| $01_H$ | Level trigger transmission | Transmits monitoring result in cycle time interval while the monitor conditions and data read match. |

(i) Monitoring conditions
This value (j) designates conditions (for transmitting monitoring results) when condition agreement transmission is designated in PLC CPU monitoring function (c).

1) Data communication in ASCII code
The value shown below is converted to a 2-digit (hexadecimal) ASCII code and sequentially transmitted beginning from the most significant digit ("0").
2) Data communication in binary code
The 1-byte value is transmitted.
3) Designate "00" or $00_H$ when constant cycle transmission is designated in PLC CPU monitoring function (c).
4) The following table summarizes the values that can be designated and the corresponding monitor condition.

| Designated value | Monitoring condition (judging criteria) | | Device for which designation is valid | |
|---|---|---|---|---|
| | | | Bit | Word |
| $01_H$ | Device value or status = device monitoring condition value or status | | ○ | ○ |
| $02_H$ | Device value or status ≠ device monitoring condition value or status | | | |
| $03_H$ | With sign | Device value ≤ Device monitoring condition value | × | |
| $04_H$ | | Device value < Device monitoring condition value | | |
| $05_H$ | | Device value ≥ Device monitoring condition value | | |
| $06_H$ | | Device value > Device monitoring condition value | | |
| $07_H$ | Without sign | Device value ≤ Device monitoring condition value | | |
| $08_H$ | | Device value < Device monitoring condition value | | |
| $09_H$ | | Device value ≥ Device monitoring condition value | | |
| $0A_H$ | | Device value > Device monitoring condition value | | |

(j) Monitoring condition value
This value designates the target status/numerical value of the monitoring condition (i) when condition agreement transmission is designated in PLC CPU monitoring function (c).

- When monitoring device is a word device: Designate the monitoring condition value by a numerical value
- When monitoring device is a bit device: Designate the monitoring condition value using a numerical value corresponding to on/off

1) Data communication in ASCII code
The value shown below is converted to an 4-digit (hexadecimal) ASCII code and sequentially transmitted beginning from the most significant byte "0."
2) Data communication in binary code
The 2-byte value shown below is transmitted from the low byte (L: bits 0 to 7).
3) Designate "0000" or $0000_H$ when constant cycle transmission is designated in PLC CPU monitoring function (c).
4) The following table summarizes the values that can be designated and the corresponding monitor condition.

| Designated value | Monitoring condition value or status | Type of monitoring device |
|---|---|---|
| $0000_H$ | OFF | Bit device |
| $0001_H$ | ON | |
| $000_H$ to $FFFF_H$ | Numerical value | Word device |

(k) Data read (device data transmitted to an external device as monitoring results)
1) This is device data for one monitoring target block read from the PLC CPU.
2) See Section 3.17.4 for the order and contents of each data.
For a bit device, the data for the number of registered points is read in word units (one points = 16 bits).
For a word device, the data for the number of registered points is read in one-point units.

(l) Data read (CPU status data transmitted to an external device as monitoring results)
1) This is status information of the PLC CPU read from the PLC CPU.
2) Data communication in ASCII code
The value shown below is transmitted in 4-digit (hexadecimal) ASCII code from the PLC CPU side. (See Section 3.17.4.)
3) Data communication in binary code
The 2-byte value below is transmitted beginning from the PLC CPU side. (See Section 3.17.4)
4) The following table shows the correspondence between the data read and the PLC CPU status.

| Data read | Status of PLC CPU | Remark |
|---|---|---|
| $0000_H$ | Operating normally | – |
| $0001_H$ | Module warning generated | Check the error description and take an action by referring to the QCPU manual. |
| $0002_H$ | Module error/module system error generated | |

REMARK

In the command message, designate the following values in the data item for designating access destination stations for a local station other than those mentioned above.

| Data item | Designated data | |
|---|---|---|
| | Data communication in ASCII code | Data communication in binary code |
| Network number | "00" | $00_H$ |
| PC number | "FF" | $FF_H$ |
| Request destination module I/O number | "03FF" | $03FF_H$ |
| Request destination module station number | "00" | $00_H$ |
| Self-station number | "00" | $00_H$ |

## 3.17.2 PLC CPU monitoring registration (command: 0630)

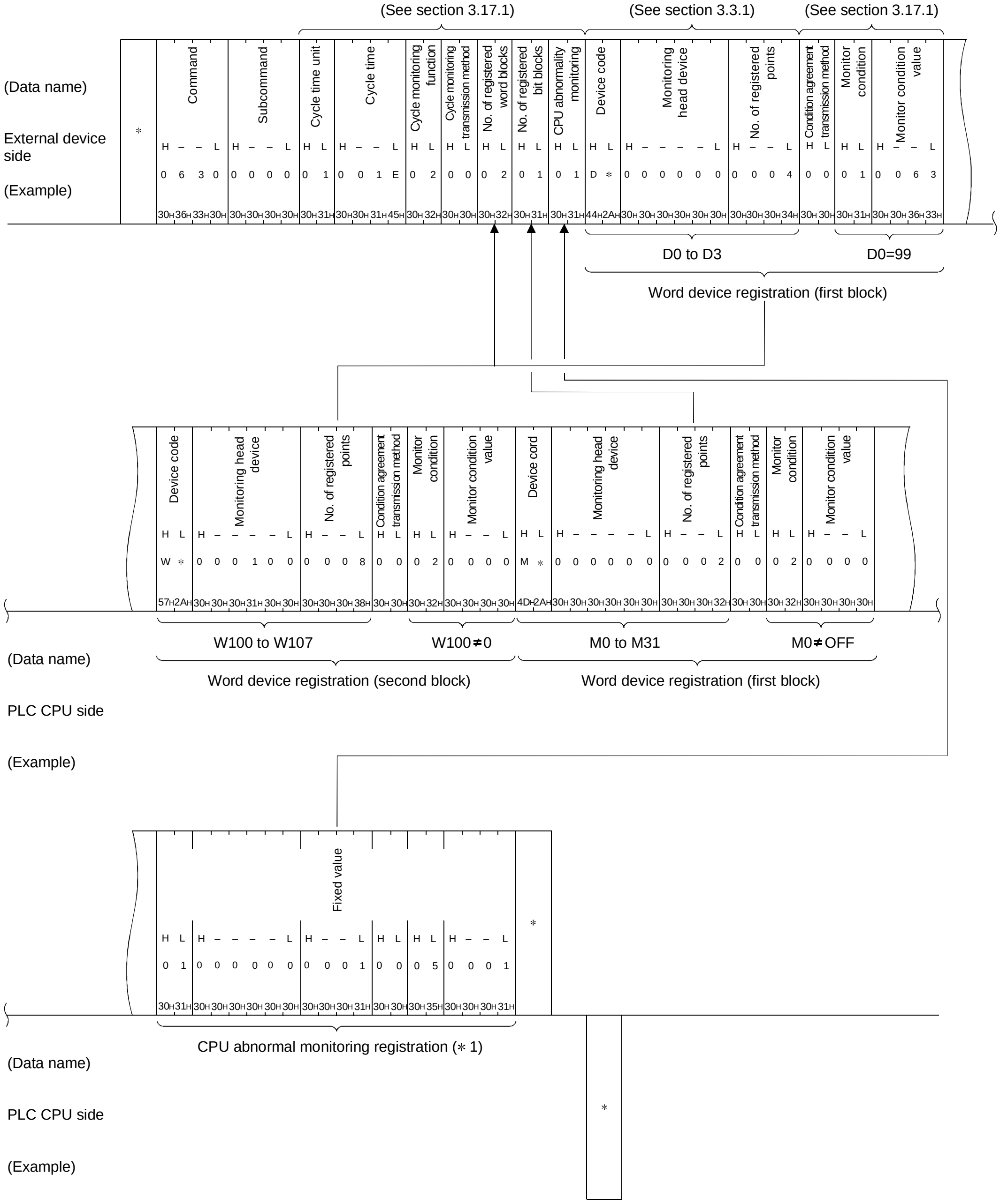
The examples shown in this section explain the control procedure for registering PLC CPU monitoring from an external device.
The order and contents of data items of the area marked by "✻" shown in the control procedure diagram differ depending on the module used as well as the frame and format used for communication.
See the detailed information described Sections 3.1.4 through 3.1.6.

[Control procedure]

When performing the following PLC CPU monitoring registration

| Data name | | | Registered value | Data name | | | Registered value |
|---|---|---|---|---|---|---|---|
| Cycle time unit, cycle time | | | 30 s | PLC CPU monitoring function | | | Condition agreement transmission |
| PLC CPU monitoring transmission method | | | Transmission of device data and CPU status data | Number of registered word blocks | | | 2 blocks |
| | | | | Number of registered bit blocks | | | 1 block |
| CPU abnormal monitoring | | | Include | – | | | – |
| Registered word device | | | (Monitoring device = D0) | Registered bit device | | | (Monitoring device = M0) |
| | First block | Device | D0 to D3 (4 points) | | Third block | Device | M0 to M31 (2 points) |
| | | Transmission method | Edge trigger transmission | | | Transmission method | Edge trigger transmission |
| | | Monitoring condition | Device value = Monitoring condition value | | | Monitoring condition | Device status ≠ Monitoring condition |
| | | Monitoring condition value | 99 | | | Monitoring status | OFF |
| Registered word device | | | (Monitoring device = W100) | – | | | – |
| | Second block | Device | W100 to W107 (8 points) | – | | | – |
| | | Transmission method | Edge trigger transmission | – | | | – |
| | | Monitoring condition | Device value ≠ Monitoring condition value | – | | | – |
| | | Monitoring condition value | 0 | – | | | – |

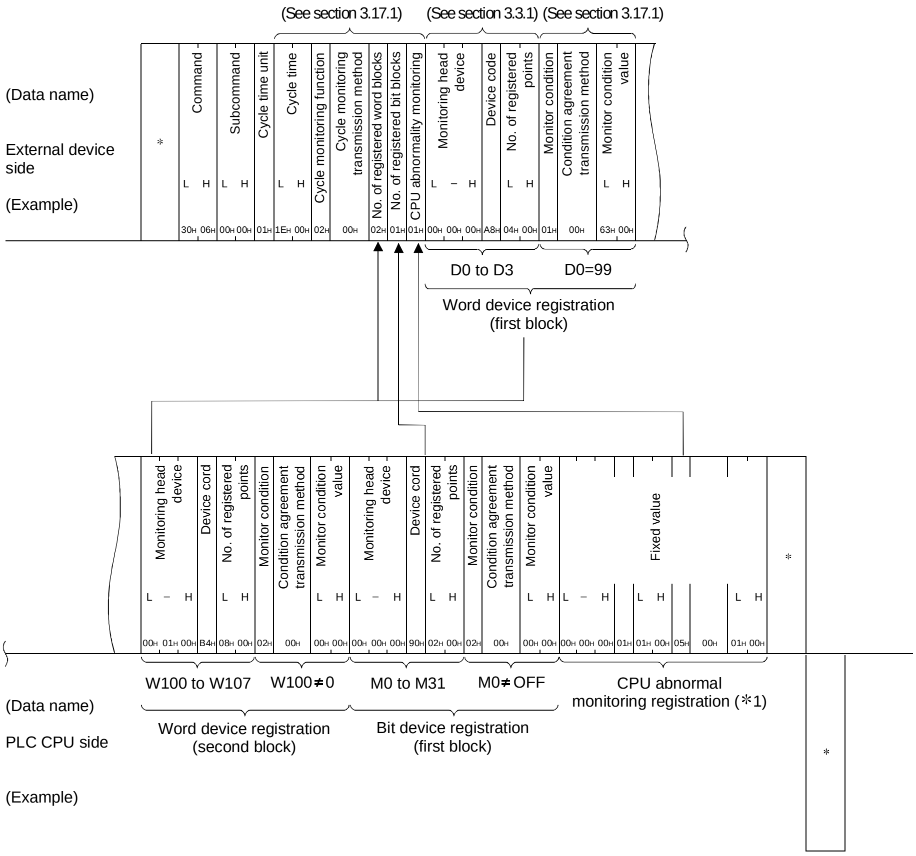
## (1) Communication in ASCII code




*1 If CPU abnormal monitoring is included in the monitor conditions, specify the following fixed values for monitor condition value items from the device code.

- Device code : "01"
- Monitoring head device : "000000"
- No. of registered points : "0001"
- Condition agreement transmission method : "00"
- Monitor condition : "05"
- Monitor condition value : "0001"

(2) Communication in binary code




*1 If CPU abnormal monitoring is included in the monitor conditions, specify the following fixed values for monitor condition value items from the device code.

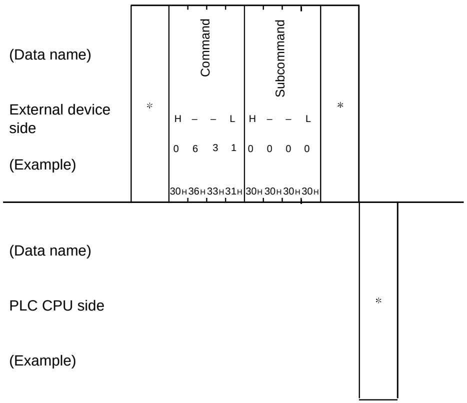
- Monitoring head device : 000000H
- Device code : 01H
- No. of registered points : 0001H
- Monitor condition : 05H
- Condition agreement transmission method : 00H
- Monitor condition value : 0001H

### 3.17.3 Canceling PLC CPU monitoring (command: 0631)

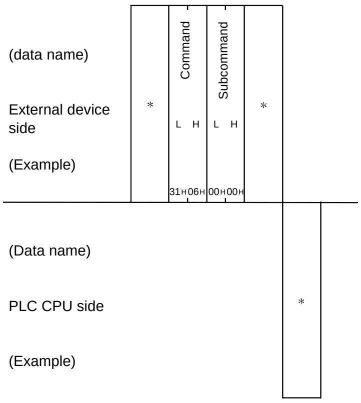
The examples shown in this section explain the control procedure for canceling the PLC CPU monitoring from an external device.
The order and contents of data items of the area marked by "∗" shown in the control procedure diagram differ depending on the module used as well as the frame and format used for communication.
See the detailed information described in Sections 3.1.4 through 3.1.6.

[Control procedure]

(1) Communication in ASCII code




(2) Communication in binary code



### 3.17.4 Data transmitted by the PLC CPU monitoring function

The examples shown in this section explain the monitor information transmitted from the Q series C24 by registering PLC CPU monitoring from an external device.

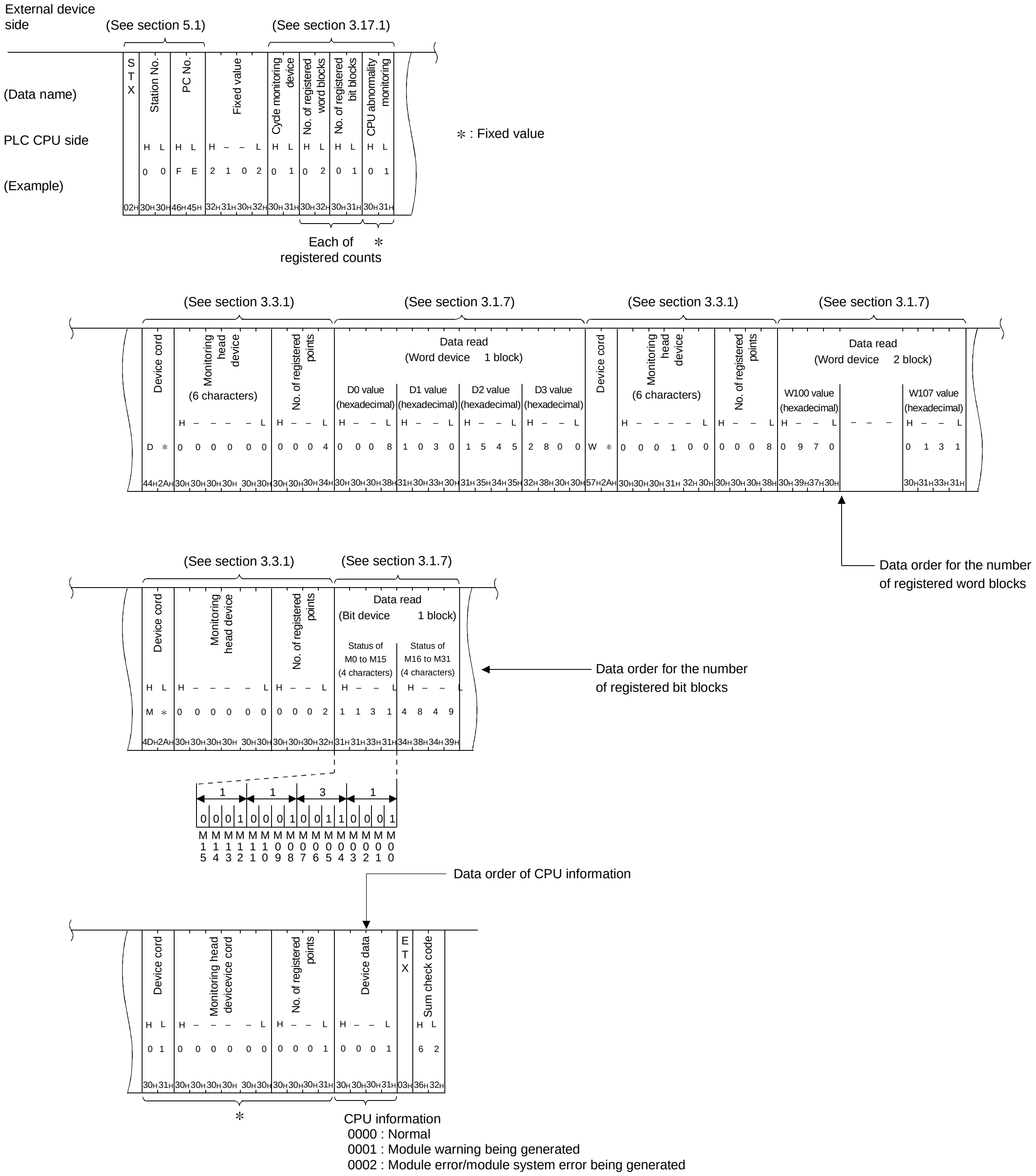
The order and contents of data items of the areas marked by "*" shown in the control procedure diagram differ depending on the module used as well as the frame and format used for communication.
When the communication protocol setting in GX Developer is "1 to 4," data is transmitted in A compatible 1C frames. See Section 5.1 for the order and contents of data items in each format of A compatible 1C frames.
When the communication protocol setting in GX Developer is "5," data is transmitted in QnA compatible 4C frame format 5. See Sections 3.1.5 and 3.1.6 for the order and contents of data items in QnA compatible 4C frames.

(1) When CPU status data and device data are transmitted in constant cycle transmission

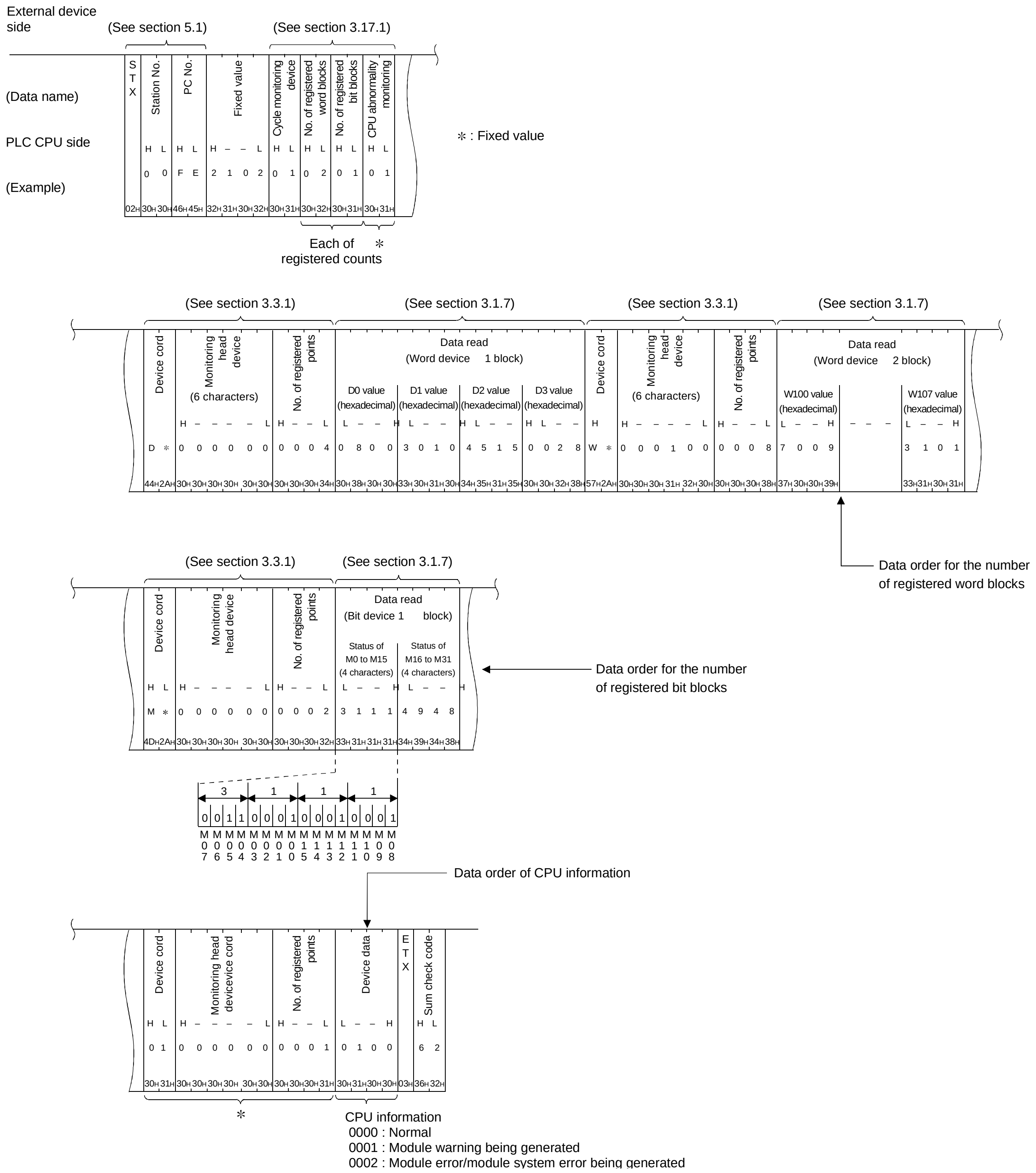
The example shown in this section explain the transmission when the device data shown in Section 3.17.2 is registered using the A compatible 1C frame format 1.
If "constant cycle transmission" is designated in the data item "PLC CPU monitoring function" when registering a PLC CPU monitoring, the registered data information is transmitted in a batch

(a) Communication in ASCII code

- When the "Word/byte units designation" in the GX Configurator-SC is set to "0" (word units)

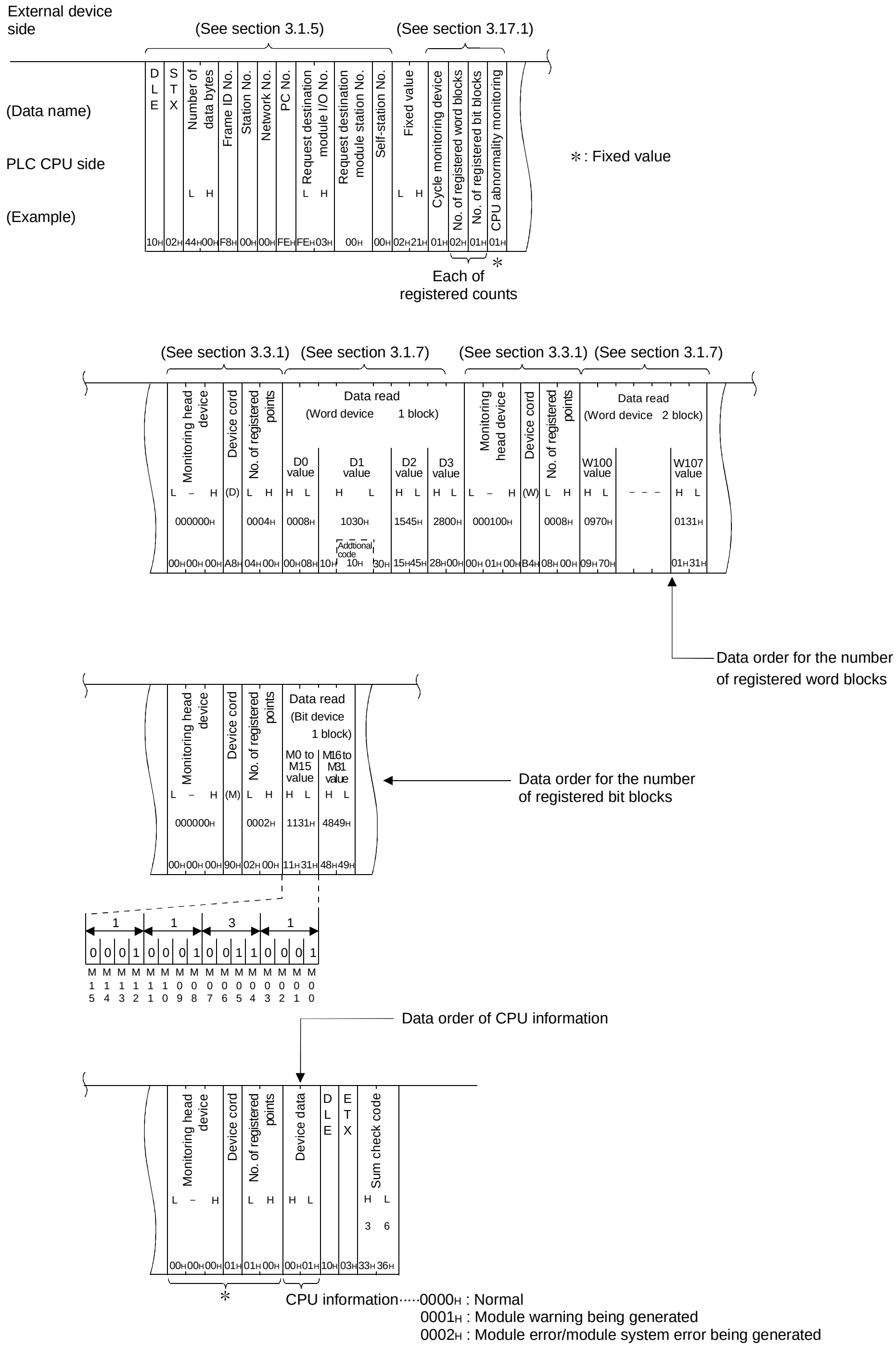


- When the "Word/byte units designation" in the GX Configurator-SC is set to "1" (byte units)

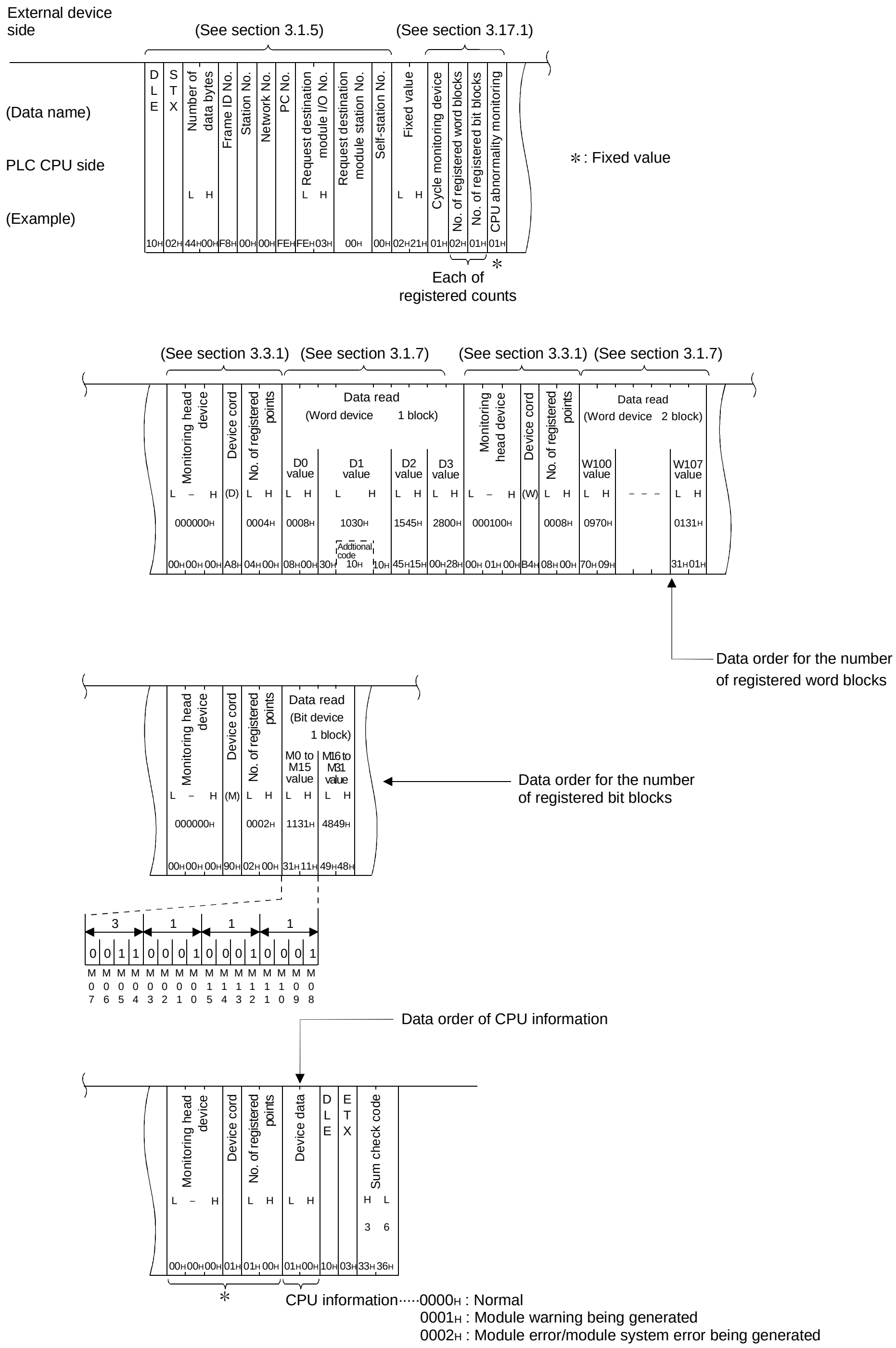


(b) Communication in binary code

- When the "Word/byte units designation" in the GX Configurator-SC is set to "0" (word units)



- When the "Word/byte units designation" in the GX Configurator-SC is set to "1" (byte units)



(2) When CPU status data or device data are transmitted in condition agreement transmission

The examples shown in this section explain the transmission when the PLC CPU monitoring is registered as shown in Section 3.17.2 using the A compatible 1C frame format 1.

If "condition agreement transmission" is designated in the data item of "PLC CPU monitoring function" when registering a PLC CPU monitoring, the registered data information is transmitted individually.

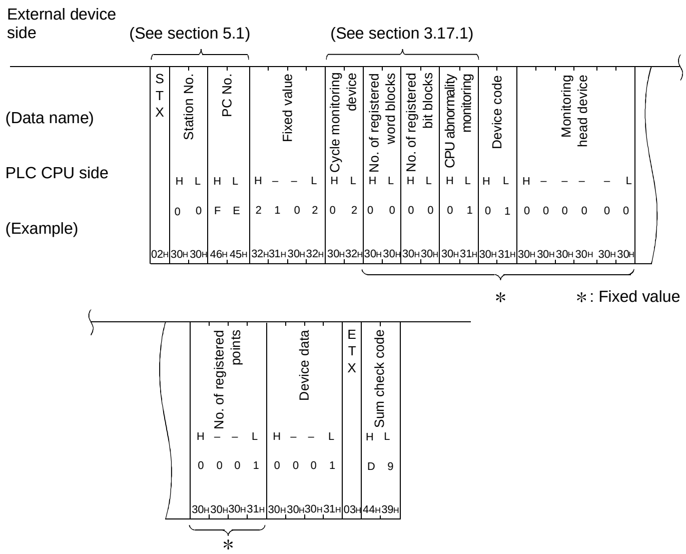
(The device data is transmitted for each block.)

For the contents of each information, see item (1) in this section.

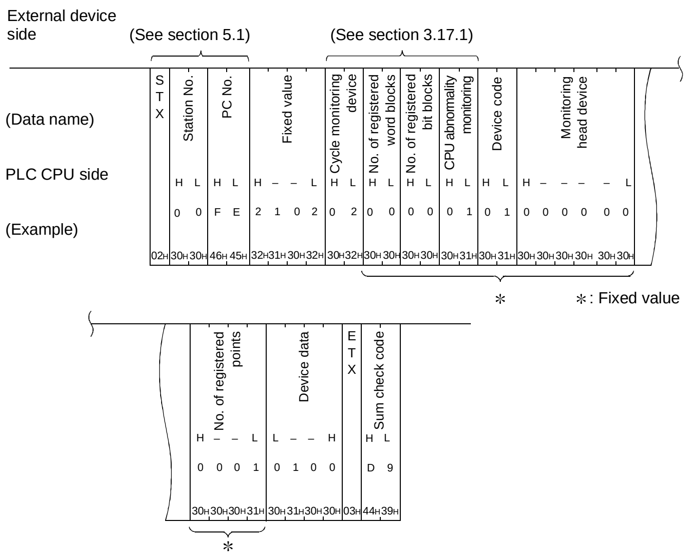
(a) Transmitting the CPU status data

1) Communication in ASCII code

• When the "Word/byte units designation" in the GX Configurator-SC is set to "0" (word units)

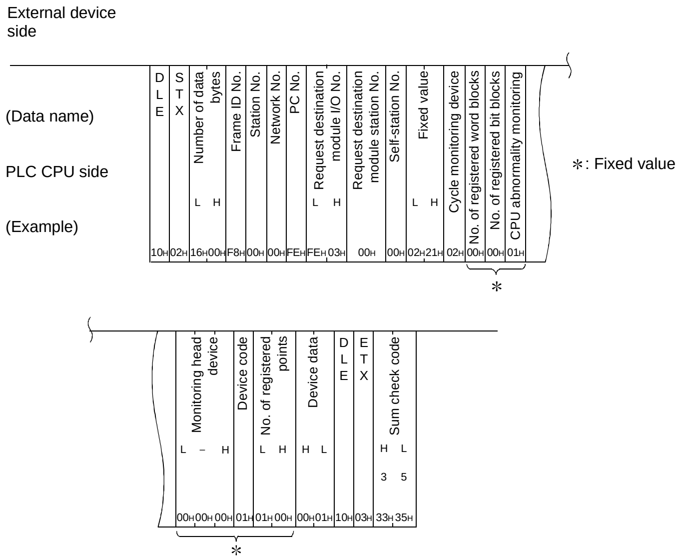



• When the "Word/byte units designation" in the GX Configurator-SC is set to "1" (byte units)

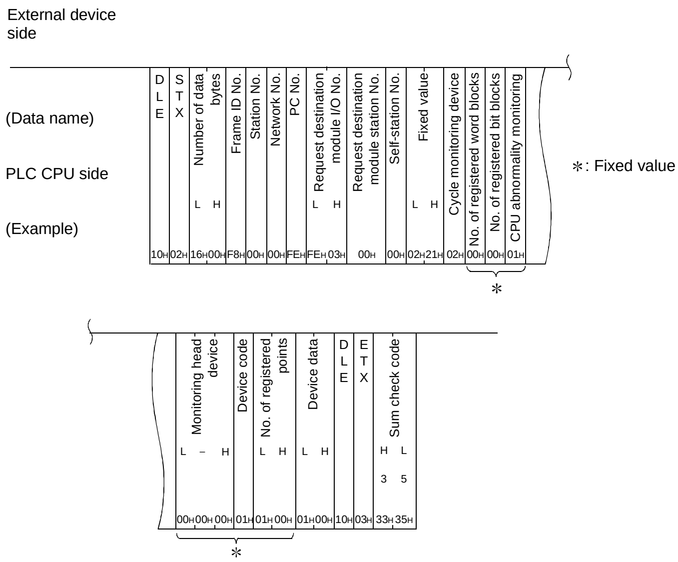


2) Communication in binary code

- When the "Word/byte units designation" in the GX Configurator-SC is set to "0" (word units)




- When the "Word/byte units designation" in the GX Configurator-SC is set to "1" (byte units)

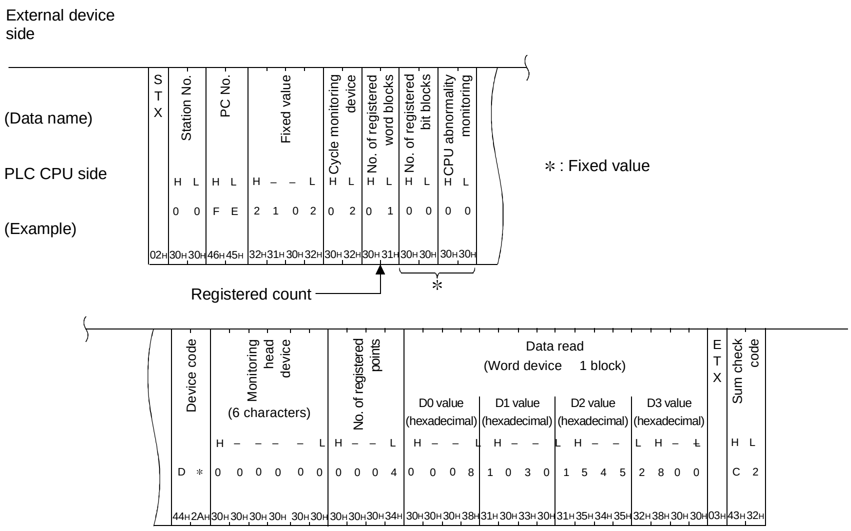


(b) Transmitting word device data

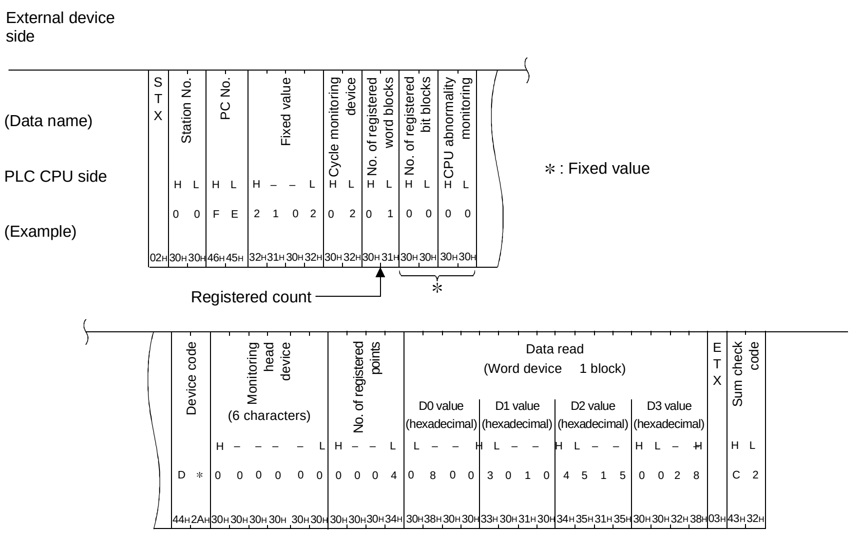
The examples shown in this section explain the transmission of word device data D0 to D3 (four points) by D0 = 99, when the PLC CPU monitoring shown in Section 3.17.2 is registered.

1) Communication in ASCII code

- When the "Word/byte units designation" in the GX Configurator-SC is set to "0" (word units)

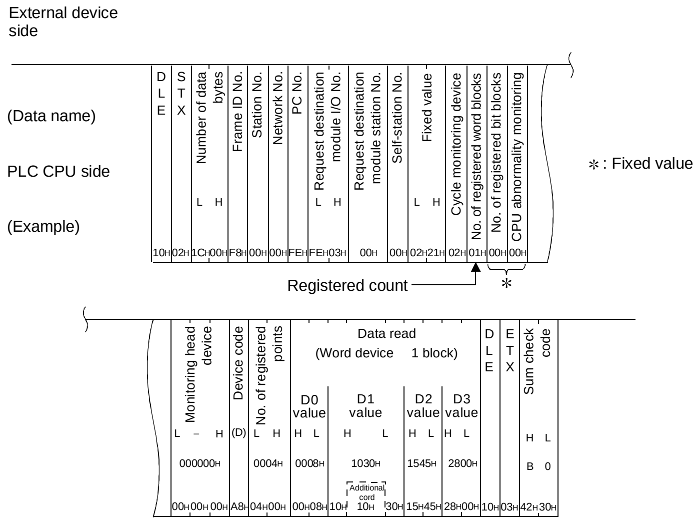



- When the "Word/byte units designation" in the GX Configurator-SC is set to "1" (byte units)

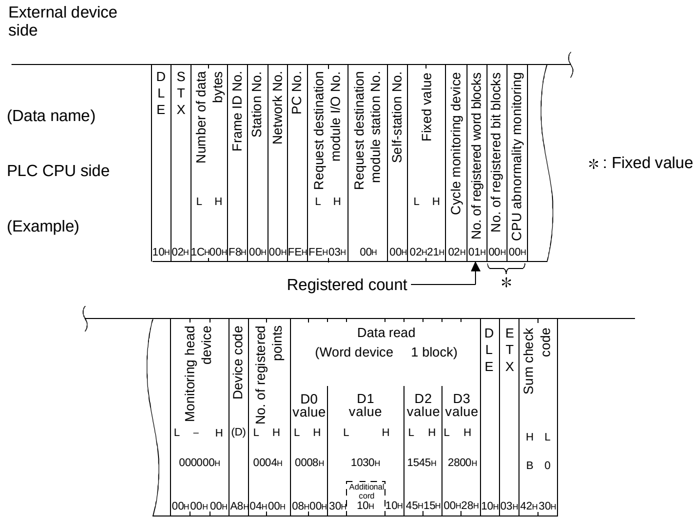


2) Communication in binary code

- When the "Word/byte units designation" in the GX Configurator-SC is set to "0" (word units)




- When the "Word/byte units designation" in the GX Configurator-SC is set to "1" (byte units)

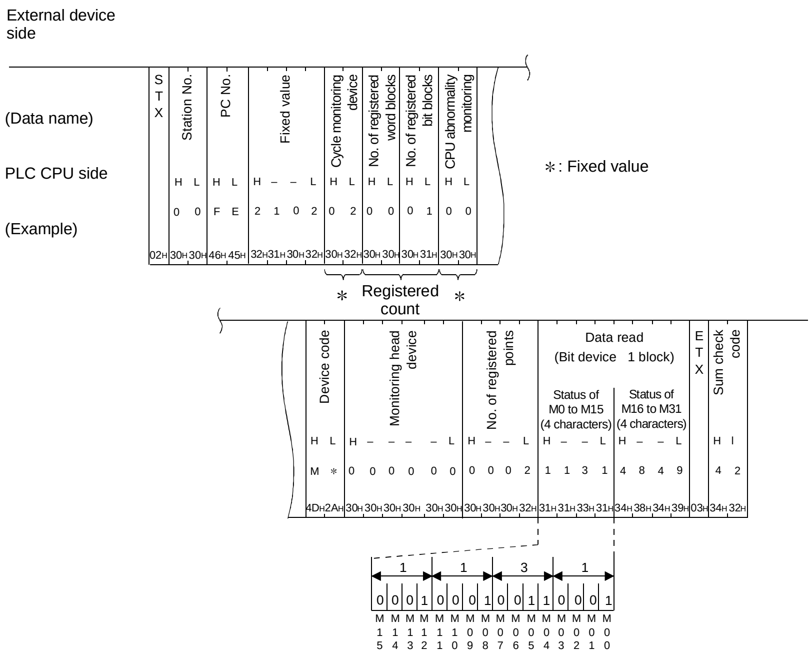


(c) Transmitting bit device data

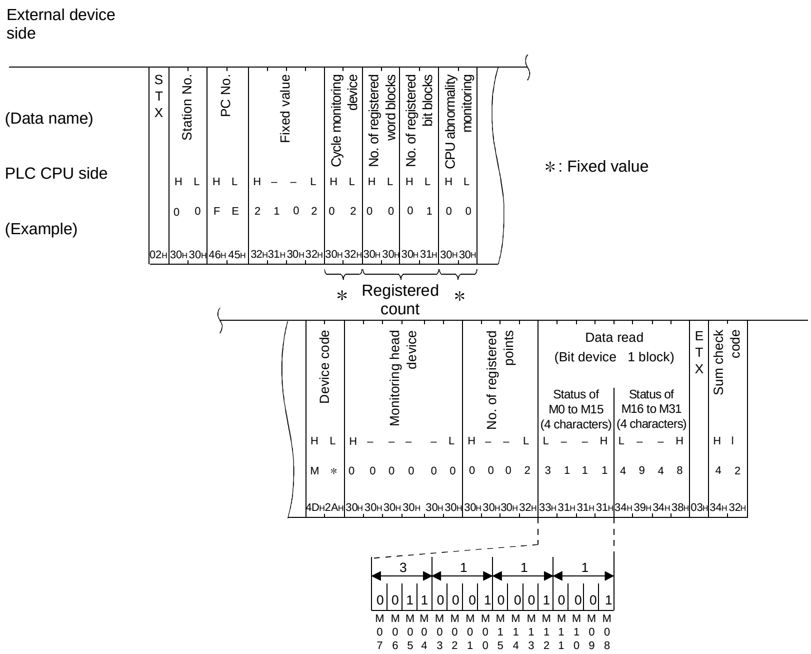
The examples shown in this section explain the transmission of bit device data M0 to M31 (two points) by M0 = ON, when the PLC CPU monitoring shown in Section 3.17.2 is registered.

1) Communication in ASCII code

- When the "Word/byte units designation" in the GX Configurator-SC is set to "0" (word units)

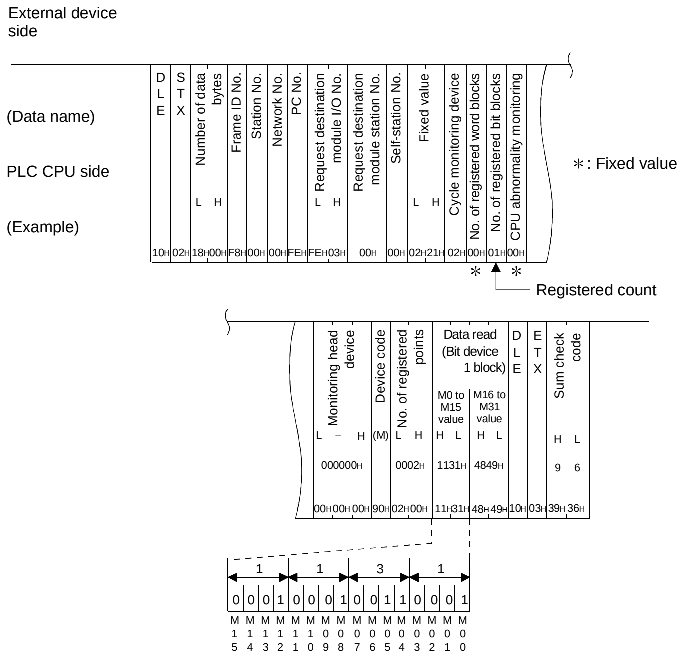



- When the "Word/byte units designation" in the GX Configurator-SC is set to "1" (byte units)

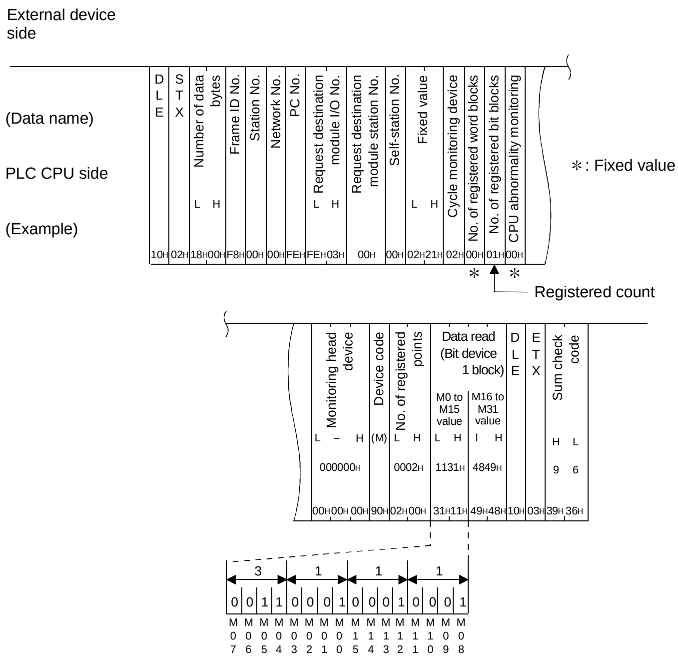


2) Communication in binary code

• When the "Word/byte units designation" in the GX Configurator-SC is set to "0" (word units)




• When the "Word/byte units designation" in the GX Configurator-SC is set to "1" (byte units)



## 3.18 Remote Password Unlock/Lock

The remote password check function prevents unauthorized access by a user at a remote location.
Performing the unlock process with respect to the remote password set in the QCPU using the GX Developer enables access from the external device.

This section explains how to use the commands that perform the unlock process/lock process for the remote password using the MC protocol.

**POINT**

(1) See the following manual for an overview and setting of the remote password.
   (a) Regarding the remote password (QCPU system configuration)
   QCPU User's Manual (Function Explanation, Program Fundamentals)
   (b) Compatibility with the remote password
   - Serial Communication Module User's Manual (Application) Section 3.3.3
   - Ethernet Interface Module User's Manual (Basic) Section 5.9
   (c) Setting the remote password
   GX Developer operating manual

(2) The commands for performing the remote password unlock process/lock process from the external device can only be used with the Q series C24 (including a multidrop connected station) or the Q series E71 connected to the external device.
These commands cannot be used with the Q series C24/E71 for other station that relays via the network system.

(3) The lock process command is a command for the Ethernet module.

### (1) Object of the remote password check function

If a remote password is set for the Q series C24/E71 loaded on the QCPU, perform the remote password unlock process using the command indicated in this section prior to performing data communication.

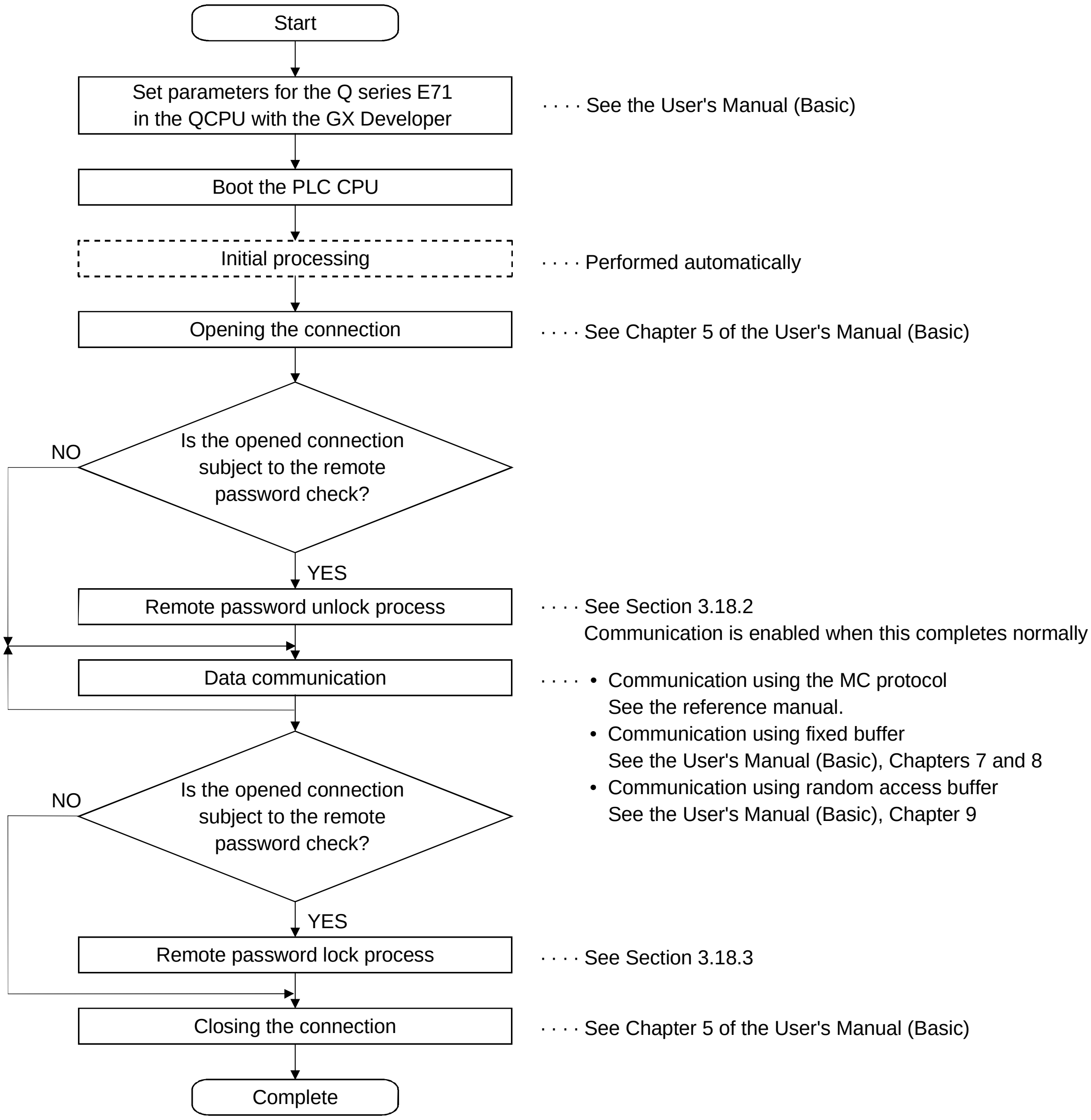
(a) Using the Q series E71
The remote password unlock process is required when the connection performing data communication per a request from the external device is the object of the remote password check.
(Example) Communication using the MC protocol
Communication using a fixed buffer (＊1)
Communication using the random access buffer (＊1)
＊1 Communication cannot be performed when the external device to be communicated is the Q series E71. (An error response will be returned if a communication request is made).

(b) Using the Q series C24
Perform data communication after doing the remote password unlock process when accessing the QCPU using the MC protocol communication from the interface that is using the modem function.

(2) Control procedure

The control procedure when a remote password is set in the QCPU for the station where the Q series C24/E71 is loaded is shown below.

(a) Accessing via the Q series E71



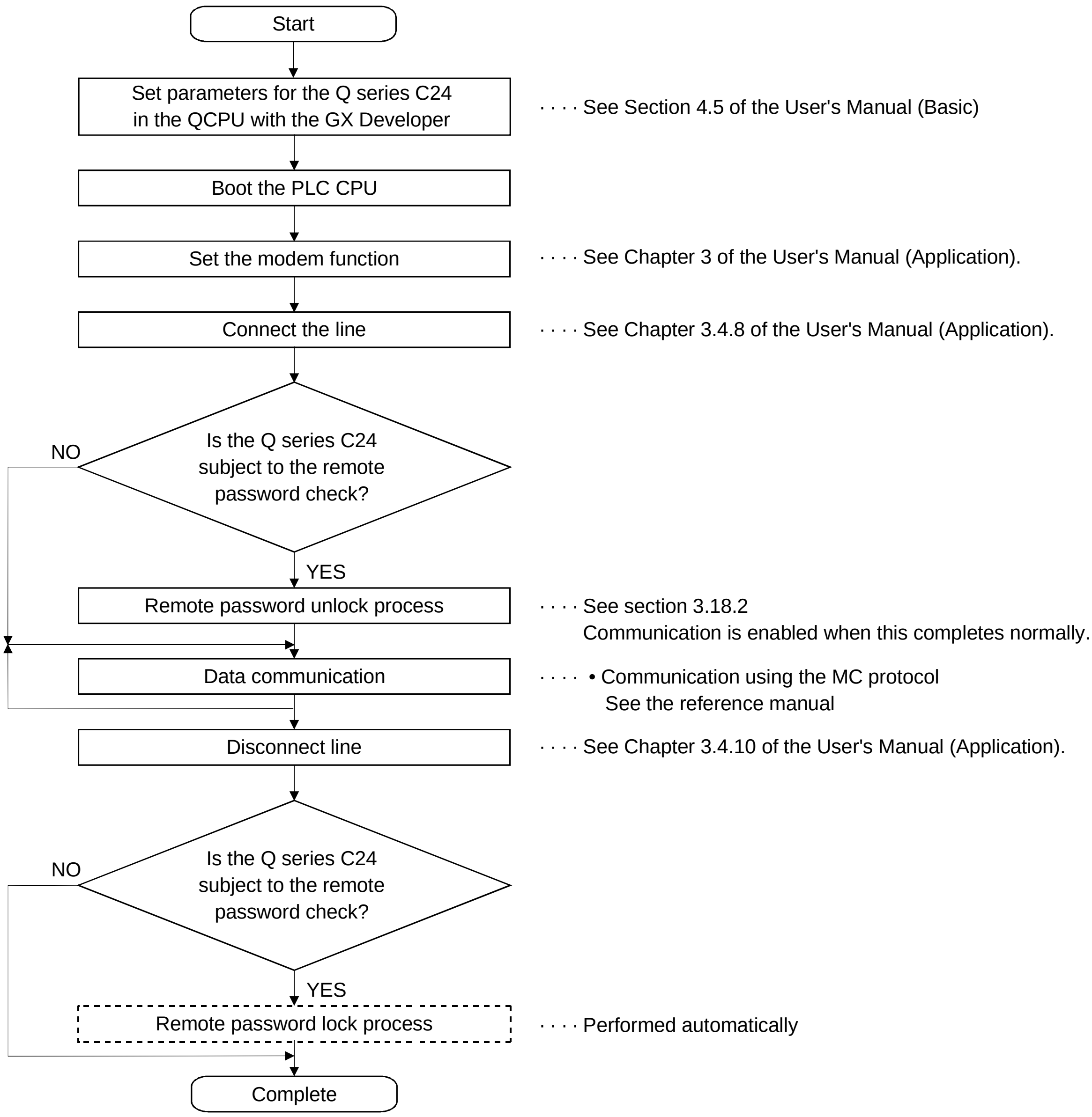

**POINT**

(1) When the connection used for data communication is set as an object of the remote password check, communication is enabled from after the completion of the unlock process until the lock process is executed.

(2) All commands received while the remote password is in locked status will generate an error response. Perform communication after executing the remote password unlock process.

(3) In TCP/IP communication, if the connection is closed without performing the remote password lock process, the lock process is automatically performed when the connection is closed. (The Q series E71 performs this automatically).

(b) Accessing via the Q series C24 (the interface using the modem function is the object)




**POINT**

(1) When the Q series C24 communicating data using the modem function is set as an object of the remote password check, communication is enabled from after the completion of the unlock process until the line is disconnected.

(2) All commands received while the remote password is in locked status will generate an error response. Perform communication after executing the remote password unlock process.

(3) The remote password lock process is automatically performed when the line is disconnected.

### 3.18.1 Contents of the command and character section

This section explains the commands and character section (the data section when communicating with binary code) of the control procedure when performing the remote password unlock process/lock process from an external device.

(1) Command

| Function | — | Command (subcommand) | Description of processing | Status of the PLC CPU | | | Reference section |
|---|---|---|---|---|---|---|---|
| | | | | During STOP | During RUN | | |
| | | | | | Write allow setting | Write prohibit setting | |
| Remote password | Unlock | 1630 (0000) | Specify the remote password and change locked status to unlocked status. (Communication with the PLC CPU is enabled.) | ○ | ○ | ○ | Section 3.18.2 |
| | Lock | 1631 (0000) | Specify the remote password and change unlocked status to locked status. (Communication with the PLC CPU is disabled.) | | | | Section 3.18.3 |

The ○ in the PLC CPU status column in the above table indicates that the process is executable.

(2) Contents of the character section

The following explains the contents of the character section when performing the remote password unlock process/lock process from an external device.

(a) Remote password length

This indicates the number of bytes (4) in the remote password.

(b) Remote password

1) This is the remote password that the user has set it with the GX Developer in the QCPU of the station where the Q series C24/E71 is loaded.

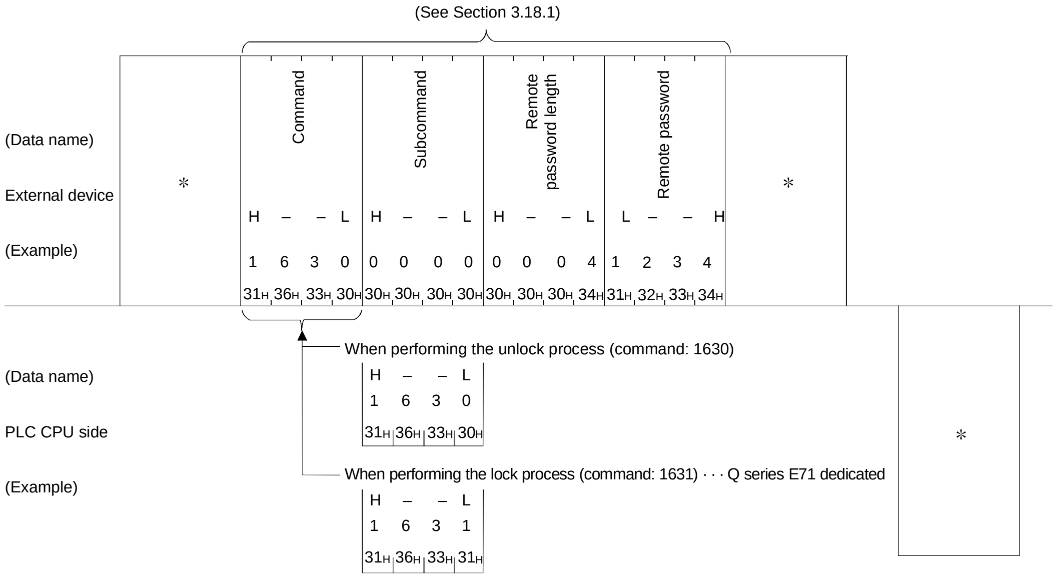
2) When data is communicated with ASCII/binary code, the remote password set in the QCPU is transmitted as is from the head character.

### 3.18.2 Remote password unlock/lock (command: 1630, 1631)

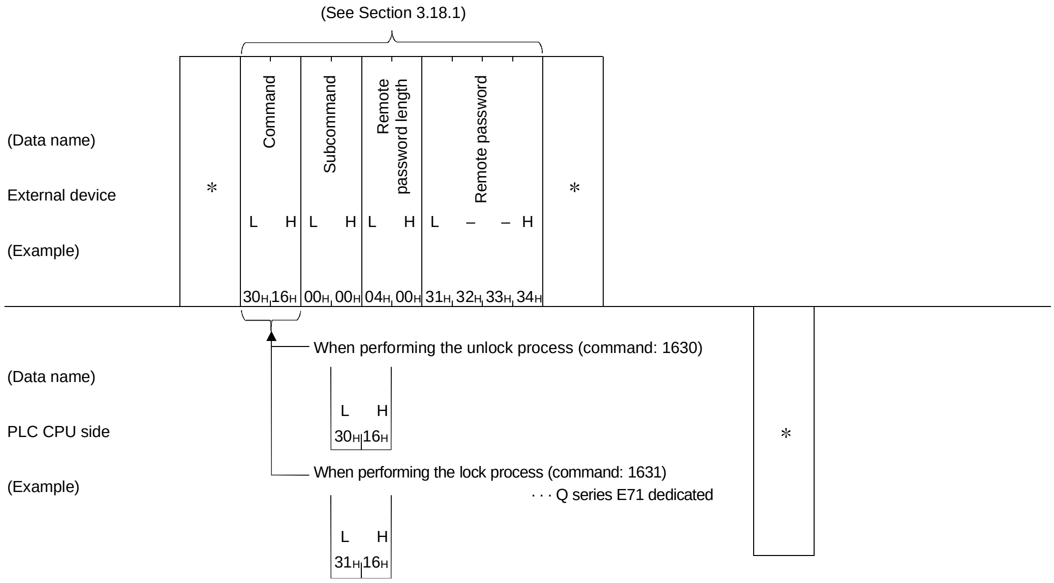
The control procedure when performing the unlock process/lock process for the remote password set for the Q series C24/E71 loaded on the QCPU, is explained using an example.
The arrangement and contents of the data items in the section marked with ∗ in the control procedure diagram will vary depending on the module used and the frame and format at the time of communication.
See the detailed explanation given in Section 3.1.

[Control procedure]

(1) Performing the unlock process/lock process in communication using ASCII code
When the remote password set in the QCPU is "1234"




(2) Performing the unlock process/lock process in communication using binary code
When the remote password set in the QCPU is "1234"



**POINT**

(1) Specify local station for the message format when performing the remote password unlock process/lock process. The unlock process/lock process cannot be done for the QCPU of other station.
(Example) Example of a message for the remote password unlock process command using the QnA compatible 4C frame (format 1)

| ENQ | Frame ID No. | | Station No. | | Network No. | | PC No. | | Request destination module I/O No. | | | | Request destination module station No. | | Self-station No. | | Command | | | | Subcommand | | | | Remote password length | | | | Remote password | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | H | L | H | L | H | L | H | L | H | – | – | L | H | L | H | L | H | – | – | L | H | – | – | L | H | – | – | L | H | – | – | L |
| | F | 8 | 0 | 0 | 0 | 0 | F | F | 0 | 3 | F | F | 0 | 0 | 0 | 0 | 1 | 6 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 4 | 1 | 2 | 3 | 4 |
| 05H | 46H | 38H | 30H | 30H | 30H | 30H | 46H | 46H | 30H | 33H | 46H | 46H | 30H | 30H | 30H | 30H | 31H | 36H | 33H | 30H | 30H | 30H | 30H | 30H | 30H | 30H | 30H | 34H | 31H | 32H | 33H | 34H |

(2) The user does not need to perform the remote password lock process for the Q series C24. The lock process is automatically performed when the modem line is disconnected.

# 4 WHEN COMMUNICATING USING THE QnA COMPATIBLE 2C FRAMES

This chapter explains the message data formats, how to designate data and the restrictions when data communication via the Q series C24 is performed with QnA compatible 2C frames using the MC protocol.

∗ The QnA compatible 2C frame has the same message format as the simplified QnA frame of the QnA serial communication module.

| POINT |
|---|
| If the Q series E71 is used for data communication, it is not necessary to read this chapter. |

## 4.1 Control Procedures and Message Formats

This section explains the control procedure and message format of each format when communicating data with QnA compatible 2C frames.

### (1) Basic format of the data communication

Four formats of the control procedures (structure, transmission/reception procedures of command messages and response messages) are available for an external device to access the PLC using the MC protocol.

Data communication can be performed in the designated format using GX Developer to set the communication protocol setting of the target interface of the Q series C24 to "1" to "4". In all formats, data communication is performed in ASCII code.

The differences between the four formats in relation to format 1 are as follows:

Format 2 : A block number is added to each message

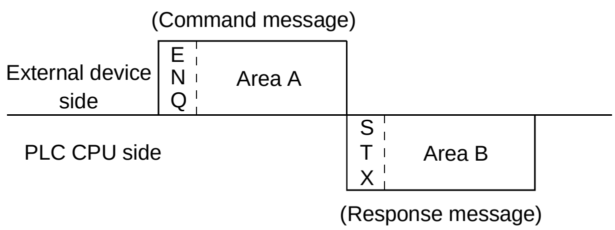
Format 3 : Each message is enclosed between STX and ETX

Format 4 : CR and LF are added to each message

### (2) How to read the MC protocol control procedure

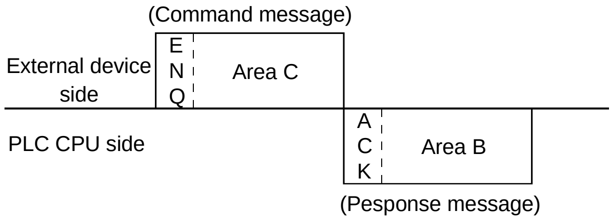
This section explains how to read the transmission data that is shown in the description of each control procedure format.

(a) When the external device reads data from the PLC




1) Area A indicates transmission from an external device to the Q series C24.
2) Area B indicates transmission from the Q series C24 to an external device.
3) The program on the external device side should be created so that the data transmissions are sequenced from left to right.
(Example: In case of area A, the data is sequentially sent from the left to the right beginning from ENQ.)

(b) When the external device writes data to PLC




1) Area C indicates transmission from an external device to the Q series C24.
2) Area B indicates transmission from the Q series C24 to an external device.
3) The program on the external device side should be created so that the data transmissions are sequenced from left to right.
(Example: In case of area C, the data is sequentially sent from the left to the right beginning from ENQ.)

4

**POINT**

When the Q series C24 receives a command message from an external device, it first completes the processing of area A in the message, transmits a response message, and then goes into the wait status.
In the wait status, the Q series C24 waits to receive the next command message or an on-demand data transmission request from the PLC CPU.

(3) Control procedure

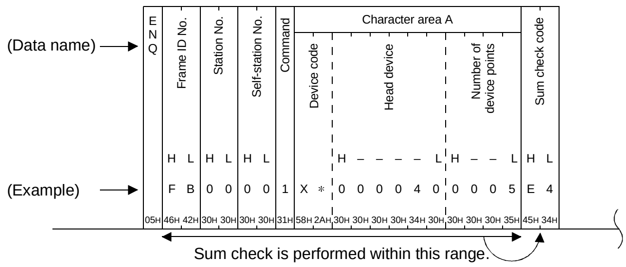
(a) Communication in format 1

1) When the external device reads data from the QCPU

(Example)

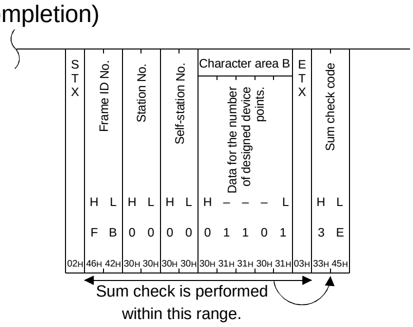
Using command 1, the external device reads five points of data from X40 to X44 in the QCPU.

External device side → PLC CPU side (Command message)

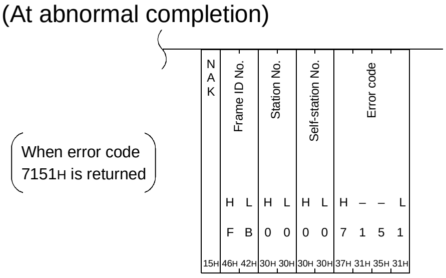



PLC CPU side → External device side (Response message)

(At normal completion)




(At abnormal completion)

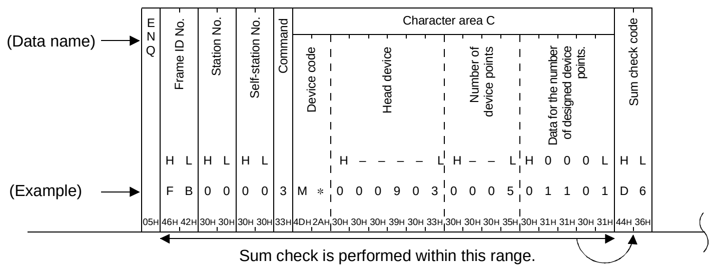



2) When the external device writes data to the QCPU

(Example)

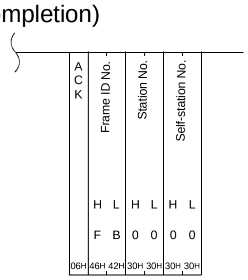
Using command 3, the external device writes five points of data to M903 to M907 in the QCPU.

External device side → PLC CPU side (Command message)

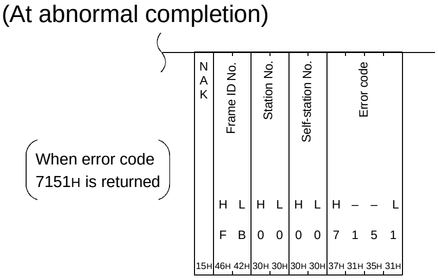



PLC CPU side → External device side (Response message)

(At normal completion)




(At abnormal completion)



(b) Communication in format 2

1) When the external device reads data from the QCPU

(Example)

Using command 1, the external device reads five points of data from X40 to X44 in the QCPU.

External device side → PLC CPU side (Command message)

| (Data name) | ENQ | Block No. | Frame ID No. | Station No. | Self-station No. | Command | Character area A: Device code | Character area A: Head device | Character area A: Number of device points | Sum check code |
|---|---|---|---|---|---|---|---|---|---|---|
| | | H L | H L | H L | H L | | | H – – – – L | H – – L | H L |
| (Example) | | 0 0 | F B | 0 0 | 0 0 | 1 | X * | 0 0 0 0 4 0 | 0 0 0 5 | 4 4 |
| | 05H | 30H 30H | 46H 42H | 30H 30H | 30H 30H | 31H | 58H 2AH | 30H 30H 30H 30H 34H 30H | 30H 30H 30H 35H | 34H 34H |

Sum check is performed within this range.

PLC CPU side → External device side (Response message)

(At normal completion)

| STX | Block No. | Frame ID No. | Station No. | Self-station No. | Character area B: Data for the number of designed device points. | ETX | Sum check code |
|---|---|---|---|---|---|---|---|
| | H L | H L | H L | H L | H – – – L | | H L |
| | 0 0 | F B | 0 0 | 0 0 | 0 1 1 0 1 | | 9 E |
| 02H | 30H 30H | 46H 42H | 30H 30H | 30H 30H | 30H 31H 31H 30H 31H | 03H | 39H 45H |

Sum check is performed within this range.

(At abnormal completion)

When error code 7151H is returned

| NAK | Block No. | Frame ID No. | Station No. | Self-station No. | Error code |
|---|---|---|---|---|---|
| | H L | H L | H L | H L | H – – L |
| | 0 0 | F B | 0 0 | 0 0 | 7 1 5 1 |
| 15H | 30H 30H | 46H 42H | 30H 30H | 30H 30H | 37H 31H 35H 31H |

2) When the external device writes data to the QCPU

(Example)

Using command 3, the external device writes five points of data to M903 to M907 in the QCPU.

External device side → PLC CPU side (Command message)

| (Data name) | ENQ | Block No. | Frame ID No. | Station No. | Self-station No. | Command | Character area C: Device code | Character area C: Head device | Character area C: Number of device points | Character area C: Data for the number of designed device points. | Sum check code |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | H L | H L | H L | H L | | | H – – – – L | H – – L | H – – – L | H L |
| (Example) | | 0 0 | F B | 0 0 | 0 0 | 3 | M * | 0 0 0 9 0 3 | 0 0 0 5 | 0 1 1 0 1 | 3 6 |
| | 05H | 30H 30H | 46H 42H | 30H 30H | 30H 30H | 33H | 4DH 2AH | 30H 30H 30H 39H 30H 33H | 30H 30H 30H 35H | 30H 31H 31H 30H 31H | 33H 36H |

Sum check is performed within this range.

PLC CPU side → External device side (Response message)

(At normal completion)

| ACK | Block No. | Frame ID No. | Station No. | Self-station No. |
|---|---|---|---|---|
| | H L | H L | H L | H L |
| | 0 0 | F B | 0 0 | 0 0 |
| 06H | 30H 30H | 46H 42H | 30H 30H | 30H 30H |

(At abnormal completion)

When error code 7151H is returned

| NAK | Block No. | Frame ID No. | Station No. | Self-station No. | Error code |
|---|---|---|---|---|---|
| | H L | H L | H L | H L | H – – L |
| | 0 0 | F B | 0 0 | 0 0 | 7 1 5 1 |
| 15H | 30H 30H | 46H 42H | 30H 30H | 30H 30H | 37H 31H 35H 31H |

(c) Communication in format 3

1) When the external device reads data from the QCPU

(Example)

Using command 1, the external device reads five points of data from X40 to X44 in the QCPU.

External device side → PLC CPU side (Command message)

| (Data name) | STX | Frame ID No. | Station No. | Self-station No. | Command | Character area A: Device code | Character area A: Head device | Character area A: Number of device points | ETX | Sum check code |
|---|---|---|---|---|---|---|---|---|---|---|
| | | H L | H L | H L | | | H – – – – L | H – – L | | H L |
| (Example) | | F B | 0 0 | 0 0 | 1 | X * | 0 0 0 0 4 0 | 0 0 0 5 | | E 7 |
| | 02H | 46H 42H | 30H 30H | 30H 30H | 31H | 58H 2AH | 30H 30H 30H 30H 34H 30H | 30H 30H 30H 35H | 03H | 45H 37H |

Sum check is performed within this range.

PLC CPU side → External device side (Response message)

(At normal completion)

| STX | Frame ID No. | Station No. | Self-station No. | (Normal completion symbol) Complete code | Character area B: Data for the number of designed device points. | ETX | Sum check code |
|---|---|---|---|---|---|---|---|
| | H L | H L | H L | | H – – – L | | H L |
| | F B | 0 0 | 0 0 | Q A C K | 0 1 1 0 1 | | 5 E |
| 02H | 46H 42H | 30H 30H | 30H 30H | 51H 41H 43H 4BH | 30H 31H 31H 30H 31H | 03H | 35H 45H |

Sum check is performed within this range.

(At abnormal completion)

When error code 7151H is returned

| STX | Frame ID No. | Station No. | Self-station No. | (Abnormal completion symbol) Complete code | Error code | ETX |
|---|---|---|---|---|---|---|
| | H L | H L | H L | | H – – L | |
| | F B | 0 0 | 0 0 | Q N A K | 7 1 5 1 | |
| 02H | 46H 42H | 30H 30H | 30H 30H | 51H 4EH 41H 4BH | 37H 31H 35H 31H | 03H |

2) When the external device writes data to the QCPU

(Example)

Using command 3, the external device writes five points of data to M903 to M907 in the QCPU.

External device side → PLC CPU side (Command message)

| (Data name) | STX | Frame ID No. | Station No. | Self-station No. | Command | Character area C: Device code | Character area C: Head device | Character area C: Number of device points | Character area C: Data for the number of designed device points. | ETX | Sum check code |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | H L | H L | H L | | | H – – – – L | H – – L | H – – – L | | H L |
| (Example) | | F B | 0 0 | 0 0 | 3 | M * | 0 0 0 9 0 3 | 0 0 0 5 | 0 1 1 0 1 | | D 9 |
| | 02H | 46H 42H | 30H 30H | 30H 30H | 33H | 4DH 2AH | 30H 30H 30H 39H 30H 33H | 30H 30H 30H 35H | 30H 31H 31H 30H 31H | 03H | 44H 39H |

Sum check is performed within this range.

PLC CPU side → External device side (Response message)

(At normal completion)

| STX | Frame ID No. | Station No. | Self-station No. | (Normal completion symbol) Complete code | ETX |
|---|---|---|---|---|---|
| | H L | H L | H L | | |
| | F B | 0 0 | 0 0 | Q A C K | |
| 02H | 46H 42H | 30H 30H | 30H 30H | 51H 41H 43H 4BH | 03H |

(At abnormal completion)

When error code 7151H is returned

| STX | Frame ID No. | Station No. | Self-station No. | (Abnormal completion symbol) Complete code | Error code | ETX |
|---|---|---|---|---|---|---|
| | H L | H L | H L | | H – – L | |
| | F B | 0 0 | 0 0 | Q N A K | 7 1 5 1 | |
| 02H | 46H 42H | 30H 30H | 30H 30H | 51H 4EH 41H 4BH | 37H 31H 35H 31H | 03H |

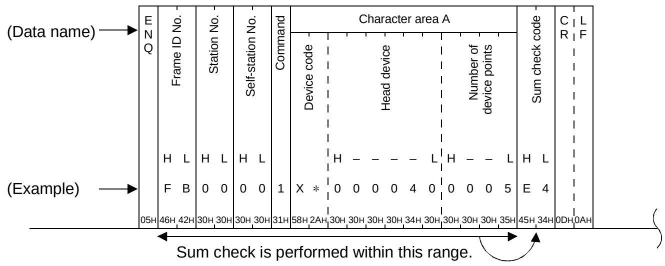
(d) Communication in format 4

1) When the external device reads data from the QCPU

(Example)

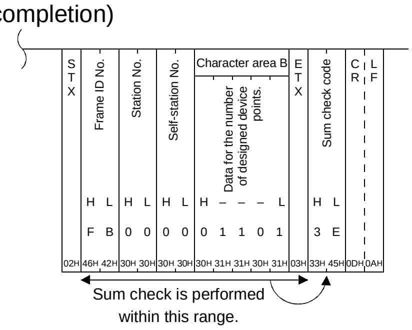
Using command 1, the external device reads five points of data from X40 to X44 in the QCPU.

External device side → PLC CPU side (Command message)

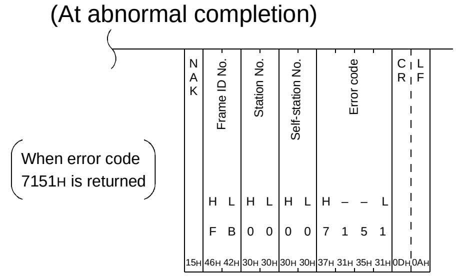



PLC CPU side → External device side (Response message)

(At normal completion)




(At abnormal completion)

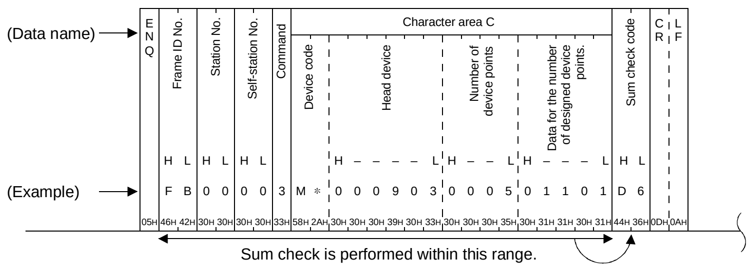



2) When the external device writes data to the QCPU

(Example)

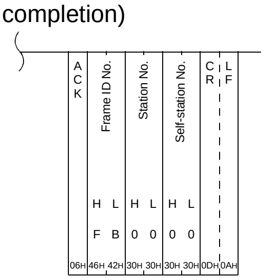
Using command 3, the external device writes five points of data to M903 to M907 in the QCPU.

External device side → PLC CPU side (Command message)

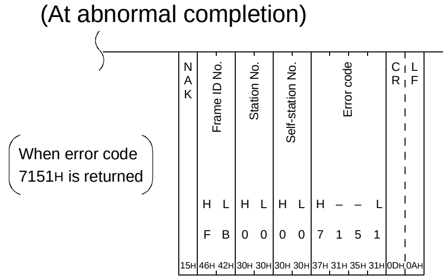



PLC CPU side → External device side (Response message)

(At normal completion)




(At abnormal completion)



## 4.2 Contents of the Data Designation Items

Each frame format contains several data items for data transmission, as defined by the control procedure. Some of the data items of the QnA compatible 2C, 3C and 4C frames are common, but not all. The following describes the data items unique to the QnA compatible 2C.

**POINT**

The data items other than those mentioned in this section are designated in the same way as when communicating using QnA compatible 3C/4C frames.
For more details, see Section 3.1.6 of this manual.
For the "message wait" time for data communication, see REMARK in Section 3.1.6 (11).

### (1) Frame ID number

"FB" should be designated as the frame ID number when communicating in QnA compatible 2C frames.

### (2) Commands

The function to be executed is designated by one of the commands from "1 to 9" (see Section 4.3), which is transmitted as a one-digit ASCII code.

### (3) Character areas (areas A, B and C)

The character areas use the same designation method and contents as those for the communication using the QnA compatible 3C/4C frames. They are determined by the commands transmitted from the external device.
The following table shows the correspondence between the commands for the QnA compatible 2C and QnA compatible 3C/4C frames for which the designation method and contents of character areas are the same.

| Command | | QnA compatible 2C frame command | QnA compatible 3C/4C frame commands corresponding to the command at the left | |
|---|---|---|---|---|
| | | | Command | Subcommand |
| Batch read | Bit units | 1 | 0401 | 0001 |
| | Word units | 2 | 0401 | 0000 |
| Batch write | Bit units | 3 | 1401 | 0001 |
| | Word units | 4 | 1401 | 0000 |
| Random read | Word units | 5 | 0403 | 0000 |
| Test (Random write) | Bit units | 6 | 1402 | 0001 |
| | Word units | 7 | 1402 | 0000 |
| Monitor data registration | Word units | 8 | 0801 | 0000 |
| Monitor | Word units | 9 | 0802 | 0000 |

Designate the character areas for the command to be used according to the instruction given in Section 3.3 of this manual.

## 4.3 List of Commands and Functions for QnA Compatible 2C Frames

The following table outlines commands and functions that are used when communicating in QnA compatible 2C frames.
All the commands are for reading and writing data from/to the device memory of the PLC CPU on which the Q series C24 is loaded using the MC protocol.

| Function | | Command | Processing | Number of points processed per communication | PLC CPU status (∗1) | | |
|---|---|---|---|---|---|---|---|
| | | | | | During STOP | During RUN | |
| | | | | | | Write allow setting | Write prohibit setting |
| Batch read | Bit units | 1 | Reads bit devices in 1-point units. (1 point = 1 bit) | 7904 points | ○ | ○ | ○ |
| | Word units | 2 | Reads bit devices in 1-point units. (1 point = 16 bits) | | | | |
| | | | Reads word devices in 1-point units. | 960 points | | | |
| Batch write (∗2) | Bit units | 3 | Writes to bit devices in 1-point units. (1 point = 1 bit) | 7904 points | ○ | ○ | × |
| | Word units | 4 | Writes to bit devices in 1-point units. (1 point = 16 bits) | 960 points | | | |
| | | | Writes to word devices in 1-point units. | | | | |
| Random read | Word units | 5 | Reads bit devices in 1-point units by randomly designating the devices and device numbers. (1 point = 16 bits) | 192 points | ○ | ○ | ○ |
| | | | Reads word devices in 1-point units by randomly designating the devices and device numbers. | | | | |
| Test (∗2) (random write) | Bit units | 6 | Writes to bit devices in 1-point units by randomly designating the devices and device numbers. (1 point = 1 bit) | 188 points | ○ | ○ | × |
| | Word units | 7 | Writes to bit devices in 1-point units by randomly designating the devices and device numbers. (1 point = 16 bits) | 1920 points (∗4) | | | |
| | | | Writes to word devices in 1-point units by randomly designating the devices and device numbers. | | | | |
| Monitor data registration (∗3) | Word units | 8 | Registers bit devices to be monitored in 1-point units. (1 point = 16 bits) | 192 points | ○ | ○ | ○ |
| | | | Registers word devices to be monitored in 1-point units. | | | | |
| Monitor (∗3) | Word units | 9 | Monitors devices with monitoring data registered. | Number of registrations points | | | |

∗1 The write allow/prohibit designation to the PLC CPU during RUN is set by the transmission setting of GX Developer.
∗2 If there is system protection on the QCPU that executes the command, an error occurs and a NAK message is returned.
∗3 The procedure for monitoring is the same as that for monitoring during communication in QnA compatible 3C/4Cframes.
∗4 Designate the number of points to be processed per communication within the following range.
1920 > = (number of word access points × 12) + (number of double word access point × 14) > = 1

## 4.4 Precautions on the Data Communication

Certain precautions should be observed while communicating data using QnA compatible 2C frames.

(1) Some precautions are common to communication in this format and in the QnA compatible 3C/4C frame formats. See Sections 3.1.4 and 3.1.5 in this manual for more details.

(2) The monitor conditions for reading data, which can be designated when communicating in the QnA compatible 3C/4C frames, cannot be designated when communicating in the QnA compatible 2C frame.
In addition, the extension setting for reading from and writing to the device memory described in Appendix 1 is not allowed.

(3) When the data communication is performed using the QnA compatible 2C frame, the number of points and the range for reading and writing for each command are the same as those of the corresponding QnA compatible 3C/4C frame command. (The designation method and contents of the character areas are the same.)
The QnA compatible 3C/4C frame commands that correspond to the QnA compatible 2C frame commands are listed in Section 4.2.

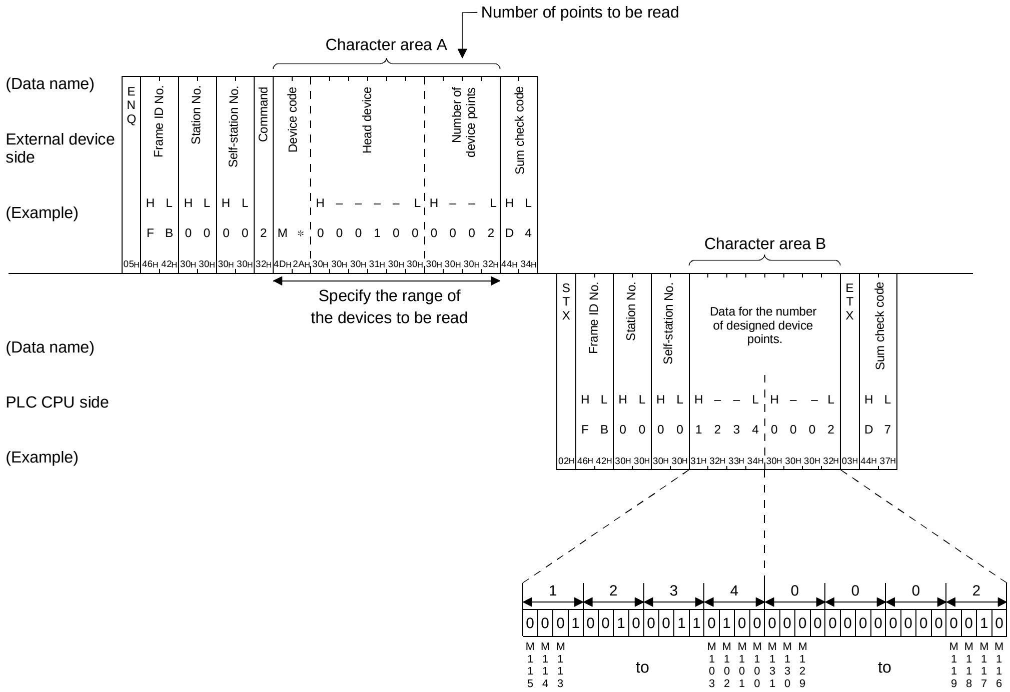
(4) Access is allowed to stations on which the Q series C24 is installed that are connected to an external device (including Q/QnACPU stations connected by multidrop link).

## 4.5 Example of Data Communication Using QnA Compatible 2C Frames

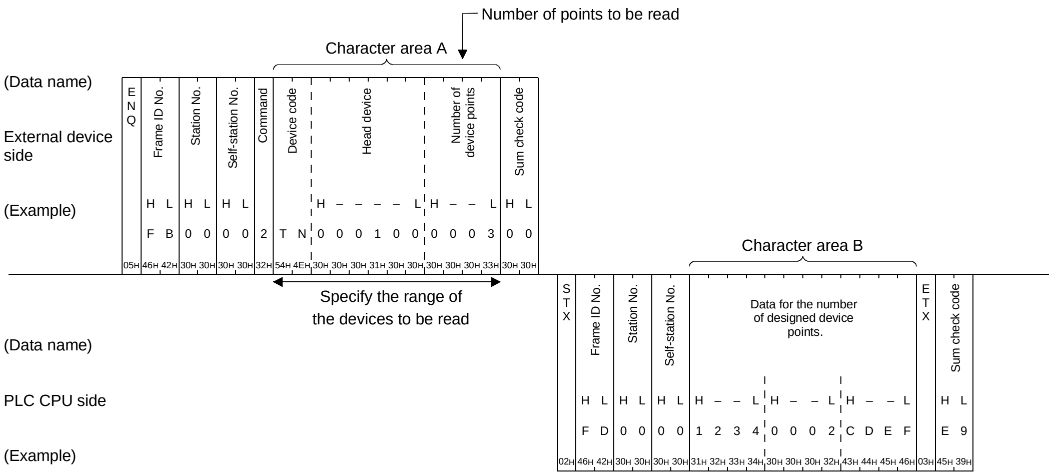
The following shows examples of the control procedure for data communication using format 1 QnA compatible 2C frames.
See Section 4.1(3) for examples of the control procedures for commands 1 and 3.

### (1) Batch read in word units (command: 2)

(a) Reading two points (32 bits) from internal relays M100 to M131




(b) Reading the current values of three points from timer T100 to T102

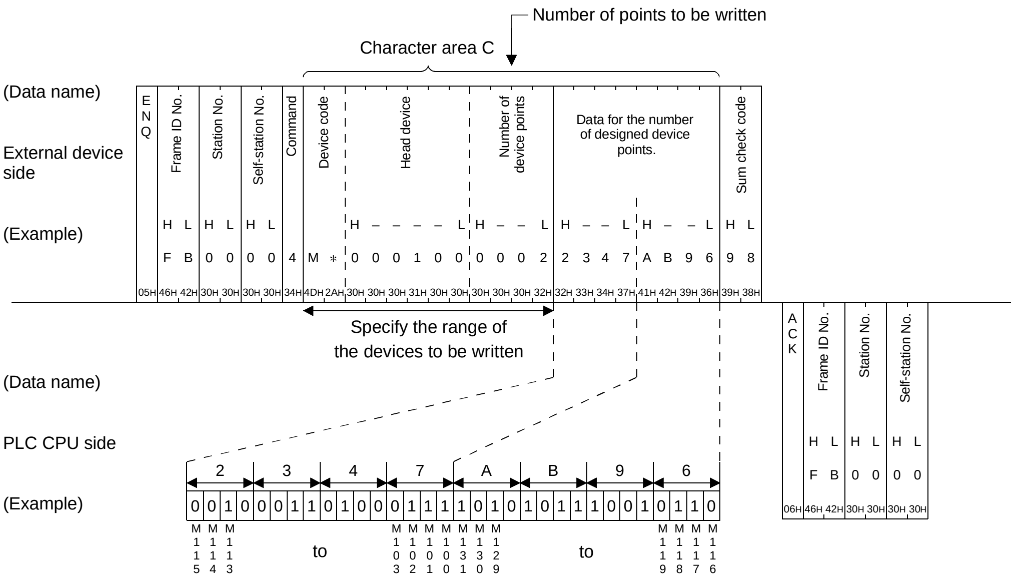



Indicate the current value of T100, 1234H hexadecimal (4660 decimal)
Indicate the current value of T101, 0002H hexadecimal (2 decimal)
Indicate the current value of T102, CDEFH hexadecimal (-12817 decimal)

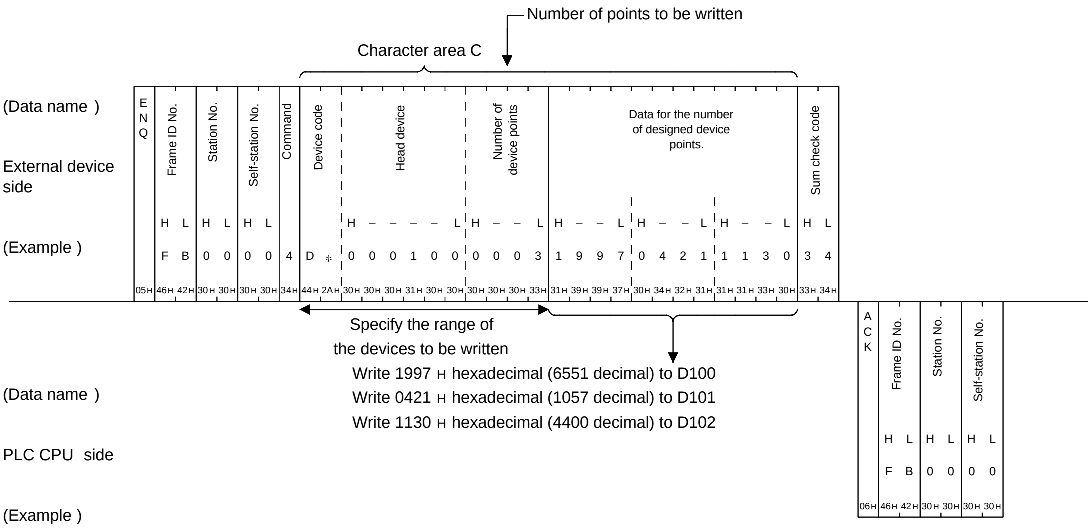
* The designation method and contents for the character areas of command 2 are the same as those of the QnA compatible 3C/4C frame command 0401 (subcommand 0000).

## (2) Batch write in word units (command: 4)

### (a) Writing two points (32 bits) to internal relays M100 to M131




### (b) Writing three points to D100 to D102

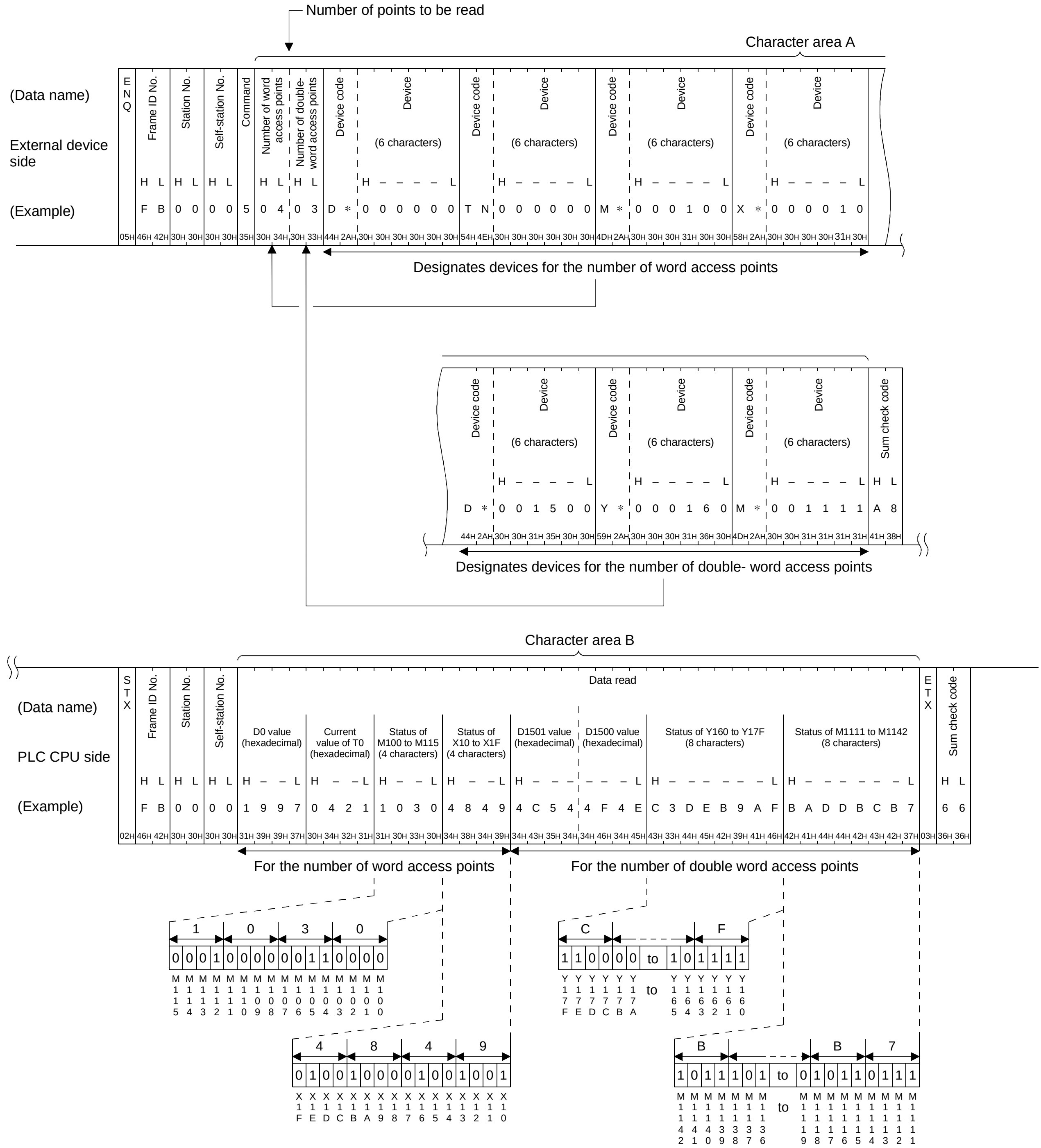



* The designation method and contents for the character areas of command 4 are the same as those of the QnA compatible 3C/4C frame command 1401 (subcommand 0000).

### (3) Random read in word units (command: 5)

Reading the following device memory

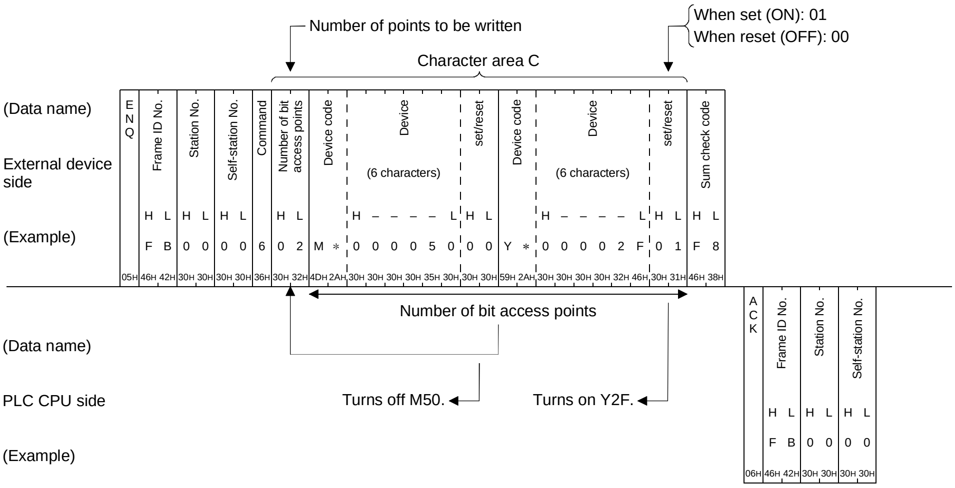
- Word access : D0, T0, M100 to M115, X10 to X1F
- Double word access : D1500 to D1501, Y160 to Y17F, M1111 to M1142




✻ The designation method and contents for the character areas of command 5 are the same as those of the QnA compatible 3C/4C frame command 0403 (subcommand 0000).

## (4) Test in bit units (command: 6)

Turning off the internal relay M50 and turning on the output relay Y2F

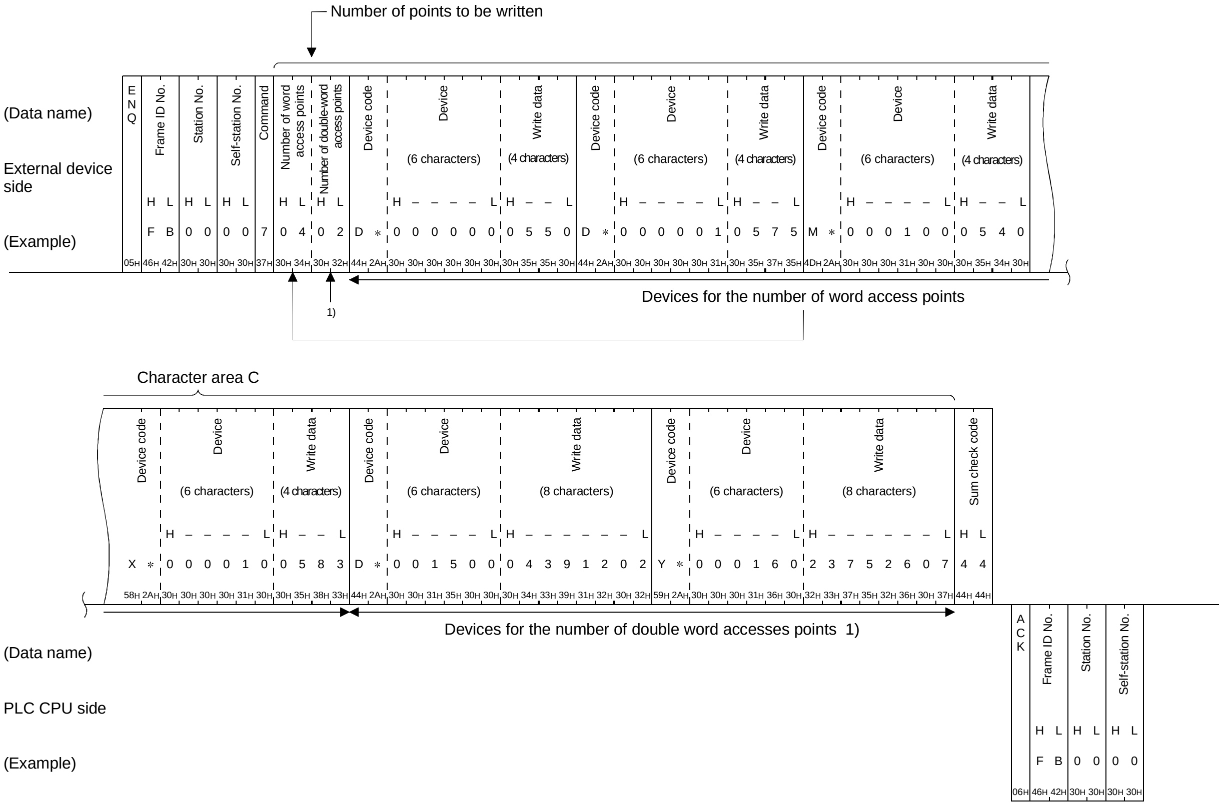



✻ The designation method and contents for the character areas of command 6 are the same as those of the QnA compatible 3C/4C frame command 1402 (subcommand 0001).

## (5) Test in word units (command: 7)

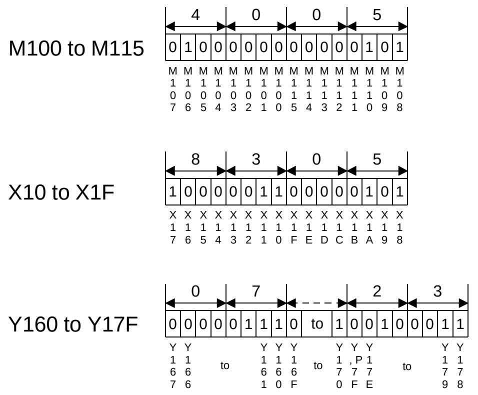
Writing to the following device memory

- Word access : D0, D1, M100 to M115, X10 to X1F
- Double word access : D1500 to D1501, Y160 to Y17F




✻ The designation method and contents for the character areas of command 7 are the same as those of the QnA compatible 3C/4C frame command 1402 (subcommand 0000).

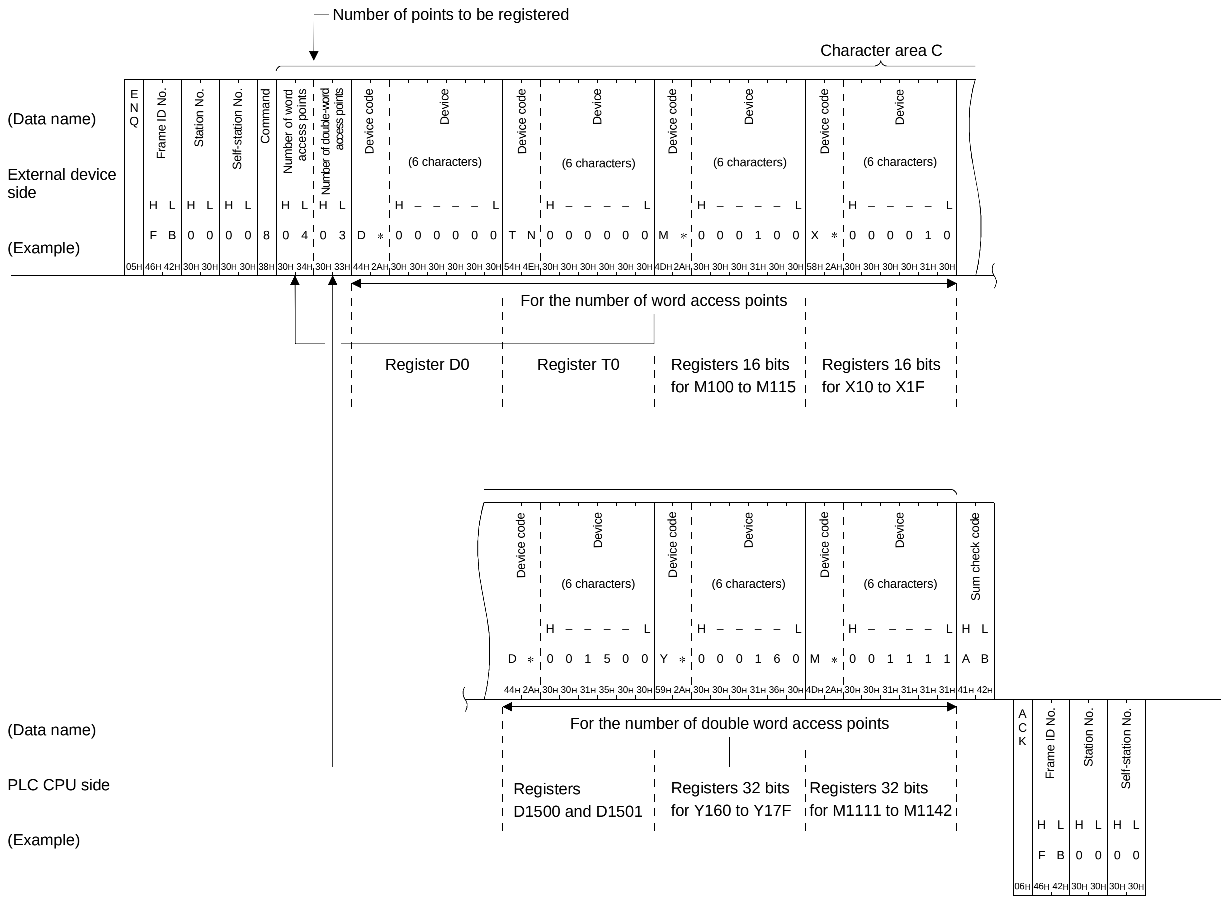
The following shows the correspondence between the data to be tested by each word unit and the bit device:




## (6) Monitor data registration in word units (command: 8)

Registering monitor data in the following device memory

- Word access : D0, T0, M100 to M115, X10 to X1F
- Double word access : D1500 to D1501, Y160 to Y17F, M1111 to M1142

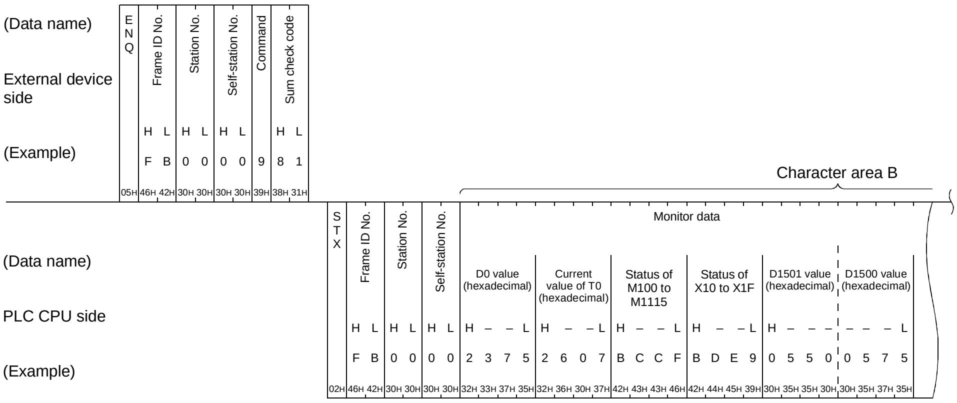


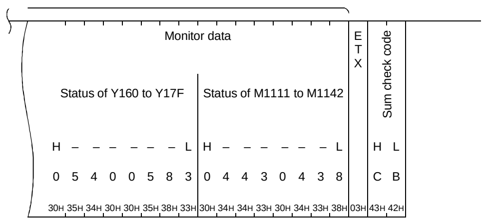

✻ The designation method and contents for the character areas of command 8 are the same as those of the QnA compatible 3C/4C frame command 0801 (subcommand 0000).

(7) Monitor in word units (command: 9)

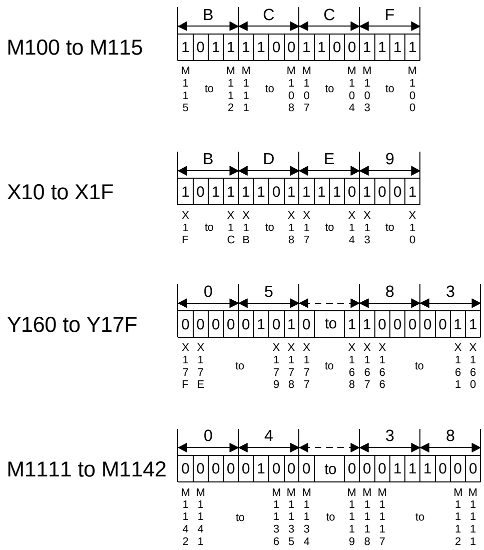
Monitoring the following device memory for which monitoring data has been registered
(Device memory with monitor data registered)

- Word access : D0, T0, M100 to M115, X10 to X1F
- Double word access : D1500 to D1501, Y160 to Y17F, M1111 to M1142




The following shows the correspondence between the bit device whose data are registered to be monitored and the data actually read:




✻ The designation method and contents for the character areas of command 9 are the same as those of the QnA compatible 3C/4C frame command 0802 (subcommand 0000).

# 5 WHEN COMMUNICATING USING THE A COMPATIBLE 1C FRAMES

This chapter explains the message data formats, how to designate data and the restrictions when data communication via the Q series C24 is performed with QnA compatible 1C frames using the MC protocol.

**POINT**

If the Q series E71 is used for data communication, it is not necessary to read this chapter.

## 5.1 Control Procedures and Message Formats

This section explains the control procedure and message format of each format when communicating data using A compatible 1C frames.
Data communication using the A compatible 1C frames is equivalent to the communication function using the dedicated protocol supported by the A series computer link modules and only the commands shown in Section 5.1.5 can be used. In all formats, data communication is performed in ASCII code.

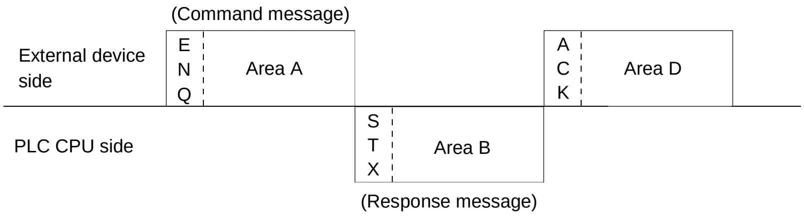
### 5.1.1 How to read the control procedure and command reference sections

This section explains how to read the control procedure and message diagrams in the Section 5.2 to 5.4.

(1) How to read the MC protocol control procedure

This section explains how to read the transmission data that is shown in the description of each control procedure format in Sections 5.2 and succeeding sections.

(a) When the external device reads data from the PLC



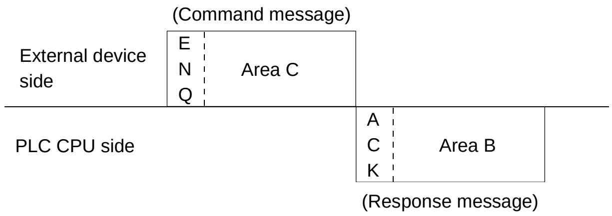

1) Area A indicates transmission from an external device to the Q series C24.
2) Area B indicates transmission from the Q series C24 to an external device.
3) The program on the external device side should be created so that the data transmissions are sequenced from left to right.
(Example: In case of area A, the data is sequentially sent from the left to the right beginning from ENQ.)

**POINT**

When an external device reads data from the PLC CPU, it does not need to send "area D" in the diagram above. (It may be omitted.)
The Q series C24 sends "area B" in the diagram above to the external device, then completes processing the read request in "area A" after which it waits for receiving the next "area A" or "area C" data read request described below (wait status).
Thus, the Q series C24 performs no operation even if it receives the data communicated in the last area, "area D."

5

(b) When the external device writes data to PLC




1) Area C indicates transmission from an external device to the Q series C24.
2) Area B indicates transmission from the Q series C24 to an external device.
3) The program on the external device side should be created so that the data transmissions are sequenced from left to right.
(Example: In case of area C, the data is sequentially sent from the left to the right beginning from ENQ.)

**POINT**

When the Q series C24 receives a command message from an external device, it first completes the processing of area A/C in the message, transmits a response message, and then goes into the wait status.
In the wait status, the Q series C24 waits to receive the next command message or an on-demand data transmission request from the PLC CPU.

## (2) Basic format of the data communication

Four formats of the control procedures (structure, transmission/reception procedures of command messages and response messages) are available for an external device to access the PLC with A compatible 1C frames using the MC protocol.
Data communication can be performed in the designated format using GX Developer to set the communication mode of the target interface of the Q series C24 to "1" to "4".
The differences between the four formats in relation to format 1 are as follows:

Format 2 : A block number is added to each message
Format 3 : Each message is enclosed between STX and ETX
Format 4 : CR and LF are added to each message

### 5.1.2 Control procedures and message formats

This section illustrates the control procedure and message format of each command when data communication is performed in each format of the A compatible 1C frames.
The □ section shown in the message explanation diagram of this section are items common to all command and corresponds to the ＊ portion of the message explanation diagrams indicated in sections 5.2.2 and after of this chapter.
See Section 5.1.3 regarding the content and specification method of the data found in □.

**POINT**

It shows the form 1 to 4 (diagram (1) to (4)) for the QnA compatible 4C frame shown in this chapter.

(a) When sum check is enabled, the sum check is performed only for the characters in the areas indicated with (＊1) in the illustrations.

(b) The contents of "Character area A," "Character area B," and "Character area C" in the diagrams depend on the processing contents. See the description of each command for more information. The contents of each character area are the same for all four formats.

(c) When the external device reads data from the PLC, the transmission of messages in the areas marked with (＊2) in the diagram may be omitted.

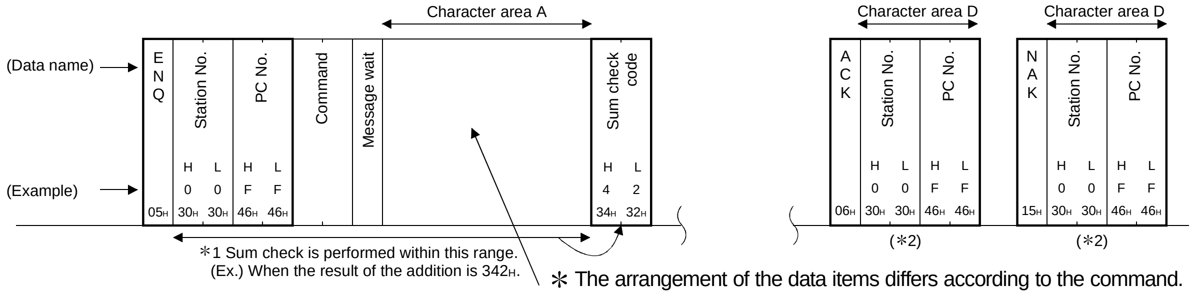
(d) When the external device and PLC CPU system configuration is m: n, the control procedure format 3 cannot be used for data communication.

(1) Communication in format 1

(a) When reading data from the local station PLC CPU on the external device side

External device side → PLC CPU side (Command message)

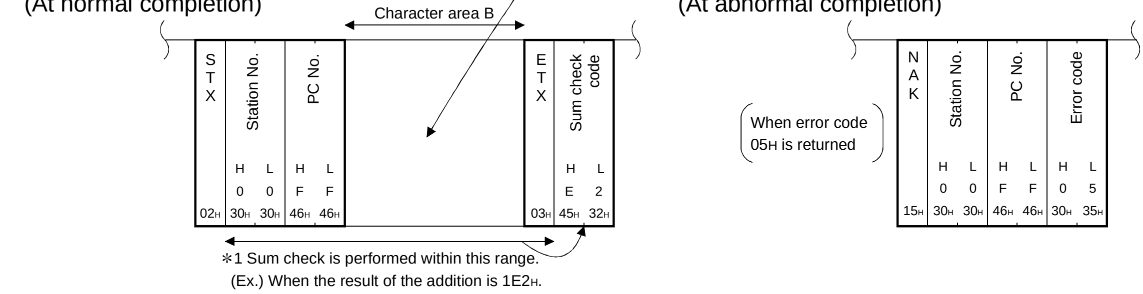
(At normal completion) (At abnormal completion)




* The arrangement of the data items differs according to the command.
See the sections explaining commands found in Section 5.2.2 and after for details

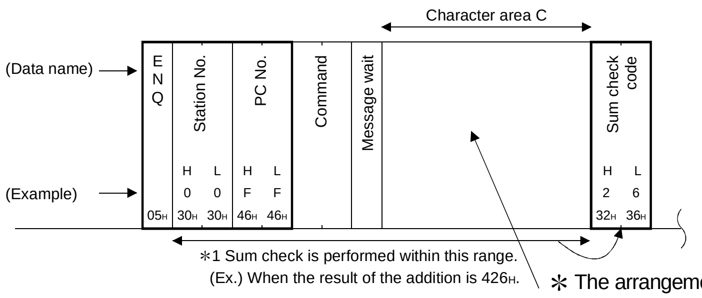
PLC CPU side → External device side (Response message)

(At normal completion) (At abnormal completion)




(b) When writing data to the local station PLC CPU from an external device side

External device side → PLC CPU side (Command message)




* The arrangement of the data items differs according to the command .
See the sections explaining commands found in Section 5.2.2 and after for details

PLC CPU side → External device side (Response message)

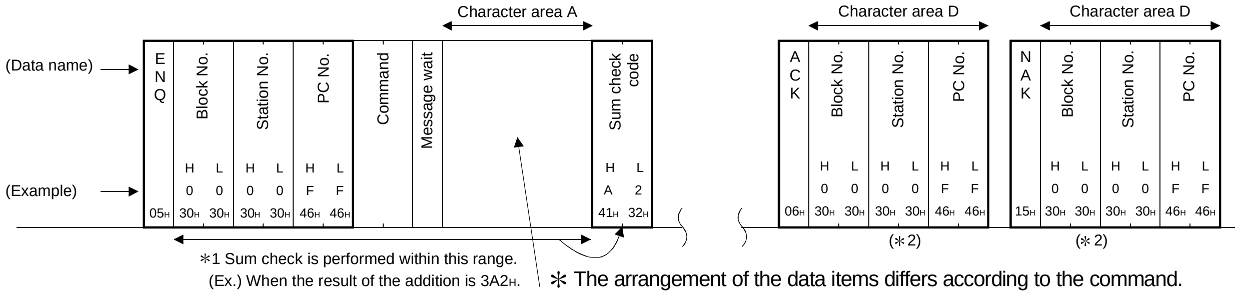
(At normal completion) (At abnormal completion)

(2) Communication in format 2

(a) When reading data from the local station PLC CPU on the external device side

External device side → PLC CPU side (Command message)

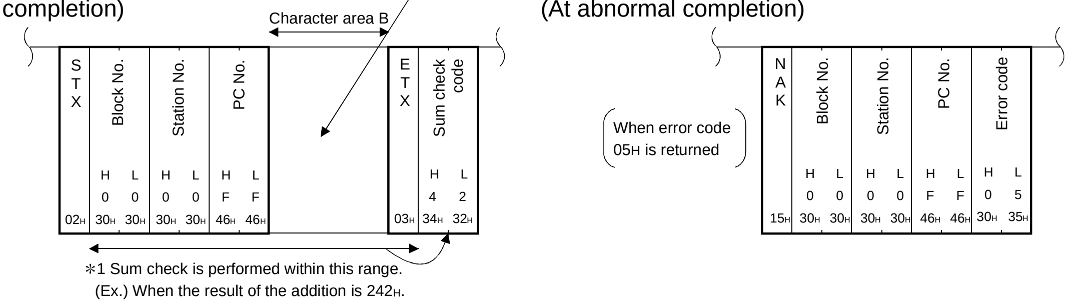
(At normal completion) (At abnormal completion)




✻ The arrangement of the data items differs according to the command.
See the sections explaining commands found in Section 5.2.2 and after for details

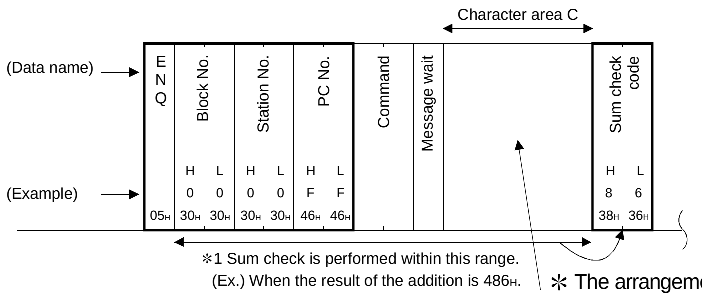
PLC CPU side → External device side (Response message)

(At normal completion) (At abnormal completion)




(b) When writing data to the local station PLC CPU from an external device side

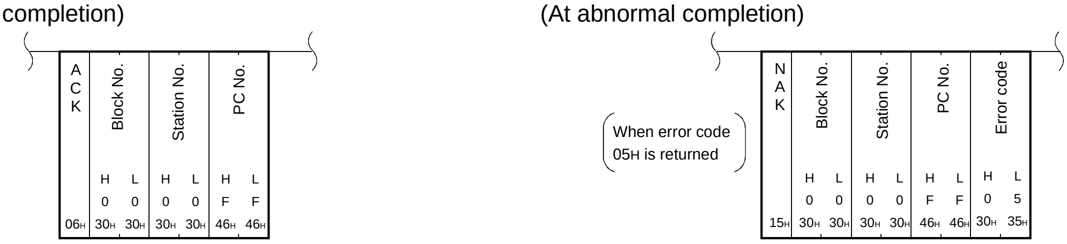
External device side → PLC CPU side (Command message)




✻ The arrangement of the data items differs according to the command.
See the sections explaining commands found in Section 5.2.2 and after for details

PLC CPU side → External device side (Response message)

(At normal completion) (At abnormal completion)

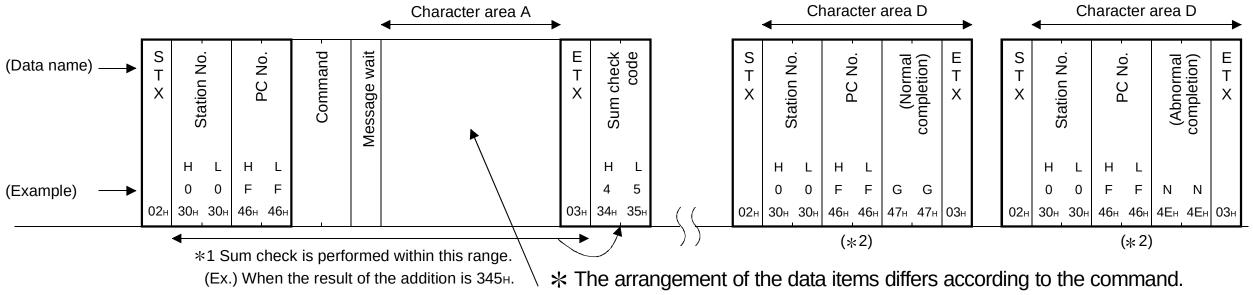


### (3) Communication in format 3

#### (a) When reading data from the local station PLC CPU on the external device side

External device side → PLC CPU side (Command message)

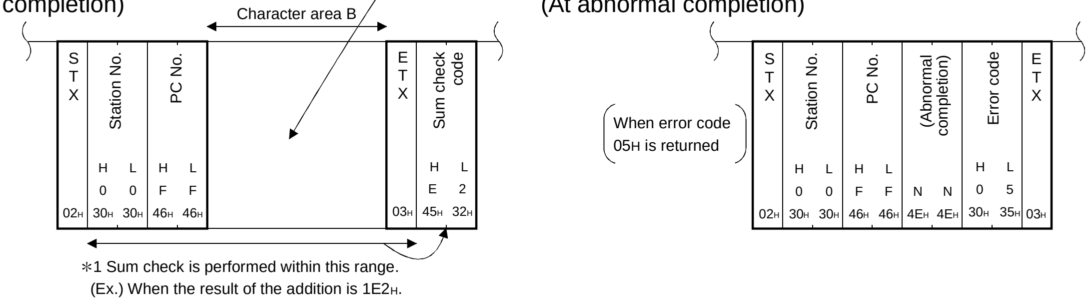
(At normal completion) (At abnormal completion)




* The arrangement of the data items differs according to the command.
See the sections explaining commands found in Section 5.2.2 and after for details

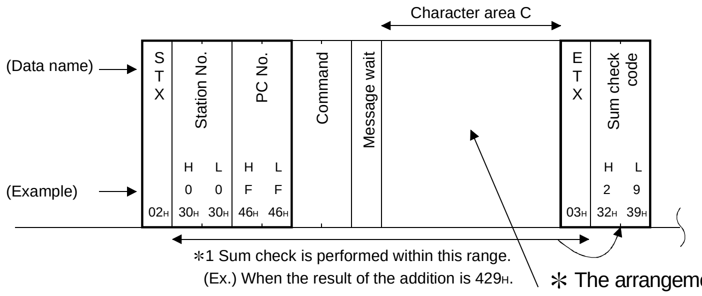
PLC CPU side → External device side (Response message)

(At normal completion) (At abnormal completion)




#### (b) When writing data to the local station PLC CPU from an external device side

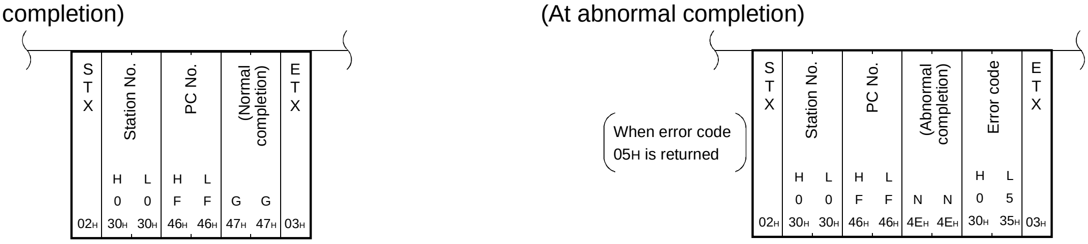
External device side → PLC CPU side (Command message)




* The arrangement of the data items differs according to the command.
See the sections explaining commands found in Section 5.2.2 and after for details

PLC CPU side → External device side (Response message)

(At normal completion) (At abnormal completion)

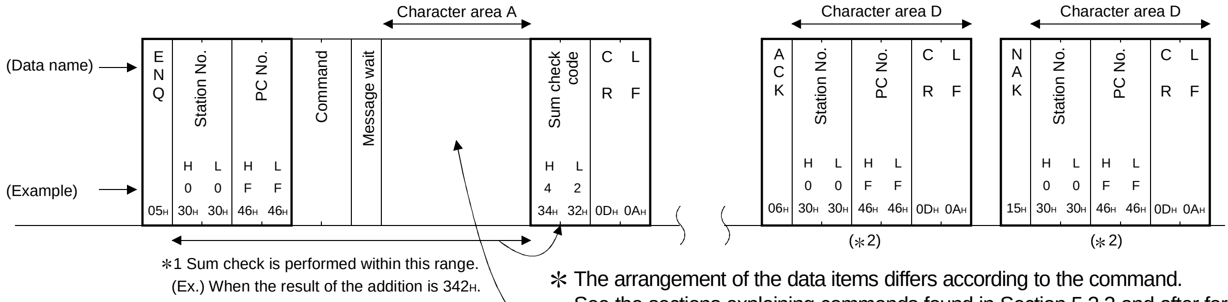


(4) Communication in format 4

(a) When reading data from the local station PLC CPU on the external device side

External device side → PLC CPU side (Command message)

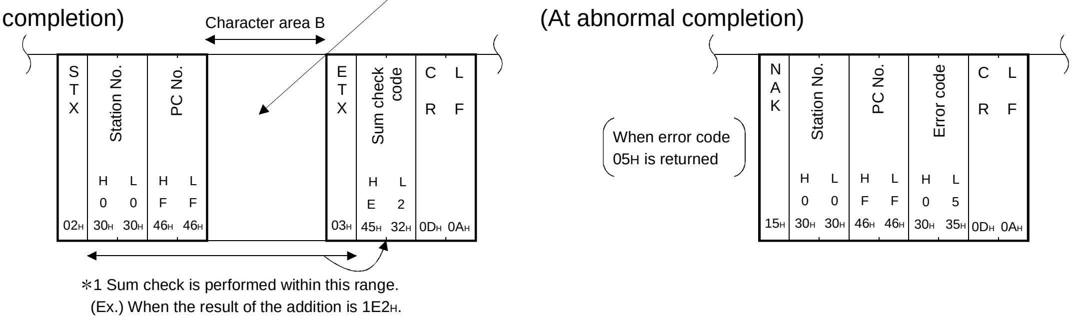
(At normal completion) (At abnormal completion)




∗ The arrangement of the data items differs according to the command.
See the sections explaining commands found in Section 5.2.2 and after for details

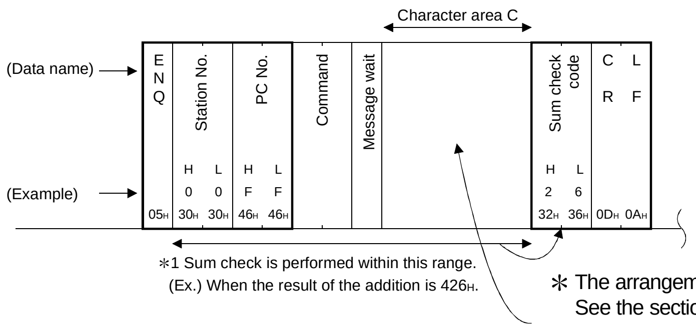
PLC CPU side → External device side (Response message)

(At normal completion) (At abnormal completion)




(b) When writing data to the local station PLC CPU from an external device side

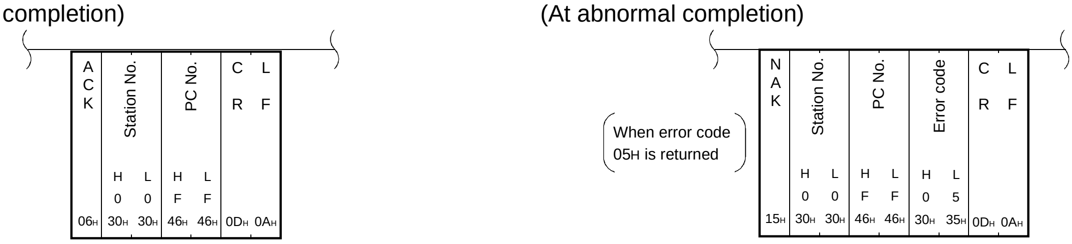
External device side → PLC CPU side (Command message)




∗ The arrangement of the data items differs according to the command.
See the sections explaining commands found in Section 5.2.2 and after for details

PLC CPU side → External device side (Response message)

(At normal completion) (At abnormal completion)



### 5.1.3 Contents of data designation items of A compatible 1C frames

This section explains the data items when communicating data via the Q series C24 with A compatible 1C frames whose contents and method of designation are different from those of QnA compatible 3C/4C frames.

#### (1) Control code

The following table outlines the control codes.

| Symbol | Code (hexadecimal) | Contents | Symbol | Code (hexadecimal) | Contents | Symbol | Code (hexadecimal) | Contents |
|---|---|---|---|---|---|---|---|---|
| NUL | $00_H$ | Null | ENQ | $05_H$ | Enquiry | CR | $0D_H$ | Carriage Return |
| STX | $02_H$ | Start of Text | ACK | $06_H$ | Acknowledge | NAK | $15_H$ | Negative Acknowledge |
| ETX | $03_H$ | End of Text | LF | $0A_H$ | Line Feed | G | $47_H$ | Good |
| EOT | $04_H$ | End of Transmission | CL | $0C_H$ | Clear | N | $4E_H$ | No Good |

(a) Null codes ($00_H$) are ignored in all messages. That means that even if there are Null codes in a message, they are not processed.

(b) In the format 3 control procedure, the control code "GG" indicates ACK (normal completion) and "NN" indicates NAK (abnormal completion).

(c) EOT and CL are codes for initializing the transmission sequence for data communication in the MC protocol and instruct the Q series C24 to go into the wait status to receive commands from an external device.
When the Q series C24 receives the EOT or CL command, it proceeds as follows (✻1):

- Terminates read/write processing performed for the PLC CPU by a request from an external device.
  In this case, the Q series C24 does not send a response message (area B shown in Section 5.1.2) for the previously received command.
- Initializes the transmission sequence of the MC protocol and enters into the command receive wait status.
- There is no response message for the received EOT and CL commands. (Nothing is sent to the external device.)
- When EOT or CL is received while the on-demand function (a function that sends data from the PLC CPU to an external device described in Section 3.11) is being executed, the Q series C24 terminates the on-demand data transmission to the external device.

When performing the following at an external device, send the EOT or CL command to the Q series C24 using the message format shown below, depending on the format used.
The following shows the message structure when sending EOT or CL.

1) Cancel a read/write request issued by the immediately preceding command. (When a write request was issued, and data was already written to the PLC CPU, the write request cannot be canceled.)
2) Before commands are sent, place the Q series C24 in the command receive wait status.
3) If data communications cannot be carried out normally, place the Q series C24 in the same status as when it was started up.

| In case of formats 1 to 3 | In case of format 4 |
|---|---|
| External device side: CL or EOT<br>PLC CPU side | External device side: CL CR LF or EOT CR LF<br>PLC CPU side |

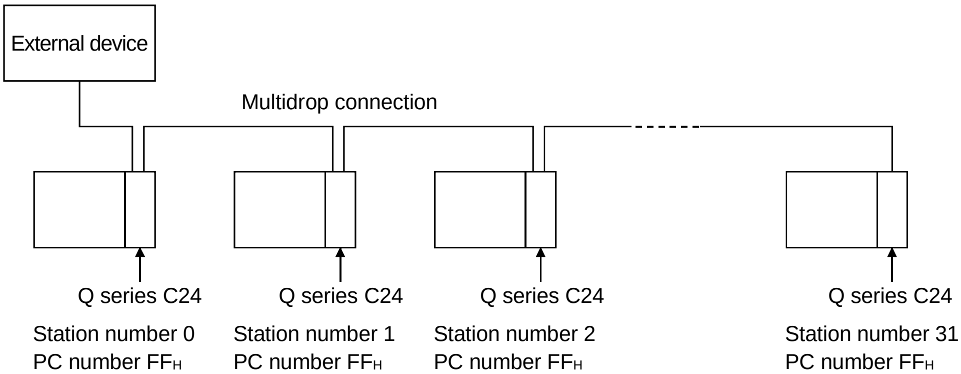
✻ When EOT or CL is sent, only the data shown in the figures is sent.
The station number, PC number, etc. do not have to be sent.

(2) Block number

Block numbers are arbitrary numbers given to the corresponding message by an external device and used as data reference numbers.
The contents and method of designation are the same as for communication using QnA compatible 3C/4C frames. For more details, see Section 3.1.6.

(3) Station number

The station numbers are set with GX Developer and used to identify to which station's Q series C24 an external device accesses.
(Example)




The contents and method of designation are the same as in communication in QnA compatible 3C/4C. For more details, see Section 3.1.6.

(4) PC number

PC numbers are used to identify which PLC on MELSECNET/H, MELSECNET/10 to be accessed.
PC numbers are designated as FF$_H$ or, for stations connected by a network module, in the station number range 00$_H$ to 40$_H$. They are obtained by converting the station numbers to 2-digit ASCII code (hexadecimal).

| | External device access station | PLC number designated by external device |
|---|---|---|
| 1 | Stations connected to an external device (local station) | FF$_H$ |
| 2 | Stations connected to an external connection station by multidrop link | FF$_H$ |
| 3 | Station on the MELSECNET/H, MELSECNET 10 (excluding 1, 2 above) (＊1) | 01$_H$ to 40$_H$ (1 to 64): Station number of access station |
| 4 | Remote master station on the MELSECNET/H (When an external device is connected to the Q series C24 in the remote I/O station) | 01$_H$ to 40$_H$ (1 to 64): Station number of access station ＊ 00$_H$ can be designated when accessing the control station. |

＊1 The module on the network designated in the "Valid module during other station access" setting by a network module (Ethernet, MELSECNET/H, MELSECNET/10) is accessed.
However, the "Valid module during other station access" set in the MELSECNET/H remote I/O station is disregarded and the other station specified by the PC number of the MELSECNET/H remote I/O station is accessed.

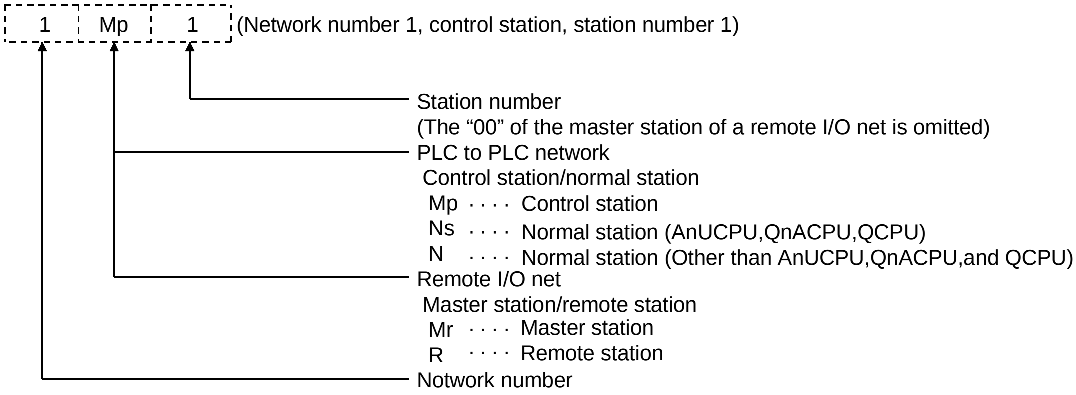
POINT

(1) The "Network parameter" must be set in the station loaded on the network module (Ethernet, MELSECNET/H, MELSECNET/10).
(2) The "Valid module during other station access" setting is required in the station loaded on the network module when accessing the other station's PLC CPU using the A compatible 1C frame.

[Example of PC number designation]

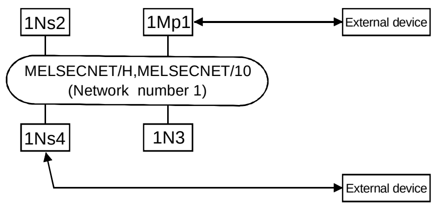
(Meaning of the symbols for each station shown in the diagrams)

- Network system (MELSECNET/H, MELSECNET/10)




In case of an PLC to PLC network net of MELSECNET/H, MELSECNET/10

- MELSECNET/H, MELSECNET/10 two-tier system

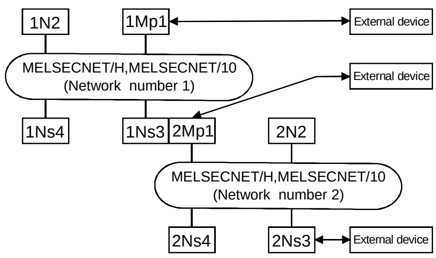



PC number to be designated

| Station connected to external device | PLC CPU accessed by an external device | | | |
|---|---|---|---|---|
| | 1Mp1 | 1Ns2 | 1N3 | 1Ns4 |
| 1Mp1 | FF | 02 | 03 | 04 |
| 1Ns4 | 01 | 02 | 03 | FF |

×: Not accessible

- MELSECNET/H, MELSECNET/10 hierarchical system



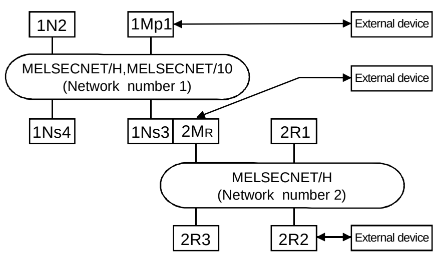

PC number to be designated

| Station connected to external device | PLC CPU accessed by an external device | | | | | | |
|---|---|---|---|---|---|---|---|
| | 1Mp1 | 1N2 | 1Ns3/ 2Mp1 | 1Ns4 | 2N2 | 2Ns3 | 2Ns4 |
| 1Mp1 | FF | 02 | 03 | 04 | × | | |
| 1Ns3/ 2Mp1 | 01 | 02 | FF | 04 | 02 | 03 | 04 |
| 2Ns3 | × | | 01 | × | 02 | FF | 04 |

×: Not accessible

In case of a MELSECNET/H, MELSECNET/10 remote I/O net

- MELSECNET/H, MELSECNET/10 hierarchical system




PC number to be designated

| Station connected to external device | PLC CPU accessed by an external device | | | | | | |
|---|---|---|---|---|---|---|---|
| | 1Mp1 | 1N2 | 1Ns3/ 2MR | 1Ns4 | 2R1 | 2R2 | 2R3 |
| 1Mp1 | FF | 02 | 03 | 04 | × | | |
| 1Ns3/ 2MR | 01 | 02 | FF | 04 | 01 | 02 | 03 |
| 2R2 | × | | 00 | × | 01 | FF | 03 |

×: Not accessible

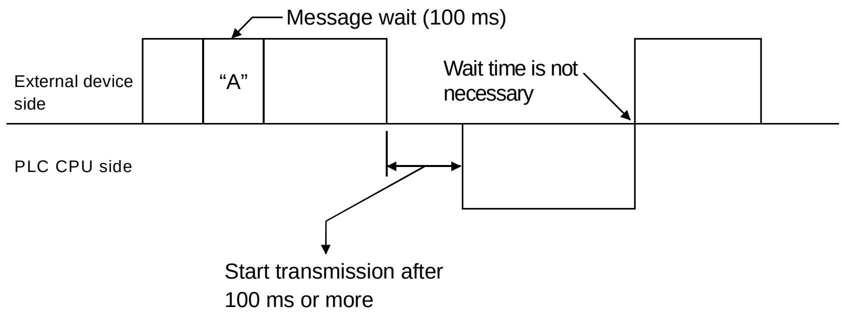
(5) Commands

A command is used to designate the type of access such as reading and writing to be done from the external device to the applicable PLC. The commands described in Sections 5.2 to 5.5 are designated by converting them to 2-digit (hexadecimal) ASCII codes.

(6) Message wait

Message wait is a data value for generating delay time in response transmission. Some external devices require a certain time to go into the receiving status after sending a command.
This value designates the minimum time that the Q series C24 must wait before sending a result after receiving a command from an external device. The wait time should be designated according to the specifications of the external device.
The wait time is designated in 10 ms units in the range from 0 to 150 ms, where every 10 ms is converted to $1_H$ in order to obtain a 1-digit ASCII code (hexadecimal) from $0_H$ to $F_H$ (0 to 15).
The following shows an example of designating a message wait value.

(Example)
The message wait time is 100 ms




(7) Character area A

This data field, which is sent in ASCII code, is used to execute a read request from the Q series C24 to the PLC CPU. The contents of character area A differ depending on the command sent from the external device. For detailed explanations, see Sections 5.2 through 5.5.

(8) Character area B

This is the data that the Q series C24 returns to an external device in response to a request designated with a command. It is sent in ASCII code. The contents of character area B differ depending on the command sent previously from an external device. For detailed explanations, see Sections 5.2 through 5.5.

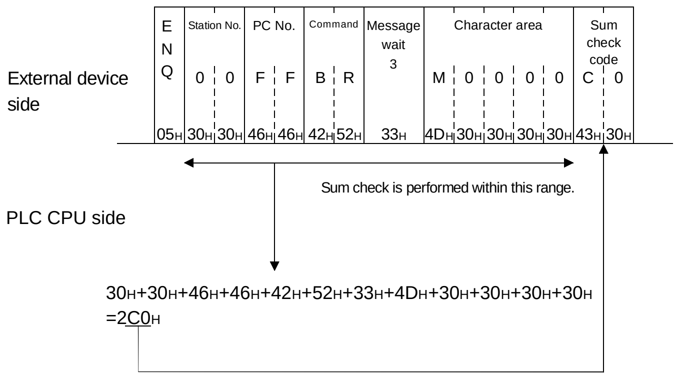
(9) Character area C

This data field, which is sent in ASCII code, is for issuing a command to the Q series C24 to execute a write request on the PLC CPU.
The contents of character area C differ depending on the command sent from the external device. For detailed explanations, see Sections 5.2 through 5.5.

### (10) Sum check code

The sum check code is obtained by converting the lower byte (8 bits) of the binary sum of the target data of the sum check into a 2-digit ASCII code (hexadecimal).
When "sum check is disabled" for GX Developer setting, the Q series C24 does not attach the sum check code to the transmission message. Also, the Q series C24 processes data received as if it does not contain a sum check code.
When "sum check is enabled," the Q series C24 attaches it to the transmission message. The Q series C24 also checks the sum check code in the received data.
The example below shows the contents of the sum check code.

(Example)
The following sum check code value is generated when character area "M0000" is transmitted in format 1, with station number 0, PC number FF, command BR (device memory batch read), and message wait 30 ms.




### (11) Error code

- An error code shows an error description when NAK is returned.
- Error codes are transmitted in 2-digit ASCII code (hexadecimal) in the range from $00_H$ to $FF_H$.
- If multiple errors are generated at the same time, the Q series C24 sends the error code that is detected first.
- For more details on the error codes, see Chapter 10, "Troubleshooting" of the User's Manual (Basic).

## 5.1.4 Understanding transmission data in the character areas

The transmission data that is handled in the character areas during data communication between an external device and the PLC CPU using various commands is the same as that when communicating using the QnA compatible 3C/4C frames.
See Section 3.1.7 for more details on the bit device data and word device data for the transmission data contained in the character areas.

# MEMO

## 5.1.5 List of Commands and functions for A compatible 1C frames

The following table lists the commands and functions when an external device accesses the PLC CPU using A compatible 1C frames.

| Function | | | Command: Symbol | Command: ASCII code | Processing | Number of points processed per communication |
|---|---|---|---|---|---|---|
| Device memory (∗6) | Batch read | Bit units | BR<br>JR | $42_H$, $52_H$<br>$4A_H$, $52_H$ | Reads bit devices (X, Y, M, etc.) in 1-point units. | 256 points |
| | | Word units | WR | $57_H$, $52_H$ | Reads bit devices (X, Y, M, etc.) in 16-point units. | 32 words (512 points) |
| | | | QR | $51_H$, $52_H$ | Reads word devices (D, R, T, C, etc.) in 1-point units. | 64 points |
| | Batch write (∗5) | Bit units | BW<br>JW | $42_H$, $57_H$<br>$4A_H$, $57_H$ | Writes to bit devices (X, Y, M, etc.) in 1-point units. | 160 points |
| | | Word units | WW | $57_H$, $57_H$ | Writes to bit devices (X, Y, M, etc.) in 16-point units. | 10 words (160 points) |
| | | | QW | $51_H$, $57_H$ | Writes to word devices (D, R, T, C, etc.) in 1-point units. | 64 points |
| | Test (random write) (∗5) | Bit units | BT<br>JT | $42_H$, $54_H$<br>$4A_H$, $54_H$ | Sets/resets bit devices (X, Y, M, etc.) in 1-point units by randomly designating the devices and device number. | 20 points |
| | | Word units | WT<br>QT | $57_H$, $54_H$<br>$51_H$, $54_H$ | Sets/resets bit devices (X, Y, M, etc.) in 16-point units by randomly designating the devices and device numbers. | 10 words (160 points) |
| | | | | | Writes to word devices (D, R, T, C, etc.) in 1-point units by randomly designating the devices and device numbers. | 10 points |
| | Monitor data registration (∗3) | Bit units | BM<br>JM | $42_H$, $4D_H$<br>$4A_H$, $4D_H$ | Registers bit devices (X, Y, M, etc.) to be monitored in 1-point units (∗2) | 40 points |
| | | Word units | WM<br>QM | $57_H$, $4D_H$<br>$51_H$, $4D_H$ | Registers bit devices (X, Y, M, etc.) to be monitored in 16-point units (∗2) | 20 words (320 points) |
| | | | | | Registers word devices (D, R, T, C, etc.) to be monitored in 1-point units. | 20 points |
| | Monitor | Bit units | MB<br>MJ | $4D_H$, $42_H$<br>$4D_H$, $4A_H$ | Monitors devices with monitor data registered. | (Number of registrations points) |
| | | Word units | MN<br>MQ | $4D_H$, $4E_H$<br>$4D_H$, $51_H$ | | |
| Extension file register | Batch read | | ER | $45_H$, $52_H$ | Reads extension file register (R) in 1-point units. | 64 points |
| | Batch write | | EW | $45_H$, $57_H$ | Writes to extension file register (R) in 1-point units. | 64 points |
| | Test (random write) (∗5) | | ET | $45_H$, $54_H$ | Writes to extension file register (R) in 1-point units by randomly designating the devices and device numbers. | 10 points |
| | Monitor data registration (∗3) | | EM | $45_H$,$4D_H$ | Registers extension file register (R) to be monitored in 1-point units. | 20 points |
| | Monitor | Word units | ME | $4D_H$, $45_H$ | Monitors extension file register (R) with monitor data registered. | (Number of registrations points) |
| | Direct read | Word units | NR | $4E_H$, $52_H$ | Reads extension file register in 1-point units by designating the device numbers by serial number regardless of the extension file register block number. | 64 points |
| | Direct write | Word units | NW | $4E_H$, $57_H$ | Writes to extension file register in 1-point units by designating the device numbers by serial number regardless of the extension file register block number. | 64 points |
| Intelligent function module | Batch read | | TR | $54_H$, $52_H$ | Reads data in the buffer memory of an intelligent function module. | 128 bytes |
| | Batch write | | TW | $54_H$, $57_H$ | Writes data to the buffer memory of an intelligent function module. | |
| Loopback test | | | TT | $54_H$, $54_H$ | Sends (returns) characters received from an external device back to the external device unchanged. | 254 bytes |

| Function | | – | Access station (＊4) | | | | PLC CPU status (＊1) | | | Reference section |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | A series CPU | QnA series CPU | Q series CPU | MELSECNET/H remote station Q series | During STOP | During RUN Write allow setting | During RUN Write prohibit setting | |
| Device memory | Batch read | Bit units | ○ | ○ | ○ | ○ | ○ | ○ | ○ | Section 5.2.2 |
| | | Word units | ○ | ○ | ○ | ○ | ○ | ○ | ○ | Section 5.2.5 |
| | Batch write | Bit units | ○ | ○ | ○ | ○ | ○ | ○ | × | Section 5.2.3 |
| | | Word units | ○ | ○ | ○ | ○ | ○ | ○ | × | Section 5.2.6 |
| | Test (random write) | Bit units | ○ | ○ | ○ | ○ | ○ | ○ | × | Section 5.2.4 |
| | | Word units | ○ | ○ | ○ | ○ | ○ | ○ | × | Section 5.2.7 |
| | Monitor data registration | Bit units | ○ | ○ | ○ | ○ | ○ | ○ | ○ | Section 5.2.8 |
| | | Word units | ○ | ○ | ○ | ○ | ○ | ○ | ○ | |
| | Monitor | Bit units / Word units | ○ | ○ | ○ | ○ | ○ | ○ | ○ | Section 5.2.8 |
| Extension file registers | Batch read | | ○ | × | × | × | ○ | ○ | ○ | Section 5.3.4 |
| | Batch write | | ○ | × | × | × | ○ | ○ | × | Section 5.3.5 |
| | Test (random write) | | ○ | × | × | × | ○ | ○ | × | Section 5.3.8 |
| | Monitor data registration | | ○ | × | × | × | ○ | ○ | ○ | Section 5.3.9 |
| | Monitor | | ○ | × | × | × | ○ | ○ | ○ | Section 5.3.9 |
| | Direct read | Word units | ○ | × | × | × | ○ | ○ | ○ | Section 5.3.6 |
| | Direct write | Word units | ○ | × | × | × | ○ | ○ | × | Section 5.3.7 |
| Intelligent function module | Batch read | | ○ | × | × | × | ○ | ○ | ○ | Section 5.4.3 |
| | Batch write | | ○ | × | × | × | ○ | ○ | × | Section 5.4.4 |
| Loopback test | | | – | – | – | – | ○ | ○ | ○ | Section 5.5 |

*1 The write allow/prohibit during RUN setting for the PLC CPU is set on the following screen of GX Developer.
- In case of the Q series C24
"Switch setting for I/O and intelligent functional module" screen

*2 For PLC CPUs other than the AnA/AnU/QnA/QCPU, device X (input) has two processing points per point. When the designated device includes X, the following condition should be satisfied:
((Designated points for an X device $\times$ 2) + designated points for other devices) $\leq$ number of points processed per communication.
When only X is designated, the number of points processed per communication becomes one half the value given in the table.

*3 The devices for the five types of command used for registering monitor data (BM, JM, WM, QM, EM) can be simultaneously registered in the Q series C24 for each interface.

*4 See Section 2.6.1 for details on the access stations shown in the table.

*5 If there is system protection on a Q/QnACPU that executes these commands, an error occurs and a NAK message is returned.

*6 Use dedicated commands for extension registers when reading and writing extension file registers to modules other than Q/QnACPU.

## 5.2 Device Memory Read/Write

This section explains the designations in the control procedure when reading from and writing to the device memory by providing an example.

### 5.2.1 Commands and device range

#### (1) Commands used for reading from and writing to the device memory

(a) ACPU common commands

| Item | | Command: Symbol | Command: ASCII code | Processing | Number of points processed per communication | PLC CPU status (＊1): During STOP | PLC CPU status (＊1): During RUN: Write allow setting | PLC CPU status (＊1): During RUN: Write prohibit setting | Reference section |
|---|---|---|---|---|---|---|---|---|---|
| Batch read | Bit units | BR | $45_H$, $52_H$ | Reads bit devices (X, Y, M, etc.) in 1-point units. | 256 points | ○ | ○ | ○ | Section 5.2.2 |
| | Word units | WR | $57_H$, $52_H$ | Reads bit devices (X, Y, M, etc.) in 16-point units. | 32 words (512 points) | | | | Section 5.2.5 |
| | | | | Reads word devices (D, R, T, C, etc.) in 1-point units. | 64 points | | | | |
| Batch write (＊4) | Bit units | BW | $42_H$, $57_H$ | Writes to bit devices (X, Y, M, etc.) in 1-point units. | 160 points | ○ | ○ | × | Section 5.2.3 |
| | Word units | WW | $57_H$, $57_H$ | Writes to bit devices (X, Y, M, etc.) in 16-point units. | 10 words (160 points) | | | | Section 5.2.6 |
| | | | | Writes to word devices (D, R, T, C, etc.) in 1-point units. | 64 points | | | | |
| Test (random write) (＊4) | Bit units | BT | $42_H$, $54_H$ | Sets/resets bit devices (X, Y, M, etc.) in 1-point units by randomly designating the devices and device number. | 20 points | ○ | ○ | × | Section 5.2.4 |
| | Word units | WT | $57_H$, $54_H$ | Sets/resets bit devices (X, Y, M, etc.) in 16-point units by randomly designating the devices and device numbers. | 10 words (160 points) | | | | Section 5.2.7 |
| | | | | Writes to word devices (D, R, T, C, etc.) in 1-point units by randomly designating the devices and device numbers. | 10 points | | | | |
| Monitor data registration (＊3) | Bit units | BM | $42_H$, $4D_H$ | Registers bit devices (X, Y, M, etc.) to be monitored in 1-point units. (＊2) | 40 points | ○ | ○ | ○ | Section 5.2.8 |
| | Word units | WM | $57_H$, $4D_H$ | Registers bit devices (X, Y, M, etc.) to be monitored in 16-point units. (＊2) | 20 words (320 points) | | | | |
| | | | | Registers word devices (D, R, T, C, etc.) to be monitored in 1-point units. | 20 points | | | | |
| Monitor | Bit units | MB | $4D_H$, $42_H$ | Monitors devices with monitor data registered. | – | ○ | ○ | ○ | Section 5.2.8 |
| | Word units | MN | $4D_H$, $4E_H$ | | | | | | |

In the PLC CPU status column of the table above, ○ indicates that the corresponding function is executable and × indicates it is not executable.

For ＊1, ＊2 and ＊3, see ＊1, ＊2 and ＊3 in Section 5.1.5.
For ＊4, see ＊5 in Section 5.1.5.

(b) AnA/AnUCPU common commands

| Item | | Command: Symbol | Command: ASCII code | Processing | Number of points processed per communication | PLC CPU status (∗1): During STOP | PLC CPU status (∗1): During RUN: Write allow setting | PLC CPU status (∗1): During RUN: Write prohibit setting | Reference section |
|---|---|---|---|---|---|---|---|---|---|
| Batch read | Bit units | JR | $4A_H$, $52_H$ | Reads bit devices (X, Y, M, etc.) in 1-point units. | 256 points | ○ | ○ | ○ | Section 5.2.2 |
| | Word units | QR | $51_H$, $52_H$ | Reads bit devices (X, Y, M, etc.) in 16-point units. | 32 words (512 points) | | | | Section 5.2.5 |
| | | | | Reads word devices (D, R, T, C, etc.) in 1-point units. | 64 points | | | | |
| Batch write | Bit units | JW | $4A_H$, $57_H$ | Writes to bit devices (X, Y, M, etc.) in 1-point units. | 160 points | ○ | ○ | × | Section 5.2.3 |
| | Word units | QW | $51_H$, $57_H$ | Writes to bit devices (X, Y, M, etc.) in 16-point units. | 10 words (160 points) | | | | Section 5.2.6 |
| | | | | Writes to word devices (D, R, T, C, etc.) in 1-point units. | 64 points | | | | |
| Test (random write) | Bit units | JT | $4A_H$, $54_H$ | Sets/resets bit devices (X, Y, M, etc.) in 1-point units by randomly designating the devices and device number. | 20 points | ○ | ○ | × | Section 5.2.4 |
| | Word units | QT | $51_H$, $54_H$ | Sets/resets bit devices (X, Y, M, etc.) in 16-point units by randomly designating the devices and device numbers. | 10 words (160 points) | | | | Section 5.2.7 |
| | | | | Writes to word devices (D, R, T, C, etc.) in 1-point units by randomly designating the devices and device numbers. | 10 points | | | | |
| Monitor data registration (∗3) | Bit units | JM | $4A_H$, $4D_H$ | Registers the bit devices (X, Y, M, etc.) to be monitored in 1-point units. (∗2) | 40 points | ○ | ○ | ○ | Section 5.2.8 |
| | Word units | QM | $51_H$, $4D_H$ | Registers bit devices (X, Y, M, etc.) to be monitored in 16-point units. (∗2) | 20 words (320 points) | | | | |
| | | | | Registers word devices (D, R, T, C, etc.) to be monitored in 1-point units. | 20 points | | | | |
| Monitor | Bit units | MJ | $4D_H$, $4A_H$ | Monitors devices with monitor data registered. | – | ○ | ○ | ○ | Section 5.2.8 |
| | Word units | MQ | $4D_H$, $51_H$ | | | | | | |

In the PLC CPU status column of the table above, ○ indicates that the corresponding function is executable and × indicates it is not executable.

For ∗1, ∗2 and ∗3, see ∗1, ∗2 and ∗3 in Section 5.1.5.

## (2) Accessible device range

### (a) Device range when using the ACPU common commands

The following table outlines the devices and device number range that can be designated when accessing the device memory.

1) Each device consists of five characters, which are designated in the order shown in the following table.
   In the device number, the "0"-sequence in the high digit (for example, the leading zeros in X0070) can be designated by a blank code (20H).

Device + Device number = 5 characters

(1 character (2 characters when T/C)) (4 characters (3 characters when T/C))

| Device | | Character number range (character) (∗1) | Decimal/hexadecimal notation | Device | | Character number range (character) (∗1) | Decimal/hexadecimal notation |
|---|---|---|---|---|---|---|---|
| Input | X | X0000 to X07FF | Hexadecimal | Timer (contact) | T | TS000 to TS255 | Decimal |
| Output | Y | Y0000 to Y07FF | | Timer (coil) | T | TC000 to TC255 | |
| Internal relay | M | M0000 to M2047 | Decimal | Timer (current value) | T | TN000 to TN255 | |
| Latch relay | L | L0000 to L2047 | | Counter (contact) | C | CS000 to CS255 | |
| Step relay | S | S0000 to S2047 | | Counter (coil) | C | CC000 to CC255 | |
| Link relay | B | B0000 to B03FF | Hexadecimal | Counter (current value) | C | CN000 to CN255 | |
| Annunciator | F | F0000 to F0255 | Decimal | Data register | D | D0000 to D1023 | |
| Special relay | M | M9000 to M9255 | | Link register | W | W0000 to W03FF | Hexadecimal |
| Special register | D | D9000 to D9255 | | File register | R | R0000 to R8191 | Decimal |

### (b) Device range when using the AnA/AnUCPU common commands

The following table outlines the devices and device number range that can be designated when accessing the device memory.

1) Each device consists of seven characters, which are designated in the order shown in the following table.
   In the device number, the "0"-sequence in the high digit (for example, the leading zeros in X000070) can be designated by a blank code (20H).

Device + Device number = 7 characters

(1 character (2 characters when T/C)) (6 characters (5 characters when T/C))

| Device | | Character number range (character) (∗1) | Decimal/hexadecimal notation | Device | | Character number range (character) (∗1) | Decimal/hexadecimal notation |
|---|---|---|---|---|---|---|---|
| Input | X | X000000 to X001FFF | Hexadecimal | Timer (contact) | T | TS00000 to TS02047 | Decimal |
| Output | Y | Y000000 to Y001FFF | | Timer (coil) | T | TC00000 to TC02047 | |
| Internal relay | M | M000000 to M008191 | Decimal | Timer (current value) | T | TN00000 to TN02047 | |
| Latch relay | L | L000000 to L008191 | | Counter (contact) | C | CS00000 to CS01023 | |
| Step relay | S | S000000 to S008191 | | Counter (coil) | C | CC00000 to CC01023 | |
| Link relay | B | B000000 to B001FFF | Hexadecimal | Counter (current value) | C | CN00000 to CN01023 | |
| Annunciator | F | F000000 to F002047 | Decimal | Data register | D | D000000 to D008191 | |
| Special relay | M | M009000 to M009255 | | Link register | W | W000000 to W001FFF | Hexadecimal |
| Special register | D | D009000 to D009255 | | File register | R | R000000 to R008191 | Decimal |

2) The range of accessible devices when accessing the MELSECNET/H remote I/O station is indicated below. For link relay (B), data register (D) and link register (W), the devices after the numbers shown in the table below cannot be accessed.

| Device | | Character number range (character) | Decimal/hexadecimal notation | Device | | Character number range (character) | Decimal/hexadecimal notation |
|---|---|---|---|---|---|---|---|
| Input | X | X000000 to X001FFF | Hexadecimal | Link relay | B | B000000 to B001FFF | Hexadecimal |
| Output | Y | Y000000 to Y001FFF | | Data register | D | D000000 to D008191 | Decimal |
| Internal relay | M | M000000 to M008191 | Decimal | Link register | W | W000000 to W001FFF | Hexadecimal |

＊1 The following precautions should be observed when reading/writing to/from Q/QnACPU.

1) Devices that have the same names as those found in AnCPU, AnNCPU, AnACPU and AnUCPU can be accessed within the range shown in the table.

The following devices cannot be accessed from the external devices:
- Devices newly added to the Q/QnACPU
- Latch relay (L) and step relay (S)

＊ For the Q/QnACPU, the latch relay (L) and the step relay (S) are separate devices from the internal relay (M). However, access will be made to the internal relay when either the latch relay or the step relay is designated.
- File register (R)

2) Special relays and special registers can be accessed in the way shown below.
- By designating M9000 to M9255, it is possible to access SM1000 to SM1255.
- By designating D9000 to D9255, it is possible to access SD1000 to SD1255.

POINT

(1) Access the PLC CPU within the device number range that can be used by the ACPU common command and AnA/AnU common command and within the device number range that can be used by the PLC CPU of the access destination.

(2) When word units are designated, always make the head device number of a bit device a multiple of 16 (0, 16... in decimal notation).
The special relays beginning from M9000 can be designated by (9000 + multiple of 16).

(3) M, L, and S can be designated by range. However, they are processed equivalently, even when the number range of M is specified by an L and S designation, or vice versa.

(4) The special relays (M9000 to M9255) and special registers (D9000 to D9255) are divided into read only, write only and system use registers.
If writing takes place outside the writing enabled range, a PLC CPU error may occur. See the ACPU Programming Manual for detailed descriptions of the special relays and special registers.

(5) When using dedicated instructions for the AnACPU and AnUCPU extension file registers, read from/write to a file register (R) using the commands described in Section 5.4.

(6) The number of device points that are read/written when the commands are executed should be designated by converting the number of processing points (number of points processed per communication), which should be within the range shown in the table in Section 5.2.1, into a 2-digit ASCII code (hexadecimal).
Note that "00" is used only when designating 256 points.
(Example: values in parenthesis indicate ASCII code)

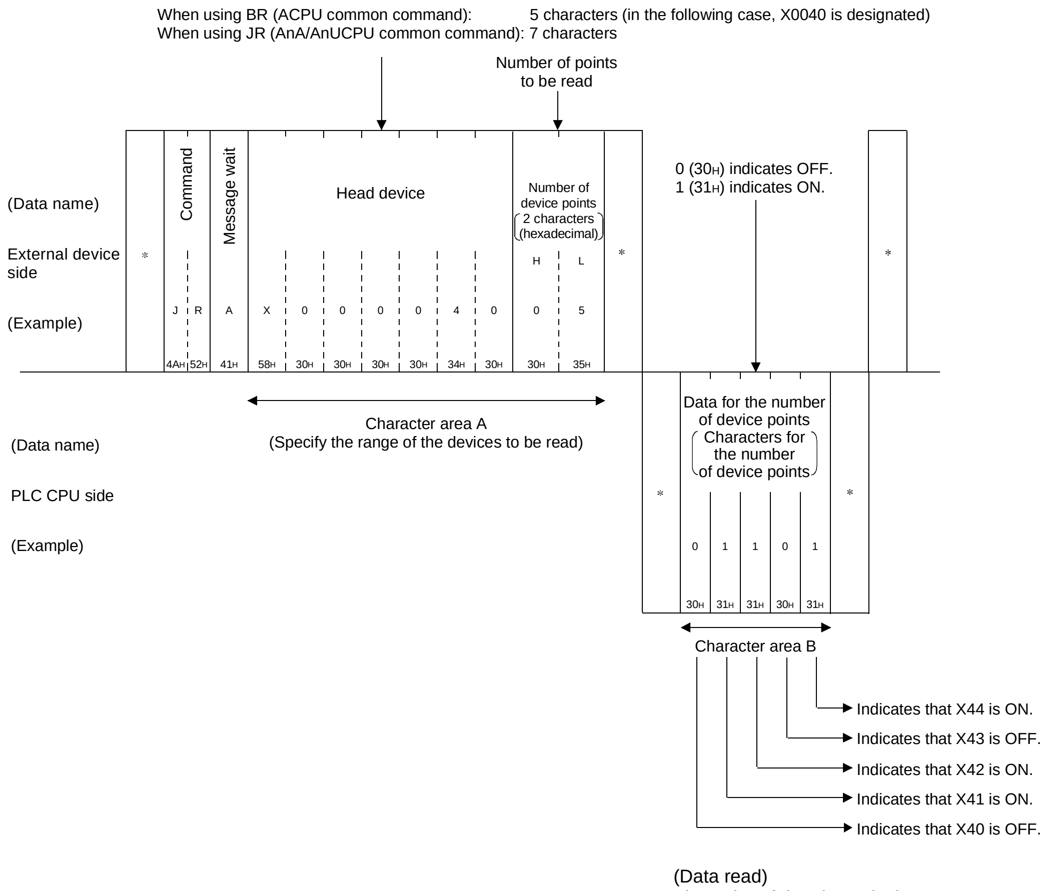
| | | |
|---|---|---|
| In case of 5 points | : | 05 ($30_H$, $35_H$) |
| In case of 10 points | : | 0A ($30_H$, $41_H$) |
| In case of 20 points | : | 14 ($31_H$, $34_H$) |
| In case of 256 points | : | 00 ($30_H$, $30_H$) |

### 5.2.2 Batch read in bit units (command: BR, JR)

The example shown in this section explain the control procedure for batch reading bit device memory using the BR and JR commands.
The order and contents of data items for the parts marked with "*" shown in the control procedure differ depending on the communication format.
See a detailed explanation in Section 5.1.

[Control procedure]

(Read conditions)
1) Message wait is 100 ms.
2) Read 5 points from X040 to X044.




* The order of data items is the same as when using both the JR and BR commands.

**POINT**

(1) Designate the number of device points within the following range:
- 1 ≤ Number of device points ≤ 256 (256 points setting is designated by 00H)
- Head device number + number of device points - 1 ≤ maximum device number

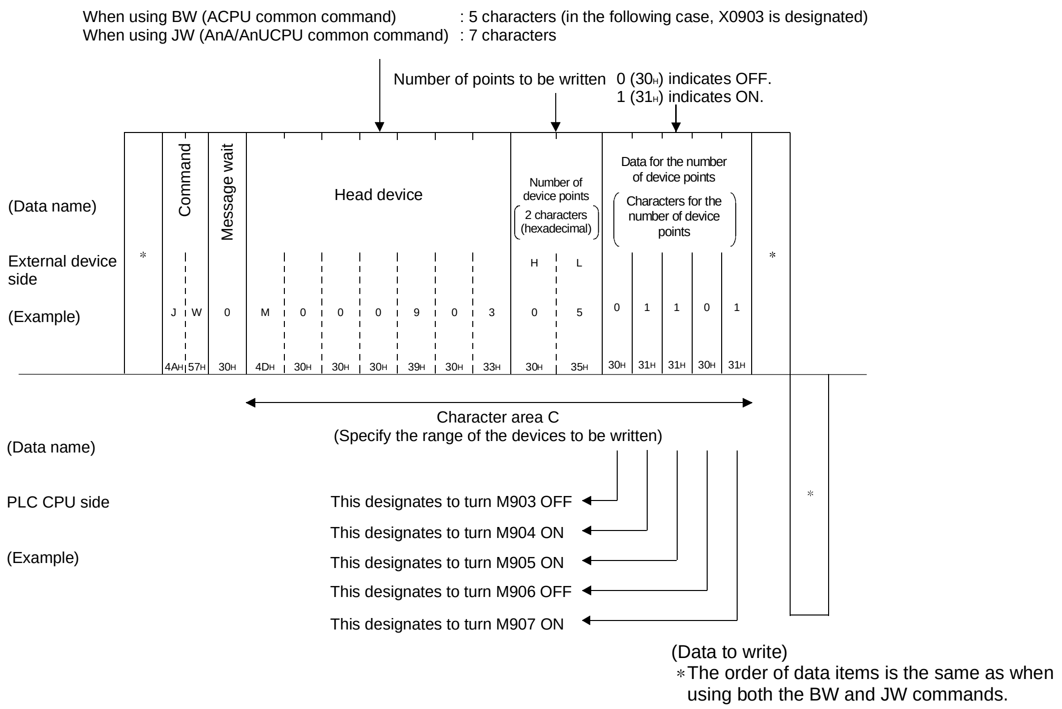
(2) The message wait time is designated in 10 ms units in the range from 0 to 150 ms and expressed in 0 to FH (hexadecimal)
Thus, 100 ms is expressed as "A."

### 5.2.3 Batch write in bit units (command: BW, JW)

The example shown in this section explain the control procedure for batch writing bit device memory using the BW and JW commands.
The order and contents of data items for the parts marked with "*" shown in the control procedure differ depending on the communication format.
See a detailed explanation in Section 5.1.

[Control procedure]

(Write conditions)
1) Message wait is 0 ms.
2) Write 5 points to M903 to M907.




**POINT**

Designate the number of device points within the following range:
- 1 ≤ Number of device points ≤ 160
- Head device number + number of device points - 1 ≤ maximum device number

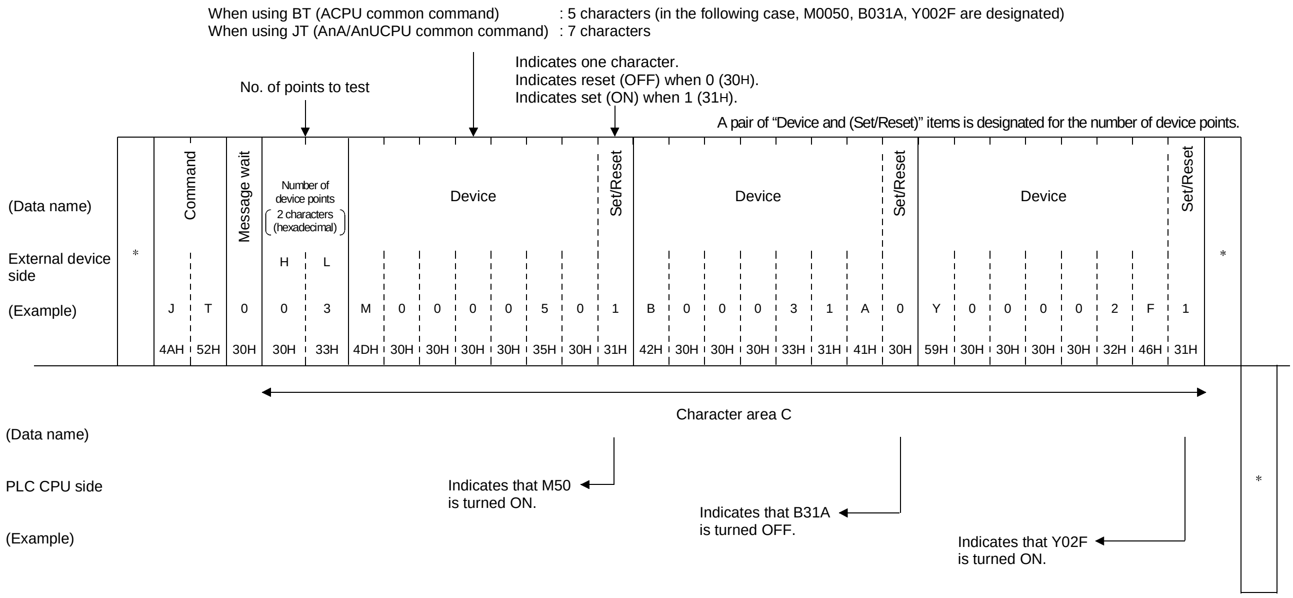
## 5.2.4 Test in bit units (random write) (command: BT, JT)

The example shown in this section explain the control procedure for writing data by randomly designating bit device memory using the BT and JT commands.
The order and contents of data items for the parts marked with "∗" shown in the control procedure differ depending on the communication format.
See a detailed explanation in Section 5.1.

[Control procedure]

(Write conditions)

1) Message wait is 0 ms.
2) Test of 3 points (3 bits) that turn M50 on, B31A off, and Y02F on.




(Data for the number of device points to test)
∗The method to designate set/reset is the same as when using both the BJ and JT commands.

| POINT |
|---|
| Designate the number of device points within the following range:<br>• 1 ≤ Number of device points ≤ 20 |

### 5.2.5 Batch read in word units (command: WR, QR)

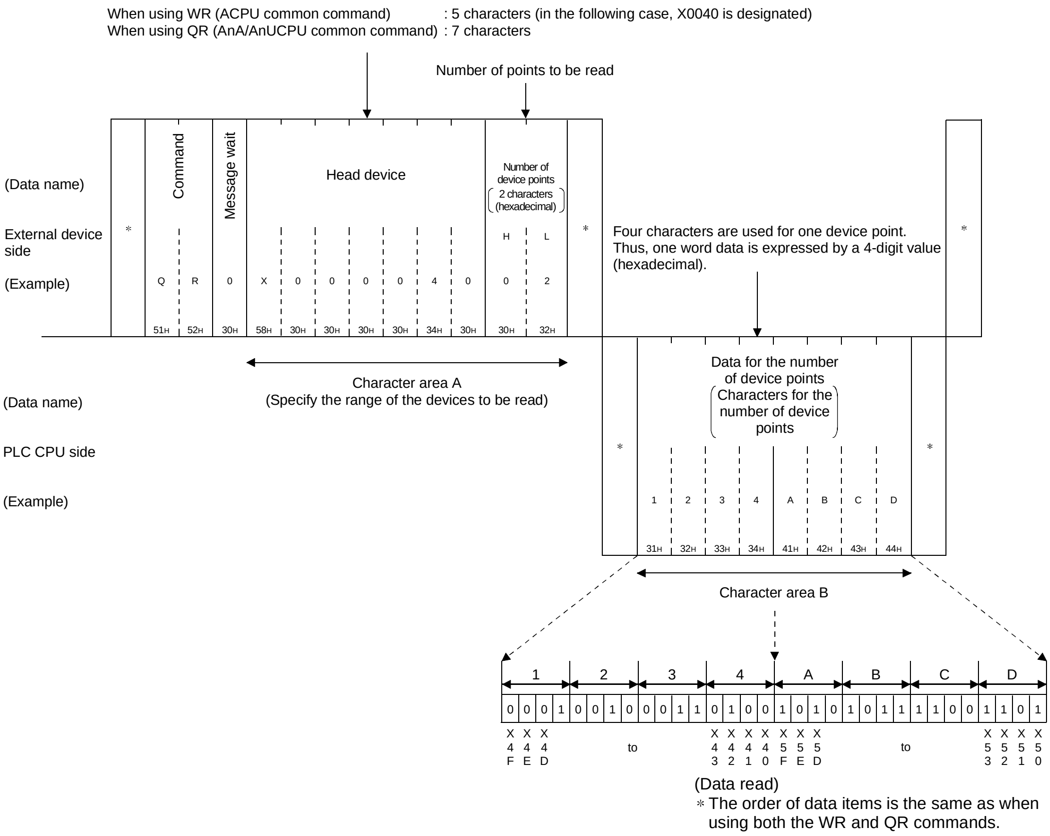
The example shown in this section explain the control procedure for batch reading bit device memory (16 bit units) and word device memory (word units) using the WR and QR commands.
The order and contents of data items for the parts marked with "∗" shown in the control procedure differ depending on the communication format.
See a detailed explanation in Section 5.1.

[Control procedure]

(Read conditions)

1) Message wait is 0 ms.
2) Read 32 points from X040 to X05F and 2 points from the current values of T123 and T124 (2 words for both).

#### (1) Reading the bit device memory

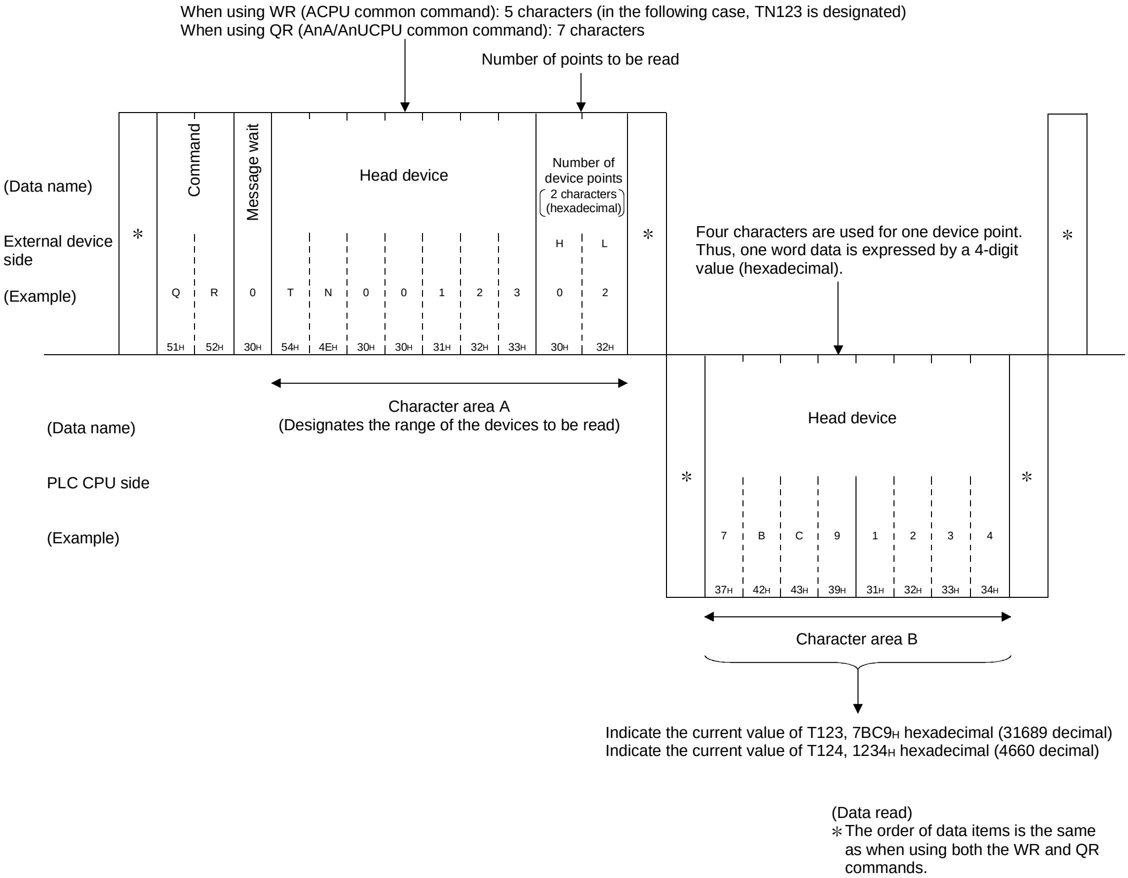



| POINT |
|---|

(1) When reading the bit device memory, designate the number of device points within the following range:
- 1 ≤ Number of device points ≤ 32
- Head device number + number of device points × 16 - 1 ≤ maximum device number

(2) The WR and QR commands are in word units. When reading 32 points from X40 to X5F, the number of device point designated is "02" (16 points are designated as 1).

## (2) Reading the word device memory




**POINT**

(1) When reading the word device memory, designate the number of device points within the following range:
- 1 ≤ Number of device points ≤ 64
- Head device number + number of device points - 1 ≤ maximum device number

(2) The WR and QR commands are in word units. When reading the current value from T123 to T124, the number of device point designated is "02" (1 point is designated as 1).

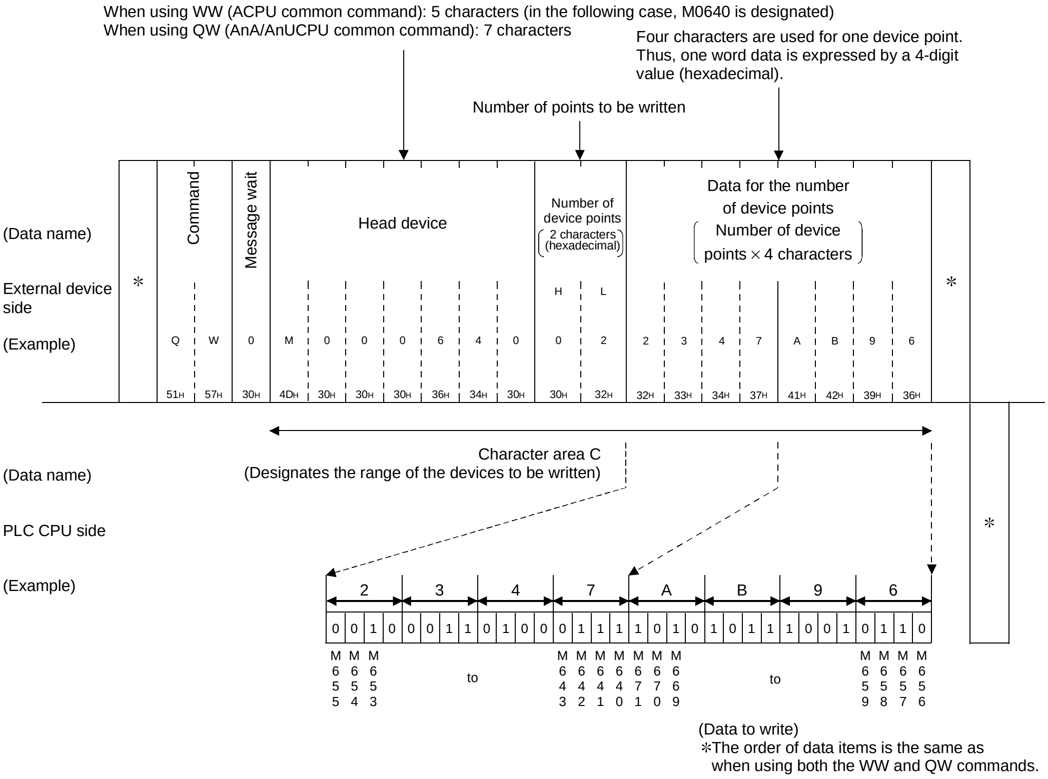
### 5.2.6 Batch write in word units (command: WW, QW)

The example shown in this section explain the control procedure for batch-writing the bit device memory (16 bit units) and the word device memory (word units) using the WW and QW commands.
The order and contents of data items for the parts marked with "✻" shown in the control procedure differ depending on the communication format.
See a detailed explanation in Section 5.1.

[Control procedure]

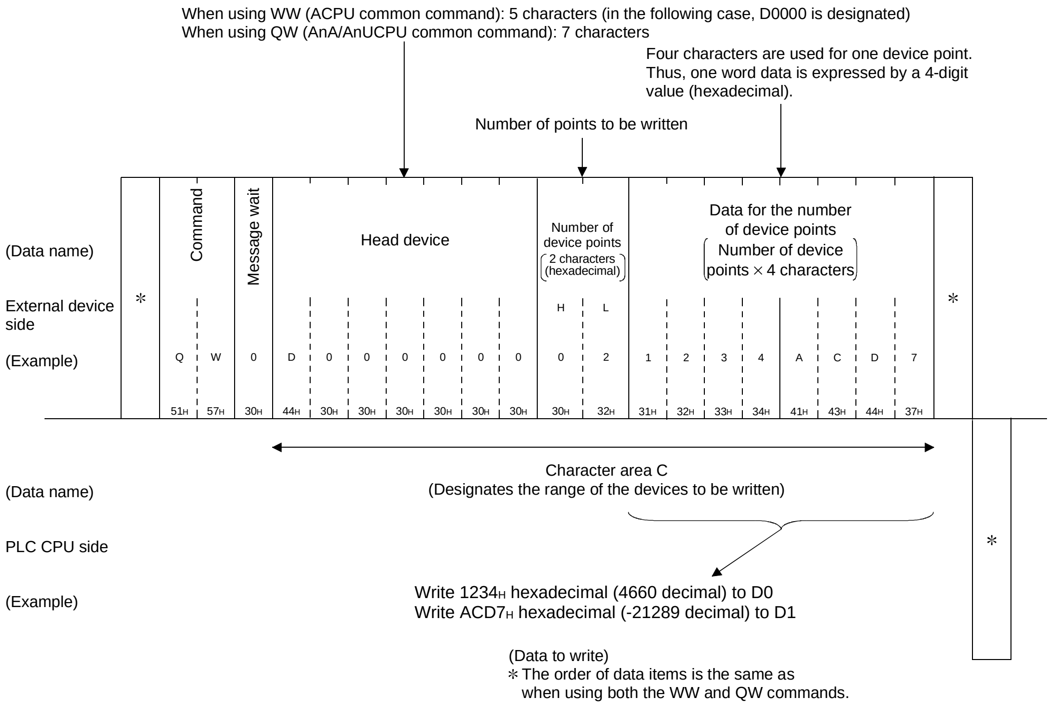
(Write conditions)
1) Message wait is 0 ms.
2) Write 32 points to M640 to M671 and 2 points to D0 and D1 (2 words for both).

(1) Writing to the bit device memory




| POINT |
|---|
| (1) When writing to the bit device memory, designate the number of device points within the following range:<br>• 1 ≤ Number of device points ≤ 10<br>• Head device number + number of device points × 16 - 1 ≤ maximum device number<br><br>(2) The WW and QW commands are in word units. When writing 32 points to M640 to M671, the number of device points designated is "02" (16 points are designated as 1). |

## (2) Writing to the word device memory




**POINT**

(1) When writing to the word device memory, designate the number of device points within the following range:
- 1 ≤ Number of device points ≤ 64
- Head device number + number of device points - 1 ≤ maximum device number

(2) The WW and QW commands are in word units. When writing two points to D0 to D1, the number of device points designated is "02" (1 points is designated as 1).

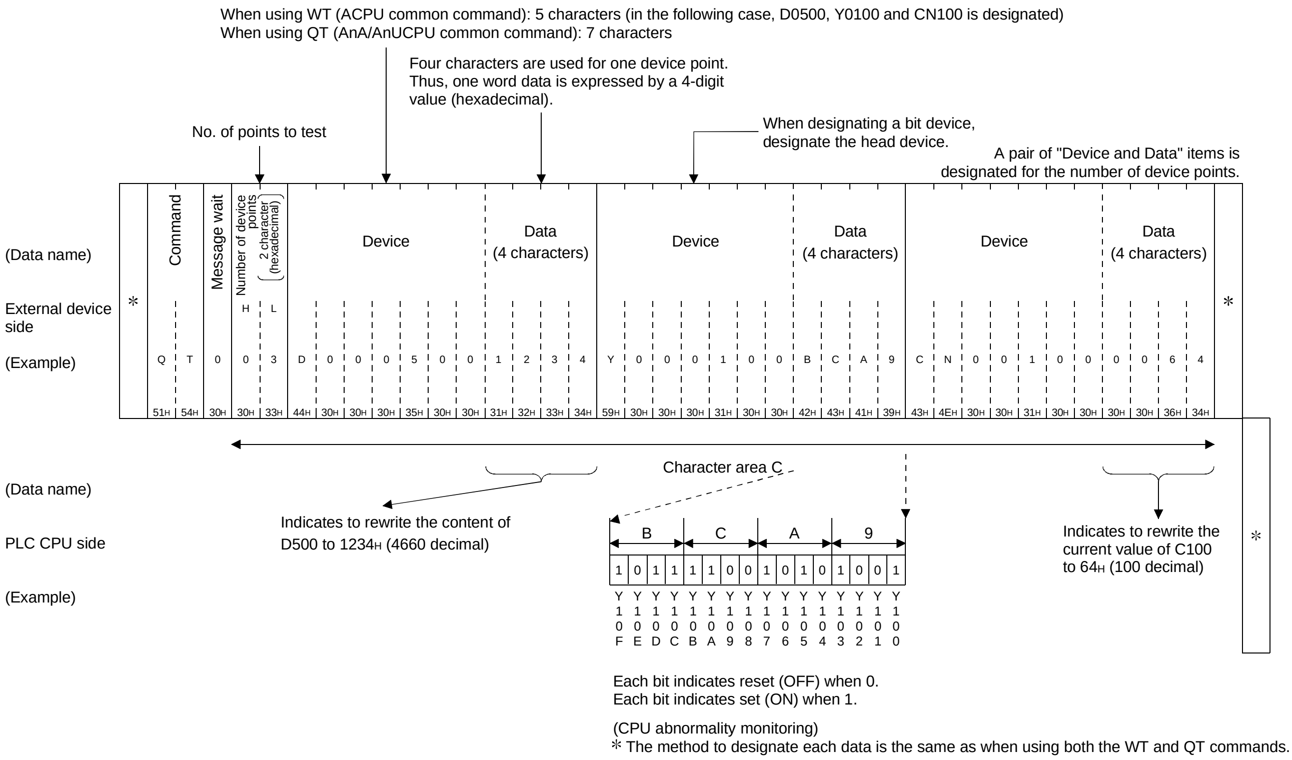
### 5.2.7 Test in word units (random write) (command: WT, QT)

The example shown in this section explain the control procedure for writing by randomly designating the word device memory and bit device memory (16 bit units) using the WT and QT commands.
The order and contents of data items for the parts marked with "*" shown in the control procedure differ depending on the communication format.
See a detailed explanation in Section 5.1.

[Control procedure]

(Write conditions)

1) Message wait is 0 ms.
2) Test 3 points (3 words) registering 1234H to D500, BCA9H to Y100 to Y10F, and 64H to the current value of C100.




**POINT**

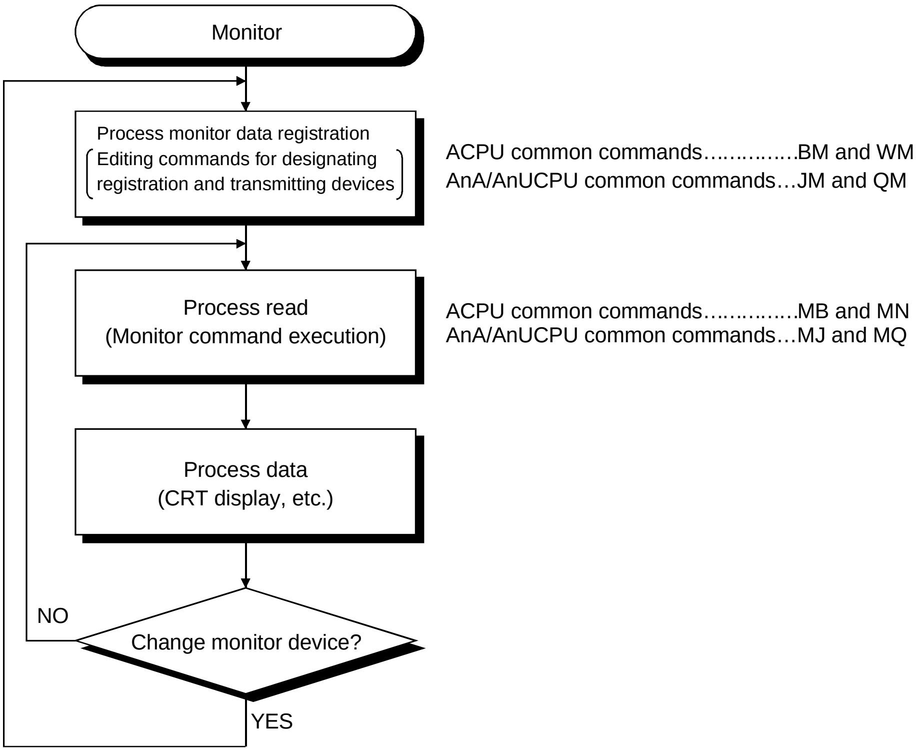
Designate the number of device points within the following range:

- 1 ≤ Number of device points ≤ 10 (for bit device : 10(16 points are designated as 1))

### 5.2.8 Monitoring device memory

The monitor data registration function registers the name and number of the devices to be monitored from an external device to the Q series C24.
The monitor function allows to read the data contents of the registered devices from the PLC CPU and process it in the external device.
When the batch read functions (BR, WR/JR or QR) are used to read the devices, the device numbers must be consecutive. However, by using the monitor data registration function, devices can be monitored by designating the device numbers freely.
The following example illustrates the control procedure for monitoring and registering name and number of the devices to be monitored to the Q series C24.

#### (1) Monitoring procedure




**POINT**

(1) Monitor data must always be registered when monitoring using the procedure shown above. If the monitoring is performed without registering the monitor data, a protocol error will be generated.

(2) The contents of registered monitor data are deleted when the Q series C24 is rebooted.

(3) Each designated devices can be registered for each command for bit units (BM or JM), word units (WM or QM), and the extension file register (EM) in the Q series C24.

(4) When multiple Q series C24s register monitor data for a device of the PLC CPU at the same station, the registration data is overwritten, and the device memory that was registered last becomes valid.

(5) See section 5.3.9 for details about monitoring the extension file registers.

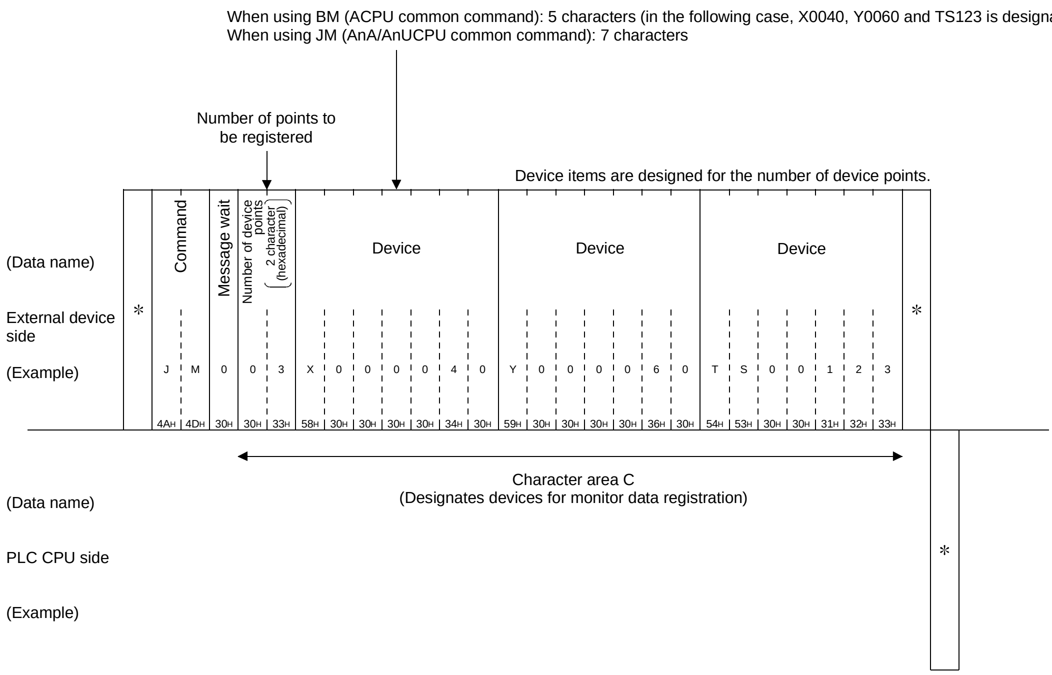
(2) Monitor data registration for device memory (commands: BM, JM, WM, QM)

The example shown in this section explain the control procedure for registering the name and number of devices to be monitored in the Q series C24 using the BM/JM/WM/QM command.
The order and contents of data items for the parts marked with "∗" shown in the control procedure differ depending on the communication format.
See a detailed explanation in Section 5.1.

(a) Registering monitor data to the bit device memory in bit units.

[Control procedure]

(Conditions for monitor data registration)
1) Message wait is 0 ms.
2) Register monitor data for 3 points (3 bits) of contacts for X40, Y060 and T123.




**POINT**

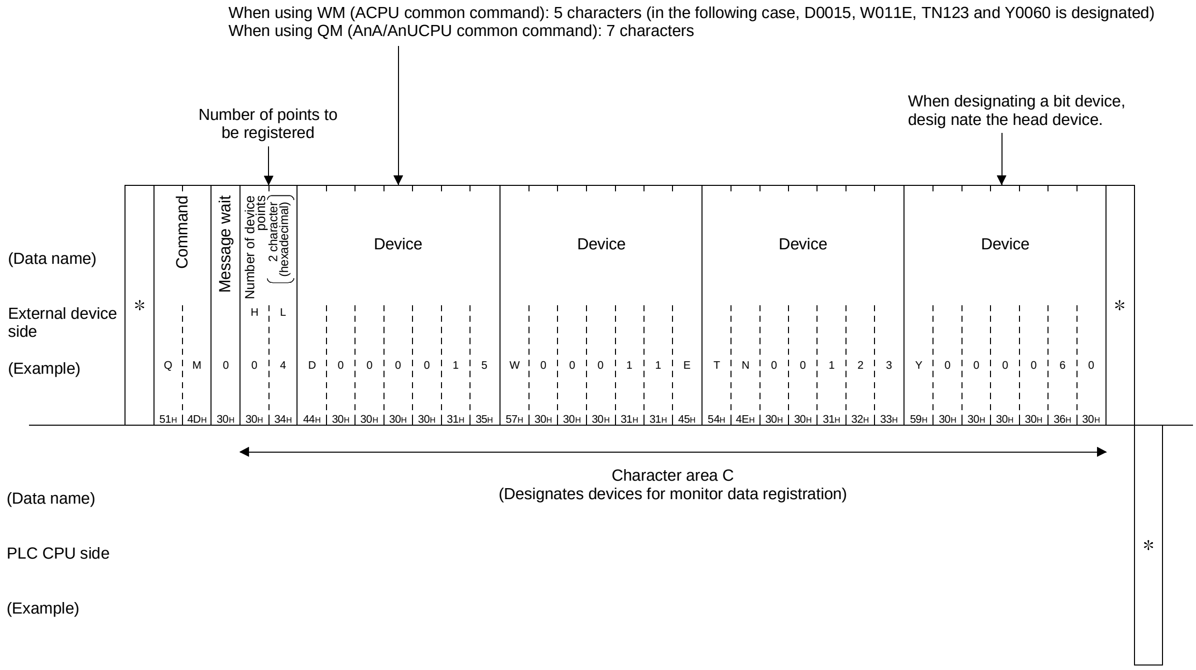
(1) Designate the number of device points within the following range:
When the BM command is used and a PLC CPU other than AnACPU/AnUCPU/QnACPU/QCPU is accessed, device X (input) has two processing points per point.
• 1 ≤ Number of device points ≤ 40

(2) Follow the explanation in this section (3) (a) to monitor device memory registered in bit units.

(b) Registering monitor data to the bit device memory and the word device memory in word units.

[Control procedure]

(Conditions for monitor data registration)
1) Message wait is 0 ms.
2) Register monitor data for the current values of D15, W11E, and T123 as well as for the 4 points (4 words) of Y060 to Y06F.




**POINT**

(1) Designate the number of device points within the following range:
When the WM command is used and a PLC CPU other than A3HCPU/AnACPU/AnUCPU/QnACPU/QCPU is accessed, device X (input) has 2 processing points per point.
- $1 \leq$ Number of device points $\leq 20$

(2) As shown in the diagram above, word devices and bit devices (16-point units) can be designated together in the monitor registration in word units.

(3) Follow the explanation in this section (3) (b) to monitor device memory registered in word units.

(3) Monitoring the device memory to which monitor data is registered (command: MB, MJ, MN, MQ)

The example shown in this section explain the control procedure for monitoring devices registered in the Q series C24 by monitor data registration of item (2) using the MB/MJ/MN/MQ command.

The order and contents of data items for the parts marked with "✻" shown in the control procedure differ depending on the communication format.

See a detailed explanation in Section 5.1.

(a) Monitoring the bit device memory to which the monitor data is registered in bit units.

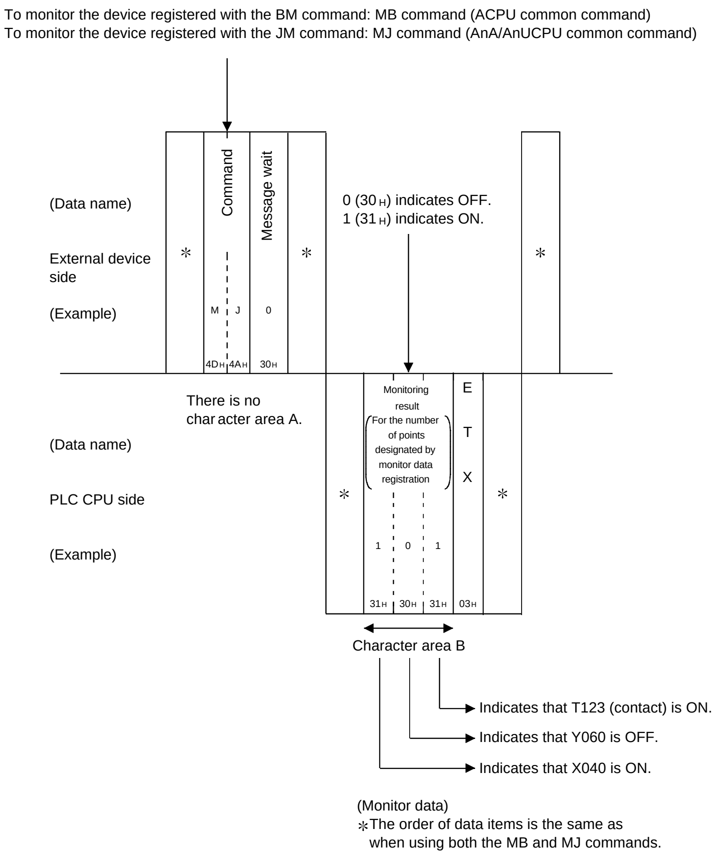
[Control procedure]

Bit device memory registered using the BM command is monitored with the MB command.

Bit device memory registered using the JM command is monitored with the MJ command.

(Monitoring conditions)

1) Message wait is 0 ms.
2) Monitor 3 points (3 bits) of contacts X040, Y060, and T123 for which monitor data is registered.



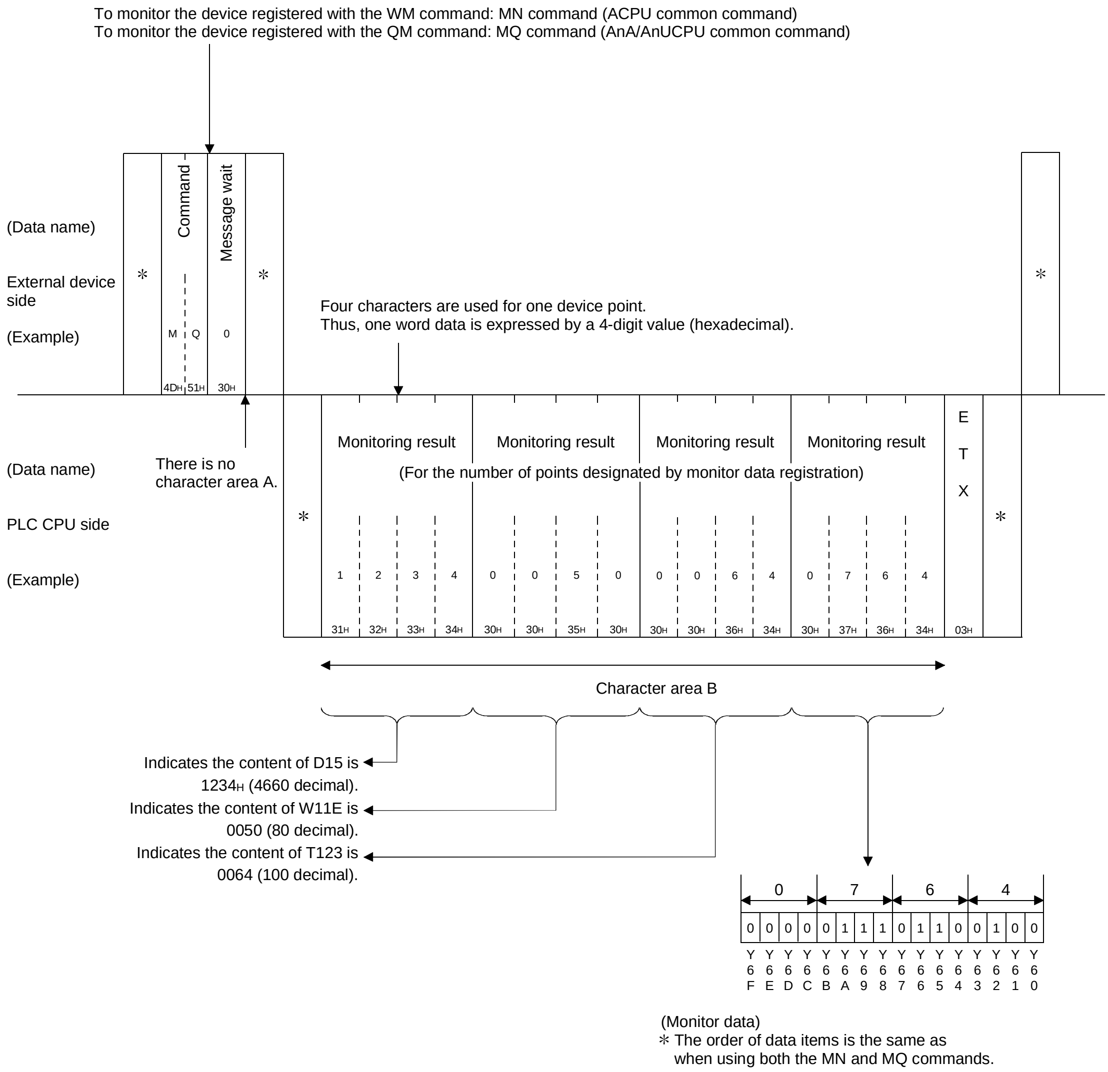
(b) Monitoring the bit device memory and the word device memory to which the monitor data is registered in word units.

[Control procedure]

Device memory registered using the WM command is monitored with the MN command.
Device memory registered using the QM command is monitored with the MQ command.

(Monitoring Conditions)

1) Message wait time is 0 ms.
2) Monitor the current values of D15, W11E and T123 as well as 4 points (4 words) of Y060 to Y06F at which monitor data is registered.



## 5.3 Extension File Register Read and Write

The extension file register uses free areas in the user memory area of the PLC CPU as a file register. It serves as a memory area for storing necessary data and results of operations in various data processing performed using the software packages for extension file registers "SWOGHP-UTLPC-FN1" (hereinafter called UTLP-FN1), as well as in the dedicated instructions for the extension file registers of AnACPU and AnUCPU.
Using the examples below, this section explains the control procedure to read and write extension file register.

### 5.3.1 ACPU common commands and addresses

(1) The following table lists the ACPU common commands that are used for reading and writing data from/to the extension file register.

| Function | Command (subcommand) | | Processing | Number of points processed per communication | PLC CPU status | | | Reference section |
|---|---|---|---|---|---|---|---|---|
| | Symbol | ASCII code | | | During STOP | During RUN | | |
| | | | | | | Write allow setting | Write prohibit setting | |
| Batch read | ER | $45_H$, $52_H$ | Reads extension file register (R) in 1-point units. | 64 points | ○ | ○ | ○ | Section 5.3.4 |
| Batch write | EW | $45_H$, $57_H$ | Writes to extension file register (R) in 1-point units. | 64 points | ○ | ○ | × | Section 5.3.5 |
| Test (Random write) | ET | $45_H$, $54_H$ | Writes to extension file register (R) in 1-point units by randomly designating the devices and device numbers. | 10 points | ○ | ○ | × | Section 5.3.8 |
| Monitor data registration | EM | $45_H$, $4D_H$ | Registers device number to be monitored in 1-point units. | 20 points | ○ | ○ | ○ | Section 5.3.9 (2) |
| Monitor | ME | $4D_H$, $45_H$ | Monitors extension file register with monitor data registered. | – | ○ | ○ | ○ | Section 5.3.9 (3) |

In the PLC CPU status column of the table above, ○ indicates that the corresponding function is executable and × indicate not executable.

(2) Addresses of the extension file register

(a) An extension file register has blocks numbered from 0 to n (the value of n differs depending on the memory cassette). Block number 0 has a number of points registered with a parameter of the PLC CPU, while block numbers 1 to n have a register of 8192 points in each block.
However, the range that can be read and written by the PLC CPU is the range of points designated by a parameter for block 0.

(b) The range of block numbers that can be designated depends on the memory cassette type and the parameter settings in the PLC CPU.
For more detailed explanations, see the operating manual of UTLP-FN1, or the User's Manuals of AnACPU and AnUCPU.

(c) Designate the address using seven characters, consisting of a block number and a device number.
- 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 No. (3 digits)" + "Device No. (4 digits)"

Example of designation

When the block number is less than 2 digits

0 5 R 8 1 9 0
- 8 1 9 0: Device number
- 0 5: Block number

When the block number is 3 digits

1 0 2 8 1 9 0
- 8 1 9 0: Device number
- 1 0 2: Block number

### 5.3.2 AnA/AnUCPU common commands and device numbers

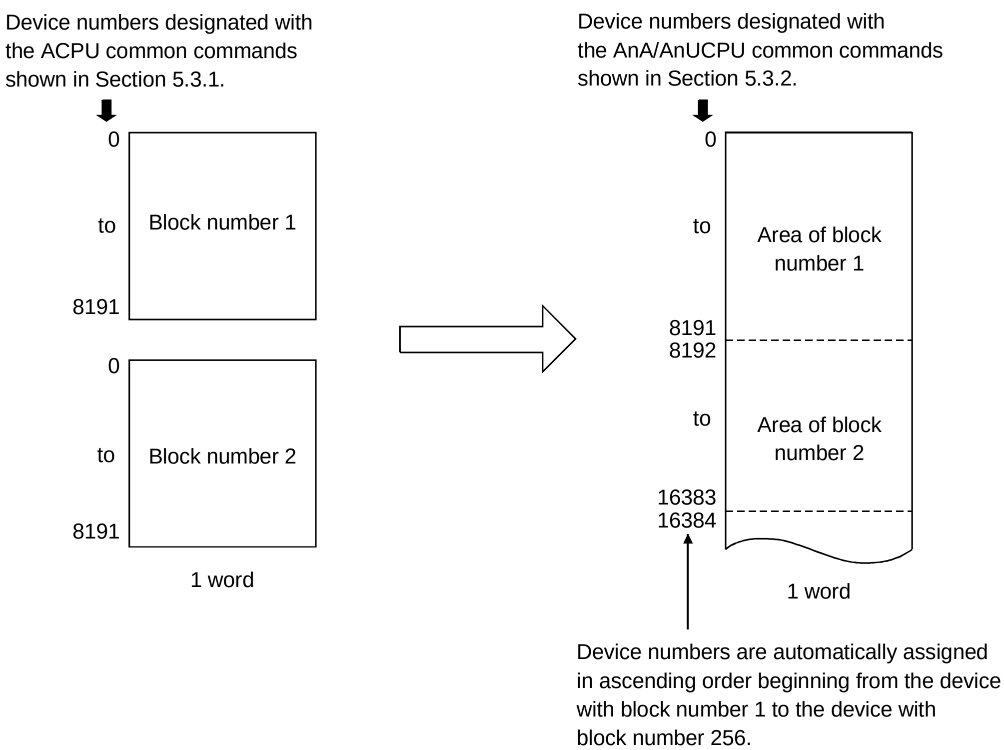
(1) This section explains AnA/AnUCPU common commands used for directly reading from and writing to the extension file register.
This command allow the access to extension file register blocks numbered from 1 to 256 by designating addresses from device number 0 of block numbered from 1 as the device number, regardless of each block number.
(Access the extension file registers for the usable block count × 8192 points using consecutive device numbers.)

| Function | Command (subcommand) | | Processing | Number of points processed per communication | PLC CPU status | | | Reference section |
|---|---|---|---|---|---|---|---|---|
| | | | | | | During RUN | | |
| | Symbol | ASCII code | | | During STOP | Write allow setting | Write prohibit setting | |
| Direct read | NR | $4E_H$, $52_H$ | Reads by designating the device numbers of the extension file register by serial number in 1-point units (1 word). | 64 points | ○ | ○ | ○ | Section 5.3.6 |
| Direct write | NW | $4E_H$, $57_H$ | Writes by designating the device numbers of extension file register by serial number in 1-point units (1 word). | 64 points | ○ | ○ | × | Section 5.3.7 |

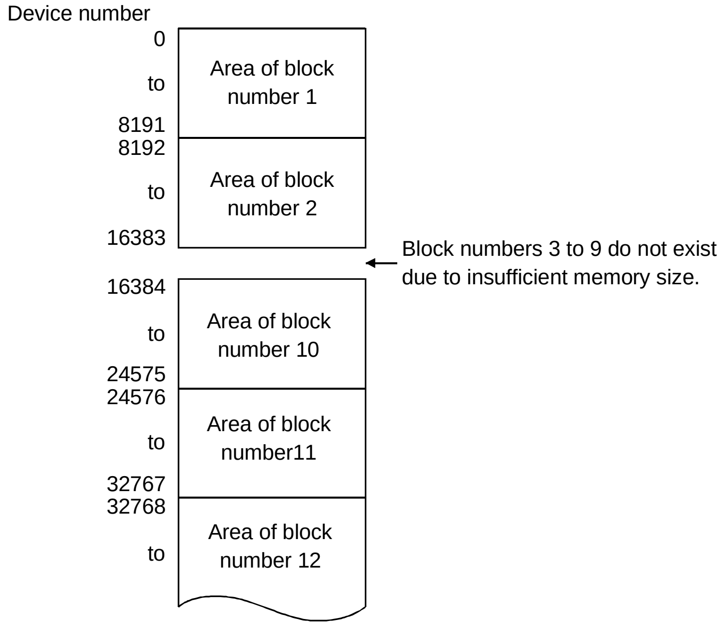
In the PLC CPU status column of the table above, ○ indicates that the corresponding function is executable and × indicate not executable.

(2) Device number of extension file register

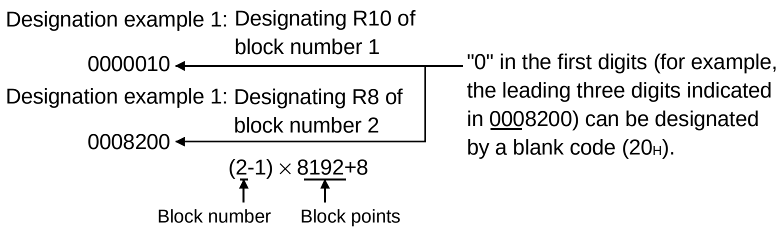
(a) The range of device numbers that can be designated is as follows.
0 to (number of available blocks × 8192) - 1



The device numbers that can be designated vary depending on the memory cassette type and the parameter settings of the PLC CPU.
(For more detailed explanations, see the operating manual of UTLP-FN1, or the User's Manual of the PLC CPU to be accessed.)
A device number is not assigned to a block number that 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.




(b) A device number is designated using 7 characters.




**POINT**

(1) The AnA/AnUCPU common commands, NR and NW can be used only when reading and writing data of block numbers 1 to 256 of the extension file register.
Also, they can be used regardless whether or not the file register parameter is set.

(2) When accessing the file register (R) set by the parameter or when accessing it by designating block number, use the commands described in Section 5.3.1.

(3) The head device number designated with the AnA/AnUCPU common commands, NR and NW, is calculated by the following expression.
(When designating device number m (0 to 8191) of block number n (the nth block from the top; "n" must be no smaller than 1))
Head device number = (n - 1) × 8192 + m

REMARK

The following table shows the range of device numbers designated when the AnA/AnUCPU common commands, NR and NW, are used for the first 28 blocks.

| 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 |

### 5.3.3 Precautions when reading and writing in the extension file register

The following precautions should be observed when reading and writing in the extension file register using the commands described in Sections 5.3.4 through 5.3.9.

(1) Only PLC CPUs that can handle an extension file register can be accessed. These functions cannot be used for PLC CPUs that cannot handle an extension file register (such as A1N).

(2) Depending on the type of memory cassette installed on the PLC CPU, an error (character area error $06_H$) may not be detected even when a read/write operation is performed on block numbers that do not exist. In this case, the data read is incorrect. Also, writing to these blocks may destroy the user memory of the PLC CPU.
Confirm 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 ($06_H$) | | |
|---|---|---|---|
| | 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 more details, see the operating manual of UTLP-FN1, or the User's Manual of the PLC CPU to be accessed.)

(3) The block numbers of the extension file register that can be handled by the A2USCPU(S1) is as follows.

- A2USCPU : No.'s 1 to 3
- A2USCPU-S1 : No.'s 1 to 8, No.'s 10 to 16

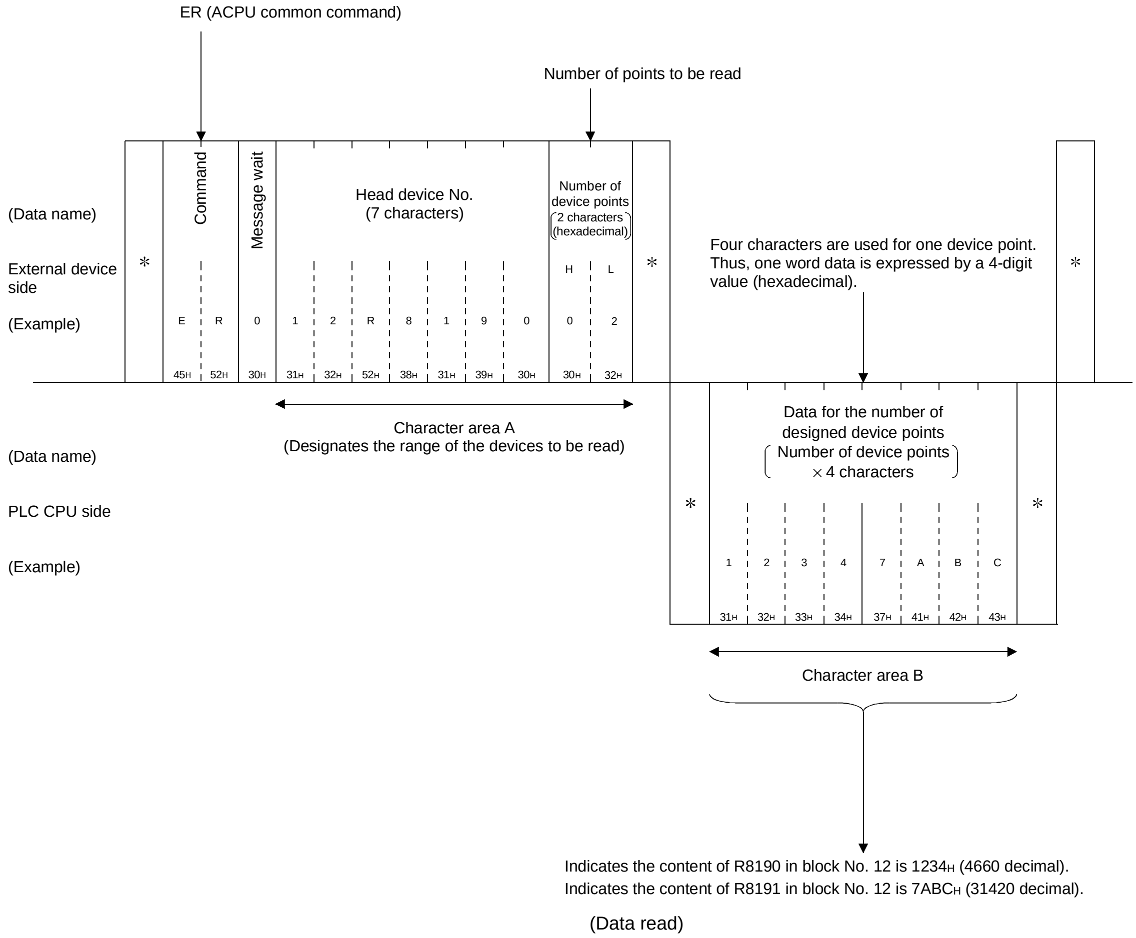
(4) The extension file register of Q/QnACPU cannot be read or written.

### 5.3.4 Batch reading of the extension file register (command: ER)

The example shown in this section explain the control procedure for batch reading from the extension file register using the ER command.
The order and contents of data items for the parts marked with "∗" shown in the control procedure differ depending on the communication format.
See a detailed explanation in Section 5.1.

[Control procedure]

(Read conditions)
1) Message wait is 0 ms.
2) Read 2 points from R8190 and R8191 of block No. 12




**POINT**

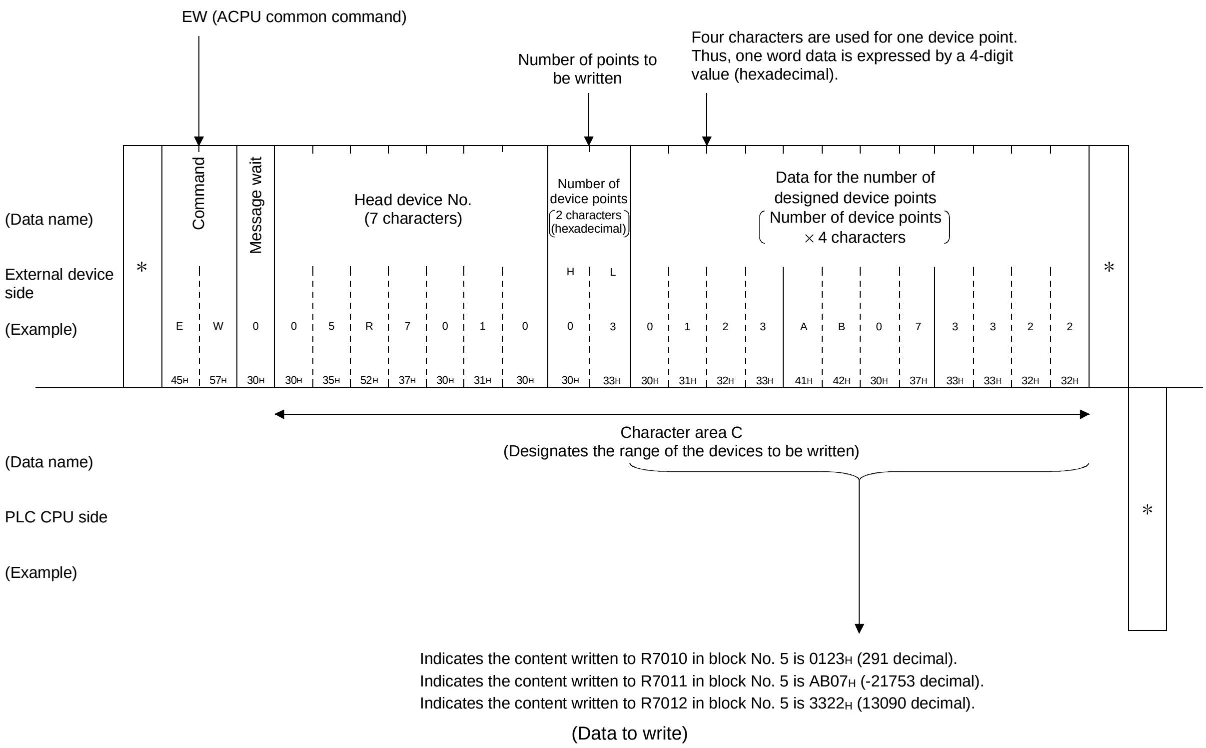
Designate the number of device points within the following range:
- 1 ≤ number of device points ≤ 64
- Head device number + number of device points - 1 ≤ maximum device number

### 5.3.5 Batch writing of the extension file register (command: EW)

The example shown in this section explain the control procedure for batch writing from the extension file register using the EW command.
The order and contents of data items for the parts marked with "∗" shown in the control procedure differ depending on the communication format.
See a detailed explanation in Section 5.1.

[Control procedure]

(Write conditions)
1) Message wait is 0 ms.
2) Read 3 points from R7010 to R7012 of block No. 5




**POINT**

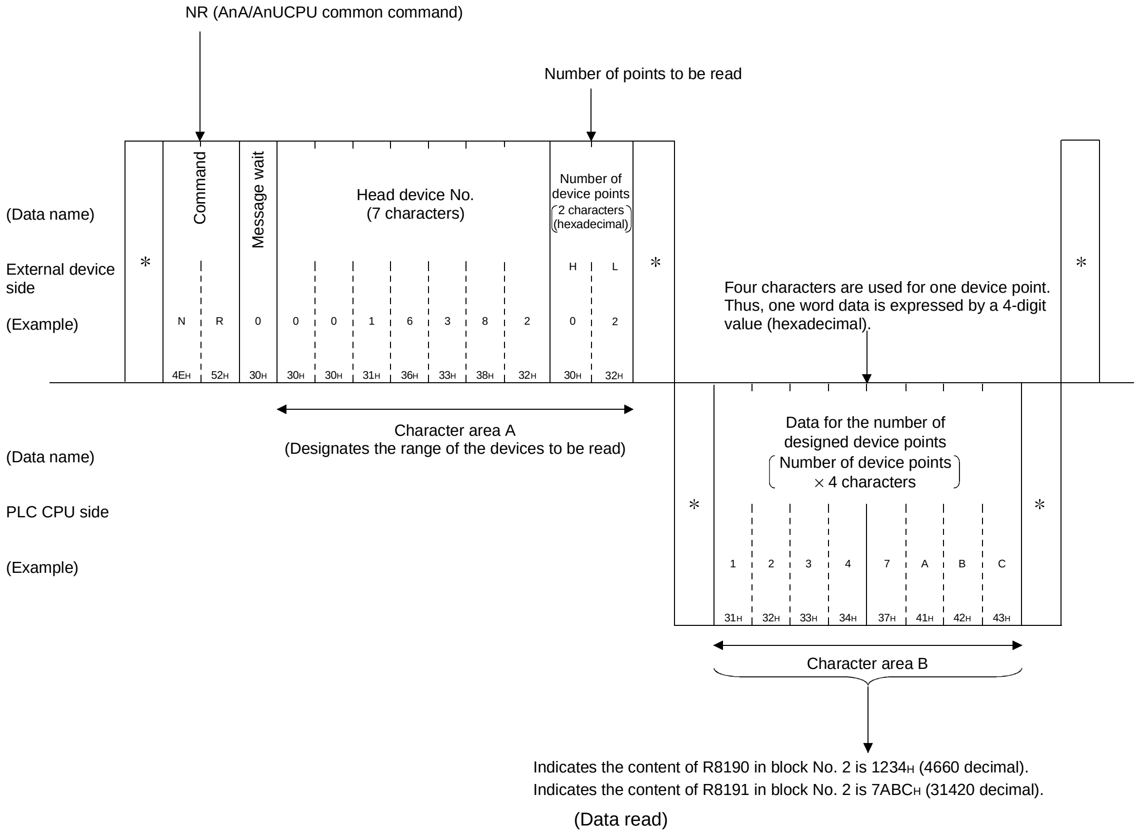
Designate the number of device points within the following range:
- 1 ≤ number of device points ≤ 64
- Head device number + number of device points - 1 ≤ maximum device number

### 5.3.6 Direct reading of the extension file register (command: NR)

The example shown in this section explain the control procedure for reading directly from the extension file register using the NR command.
The order and contents of data items for the parts marked with "∗" shown in the control procedure differ depending on the communication format.
See a detailed explanation in Section 5.1.

[Control procedure]

(Read conditions)
1) Message wait is 0 ms.
2) Read 2 points from R8190 and R8191 of block No. 2.




| POINT |
| --- |
| Designate the number of device points within the following range:<br>• 1 ≤ number of device points ≤ 64<br>• Head device number + number of device points - 1 ≤ maximum device number |

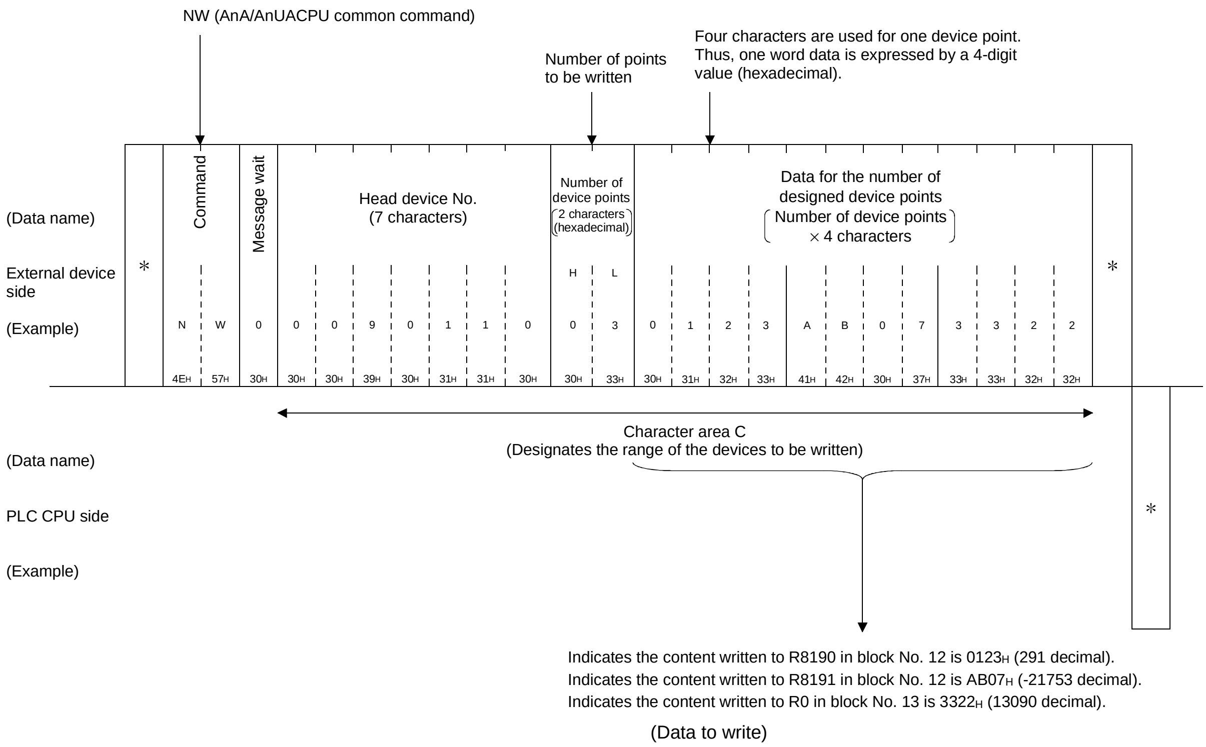
## 5.3.7 Direct writing of the extension file register (command: NW)

The example shown in this section explain the control procedure for writing directly from the extension file register using the NW command.
The order and contents of data items for the parts marked with "∗" shown in the control procedure differ depending on the communication format.
See a detailed explanation in Section 5.1.

[Control procedure]

(Write conditions)

1) Message wait is 0 ms.
2) Write 3 points to R8190 and R8191 of block No. 12 and R0 of block No.13.
   (When the extension file register of block No. 1 to 8 and 10 to 13 exist.)




**POINT**

Designate the number of device points within the following range:
- 1 ≤ number of device points ≤ 64
- Head device number + number of device points - 1 ≤ maximum device number

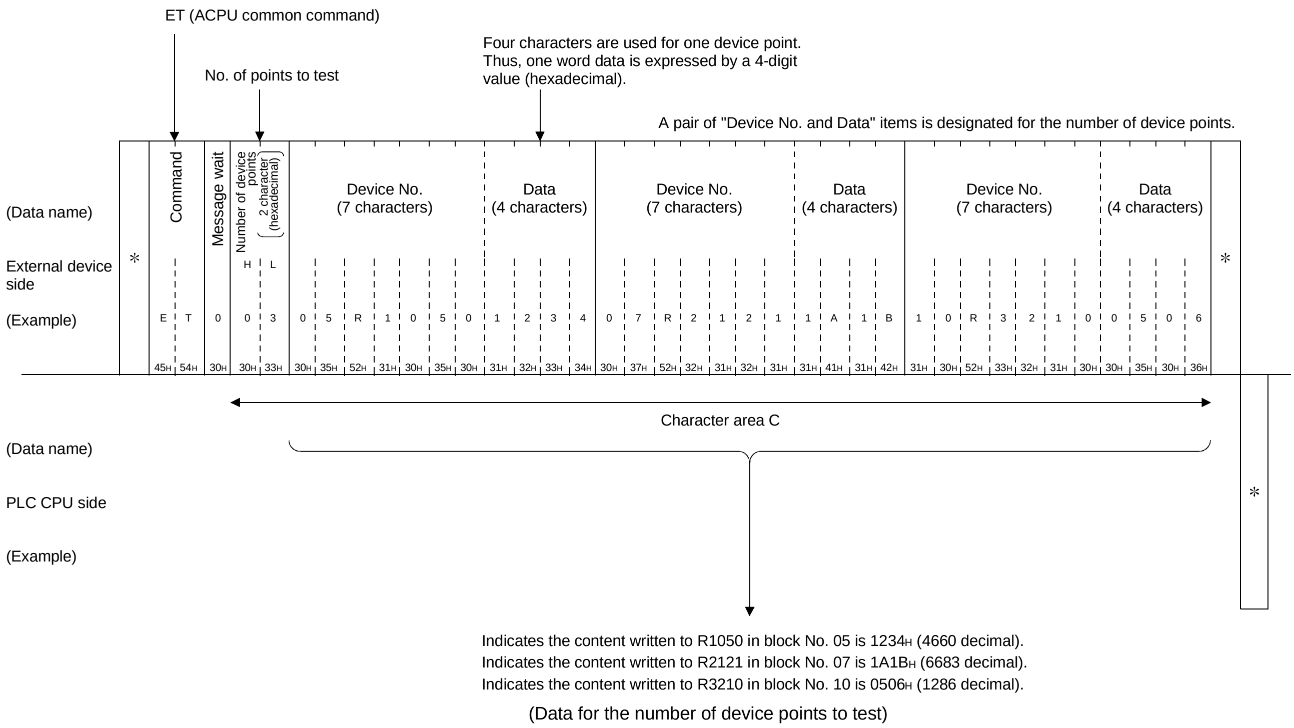
### 5.3.8 Test of extension file register (random write) (command: ET)

The example shown in this section explain the control procedure for writing data by randomly designating device numbers in the extension file register with the ET command.
The order and contents of data items for the parts marked with "*" shown in the control procedure differ depending on the communication format.
See a detailed explanation in Section 5.1.

[Control procedure]

(Write conditions)

1) Message wait is 0 ms.
2) Test 3 points (3 words) by registering R1234H to R1050 of block No. 5, 1A1BH to R2121 of block No. 7, and 506H to R3210 of block No.10.
(When the extension file register of block No. 1 to 8 and 10 exist.)

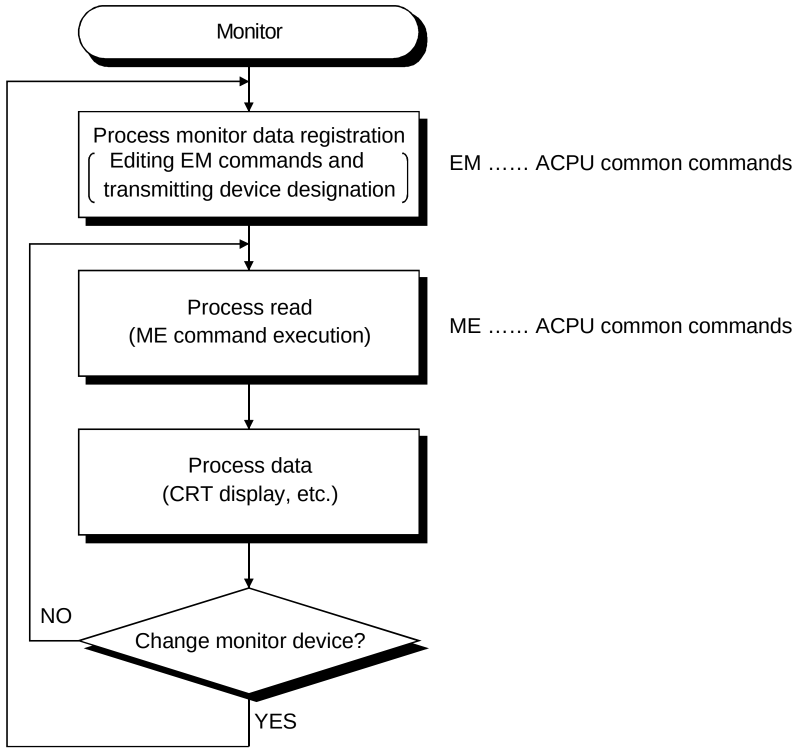



| POINT |
|---|
| Designate the number of device points within the following range:<br>• 1 ≤ number of device points ≤ 10 |

### 5.3.9 Monitoring extension file register

The monitor data registration function registers the name and number of the devices to be monitored from an external device to the Q series C24.
The monitor function allows to read the data contents of the registered devices from the PLC CPU and processing it in the external device.
When the batch read (ER) or direct read (NR) is used to read the devices, the device numbers must be consecutive. However, by using the monitor data registration function, devices can be monitored by designating the device numbers freely.
The following example illustrates the control procedure for monitoring and registering name and number of the devices to be monitored to the Q series C24.

(1) Monitoring procedure




| POINT |
|---|

(1) Monitor data must always be registered when monitoring using the procedure shown above. If the monitoring is performed without registering the monitor data, a protocol error will be generated.

(2) The contents of registered monitor data are deleted when the Q series C24 is rebooted.

(3) Five kinds of monitor data can be registered for each command of the extension file register (EM), and device memory in bit units (BM or JM) and word units (WM or QM).

(4) When multiple external devices registers the monitor data in the PLC CPU device memory of the same station, the registered data is overwritten, and the device memory that was registered last becomes valid.
See section 5.2.8 for details about monitoring the device memory.

(2) Monitor data registration of extension file register (command: EM)

The example shown in this section explain the control procedure for registering device numbers of the extension file register to be monitored to the Q series C24 with the EM command.

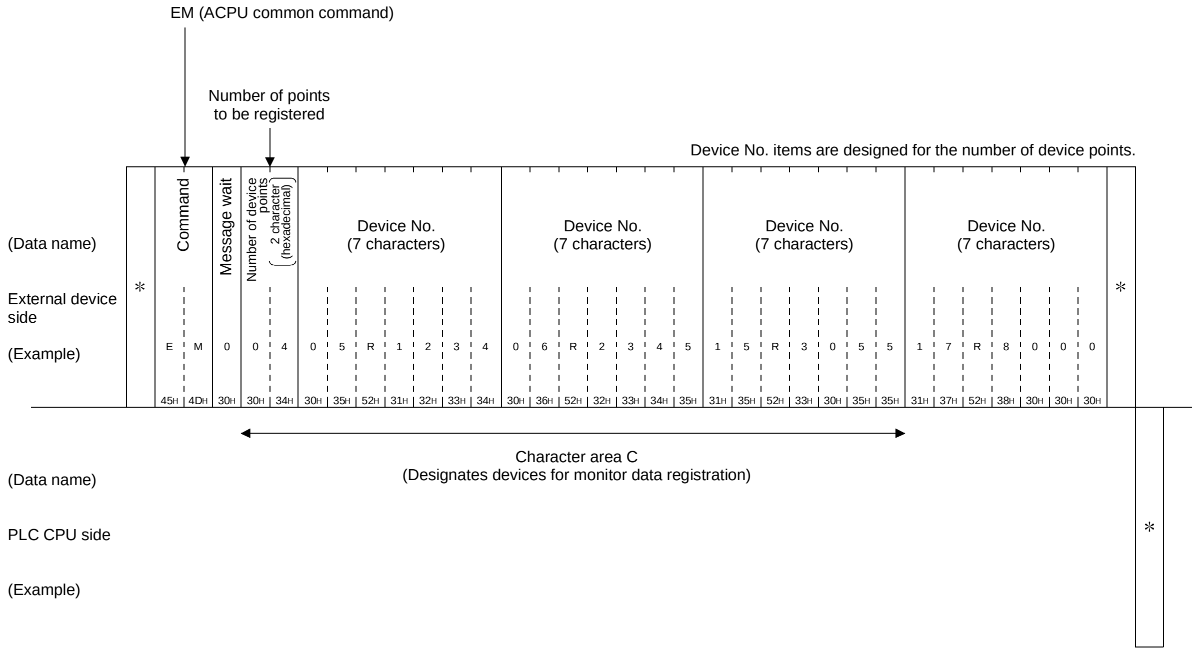
The order and contents of data items for the parts marked with "✻" shown in the control procedure differ depending on the communication format.

See a detailed explanation in Section 5.1.

[Control procedure]

(Monitoring data registration conditions)

1) Message wait is 0 ms.
2) Register monitor data for 4 points (4 words): R1234 of block No. 5, R2345 of block No. 6, R3055 of block No. 15, and R8000 of block No. 17.
(When the extension file register of block No. 1 to 8 and 10 to 17 exist.)




**POINT**

(1) Designate the number of device points within the following range:
- $1 \leq$ number of device points $\leq 20$

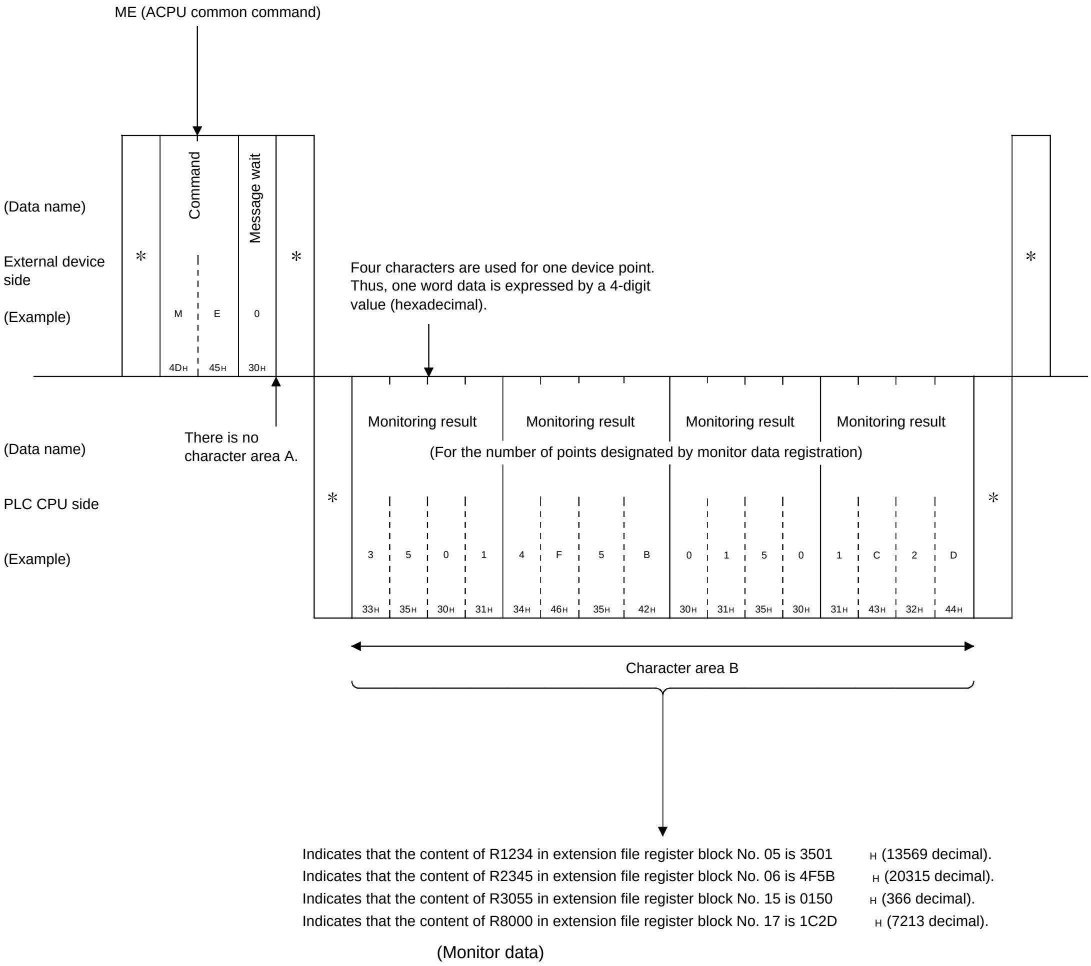
(2) Follow the explanation in item (3) of this section to monitor the extension register registered with the EM command.

(3) Monitor extension file register (command: ME)

The example shown in this section explain the control procedure for monitoring the extension file register that is registered in Q series C24 in item (2), "Monitor data registration (EM)" using the ME command.
The order and contents of data items for the parts marked with "∗" shown in the control procedure differ depending on the communication format.
See a detailed explanation in Section 5.1.

[Control procedure]

(Monitoring conditions)
1) Message wait is 0 ms.
2) Monitor 4 points (4 words): R1234 of block No. 5, R2345 of block No. 6, R3055 of block No. 15, and R8000 of block No. 17, for which monitor data is registered.
(When the extension file register of block No. 1 to 8 and 10 to 17 exist.)



## 5.4 Reading and Writing in the Buffer Memory of an Intelligent Function Module

The following examples explain the control procedure when reading data from and writing data to the buffer memory of an intelligent module (special function module) in an A series PLC CPU station or a MELSECNET/H, MELSECNET/10 remote I/O station.
This command accesses the buffer memory of an intelligent function module in byte units.

### 5.4.1 Commands and processing

(1) ACPU common commands

| Item | Command | | Processing | Number of points processed per communication | PLC CPU status | | | Reference section |
|---|---|---|---|---|---|---|---|---|
| | | | | | During STOP | During RUN | | |
| | Symbol | ASCII code | | | | Write allow setting | Write prohibit setting | |
| Batch read | TR | $54_H$, $52_H$ | Reads the buffer memory of an intelligent function module. | 128 bytes (64 words) | ○ | ○ | ○ | Section 5.4.3 |
| Batch write | TW | $54_H$, $57_H$ | Writes to the buffer memory of an intelligent function module. | | ○ | ○ | × | Section 5.4.4 |

In the PLC CPU status column of the table above, ○ indicates that the corresponding function is executable and × indicates that it is not executable.

(2) Accessible modules and addresses of the buffer memory

The module models in the A/QnA series that can be accessed using this function and the designated head address of the buffer memory are the same as when accessing an intelligent function module while communicating through QnA compatible 3E/3C/4C frames. See Sections 3.5.1 and 3.5.2. (Intelligent function modules in the Q series cannot be accessed.)
This function reads and writes data in byte units regardless of the word/byte units designation by GX Configurator-SC.

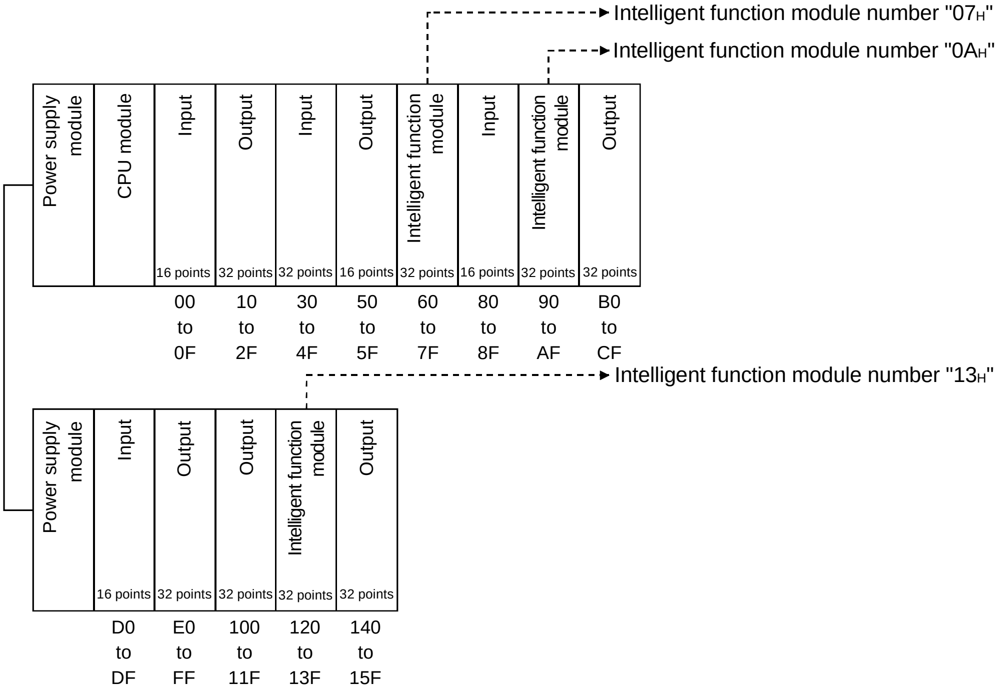
**POINT**

The buffer memory of each intelligent function module has read/write areas, read only areas, write only areas, and areas that may used by the OS but not by the user.
Execute this function according to the explanation in each module's manual.
If the read/write operations are not performed properly, an error may occur in the PLC CPU and each of the intelligent function modules.

### 5.4.2 Understanding the intelligent function module number in the control procedure

(1) Module number of an intelligent function module that occupies 1 slot

The intelligent function module number designated in the control procedure is the first 2 digits of the last (3-digit) number of the input/output signal (I/O address) of the slot on which the module is loaded.



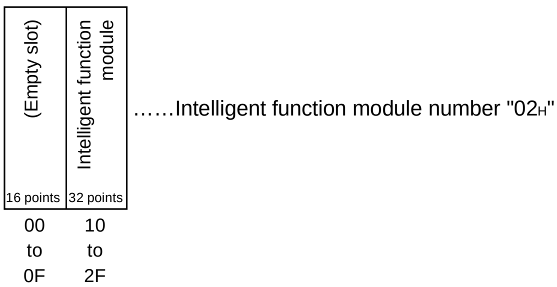

(2) Module number of an intelligent function module that occupies 2 slots

For an intelligent function module that occupies two slots, the number of occupied points for each slot is determined for each module.

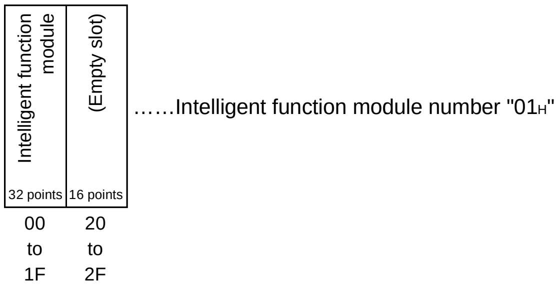
Out of the slots on which the module is loaded, the intelligent function module number designated in the control procedure is the first 2 digits of the last (3-digit) number of the input/output signal (I/O address) of the slots on the side assigned to the intelligent functional module.

For details on how each slot is assigned in each module, see the manual of the applicable module.

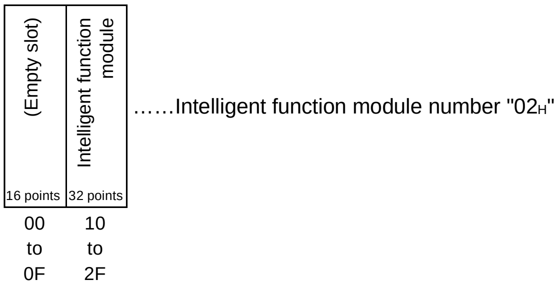
1) In case of a module that assign the first half slots as empty slots (AD72, A84AD, etc.)



2) In case of a module that assign the last half slots as empty slots (A61LS, etc.)

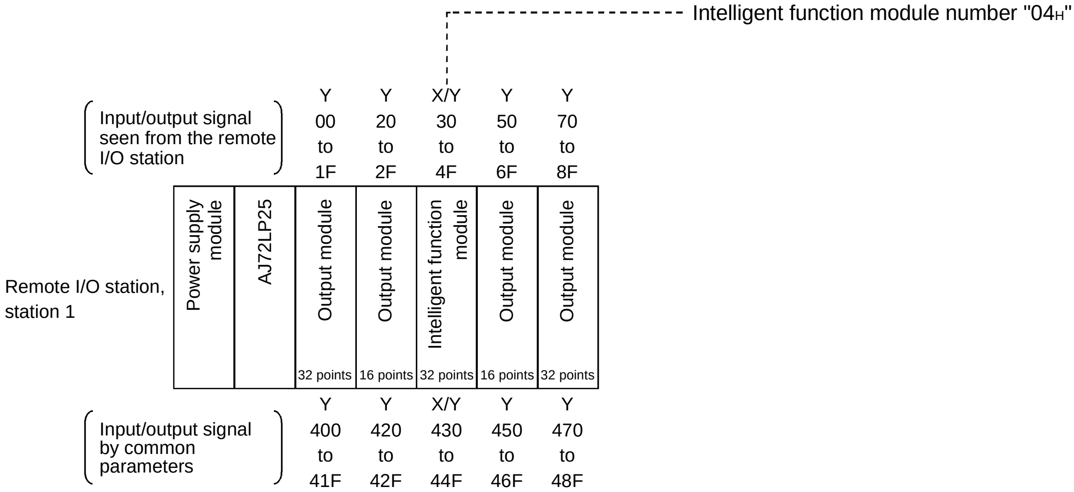



3) In case of a module in which both an intelligent function module and an input/output module are assigned
(In case of the A81CPU)




## (3) Intelligent function module number of the MELSECNET/H, MELSECNET/10 remote I/O station

All intelligent function module numbers of the MELSECNET/H, MELSECNET/10 remote I/O station are given by the first 2 digits of the last (3-digit) number of the "input/output signal seen from the remote I/O station" shown below.
Designate the intelligent function module number using the "input/output signal seen from the remote I/O station" regardless of the common parameters set in the master station of MELSECNET/H, MELSECNET/10 remote I/O net.



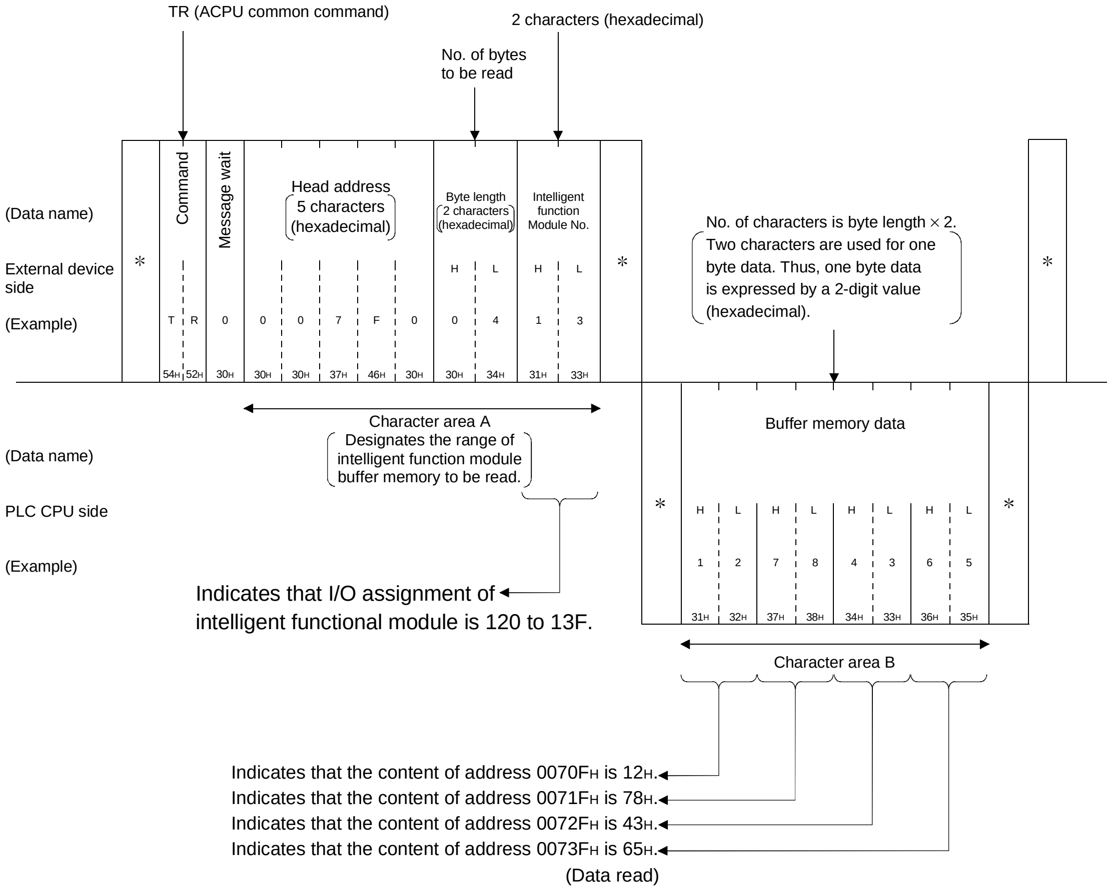
### 5.4.3 Reading from the buffer memory of an intelligent function module (command: TR)

The example shown in this section explain the control procedure for reading from the buffer memory of an intelligent module with the TR command.
The order and contents of data items for the parts marked with "*" shown in the control procedure differ depending on the communication format.
See a detailed explanation in Section 5.1.

[Control procedure]

(Read conditions)

1) Message wait is 0 ms.
2) Read 4 bytes from the buffer memory addresses 7F0H to 7F3H on an intelligent function module whose input/output signals are 120H to 13FH (module No. 13H).




**POINT**

(1) Designate the byte length within the following range:
- 1 ≤ byte length ≤ 128

(2) The content of one data may take up 2 or 3 bytes depending on the intelligent function module; designate the byte length by referring to the manual of the applicable module.

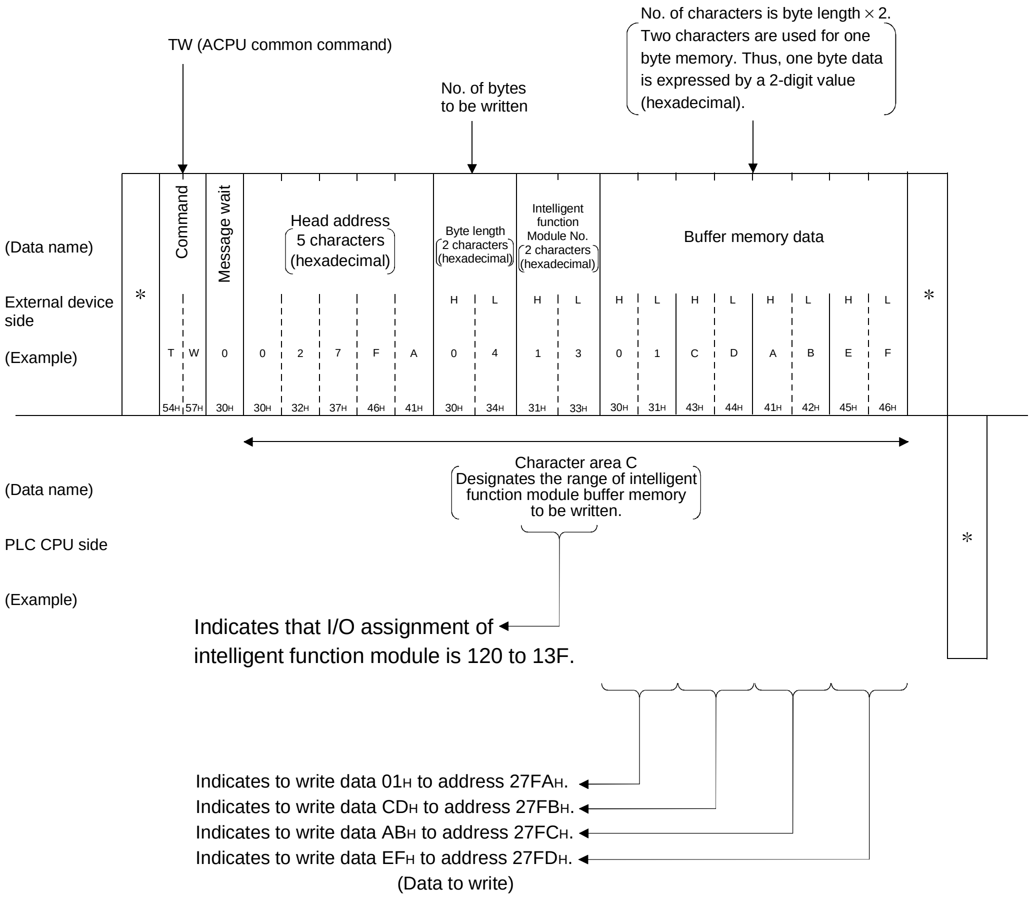
### 5.4.4 Writing to the buffer memory of an intelligent function module (command: TW)

The example shown in this section explain the control procedure for writing to the buffer memory of an intelligent module with the TW command.
The order and contents of data items for the parts marked with "✻" shown in the control procedure differ depending on the communication format.
See a detailed explanation in Section 5.1.

[Control procedure]

(Read conditions)

1) Message wait is 0 ms.
2) Write 4 bytes to the buffer memory addresses 27FAH to 27FDH on an intelligent function module whose input/output signals are 120H to 13FH (module No. 13H).




**POINT**

(1) Designate the byte length within the following range:
- 1 ≤ byte length ≤ 128

(2) The content of one data may take up 2 or 3 bytes depending on the intelligent function module; designate the byte length by referring to the manual of the applicable module.

## 5.5 Loopback Test

A loopback test checks whether or not the communication function between an external device and the Q series C24 module operates normally. The examples show the control procedure using this function.
The order and contents of data items for the parts marked with "∗" in the control procedure differ depending on the communication format.
See a detailed explanation in Section 5.1.

### (1) ACPU common commands and processing

The following table lists the command when the loopback test is performed with the TT command.

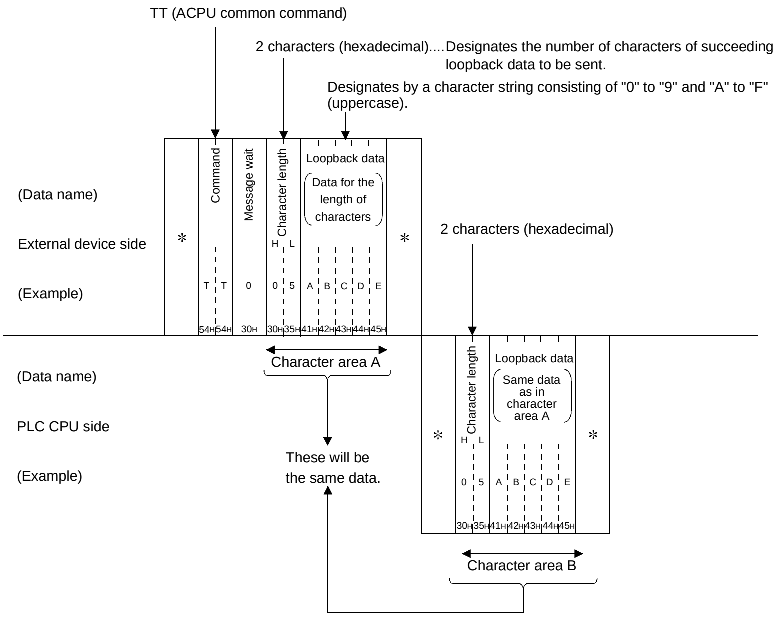
| Item | Command | | Processing | Number of points processed per communication | PLC CPU status | | |
|---|---|---|---|---|---|---|---|
| | | | | | | During RUN | |
| | Symbol | ASCII code | | | During STOP | Write allow setting | Write prohibit setting |
| Loopback test | TT | $54_H$, $54_H$ | Returns characters received from an external device to the external device unchanged. | 254 characters | ○ | ○ | ○ |

○ in the PLC CPU status column of the table above indicates that the corresponding function is executable.

### (2) The control procedure of a loopback test

[Control procedure]

(Loopback test conditions)
1) Message wait is 0 ms.
2) Send/receive the five characters in the string "ABCDE" as the loopback data




| POINT |
|---|
| (1) Designate the character length within the following range:<br>• 1 ≤ character length ≤ 254<br>(2) Designate "FF" for the PC number. |

# 6 WHEN COMMUNICATING USING THE A COMPATIBLE 1E FRAMES

This chapter explains the message format, how to designate data items in a message and restrictions when communicating data via the Q series E71 using the MC protocol and A compatible 1E frames.

| POINT |
|---|
| If the Q series C24 is used for data communication, it is not necessary to read this chapter. |

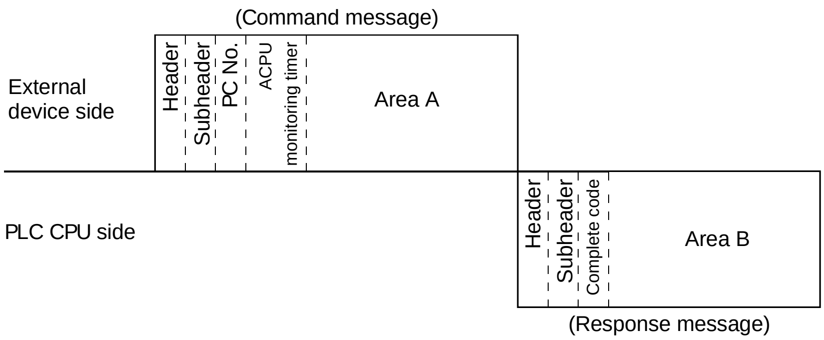
## 6.1 Message Formats and Control Procedures

This section explains the message format and control procedure for each command when data communication is performed using A compatible 1E frames.
Data communication using A compatible 1E frames is equivalent to the functions for reading/writing data from/to the PLC CPU supported by A series Ethernet modules, and only the commands mentioned in Section 6.2 can be used.

### 6.1.1 How to read the command reference section

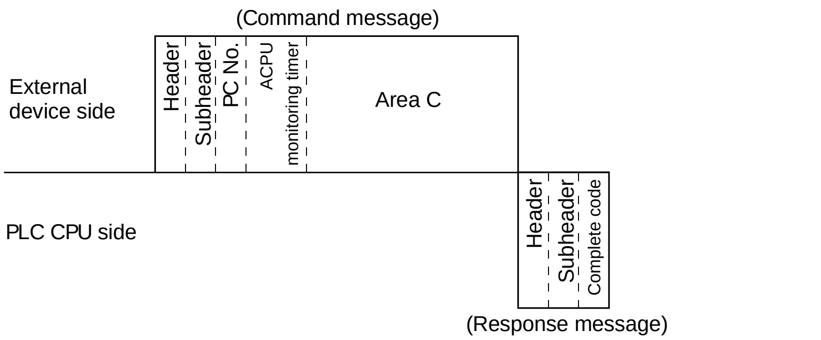
The following explains how to read the message explanation diagrams shown in each of the command description Sections 6.3 through 6.5.

(1) When an external device reads data from the PLC CPU




1) Area A indicates transmission from the external device to the PLC CPU.
2) Area B indicates transmission from the PLC CPU to the external device.
3) The program of the external device is generated so that the data is transmitted sequentially from left to right. (For example: in case of area A, data should be sequentially sent from Header to the right.)

(2) When an external device writes data to the PLC CPU




1) Area C indicates transmission from the external device to the PLC CPU.
2) Area B indicates transmission from the PLC CPU to the external device.
3) The program of the external device is generated so that the data is transmitted sequentially from left to right. (For example: in case of area C, data should be sequentially sent from Header to the right.)

**POINT**

When the PLC CPU receives a command message from an external device, it completes processing of the data in area A/C, then sends a response message and waits for the next command message (neutral state).

### 6.1.2 Message format and control procedure

This section explains the message format and control procedure when data communication is performed using A compatible 1E frames for the Q series E71.

#### (1) Message format

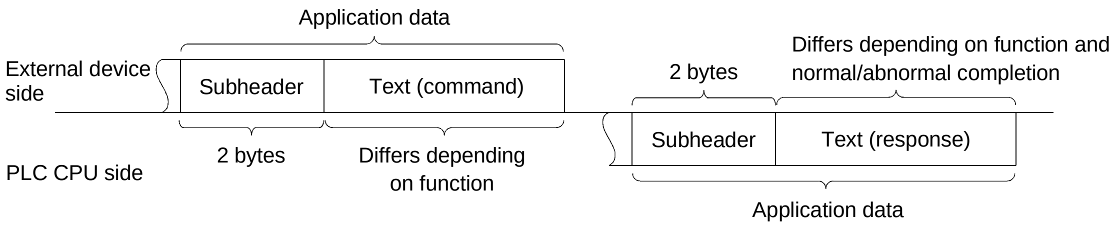
This section explains the message format for transmission between the Q series E71 and an external device.
The communication data consists of "header" and "application data" as shown below.

| Header | Application data |
|---|---|

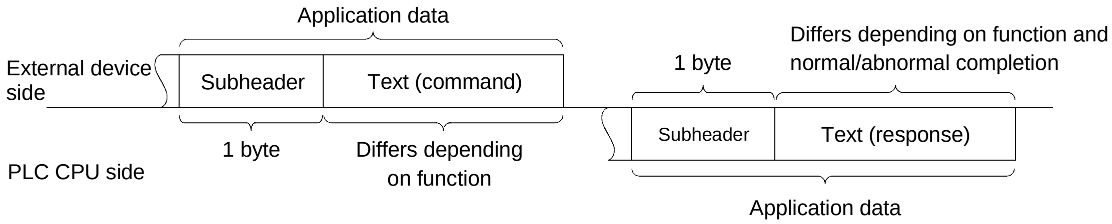
(a) Header
The header for TCP/IP or UDP/IP is used. The user does not need to specify it; the Q series E71 attaches it.

(b) Application data
The application data is largely divided into "subheader" and "text" as shown below.
The subheader represents command/response and the setting value is predetermined.
The text contains the request data (command) and response data (response) for each function and this data is determined by the prescribed format (for a more detailed description, see section 6.3 and succeeding sections).

(c) Format in the application data field
- Communication in ASCII code




- Communication in binary code




**REMARK**

When communicating in the MC protocol, the user does not need to specify a response for a command from an external device; the Q series E71 generates it and then responds.

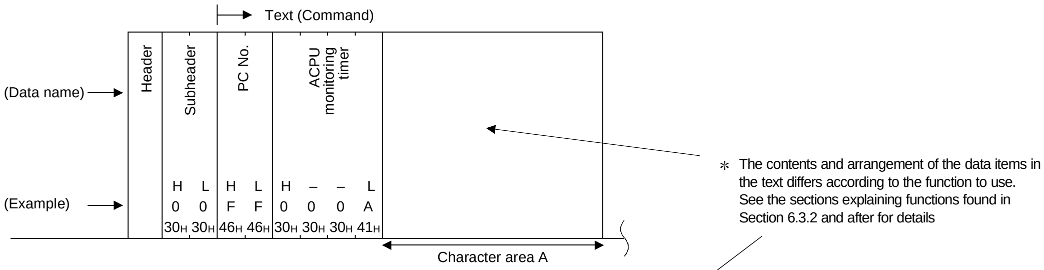
(2) Control procedure

The following diagrams illustrate the control procedure for communicating with A compatible 1E frame and the order of data items in the application data field.
The header section shown in the message explanation diagram of this section corresponds to the ✻ portion of the message explanation diagrams indicated in Section 6.3.2 of this chapter and after.
See Section 6.1.3 regarding the content of data items in the message format and data specification method.

(a) Communication in ASCII code

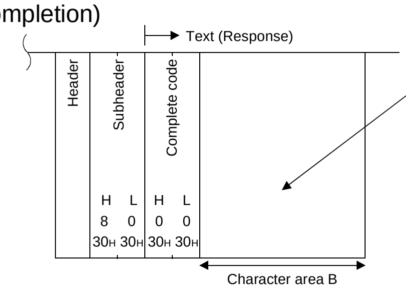
1) When reading data from the local station PLC CPU at the external device side

External device side → PLC CPU side (Command message)

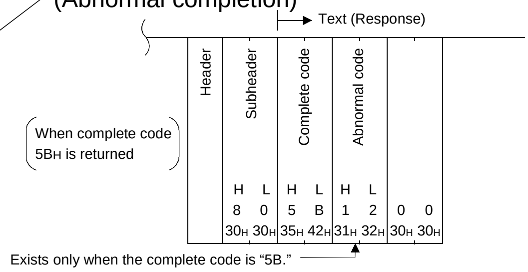



PLC CPU side → External device side (Response message)

(Normal completion)

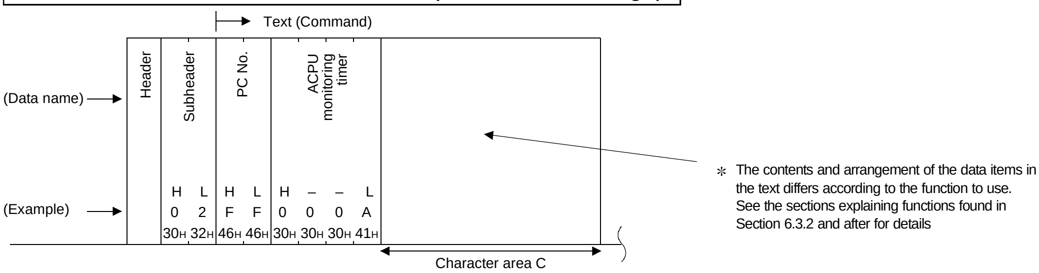



(Abnormal completion)



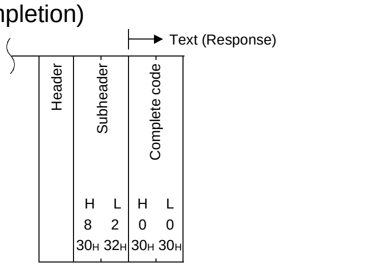

2) When writing data to the local station PLC CPU from the external device side

External device side → PLC CPU side (Command message)

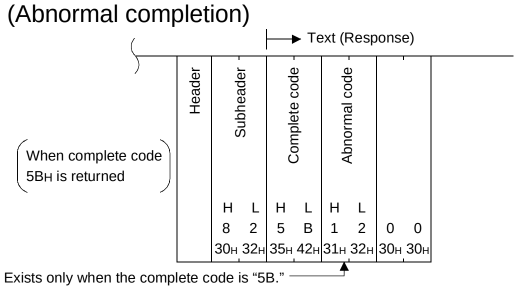



PLC CPU side → External device side (Response message)

(Normal completion)

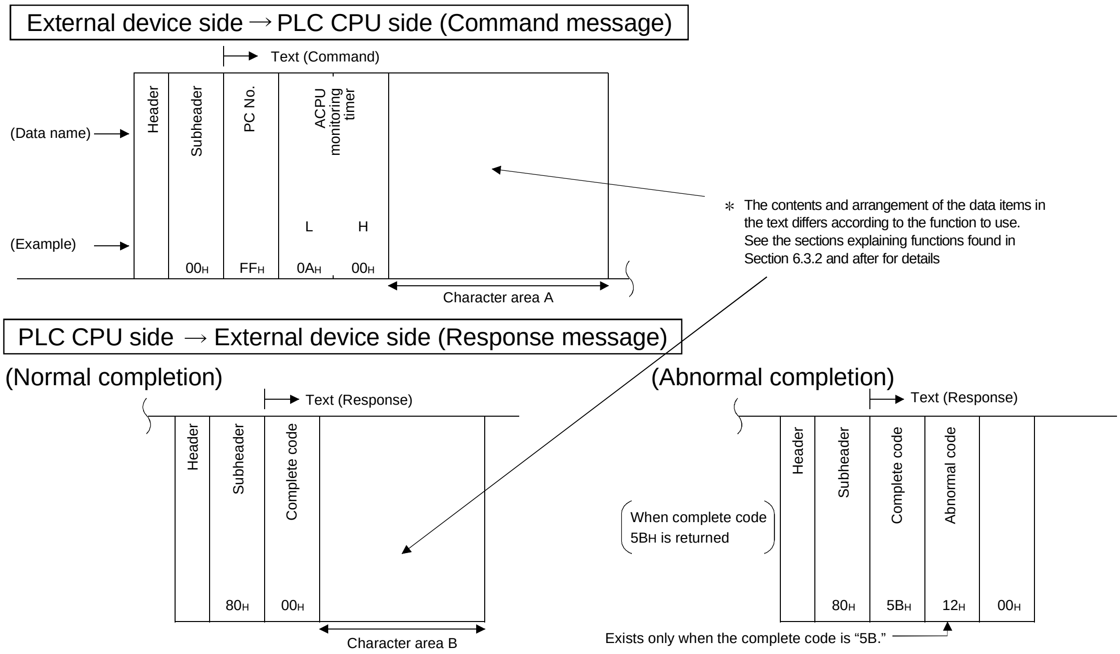



(Abnormal completion)

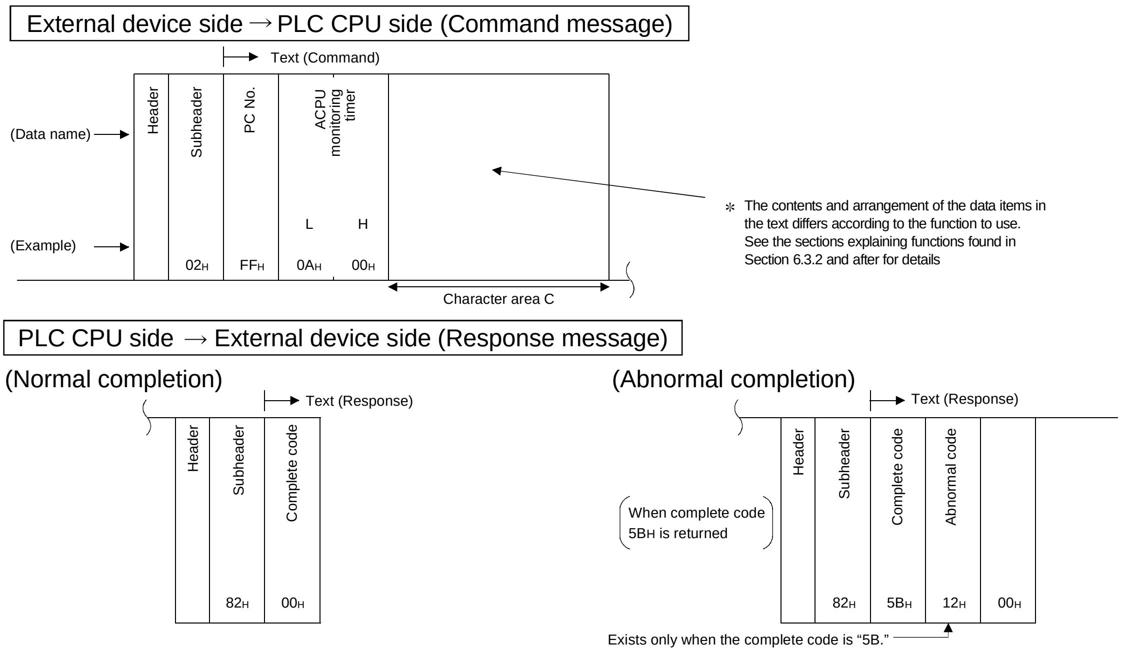


(b) Communication in binary code

1) When reading data from the local station PLC CPU at the external device side




2) When writing data to the local station PLC CPU from the external device side



### 6.1.3 Contents of data designation items of A compatible 1E frames

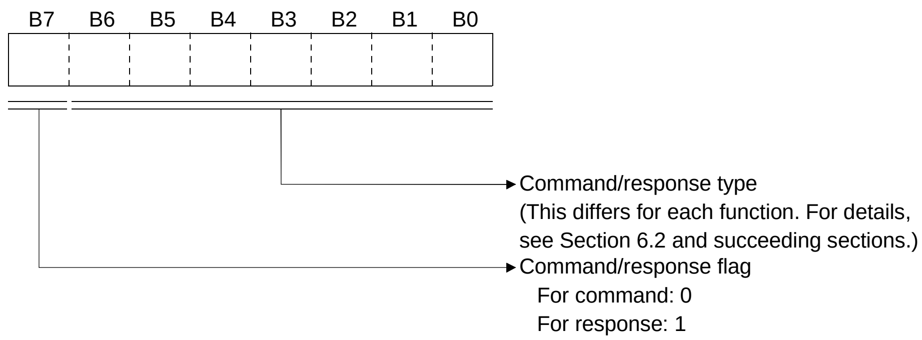
This section explains the data items of commands and responses when communicating using the MC protocol and A compatible 1E frames.
In the response that is returned by the Q series E71 to an external device, the data is automatically specified by the Q series E71; the user does not need to specify it.

(1) Header

The header for TCP/IP or UDP/IP is used. The user does not need to specify it at the PLC CPU side; the Q series E71 attaches it.

(2) Subheader

The format of the subheader is configured as follows.




(3) PC number

The PC numbers are used to identify which PLC on MELSECNET/H, MELSECNET/10 to be accessed.
The PC numbers are designated by FFH or in the station number range of the stations connected to the network module (maximum range 00H to 40H).

| | External device access station | Number designated by external device |
|---|---|---|
| 1 | Stations connected to an external device (local station) | FFH |
| 2 | Stations connected to an external connection station by multidrop link | FFH |
| 3 | Station on MELSECNET/H, MELSECNET/10<br>(excluding 1 and 2 above) ( ＊1) | 01H to 40H (1 to 64): Station number of access station.<br>＊ 00H can be designated when accessing the control station. |
| 4 | Remote master station on the MELSECNET/H (when an external device is connected to the Q series E71 in the remote I/O station) | (Not accessible) |

＊1 Applicable when accessing a modules on the network designated in the "Valid module during other station access" setting by a network module (Ethernet, MELSECNET/H, MELSECNET/10).

(a) When communicating in binary code, the PC number is expressed in binary code.
(b) When communicating in ASCII code, the PC number is expressed in hexadecimal ASCII code.

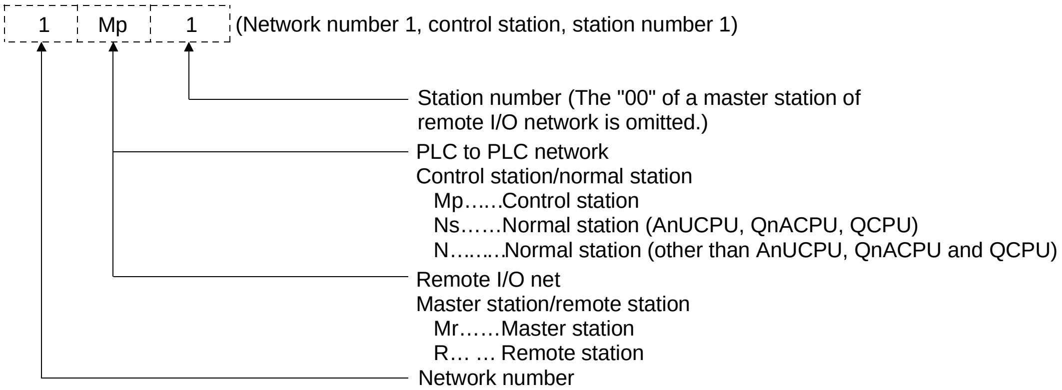
**POINT**

In order to access PLC CPUs of other stations, the appropriate network parameters or "Valid module during other station access" must be set for the Q series E71 or MELSECNET/H, MELSECNET/10.
＊ For detailed information, see the applicable manual for the Q series E71 or MELSECNET/H, MELSECNET/10.

[Example of PC number designation]

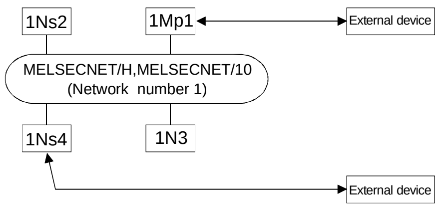
(Meaning of the symbols for each station shown in the diagrams)

- Network system (MELSECNET/H, MELSECNET/10)




In case of an PLC to PLC network of MELSECNET/H, MELSECNET/10

- MELSECNET/H, MELSECNET/10 two-tier system

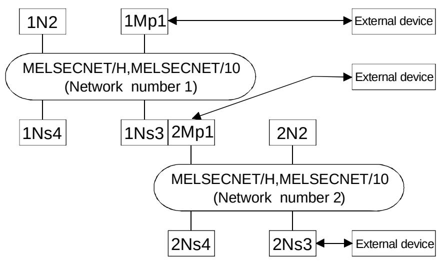



PC number to be designated

| Station connected to external device | PLC CPU accessed by an external device | | | |
|---|---|---|---|---|
| | 1Mp1 | 1Ns2 | 1N3 | 1Ns4 |
| 1Mp1 | FF | 02 | 03 | 04 |
| 1Ns4 | 01 | 02 | 03 | FF |

×: Not accessible

- MELSECNET/H, MELSECNET/10 hierarchical system



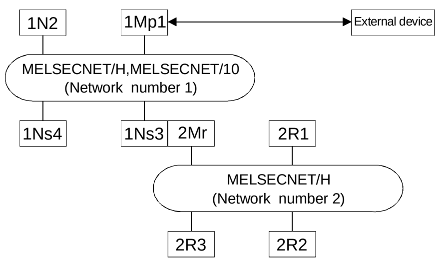

PC number to be designated

| Station connected to external device | PLC CPU accessed by an external device | | | | | | |
|---|---|---|---|---|---|---|---|
| | 1Mp1 | 1N2 | 1Ns3/ 2Mp1 | 1Ns4 | 2N2 | 2Ns3 | 2Ns4 |
| 1Mp1 | FF | 02 | 03 | 04 | × | | |
| 1Ns3/ 2Mp1 | 01 | 02 | FF | 04 | 02 | 03 | 04 |
| 2Ns3 | × | | 01 | × | 02 | FF | 04 |

×: Not accessible

In case of a MELSECNET/H remote I/O net

- MELSECNET/H hierarchical system




PC number to be designated

| Station connected to external device | PLC CPU accessed by an external device | | | | | | |
|---|---|---|---|---|---|---|---|
| | 1Mp1 | 1N2 | 1Ns3/ 2MR | 1Ns4 | 2R1 | 2R2 | 2R3 |
| 1Mp1 | FF | 02 | 03 | 04 | 01 | × | |
| 1Ns3/ 2MR | 01 | 02 | FF | 04 | × | | |
| 2R2 | × | | | | | | |

×: Not accessible

(4) ACPU monitoring timer

This is a timer for setting the period of time that the Q series E71 (that has received request data from an external device) should wait after outputting a read/write request to the PLC CPU until the result is returned.

(a) Specify the value as shown below.

0000H (0): : Waits infinitely (∗1)
0001 to FFFFH (1 to 65535) : Wait time (unit is 250 ms)

∗1 It keeps waiting until a response is returned from the PLC CPU.

(b) For normal data communications, using a value within the following setting range is recommended.

| Setting range | Destination |
|---|---|
| 1 to 40H (0.25 to 10s) | Local station |
| 2 to 240H (0.5 to 60s) | Other stations accessed via MELSECNET/H or MELSECNET/10, or by router relay |

POINT

Accessing the QnACPU or ACPU for the first time requires full time duration preset for the CPU monitoring timer before receiving a response because the CPU type must be identified. Be sure to set a value within the setting range shown in the above (b).

(5) Text (command)

This data contains the Q series E71 commands that indicate functions for when an external device reads/writes data from/to the target PLC station.
The contents and order of data in the text (command) field differ depending on the function used.
The order of data items for each function is explained in each function's reference section from Section 6.2.

(6) Text (response)

This data contains data read/processing result when an external device reads/writes data from/to the target PLC station.
The contents and order of data in the text (response) field differ depending on the function used.
The order or data items at normal completion for each function is explained in each function's reference section from Section 6.2.

(7) Complete code

The result of processing when an external device reads/writes data from/to the target PLC station is indicated by the following values.
00H : Normal completion
Other than 00H : Abnormal completion (01H to B001H)

(a) When communicating in binary code, the complete code is expressed in binary values.

(b) When communicating in ASCII code, the complete code is expressed in hexadecimal ASCII code.

(c) When the complete code indicates abnormal completion, check the content and take an action according to the troubleshooting section of the Ethernet User's Manual (Basic).
When the complete code is 5BH/"5B," the data of the abnormal code (10H to 21H) and 00H/"00" are included immediately after.

(8) Abnormal code

This value indicates the nature of the error when the processing result of reading/writing data from/to the target PLC station by an external device is faulty and the complete code is $5B_H$/"5B." (Abnormal code: $10_H$ to $21_H$)

(a) When communicating in binary code, the abnormal code is expressed in binary values.

(b) When communicating in ASCII code, the abnormal code is expressed in hexadecimal ASCII code.

(c) Check the content and take an action according to the troubleshooting section of the Ethernet User's Manual (Basic).

**POINT**

The data code (ASCII/binary) when sending/receiving commands and responses between the Q series E71 and an external device is determined in the operation settings with GX Developer.
The external device communicating with Q series E71 should send the values specified in each data item in the commands and responses in the above setting using the code shown below. Also, it should receive the corresponding values in the code shown below.

In the explanation hereafter in this section, the values specified in each item in a command and response are shown in binary values.

(1) Communication in binary code
Unless specifically stated, the value shown in each explanation is sent/received in the designated order (L to H) since it is in binary.

(2) Communication in ASCII code
Unless specifically stated, the value shown in each explanation is converted to hexadecimal ASCII code and sent/received in the designated order (H to L).

**REMARK**

The following example shows the designation of the subheader to the ACPU monitoring timer when communicating using the MC protocol under the following conditions.

(Designated value)

| | | | | |
|---|---|---|---|---|
| • Target station | : | PLC CPU station on which the Q series E71 is loaded (local station) | : | FFH |
| • Function used | : | Device memory batch read (bit units) (command for A compatible 1E frames) | : | 00H |
| • ACPU monitoring timer value | : | 2500 ms | : | 000AH |

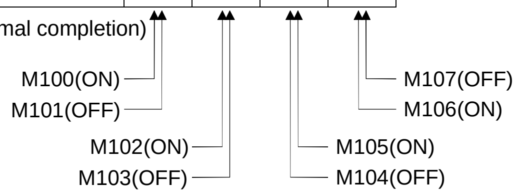
## (1) Format when communicating in binary code

### (a) The order when sending a command (external device → Q series E71)

| Header | Application data | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Subheader | PC number | ACPU monitoring timer | | Text (command) | | | | | | | |
| | | | | | (Head device number) | | | | (Device name) | | Number of device points | |
| | | | (L) | (H) | (L) | – | – | (H) | (L) | (H) | | |
| | 00H | FFH | 0AH | 00H | 64H | 00H | 00H | 00H | 20H | 4DH | 08H | 00H |
| | | (Local station) | (2500 ms) | | (100) | | | | (M) | | (8 points) | |

### (b) The order when receiving a response (external device ← Q series E71)

| Header | Application data | | | | | |
|---|---|---|---|---|---|---|
| | Subheader | Complete code | Text (response) on/off status of the designated device | | | |
| | 80H | 00H | 10H | 10H | 01H | 10H |
| | | (Normal completion) | | | | |



(2) Format when communicating in ASCII code

(a) The order when sending a command (external device → Q series E71)

| Header | Application data | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | Subheader | | PC number | | ACPU monitoring timer | | | |
| | (H) | (L) | (H) | (L) | (H) | | (L) | |
| | "0" | "0" | "F" | "F" | "0" | "0" | "0" | "A" |
| | 30H | 30H | 46H | 46H | 30H | 30H | 30H | 41H |
| | | | (Local station) | | (2500 ms) | | | |

| Application data | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Text (command) | | | | | | | | | | | | | | | |
| (Device name) | | | | (Head device number) | | | | | | | | Number of device points | | | |
| (H) | – | – | (L) | (H) | – | – | – | – | – | – | (L) | (H) | (L) | | |
| "4" | "D" | "2" | "0" | "0" | "0" | "0" | "0" | "0" | "0" | "6" | "4" | "3" | "8" | "0" | "0" |
| 34H | 44H | 32H | 30H | 30H | 30H | 30H | 30H | 30H | 30H | 36H | 34H | 30H | 38H | 30H | 30H |
| (M) | | | | (100) | | | | | | | | (8 points) | | | |

(b) The order when receiving a response (external device ← Q series E71)

| Header | Application data | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Subheader | | Complete code | | Text (response) ON/OFF status of the designated device | | | | | | | |
| | (H) | (L) | (H) | (L) | (H) | – | – | – | – | – | – | (L) |
| | "8" | "0" | "0" | "0" | "1" | "0" | "1" | "0" | "0" | "1" | "1" | "0" |
| | 38H | 30H | 30H | 30H | 31H | 30H | 31H | 30H | 30H | 31H | 31H | 30H |
| | | | (Normal completion) | | M100(ON) | M101(OFF) | M102(ON) | M103(OFF) | M104(OFF) | M105(ON) | M106(ON) | M107(OFF) |

## 6.1.4 Understanding transmission data in the character areas

The transmission order of bit device data and word device data indicated in the transmission data (each character area) when communicating using the MC protocol and A compatible 1E frames is the same as when communicating using QnA compatible 3E/3C/4C frames.
For more details, see Section 3.1.7.

**POINT**

The "additional code" data shown in the explanation of transmission data in Section 3.1.7 is the data specified by the Q series C24.
When communicating data with the Q series E71, ignore the explanation and diagrams related to the "additional code."

# MEMO

## 6.2 List of Commands and Functions for A Compatible 1E Frames

The following table lists the commands and functions when an external device accesses the PLC CPU using A compatible 1E frames.

| Function | | | Command/response type | Processing | Number of points processed per communication |
|---|---|---|---|---|---|
| Device memory (∗6) | Batch read | Bit units | 00H | Reads bit devices (X, Y, M, etc.) in 1-point units. | 256 points |
| | | Word units | 01H | Reads bit devices (X, Y, M, etc.) in 16-point units. | 128 words (2048 points) |
| | | | | Reads word devices (D, R, T, C, etc.) in 1-point units. | 256 points |
| | Batch write (∗5) | Bit units | 02H | Writes to bit devices (X, Y, M, etc.) in 1-point units. | 256 points |
| | | Word units | 03H | Writes to bit devices (X, Y, M, etc.) in 16-point units. | 40 words (640 points) |
| | | | | Writes to word devices (D, R, T, C, etc.) in 1-point units. | 256 points |
| | Test (random write) (∗5) | Bit units | 04H | Sets/resets bit devices (X, Y, M, etc.) in 1-point units by randomly designating the devices and device number. | 80 points |
| | | Word units | 05H | Sets/resets bit devices (X, Y, M, etc.) in 16-point units by randomly designating the devices and device numbers. | 40 words (640 points) |
| | | | | Writes to word devices (D, R, T, C, etc.) in 1-point units by randomly designating the devices and device numbers. | 40 points |
| | Monitor data registration (∗3) | Bit units | 06H | Registers bit devices (X, Y, M, etc.) to be monitored in 1-point units. (∗2) | 40 points |
| | | Word units | 07H | Registers bit devices (X, Y, M, etc.) to be monitored in 16-point units. (∗2) | 20 words (320 points) |
| | | | | Registers word devices (D, R, T, C, etc.) to be monitored in 1-point units. | 20 points |
| | Monitor | Bit units | 08H | Monitors devices with monitor data registered. | (Number of registrations points) |
| | | Word units | 09H | | |
| Extension file registers | Batch read | | 17H | Reads extension file register (R) in 1-point units. | 256 points |
| | Batch write | | 18H | Writes to extension file register (R) in 1-point units. | 256 points |
| | Test (random write) | | 19H | Writes to extension file register (R) in 1-point units by randomly designating the devices and device numbers. | 40 points |
| | Monitor data registration (∗3) | | 1AH | Registers extension file register (R) to be monitored in 1-point units. | 20 points |
| | Monitor | | 1BH | Monitors extension file register (R) that were performed for monitor data registration. | (Number of registrations points) |
| | Direct read | | 3BH | Reads extension file register (R) in 1-point units with direct designation. | 256 points |
| | Direct write | | 3CH | Writes to extension file register (R) in 1-point units with direct designation. | 256 points |
| Intelligent function module | Batch read | | 0EH | Reads the content in the buffer memory of an intelligent function module. | 256 bytes (128 words) |
| | Batch write | | 0FH | Writes data to the buffer memory of an intelligent function module. | |

| Function | | – | Access station (＊4) | | | | | | Status of the PLC CPU (＊1) | | | Reference section |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | A series CPU | QnA series CPU | Q series CPU | MELSECNET/10 remote station A series | MELSECNET/10 remote station QnA series | MELSECNET/H remote station Q series | During STOP | During RUN Write allow | During RUN Write prohibit | |
| Device memory (＊5) | Batch read | Bit units | ○ | ○ | ○ | × | × | × | ○ | ○ | ○ | Section 6.3.2 |
| | | Word units | ○ | ○ | ○ | × | × | × | | | | Section 6.3.5 |
| | Batch write | Bit units | ○ | ○ | ○ | × | × | × | ○ | ○ | × | Section 6.3.3 |
| | | Word units | ○ | ○ | ○ | × | × | × | | | | Section 6.3.6 |
| | Test (random write) | Bit units | ○ | ○ | ○ | × | × | × | ○ | ○ | × | Section 6.3.4 |
| | | Word units | ○ | ○ | ○ | × | × | × | | | | Section 6.3.7 |
| | Monitor data registration (＊3) | Bit units | ○ | ○ | ○ | × | × | × | ○ | ○ | ○ | Section 6.3.8 |
| | | Word units | ○ | ○ | ○ | × | × | × | | | | |
| | Monitor | Bit units | ○ | ○ | ○ | × | × | × | ○ | ○ | ○ | |
| | | Word units | ○ | ○ | ○ | × | × | × | | | | |
| Extension file registers | Batch read | | ○ | × | × | × | × | × | ○ | ○ | ○ | Section 6.4.3 |
| | Batch write | | ○ | × | × | × | × | × | ○ | ○ | × | Section 6.4.4 |
| | Test (random write) | | ○ | × | × | × | × | × | ○ | ○ | × | Section 6.4.5 |
| | Monitor data registration (＊3) | | ○ | × | × | × | × | × | ○ | ○ | ○ | Section 6.4.6 |
| | Monitor | | ○ | × | × | × | × | × | | | | |
| | Direct read | | ○ | × | × | × | × | × | | | | Section 6.4.7 |
| | Direct write | | ○ | × | × | × | × | × | | | | Section 6.4.7 |
| Intelligent function module | Batch read | | ○ | × | × | ○ | ○ | × | ○ | ○ | ○ | Section 6.5.3 |
| | Batch write | | ○ | × | × | ○ | ○ | × | ○ | ○ | × | Section 6.5.4 |

*1 The write allow/prohibit during RUN setting for the PLC CPU is set on the following screen of GX Developer.
- In case of the Q series E71, use the "Ethernet operation setting" screen.

*2 For PLC CPUs other than the AnA/AnU/QnA/QCPU, device X (input) has two processing points per point. When the designated device includes X, the following condition should be met:
((designated points for an X device × 2) + designated points for other devices) ≤ number of points processed per communication.
When only X is designated, the number of points processed per communication becomes one half the value given in the table.

*3 The device for 1 command of the three types of commands ($06_H$, $07_H$, $1A_H$) can be registered in the Q series E71.
Devices designated by the command used at the end of one of the above are registered in the Q series E71.

*4 See section 2.6.1 for the details of the access stations shown in the table.

*5 If there is system protection on the Q/QnACPU that executes these commands, an error occurs and an abnormal completion response is returned.

*6 Use dedicated commands for extension registers when reading and writing extension file registers to modules other than Q/QnACPU.

## 6.3 Device Memory Read/Write

This section explains the designations in the control procedure when reading from and writing to the device memory by providing an example.

### 6.3.1 Commands and device range

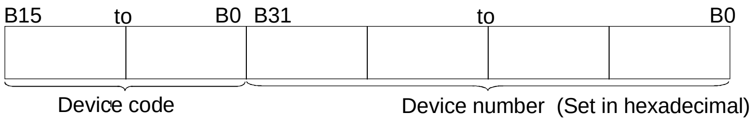
(1) Commands used for reading from and writing to the device memory

| Item | | Command/response type | Processing | Number of points processed per communication | PLC CPU status (∗1) During STOP | PLC CPU status (∗1) During RUN: Write allow setting | PLC CPU status (∗1) During RUN: Write prohibit setting |
|---|---|---|---|---|---|---|---|
| Batch read | Bit units | $00_H$ | Reads bit devices (X, Y, M, etc.) in 1-point units. | 256 points | ○ | ○ | ○ |
| | Word units | $01_H$ | Reads bit devices (X, Y, M, etc.) in 16-point units. | 128 words (2048 points) | | | |
| | | | Reads word devices (D, R, T, C, etc.) in 1-point units. | 256 points | | | |
| Batch write (∗4) | Bit units | $02_H$ | Writes to bit devices (X, Y, M, etc.) in 1-point units. | 256 points | ○ | ○ | × |
| | Word units | $03_H$ | Writes to bit devices (X, Y, M, etc.) in 16-point units. | 40 words (640 points) | | | |
| | | | Writes to word devices (D, R, T, C, etc.) in 1-point units. | 256 points | | | |
| Test (random write) (∗4) | Bit units | $04_H$ | Sets/resets bit devices (X, Y, M, etc.) in 1-point units by randomly designating the devices and device number. | 80 points | ○ | ○ | × |
| | Word units | $05_H$ | Sets/resets bit devices (X, Y, M, etc.) in 16-point units by randomly designating the devices and device numbers. | 40 words (640 points) | | | |
| | | | Writes to word devices (D, R, T, C, etc.) in 1-point units by randomly designating the devices and device numbers. | 40 points | | | |
| Monitor data registration (∗3) | Bit units | $06_H$ | Registers bit devices (X, Y, M, etc.) to be monitored in 1-point units (∗2) | 40 points | ○ | ○ | ○ |
| | Word units | $07_H$ | Registers bit devices (X, Y, M, etc.) to be monitored in 16-point units (∗2) | 20 words (320 points) | | | |
| | | | Registers word devices (D, R, T, C, etc.) to be monitored in 1-point units. | 20 points | | | |
| Monitor | Bit units | $08_H$ | Monitors devices with monitor data registered. | (Number of registrations points) | ○ | ○ | ○ |
| | Word units | $09_H$ | | | | | |

For ∗1, ∗2 and ∗3, see ∗1, ∗2 and ∗3 in Section 6.2.
For ∗4, see ∗5 in Section 6.2.

(2) Designation method and accessible range of devices.

(a) In device read/write, each device is set by a device code and number as shown in the diagram below.




[Example] In case of D100
D100 = 4420 00000064H
→ Device number (hexadecimal)
→ Device code

(b) The following table outlines the device codes and numbers.

## Device list (CPU module without restriction)

○ : Accessible × : Not accessible – : No device.

| Device (＊1) | | Device code | Device range (＊1) | Device number | A1S<br>A1SH<br>A1SJ<br>A1SJH<br>A1<br>A1N | A2S<br>A2SH<br>A2<br>A2N<br>A2C<br>A2CJ<br>A0J2H | A2-S1<br>A2N-S1 | A3<br>A3N | A2A | A2A-S1 | A3A |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Data register | | D0<br>(44H, 20H) | D0 to D1023 | 0000H to 03FFH | ○ | ○ | | | ○ | | |
| | | | D1024 to D6143 | 0400H to 17FFH | – | – | | | ○ | | |
| | | | D9000 to D9255 | 2328H to 2427H | ○ | ○ | | | ○ | | |
| Link register | | W0<br>(57H, 20H) | W0 to W3FF | 0000H to 03FFH | ○ | ○ | | | ○ | | |
| | | | W400 to WFFF | 0400H to 0FFFH | – | – | | | ○ | | |
| File register | | R0<br>(52H, 20H) | R0 to R4095 | 0000H to 0FFFH | – | ○ | | | ○ | | |
| | | | R4096 to R8191 | 1000H to 1FFFH | – | – | | ○ | ○ | | |
| Timer | Current value | TN<br>(54H, 4EH) | T0 to T255 | 0000H to 00FFH | ○ | ○ | | | ○ | | |
| | | | T256 to T2047 | 0100H to 07FFH | – | – | | | ○ | | |
| | Contact | TS<br>(54H, 53H) | T0 to T255 | 0000H to 00FFH | ○ | ○ | | | ○ | | |
| | | | T256 to T2047 | 0100H to 07FFH | – | – | | | ○ | | |
| | Coil | TC<br>(54H, 43H) | T0 to 255 | 0000H to 00FFH | ○ | ○ | | | ○ | | |
| | | | T256 to T2047 | 0100H to 07FFH | – | – | | | ○ | | |
| Counter | Current value | CN<br>(43H, 4EH) | C0 to C255 | 0000H to 00FFH | ○ | ○ | | | ○ | | |
| | | | C256 to C1023 | 0100H to 03FFH | – | – | | | ○ | | |
| | Contact | CS<br>(43H, 53H) | C0 to C255 | 0000H to 00FFH | ○ | ○ | | | ○ | | |
| | | | C256 to C1023 | 0100H to 03FFH | – | – | | | ○ | | |
| | Coil | CC<br>(43H, 43H) | C0 to C255 | 0000H to 00FFH | ○ | ○ | | | ○ | | |
| | | | C256 to C1023 | 0100H to 03FFH | – | – | | | ○ | | |
| Input | | X0<br>(58H, 20H) | 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<br>(59H, 20H) | 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<br>＊Including latch relay and step relay | | M0<br>(4DH, 20H) | M0 to M2047 | 0000H to 07FFH | ○ | ○ | | | ○ | | |
| | | | M2048 to M8191 | 0800H to 1FFFH | – | – | | | ○ | | |
| | | | M9000 to M9255 | 2328H to 2427H | ○ | ○ | | | ○ | | |
| Link relay | | B0<br>(42H, 20H) | B0 to B3FF | 0000H to 03FFH | ○ | ○ | | | ○ | | |
| | | | B400 to BFFF | 0400H to 0FFFH | – | – | | | ○ | | |
| Annunciator | | F0<br>(46H, 20H) | F0 to F255 | 0000H to 00FFH | ○ | ○ | | | ○ | | |
| | | | F256 to F2047 | 0100H to 07FFH | – | – | | | ○ | | |

(Continued to the next page)

## Device list (CPU module with restriction)

○ : Accessible  × : Not accessible  – : No device.

| Device (＊1) | | Device code | Device range (＊1) | Device number | Q00J Q00 Q01 | Q02, Q02H, Q06H, Q12H, Q25H, Q12PH, Q25PH, Q12PRH, Q25PRH | QJ72LP25-25 QJ72LP25GE QJ72BR15 |
|---|---|---|---|---|---|---|---|
| Data register | | D0 (44H, 20H) | D0 to D6143 | 0000H to 17FFH | ○ | ○ | × |
| Data register | | D0 (44H, 20H) | D6144 or more | 1800H or more | × | × | × |
| Data register | | D0 (44H, 20H) | D9000 to D9255 (SD1000 to SD1255) | 2328H to 2427H | ○ | ○ | – |
| Data register | | D0 (44H, 20H) | (SD1256 to SD2047) | | × | × | – |
| Link register | | W0 (57H, 20H) | W0 to W7FF | 0000H to 7FFFH | ○ | ○ | × |
| Link register | | W0 (57H, 20H) | W800 to WFFF | 800H to 0FFFH | – | ○ | × |
| Link register | | W0 (57H, 20H) | W1000 or more | 1000H or more | × | × | × |
| File register | | R0 (52H, 20H) | R0 or more | 0000H or more | × | × | – |
| Timer | Current value | TN (54H, 4EH) | T0 to T511 | 0000H to 01FFH | ○ | ○ | – |
| Timer | Current value | TN (54H, 4EH) | T512 to T2047 | 0200H to 07FFH | – | ○ | – |
| Timer | Current value | TN (54H, 4EH) | T2048 or more | 0800H or more | × | × | – |
| Timer | Contact | TS (54H, 53H) | T0 to T511 | 0000H to 01FFH | ○ | ○ | – |
| Timer | Contact | TS (54H, 53H) | T512 to T2047 | 0200H to 07FFH | – | ○ | – |
| Timer | Contact | TS (54H, 53H) | T2048 or more | 0800H or more | × | × | – |
| Timer | Coil | TC (54H, 43H) | T0 to T511 | 0000H to 01FFH | ○ | ○ | – |
| Timer | Coil | TC (54H, 43H) | T512 to T2047 | 0200H to 07FFH | – | ○ | – |
| Timer | Coil | TC (54H, 43H) | T2048 or more | 0800H or more | × | × | – |
| Counter | Current value | CN (43H, 4EH) | C0 to C511 | 0000H to 01FFH | ○ | ○ | – |
| Counter | Current value | CN (43H, 4EH) | C512 to C1023 | 0200H to 03FFH | – | ○ | – |
| Counter | Current value | CN (43H, 4EH) | C1024 or more | 0400H or more | × | × | – |
| Counter | Contact | CS (43H, 53H) | C0 to C511 | 0000H to 01FFH | ○ | ○ | – |
| Counter | Contact | CS (43H, 53H) | C512 to C1023 | 0200H to 03FFH | – | ○ | – |
| Counter | Contact | CS (43H, 53H) | C1024 or more | 0400H or more | × | × | – |
| Counter | Coil | CC (43H, 43H) | C0 to C511 | 0000H to 01FFH | ○ | ○ | – |
| Counter | Coil | CC (43H, 43H) | C512 to C1023 | 0200H to 03FFH | – | ○ | – |
| Counter | Coil | CC (43H, 43H) | C1024 or more | 0400H or more | × | × | – |
| Input | | X0 (58H, 20H) | X0 to X7FF | 0000H to 07FFH | ○ | ○ | × |
| Input | | X0 (58H, 20H) | X800 or more | 800H or more | × | × | – |
| Output | | Y0 (59H, 20H) | Y0 to Y7FF | 0000H to 07FFH | ○ | ○ | × |
| Output | | Y0 (59H, 20H) | Y800 or more | 800H or more | × | × | – |
| Internal relay | | M0 (4DH, 20H) | M0 to M8191 | 0000H to 1FFFH | ○ | ○ | × |
| Internal relay | | M0 (4DH, 20H) | M8192 or more | 2000H or more | × | × | – |
| Internal relay | | M0 (4DH, 20H) | M9000 to M9255 (SM1000 to SM1255) | 2328H to 2427H | ○ | ○ | – |
| Internal relay | | M0 (4DH, 20H) | (SM1256 to SM2047) | | × | × | – |
| Latch relay | | | | | × ＊ Even if the latch relay (L) is designated, it accesses to the internal relay (M) | × ＊ Even if the latch relay (L) is designated, it accesses to the internal relay (M) | – |
| Step relay | | | | | × ＊ Even if the latch relay (L) is designated, it accesses to the internal relay (M) | × ＊ Even if the latch relay (L) is designated, it accesses to the internal relay (M) | – |
| Link relay | | B0 (42H, 20H) | B0 to B7FF | 0000H to 07FFH | ○ | ○ | × |
| Link relay | | B0 (42H, 20H) | B800 to BFFF | 0800H to 0FFFH | – | ○ | × |
| Link relay | | B0 (42H, 20H) | B1000 or more | 1000H or more | × | × | × |
| Annunciator | | F0 (46H, 20H) | F0 to F1023 | 0000H to 03FFH | ○ | ○ | – |
| Annunciator | | F0 (46H, 20H) | F1024 to F2047 | 0400H to 07FFH | – | ○ | – |
| Annunciator | | F0 (46H, 20H) | F2048 or more | 0800H or more | × | × | – |

(Continue to the next page)

## Device list (CPU module with restriction)

○ : Accessible × : Not accessible – : No device.

<table>
<tr><th colspan="2">Device ( * 1)</th><th>Device code</th><th>Device range ( * 1)</th><th>Device number</th><th>A2AS<br>A2U</th><th>A2AS-S1<br>A2U-S1</th><th>A3U<br>A4U</th><th>Q2A<br>Q2AS<br>Q2ASH</th><th>Q2A-S1<br>Q2AS-S1<br>Q2ASH-S1</th><th>Q3A<br>Q4A<br>Q4AR</th></tr>
<tr><td colspan="2" rowspan="5">Data register</td><td rowspan="5">D0<br>(44$_H$, 20$_H$)</td><td>D0 to D6143</td><td>0000$_H$ to 17FF$_H$</td><td colspan="3">○</td><td colspan="3">○</td></tr>
<tr><td>D6144 to D8191</td><td>1800$_H$ to 1FFF$_H$</td><td colspan="3">×</td><td colspan="3">×</td></tr>
<tr><td>D8192 or more</td><td>2000$_H$ or more</td><td colspan="3">–</td><td colspan="3">×</td></tr>
<tr><td>D9000 to D9255<br>(SD1000 to SD1255)</td><td>2328$_H$ to 2427$_H$</td><td colspan="3">○</td><td colspan="3">○</td></tr>
<tr><td>(SD1256 to SD2047)</td><td></td><td colspan="3">–</td><td colspan="3">×</td></tr>
<tr><td colspan="2" rowspan="3">Link register</td><td rowspan="3">W0<br>(57$_H$, 20$_H$)</td><td>W0 to WFFF</td><td>0000$_H$ to 0FFF$_H$</td><td colspan="3">○</td><td colspan="3">○</td></tr>
<tr><td>W1000 to W1FFF</td><td>1000$_H$ to 1FFF$_H$</td><td colspan="3">×</td><td colspan="3">×</td></tr>
<tr><td>W2000 or more</td><td>2000$_H$ or more</td><td colspan="3">–</td><td colspan="3">×</td></tr>
<tr><td colspan="2" rowspan="2">File register</td><td rowspan="2">R0<br>(52$_H$, 20$_H$)</td><td>R0 to R8191</td><td>0000$_H$ to 1FFF$_H$</td><td colspan="3">○</td><td colspan="3">×</td></tr>
<tr><td>R8192 or more</td><td>2000$_H$ or more</td><td colspan="3">–</td><td colspan="3">×</td></tr>
<tr><td rowspan="6">Timer</td><td rowspan="2">Current value</td><td rowspan="2">TN<br>(54$_H$, 4E$_H$)</td><td>T0 to T2047</td><td>0000$_H$ to 07FF$_H$</td><td colspan="3">○</td><td colspan="3">○</td></tr>
<tr><td>T2048 or more</td><td>0800$_H$ or more</td><td colspan="3">–</td><td colspan="3">×</td></tr>
<tr><td rowspan="2">Contact</td><td rowspan="2">TS<br>(54$_H$, 53$_H$)</td><td>T0 to T2047</td><td>0000$_H$ to 07FF$_H$</td><td colspan="3">○</td><td colspan="3">○</td></tr>
<tr><td>T2048 or more</td><td>0800$_H$ or more</td><td colspan="3">–</td><td colspan="3">×</td></tr>
<tr><td rowspan="2">Coil</td><td rowspan="2">TC<br>(54$_H$, 43$_H$)</td><td>T0 to T2047</td><td>0000$_H$ to 07FF$_H$</td><td colspan="3">○</td><td colspan="3">○</td></tr>
<tr><td>T2048 or more</td><td>0800$_H$ or more</td><td colspan="3">–</td><td colspan="3">×</td></tr>
<tr><td rowspan="6">Counter</td><td rowspan="2">Current value</td><td rowspan="2">CN<br>(43$_H$, 4E$_H$)</td><td>C0 to C1023</td><td>0000$_H$ to 03FF$_H$</td><td colspan="3">○</td><td colspan="3">○</td></tr>
<tr><td>C1024 or more</td><td>0400$_H$ or more</td><td colspan="3">–</td><td colspan="3">×</td></tr>
<tr><td rowspan="2">Contact</td><td rowspan="2">CS<br>(43$_H$, 53$_H$)</td><td>C0 to C1023</td><td>0000$_H$ to 03FF$_H$</td><td colspan="3">○</td><td colspan="3">○</td></tr>
<tr><td>C1024 or more</td><td>0400$_H$ or more</td><td colspan="3">–</td><td colspan="3">×</td></tr>
<tr><td rowspan="2">Coil</td><td rowspan="2">CC<br>(43$_H$, 43$_H$)</td><td>C0 to C1023</td><td>0000$_H$ to 03FF$_H$</td><td colspan="3">○</td><td colspan="3">○</td></tr>
<tr><td>C1024 or more</td><td>0400$_H$ or more</td><td colspan="3">–</td><td colspan="3">×</td></tr>
<tr><td colspan="2" rowspan="5">Input</td><td rowspan="5">X0<br>(58$_H$, 20$_H$)</td><td>X0 to X1FF</td><td>0000$_H$ to 01FF$_H$</td><td colspan="3">○</td><td colspan="3">○</td></tr>
<tr><td>X200 to X3FF</td><td>0200$_H$ to 03FF$_H$</td><td>×</td><td colspan="2">○</td><td>×</td><td colspan="2">○</td></tr>
<tr><td>X400 to X7FF</td><td>0400$_H$ to 07FF$_H$</td><td colspan="2">×</td><td>○</td><td colspan="2">×</td><td>○</td></tr>
<tr><td>X800 to X1FFF</td><td>0800$_H$ to 1FFF$_H$</td><td colspan="3">×</td><td colspan="3">×</td></tr>
<tr><td>X2000 or more</td><td>2000$_H$ or more</td><td colspan="3">–</td><td colspan="3">×</td></tr>
<tr><td colspan="2" rowspan="5">Output</td><td rowspan="5">Y0<br>(59$_H$, 20$_H$)</td><td>Y0 to Y1FF</td><td>0000$_H$ to 01FF$_H$</td><td colspan="3">○</td><td colspan="3">○</td></tr>
<tr><td>Y200 to Y3FF</td><td>0200$_H$ to 03FF$_H$</td><td>×</td><td colspan="2">○</td><td>×</td><td colspan="2">○</td></tr>
<tr><td>Y400 to Y7FF</td><td>0400$_H$ to 07FF$_H$</td><td colspan="2">×</td><td>○</td><td colspan="2">×</td><td>○</td></tr>
<tr><td>Y800 to Y1FFF</td><td>0800$_H$ to 1FFF$_H$</td><td colspan="3">×</td><td colspan="3">×</td></tr>
<tr><td>Y2000 or more</td><td>2000$_H$ or more</td><td colspan="3">–</td><td colspan="3">×</td></tr>
<tr><td colspan="2" rowspan="4">Internal relay</td><td rowspan="4">M0<br>(4D$_H$, 20$_H$)</td><td>M0 to M8191</td><td>0000$_H$ to 1FFF$_H$</td><td colspan="3">○<br>* Including latch relay (L)/<br>Step relay (S)</td><td colspan="3">○</td></tr>
<tr><td>M8192 or more</td><td>2000$_H$ or more</td><td colspan="3">–</td><td colspan="3">×</td></tr>
<tr><td>M9000 to M9255<br>(SM1000 to SM1255)</td><td>2328$_H$ to 2427$_H$</td><td colspan="3">○</td><td colspan="3">○</td></tr>
<tr><td>(SM1256 to SM2047)</td><td></td><td colspan="3">–</td><td colspan="3">×</td></tr>
<tr><td colspan="2">Latch relay</td><td></td><td></td><td></td><td colspan="3" rowspan="2">(Depends on the description above.)</td><td colspan="3" rowspan="2">×<br>* Even if the latch relay (L) is designated,<br>it accesses to the internal relay (M)</td></tr>
<tr><td colspan="2">Step relay</td><td></td><td></td><td></td></tr>
<tr><td colspan="2" rowspan="3">Link relay</td><td rowspan="3">B0<br>(42$_H$, 20$_H$)</td><td>B0 to BFFF</td><td>0000$_H$ to 0FFF$_H$</td><td colspan="3">○</td><td colspan="3">○</td></tr>
<tr><td>B1000 to B1FFF</td><td>1000$_H$ to 1FFF$_H$</td><td colspan="3">×</td><td colspan="3">×</td></tr>
<tr><td>B2000 or more</td><td>2000$_H$ or more</td><td colspan="3">–</td><td colspan="3">×</td></tr>
<tr><td colspan="2" rowspan="2">Annunciator</td><td rowspan="2">F0<br>(46$_H$, 20$_H$)</td><td>F0 to F2047</td><td>0000$_H$ to 07FF$_H$</td><td colspan="3">○</td><td colspan="3">○</td></tr>
<tr><td>F2048 or more</td><td>0800$_H$ or more</td><td colspan="3">–</td><td colspan="3">×</td></tr>
</table>

＊1 The following precautions should be observed when reading/writing to/from the Q/QnACPU.

1) Only the devices with the same names as those found in the AnCPU, AnNCPU, AnACPU and AnUCPU can be read/written within the device range of AnACPU.
The following devices cannot be accessed from the external devices:
- Devices newly added to the Q/QnACPU
- Latch relay (L) and step relay (S)
＊ For the Q/QnACPU, the latch relay (L) and the step relay (S) are separate devices from the internal relay (M). However, access will be made to the internal relay when either the latch relay or the step relay is not designated.
- File register (R)

2) Special relays and special registers can be accessed in the way shown below.
- By designating M9000 to M9255, it is possible to access SM1000 to SM1255.
- By designating D9000 to D9255, it is possible to access SD1000 to SD1255.

**POINT**

(1) Access the PLC CPU within the device number range that can be used by the PLC CPU of the access destination (within the range of the AnACPU for the Q/QnACPU).

(2) Bit devices and word devices are classified according to the following.
Bit devices : X, Y, M, L, B, F, T (contact), T (coil), C (contact), C (coil)
Word devices : T (current value), C (current value), D, W, R

(3) When word units are designated, always make the head device number of a bit device a multiple of 16 (0, 16... in decimal representation).
The special relays beginning from M9000 can be designated by (9000 + multiple of 16).

(4) When accessing to the internal relay (M), the latch relay (L) or the step relay (S) for other than the Q/QnACPU, designate the internal relay (M) and the target access device number. (To access to L100, designate M100.)

(5) The special relays (M9000 to M9255) and special registers (D9000 to D9255) are divided into read only, write only and system use registers.
If writing takes place outside the writing enabled range, a PLC CPU error may occur.
See the ACPU Programming Manual for detailed descriptions of the special relays and the special registers.

(6) When using dedicated instructions for the AnACPU and AnUCPU extension file registers, read/write from/to the file register (R) using the commands described in Section 6.4.

(7) The number of device points that are read/written when the commands are executed should be designated by converting the number of processing points (number of points processed per communication), which should be within the range shown in the table in Section 6.3.1, into a 2-digit ASCII code (hexadecimal).
Note that "00" is used only when designating 256 points.
(Example: The values in parenthesis indicate designated data when communicating in ASCII code.)
In case of 5 points : $05_H$/"05"
In case of 10 points : $0A_H$ /"0A"
In case of 20 points : $14_H$/"14"
In case of 256 points : $00_H$/"00"

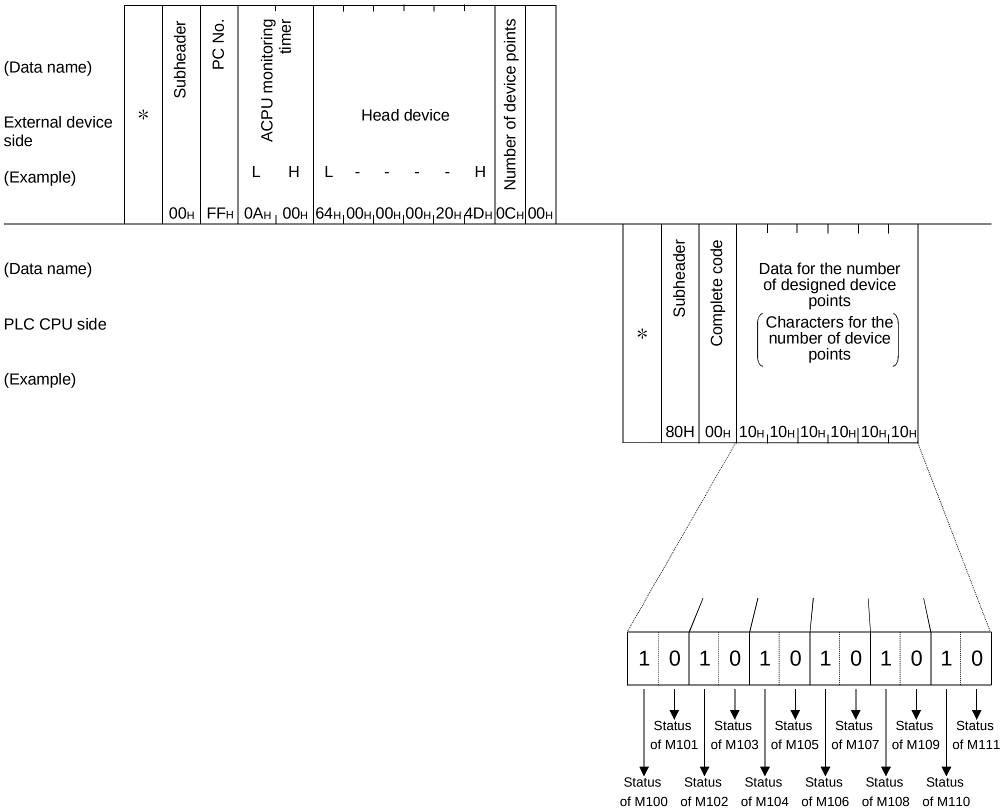
### 6.3.2 Batch read in bit units (command: 00)

The examples shown in this section explain the command/response format when batch-reading the bit device memory.
For more details on the order and contents of data items of the areas marked by "∗" shown in the control procedure diagram, see Section 6.1.

[Control procedure]

Reading the on/off status of M100 to M111 of the PLC CPU on which the Q series E71 is loaded.

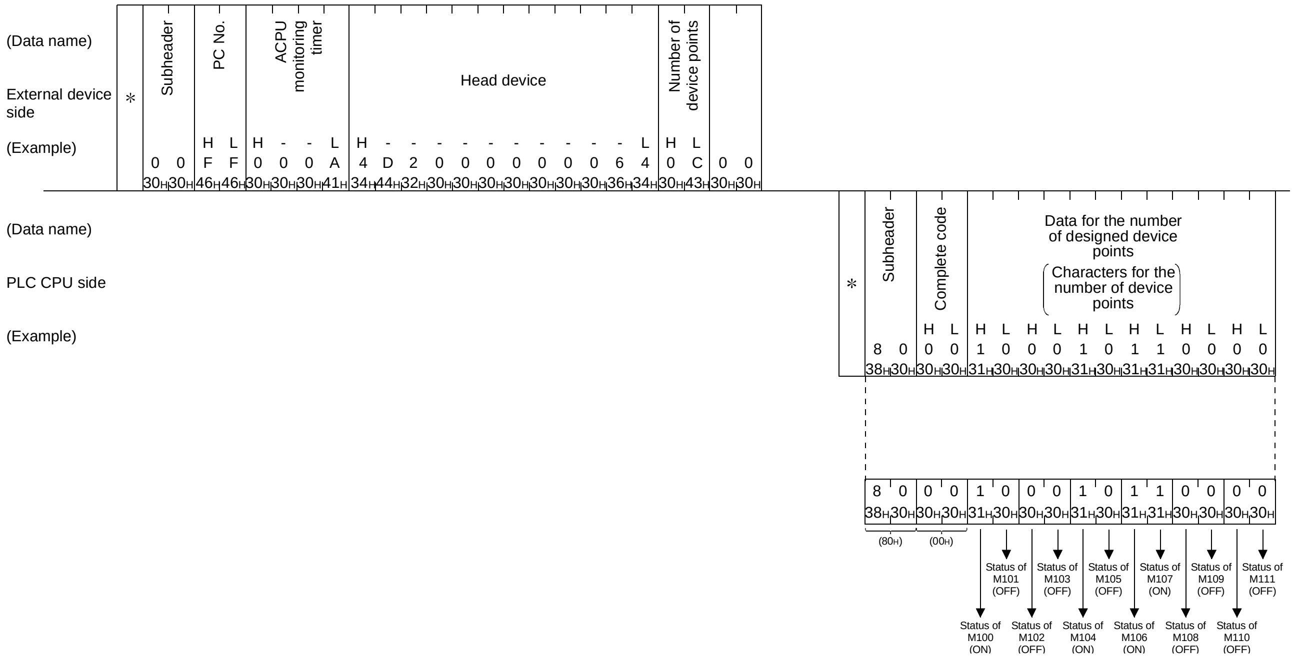
(1) Communication in binary code




REMARK

Use the designation "00H" when the number of device points is 256 points.

(2) Communication in ASCII code




REMARK

(1) Use the designation "00" when the number of device points is 256 points.
(2) If the number of device points designated is an odd number, one byte of dummy data (30H) will be added to the response data. For example, if three points are read, data for four points is returned. The last byte is dummy data.

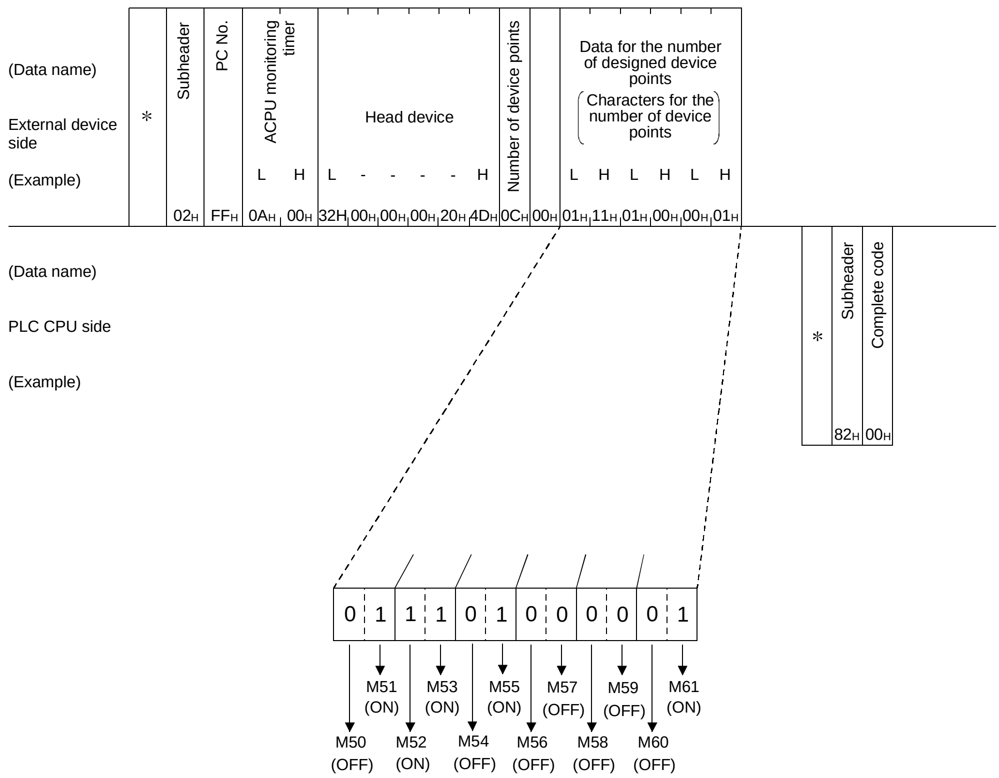
### 6.3.3 Batch write in bit units (command: 02)

The examples shown in this section explain the command/response format when batch writing to the bit device memory.
For more details on the order and contents of data items of the areas marked by "*" shown in the control procedure diagram, see Section 6.1.

[Control procedure]

Writing the on/off status of M50 to M61 of the PLC CPU on which the Q series E71 is loaded.

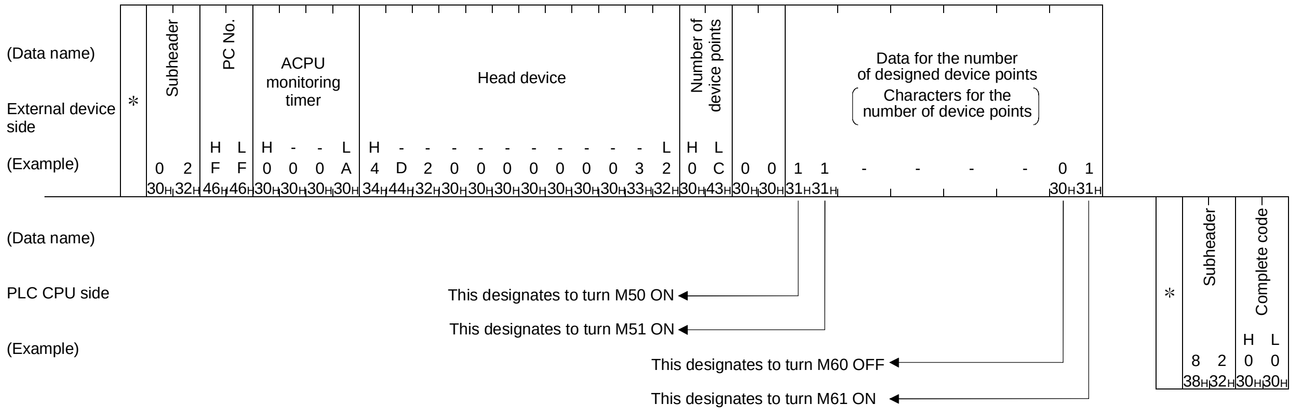
(1) Communication in binary code




**REMARK**

Use the designation "00H" when the number of device points is 256 points.

(2) Communication in ASCII code




**REMARK**

(1) Use the designation "00" when the number of device points is 256 points.
(2) If the number of device points to be set is an odd number, add one byte of dummy data (30H) at the end of data written. For example, when writing three points, add the dummy data (30H) at the end.

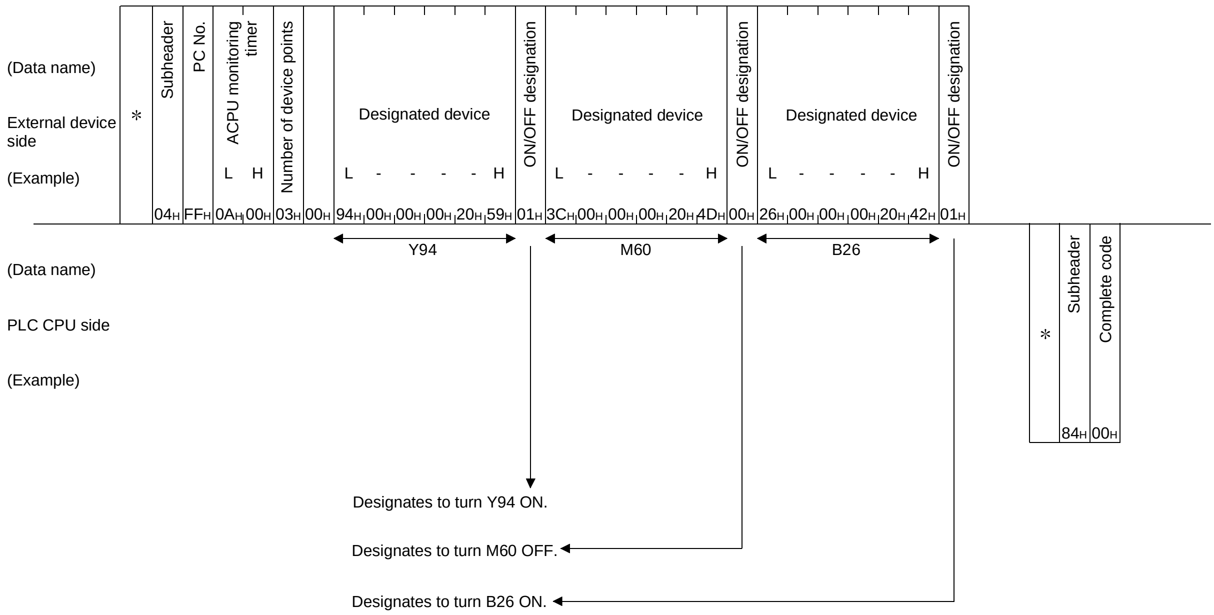
### 6.3.4 Test in bit units (random write) (command: 04)

The examples shown in this section explain the command/response format when writing data by designating bit device memories at random.
For more details on the order and contents of data items of the areas marked by "∗" shown in the control procedure diagram, see Section 6.1.

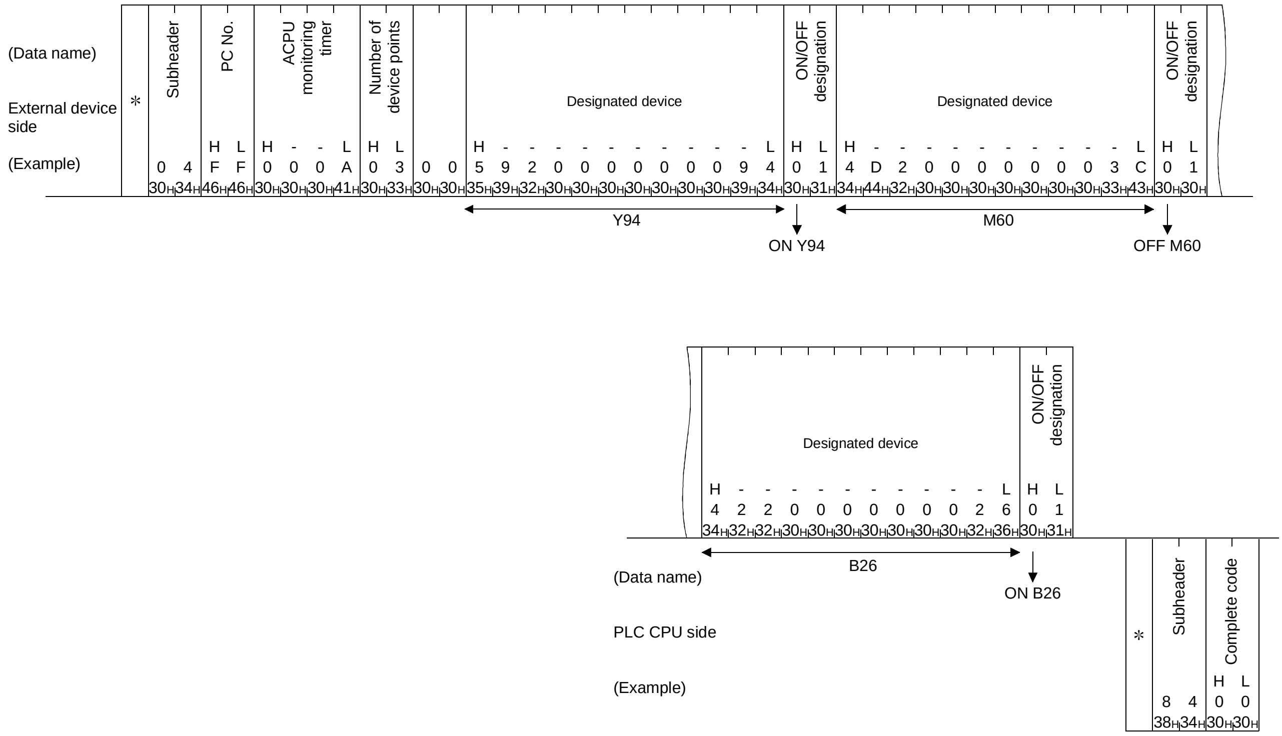
[Control procedure]

Specifying Y94 to ON, M60 to OFF, and B26 to ON at the PLC CPU on which the Q series E71 is loaded.

(1) Communication in binary code



## (2) Communication in ASCII code



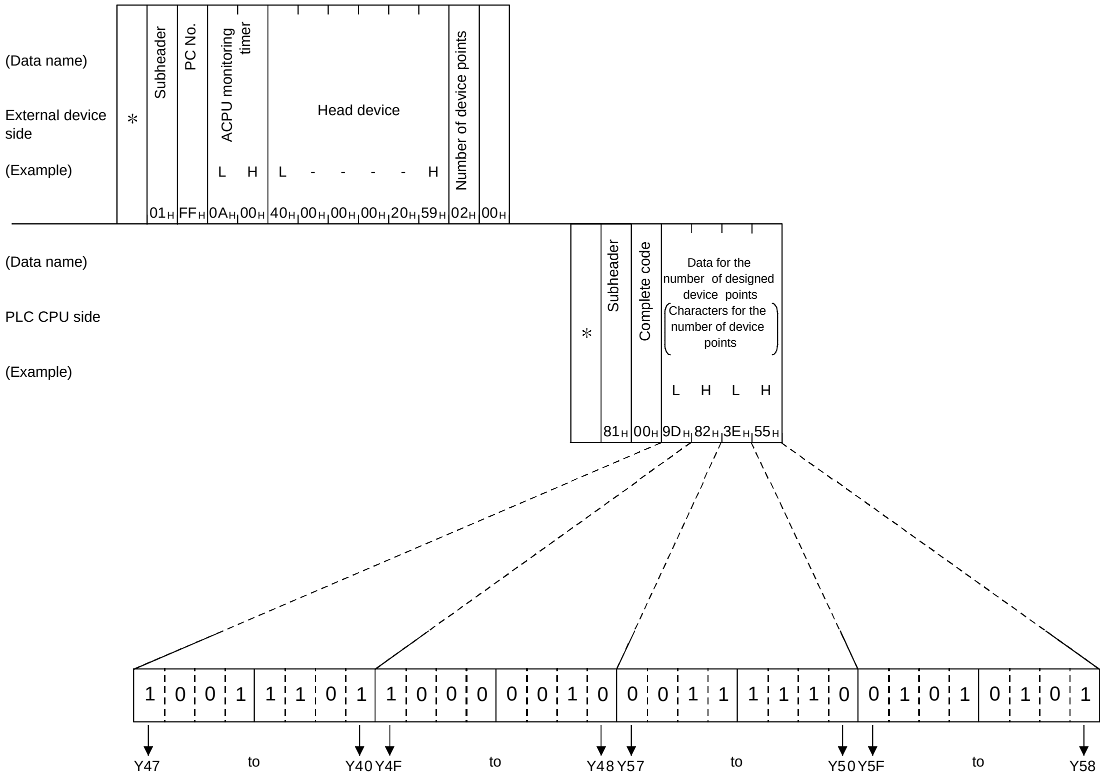
## 6.3.5 Batch read in word units (command: 01)

The examples shown in this section explain the command/response format when batch reading the word device memory and the bit device memory (16 point units).
For more details on the order and contents of data items of the areas marked by "∗" shown in the control procedure diagram, see Section 6.1.

[Control procedure]

Reading the on/off status of Y40 to 5F (32 points) of the PLC CPU on which the Q series E71 is loaded.

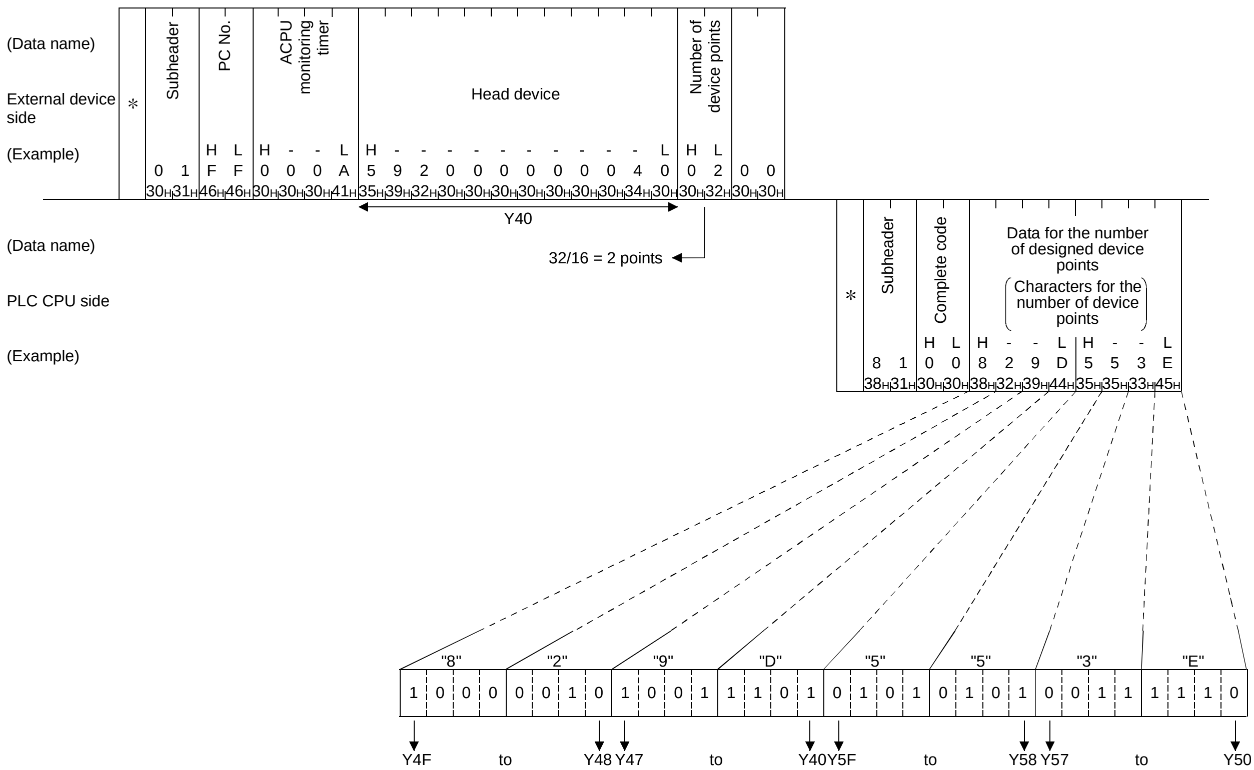
### (1) Communication in binary code




**REMARK**

Use the designation "00H" when the number of device points is 256 points.

(2) Communication in ASCII code




REMARK

Use the designation "00" when the number of device points is 256 points.

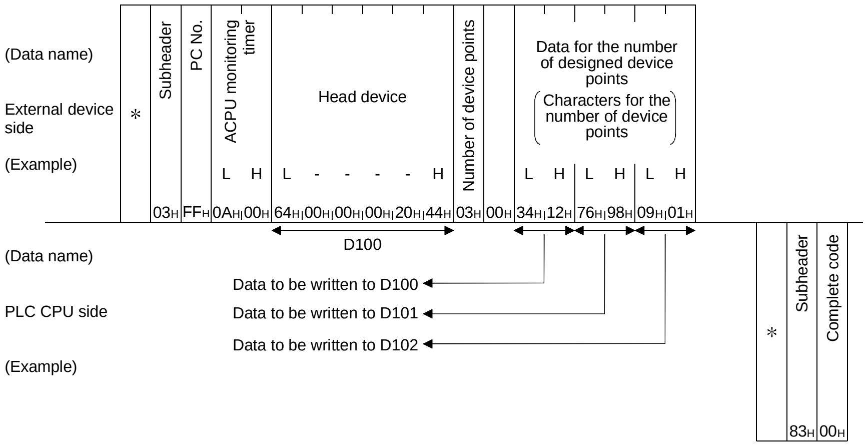
## 6.3.6 Batch write in word units (command: 03)

The examples shown in this section explain the command/response format when batch writing to a word device memory and bit device memory (16 point units).
For more details on the order and contents of data items of the areas marked by "*" shown in the control procedure diagram, see Section 6.1.

[Control procedure]

Writing data to D100 to 102 of the PLC CPU on which the Q series E71 is loaded.

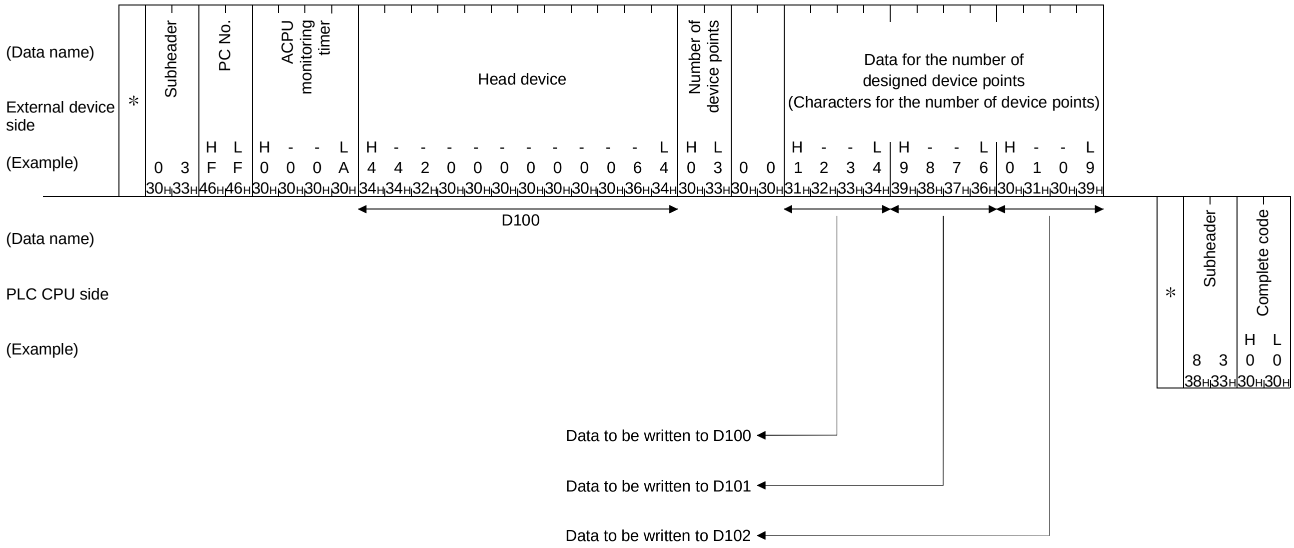
### (1) Communication in binary code




REMARK

Use the designation "00H" when the number of device points is 256 points.

(2) Communication in ASCII code




REMARK

Use the designation "00" when the number of device points is 256 points.

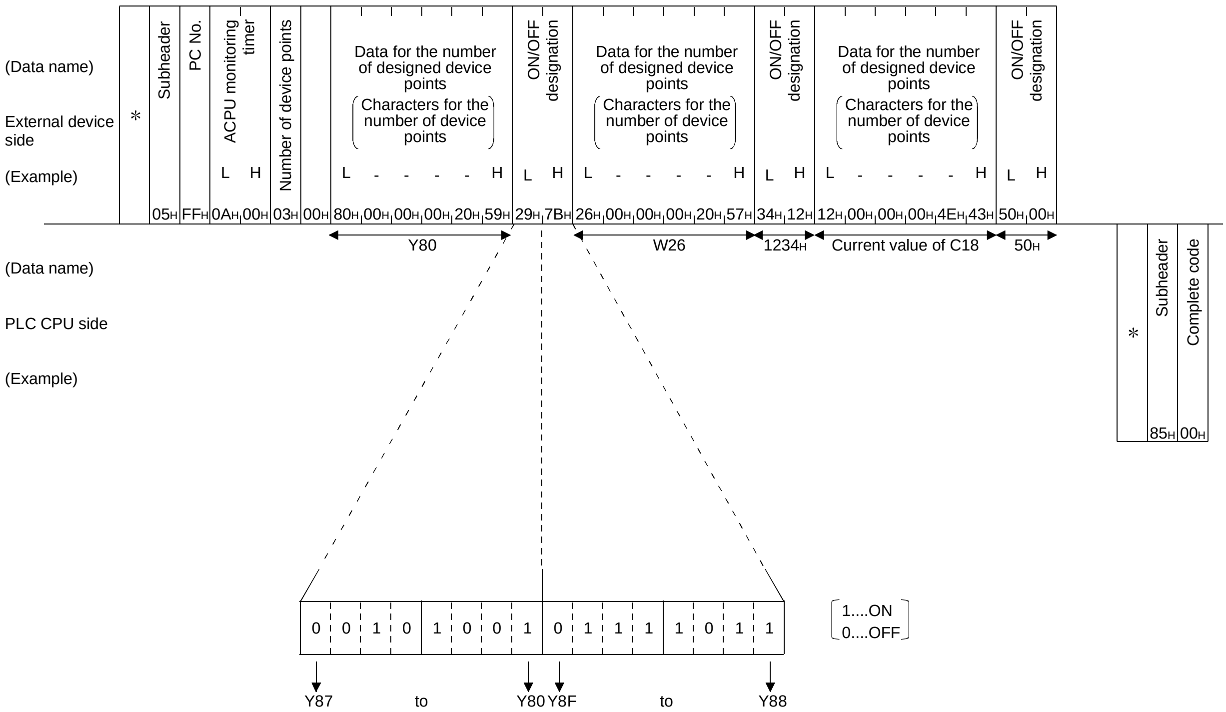
### 6.3.7 Test in word units (random write) (command: 05)

The examples shown in this section explain the command/response format when writing data by designating word device memories and bit device memories (16 point units) at random.
For more details on the order and contents of data items of the areas marked by "∗" shown in the control procedure diagram, see Section 6.1.

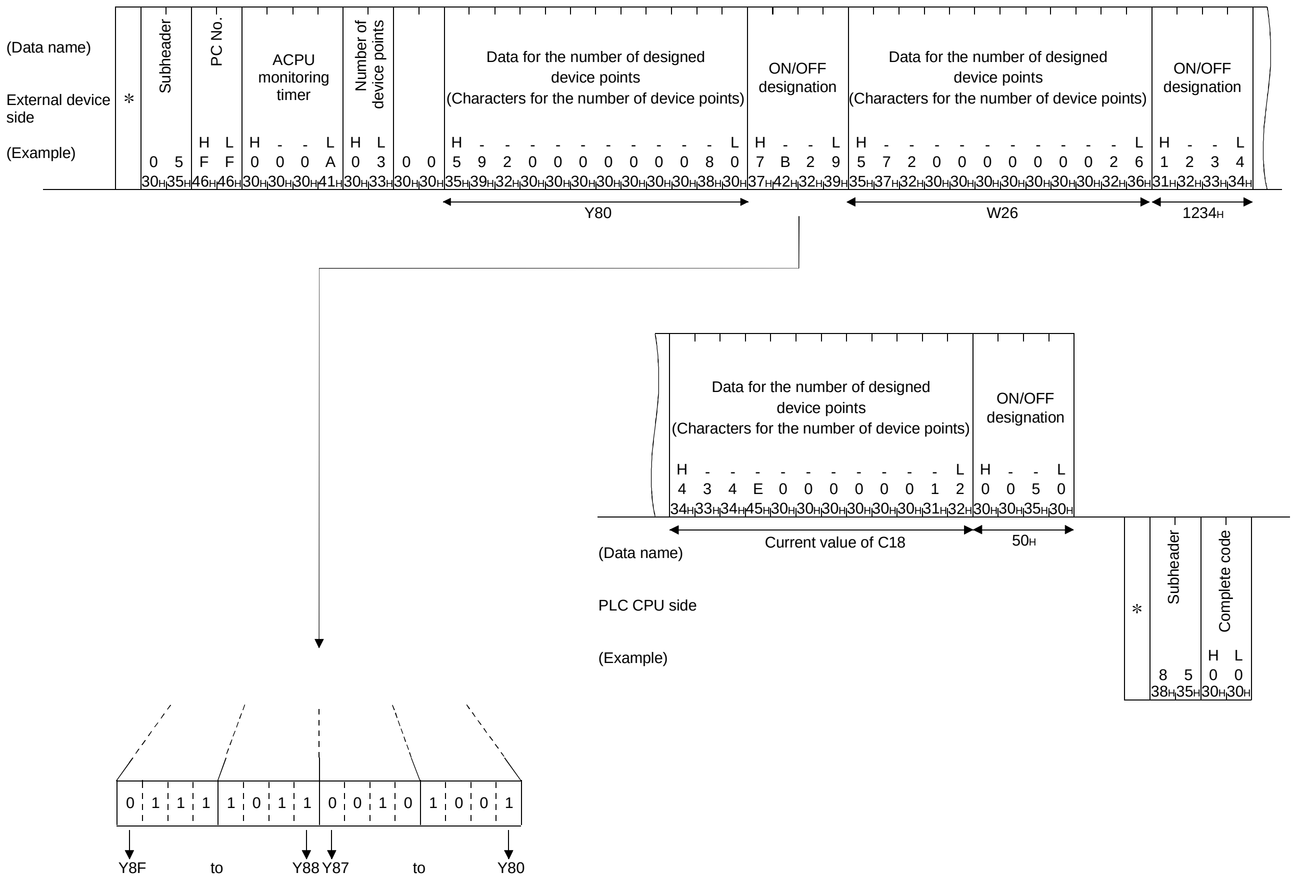
[Control procedure]

Specifying Y80 to 8F to on/off, W26 to "1234H," and the current value of C18 to "50H" at the PLC CPU on which the Q series E71 is loaded.

(1) Communication in binary code



## (2) Communication in ASCII code

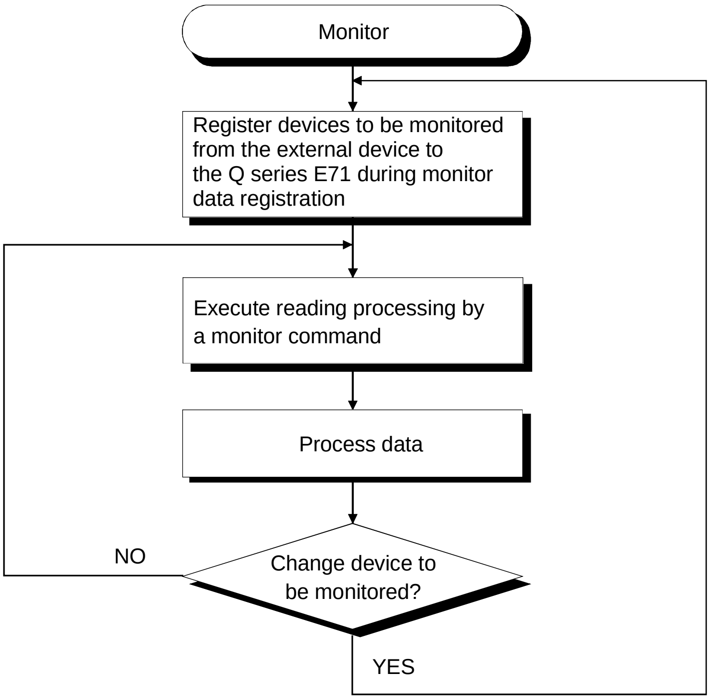


## 6.3.8 Monitoring device memory

The on/off status and contents of devices in the PLC CPU can be monitored from an external device by registering the devices and device numbers to be monitored by the external device to the Q series E71 in advance and issuing a monitor command from the external device.
When reading the device memory using the device memory batch read command, the processed device numbers must be consecutive. However, when reading it using the monitor command, device memory can be monitored by arbitrarily designating the devices and device numbers.

### (1) Monitoring procedure

The operation procedure for monitoring is shown below.




**POINT**

(1) When a monitoring is performed using the procedure above, the monitor data must always be registered. If a monitoring is performed without registering the monitor data, an error occurs.

(2) The contents of registered monitor data are deleted when power supply is turned off or the PLC CPU is reset.

(3) The monitor data for 1 command registered at the end of the three types of commands, expansion file register, device memory bit unit and device memory word unit can be registered in the Q series E71.
See Section 6.3.8 about monitoring the device momery.

(4) When multiple external devices registers the monitor data in the PLC CPU device memory of the same station, the registered data is overwritten, and the device memory that was registered last becomes valid.

(5) See Section 6.4.6 for details about monitoring the extension file register.

## (2) Monitor data registration (command: 06, 07)

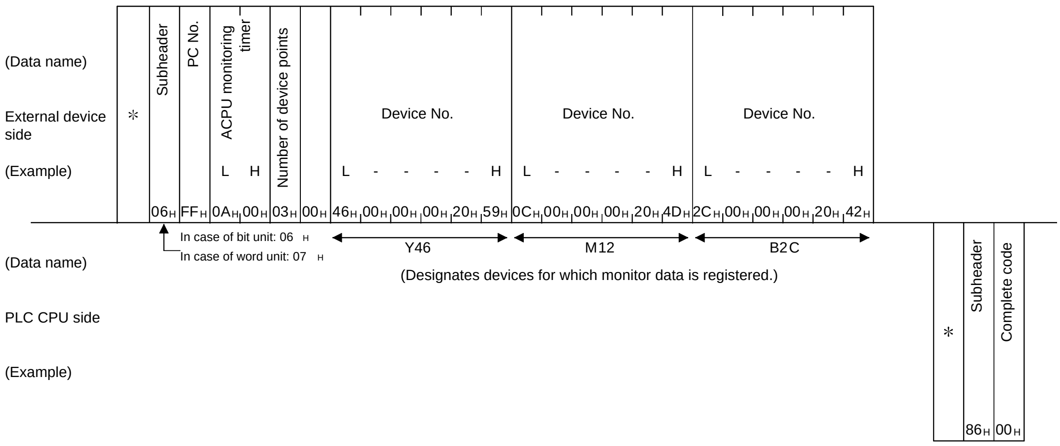
The examples shown in this section explain the command/response format when registering devices to be monitored.

For more details on the order and contents of data items of the areas marked by "✻" shown in the control procedure diagram, see Section 6.1.

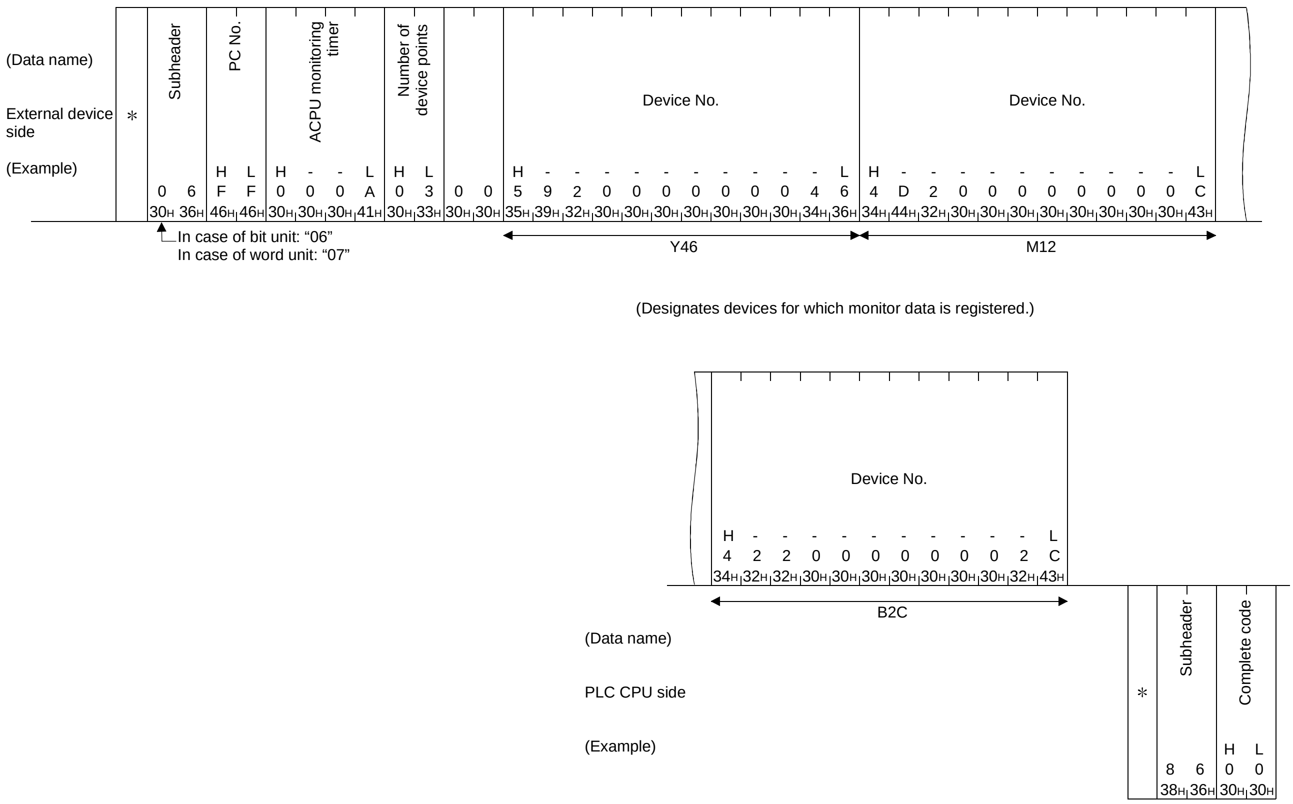
[Control procedure]

Registering Y46, M12, and B2C of the PLC CPU on which the Q series E71 is loaded.

(a) Communication in binary code




(b) Communication in ASCII code



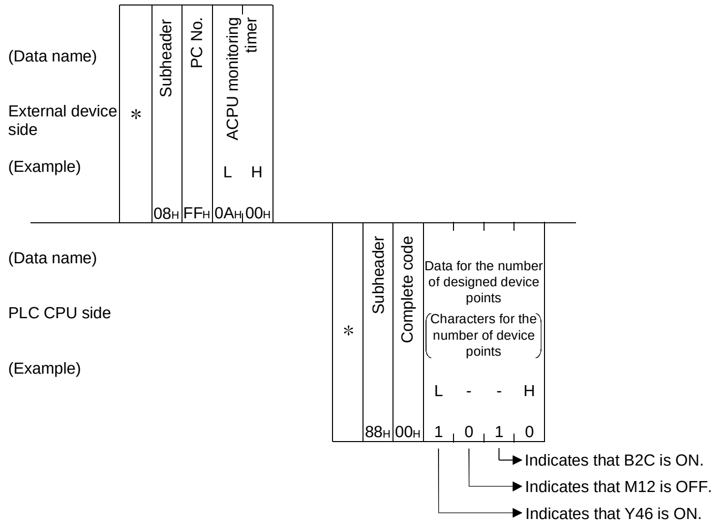
(3) Monitor in bit units (command: 08)

The examples shown in this section explain the command/response format when monitoring bit devices for which monitor data is registered.
For more details on the order and contents of data items of the areas marked by "✻" shown in the control procedure diagram, see Section 6.1.

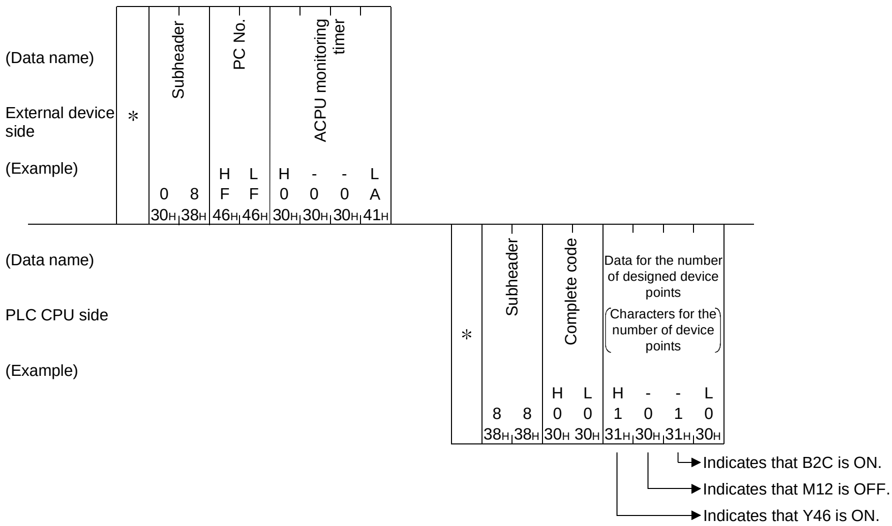
[Control procedure]

Monitoring "Y46," "M12," and "B2C" for which monitor data is registered at the PLC CPU on which the Q series E71 is loaded.

(a) Communication in binary code




(b) Communication in ASCII code




REMARK

If the number of device points registered to be monitored is an odd number, dummy data (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 is dummy data.

(4) Monitor in word units (command: 09)

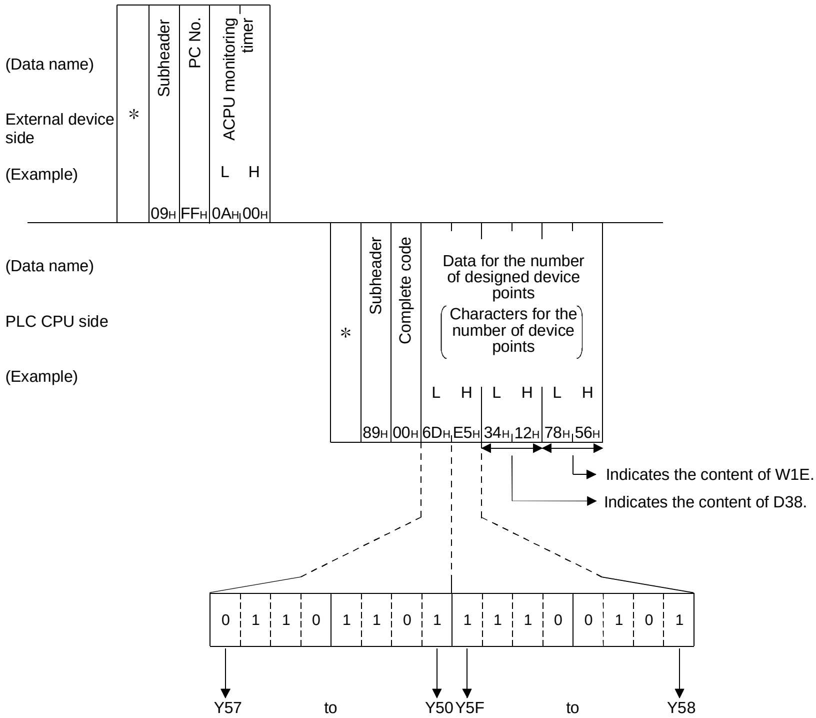
The examples shown in this section explain the command/response format when monitoring word devices and bit devices (16 point units) for which monitor data is registered.

For more details on the order and contents of data items of the areas marked by "*" shown in the control procedure diagram, see Section 6.1.

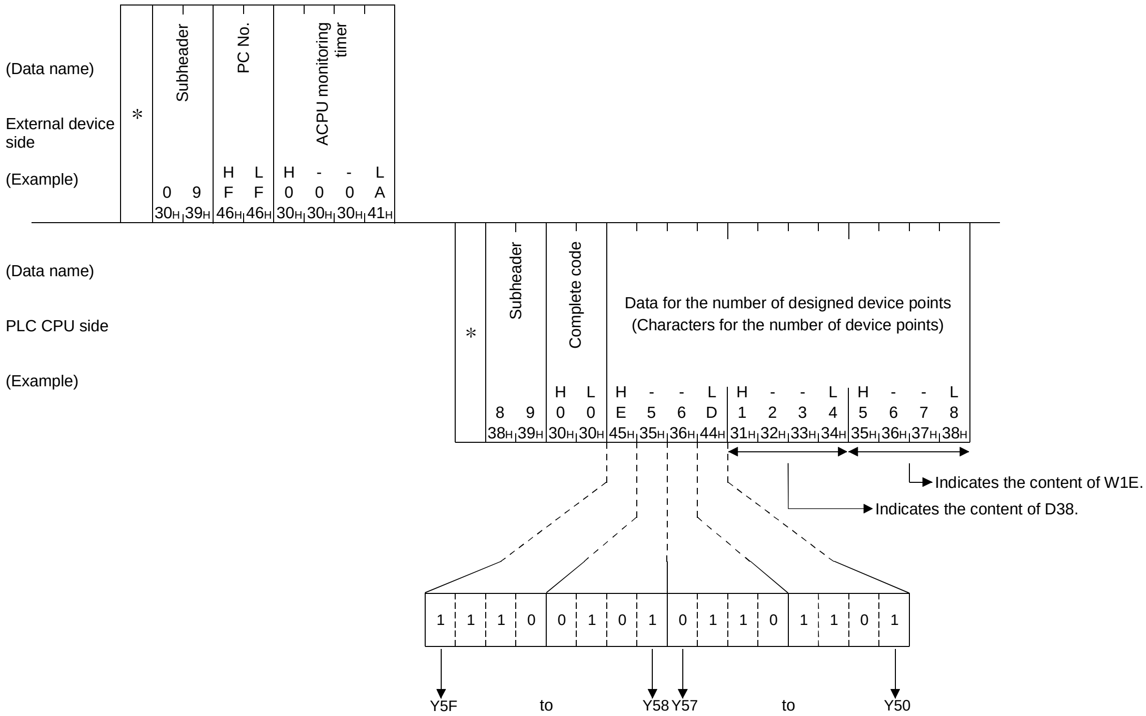
[Control procedure]

Monitoring "Y50 to 5F," "D38," and "W1E" for which monitor data is registered at the PLC CPU on which the Q series E71 is loaded.

(a) Communication in binary code



(b) Communication in ASCII code



## 6.4 Extension File Register Read and Write

The extension file register uses free areas in the user memory area of the PLC CPU as a file register. It serves as a memory area for storing necessary data and results of operations in various data processing performed using the software packages for extension file register "SWOGHP-UTLPC-FN1" (hereinafter called UTLP-FN1), as well as in the dedicated instructions for the extension file register of the AnACPU and AnUCPU.
Using the examples below, this section explains the control procedure to read and write extension file register.

### 6.4.1 ACPU common commands and addresses

(1) The following table lists the ACPU common commands that are used for reading and writing data from/to the extension file register.

| Item | Command/response format | Processing | Number of points processed per communication | PLC CPU status | | |
|---|---|---|---|---|---|---|
| | | | | During STOP | During RUN: Write allow setting | During RUN: Write prohibit setting |
| Batch read | $17_H$ | Reads extension file register (R) in 1-point units. | 256 points | ○ | ○ | ○ |
| Batch write | $18_H$ | Writes to extension file register (R) in 1-point units. | 256 points | ○ | ○ | × |
| Test (random write) | $19_H$ | Writes to extension file register (R) in 1-point units by randomly designating the devices and device numbers. | 40 points | ○ | ○ | × |
| Monitor data registration | $1A_H$ | Registers device number to be monitored in 1-point units. | 20 points | ○ | ○ | ○ |
| Monitor | $1B_H$ | Monitors extension file register with monitor data registered. | – | ○ | ○ | ○ |

In the PLC CPU status column of the table above, ○ indicates that the corresponding function is executable and × indicate that it is not executable.

(2) Addresses of the extension file register

(a) An extension file register has blocks numbered from 0 to n (the value of n differs depending on the memory cassette). Block number 0 has a number of points registered with a parameter of the PLC CPU, while block numbers 1 to n have a register of 8192 points in each block.
However, the range that can be read and written by the PLC CPU is the range of points designated with a parameter for block 0.

(b) The range of block numbers that can be designated depends on the memory cassette type and the parameter setting in the PLC CPU.
For more detailed explanations, refer to the operating manual for UTLP-FN1, or the User's Manuals for the AnACPU and AnUCPU.

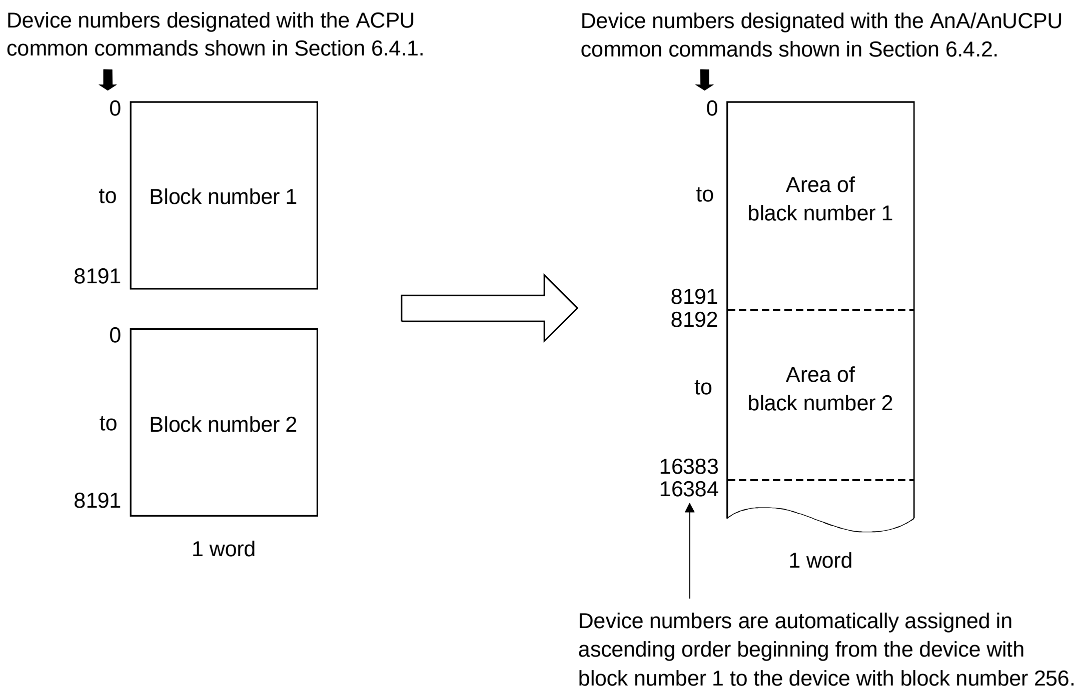
### 6.4.2 AnA/AnUCPU common commands and device numbers

(1) This section explains the AnUCPU dedicated commands used for directly reading from and writing to the extension file register.
This command allow the access to extension file register blocks numbered from 0 to 256 by designating addresses from device number 0 of block number 1 as the device number, regardless of each block number.
(Access the extension file registers for the usable block count × 8192 points using consecutive device numbers.)

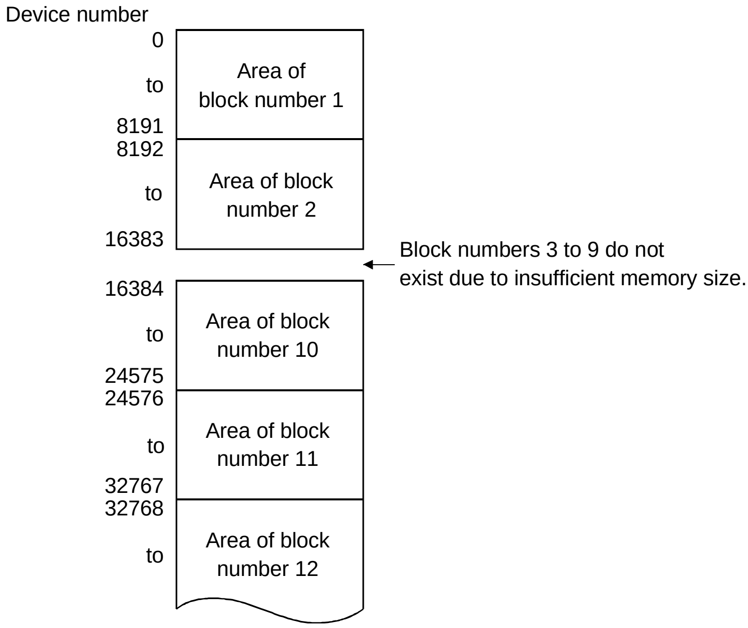
| Item | Command/response format | Processing | Number of points processed per communication | PLC CPU status | | |
|---|---|---|---|---|---|---|
| | | | | During STOP | During RUN | |
| | | | | | Write allow setting | Write prohibit setting |
| Direct read | 3BH | Reads extension file register (R) in 1-point units. | 256 points | ○ | ○ | ○ |
| Direct write | 3CH | Writes to extension file register (R) in 1-point units. | 256 points | ○ | ○ | × |

In the PLC CPU status column of the table above, ○ indicates that the corresponding function is executable and × indicate that it is not executable.

(2) Device number of extension file register
The range of device numbers that can be designated is as follows.
0 to (number of available blocks × 8192) - 1



The device numbers that can be designated vary depending on the memory cassette type and the parameter setting of the PLC CPU.
(For a more detailed explanation, refer to the operating manual for UTLP-FN1, or the User's Manual for the PLC CPU to be accessed.)
A device number is not assigned to a block number that 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.




**POINT**

(1) The AnA/AnUCPU common commands can be used only when reading and writing data of blocks numbered 1 to 256 from/to the extension file register. Also, they can be used regardless whether or not the file register parameter is set.

(2) When accessing the file register (R) set by the parameter or when accessing it by designating block number, use the commands described in Section 6.4.1.

(3) The head device number designated with the AnA/AnUCPU common commands is calculated by the following expression.
When designating device numbers m (0 to 8191) of the nth block from the top:
Head device number = $(n - 1) \times 8192 + m$

REMARK

The following table shows the range of device numbers designated when the AnA/AnUCPU common commands ($3B_H$, $3C_H$) are used for the first 28 blocks.

| 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 |

### 6.4.3 Precautions when reading and writing from/to the extension file register

The following precautions should be observed when reading and writing from/to the extension file register using the commands described in Sections 6.4.4 through 6.4.9.

(1) Only PLC CPUs that can handle an extension file register can be accessed. These functions cannot be used for PLC CPUs that cannot handle an extension file register (such as A1N).

(2) Depending on the type of memory cassette installed on the PLC CPU, an error (complete code 58H) may not be detected even when a read/write operation is performed on block numbers that do not exist. In this case, the data read is incorrect. Also, writing to these blocks may destroy the user memory of the PLC CPU.
Confirm the type of memory cassette and parameter settings before using these functions.

| Memory cassette model name | Block numbers that do not cause an error (58H) | | |
|---|---|---|---|
| | 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 more details, see the operating manual for UTLP-FN1, or the User's Manual for the PLC CPU to be accessed.)

(3) The block numbers of the extension file register that can be handled by the A2USCPU(S1) is as follows.

- A2USCPU : No.'s 1 to 3
- A2USCPU-S1 : No.'s1 to 8, No.'s 10 to 16

(4) The extension file register of the Q/QnACPU cannot be read or written.

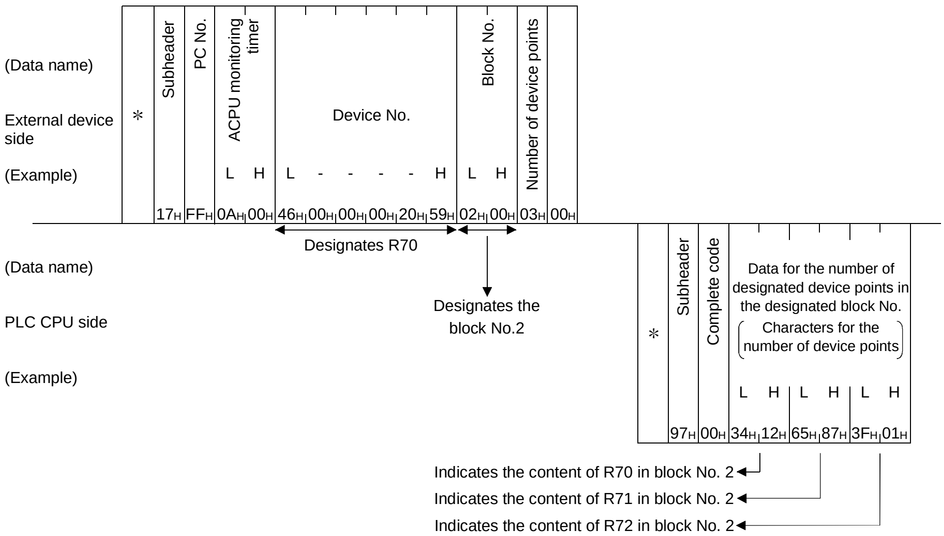
## 6.4.4 Batch reading of the extension file register (command: 17)

The examples shown in this section explain the command/response format when batch-reading extension file registers.
For more details on the order and contents of data items of the areas marked by "✻" shown in the control procedure diagram, see Section 6.1.

[Control procedure]

Reading the contents of R70 to R72 in extension file register number 2 block of the PLC CPU on which the Q series E71 is loaded.

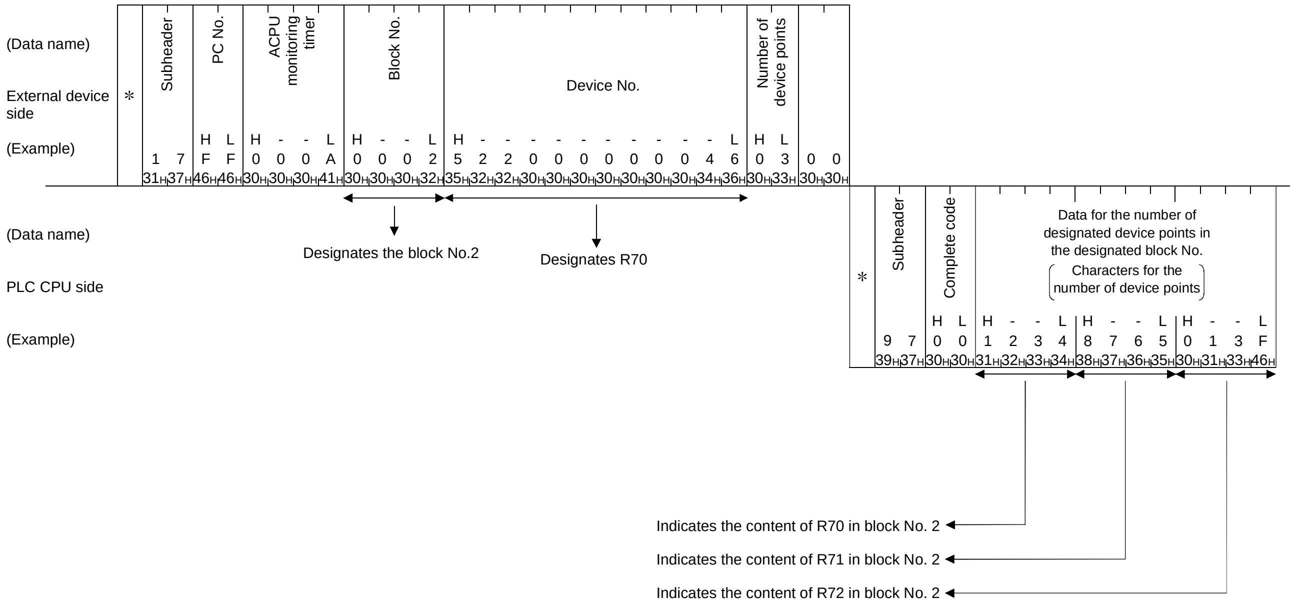
### (1) Communication in binary code




REMARK

Use the designation "00H" when the number of device points is 256 points.

(2) Communication in ASCII code




REMARK

Use the designation "00" when the number of device points is 256 points.

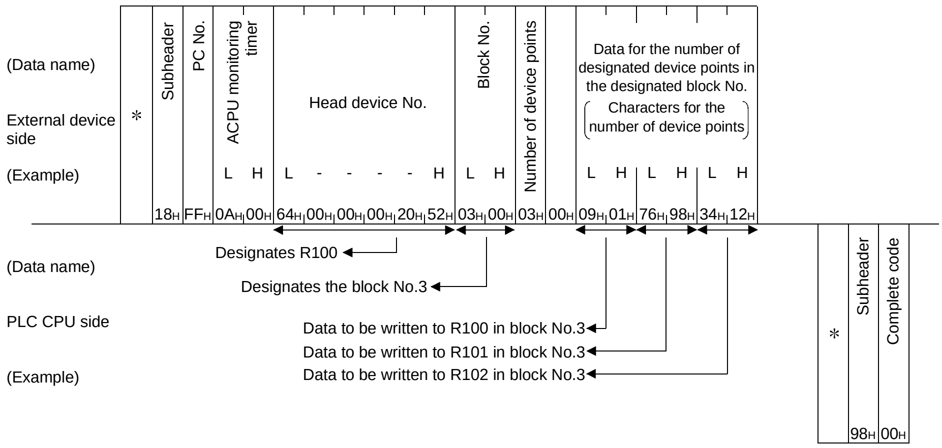
### 6.4.5 Batch writing of the extension file register (command: 18)

The examples shown in this section explain the command/response format when batch-writing extension file registers.
For more details on the order and contents of data items of the areas marked by "∗" shown in the control procedure diagram, see Section 6.1.

[Control procedure]

Writing the contents of R100 to R102 in extension file register number 3 block of the PLC CPU on which the Q series E71 is loaded.

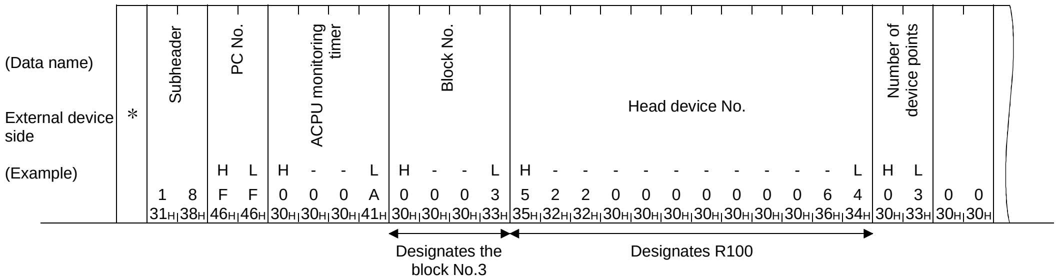
(1) Communication in binary code

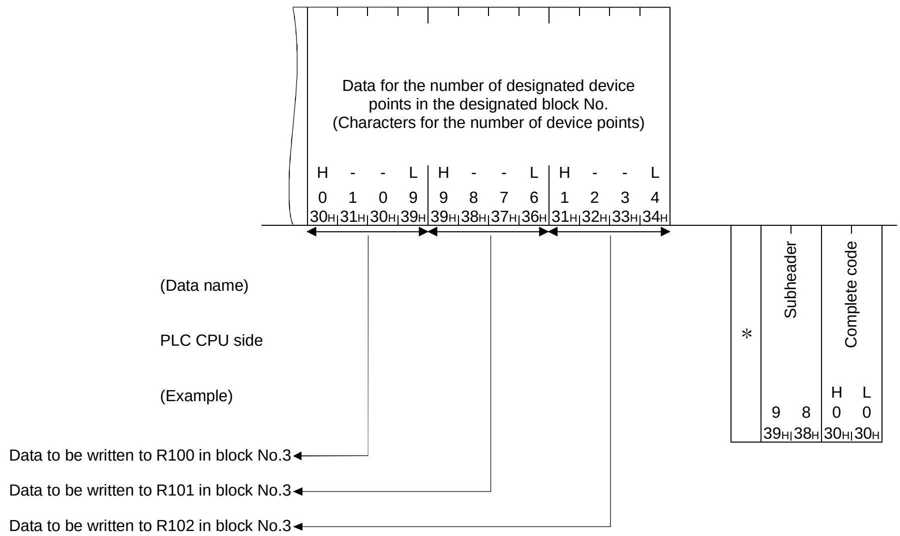



**REMARK**

Use the designation "00H" when the number of device points is 256 points.

(2) Communication in ASCII code




REMARK

Use the designation "00" when the number of device points is 256 points.

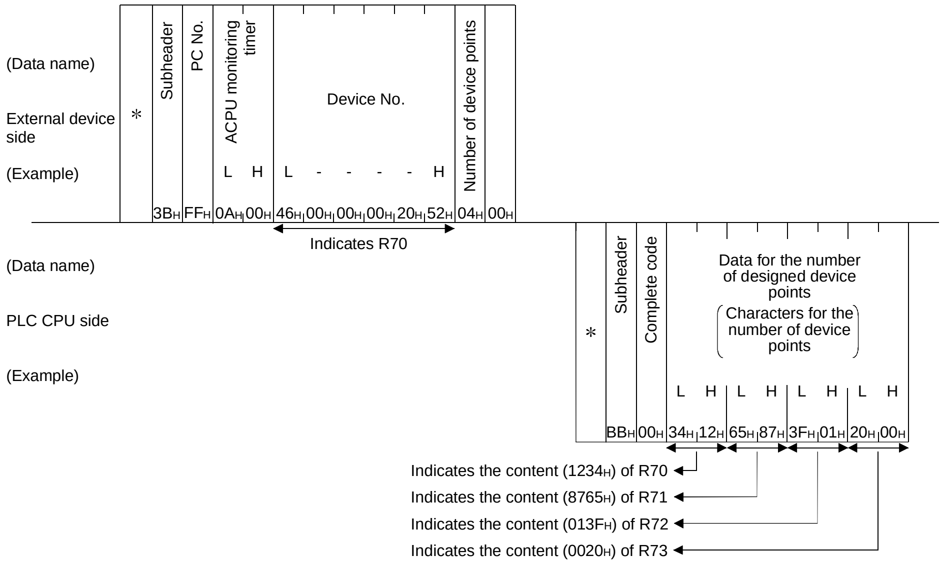
### 6.4.6 Direct reading of the extension file register (command: 3B)

The examples shown in this section explain the command/response format when directly reading extension file registers.
For more details on the order and contents of data items of the areas marked by "*" shown in the control procedure diagram, see Section 6.1.

[Control procedure]

Reading the contents of extension file registers R70 to R73 of the PLC CPU on which the Q series E71 is loaded.

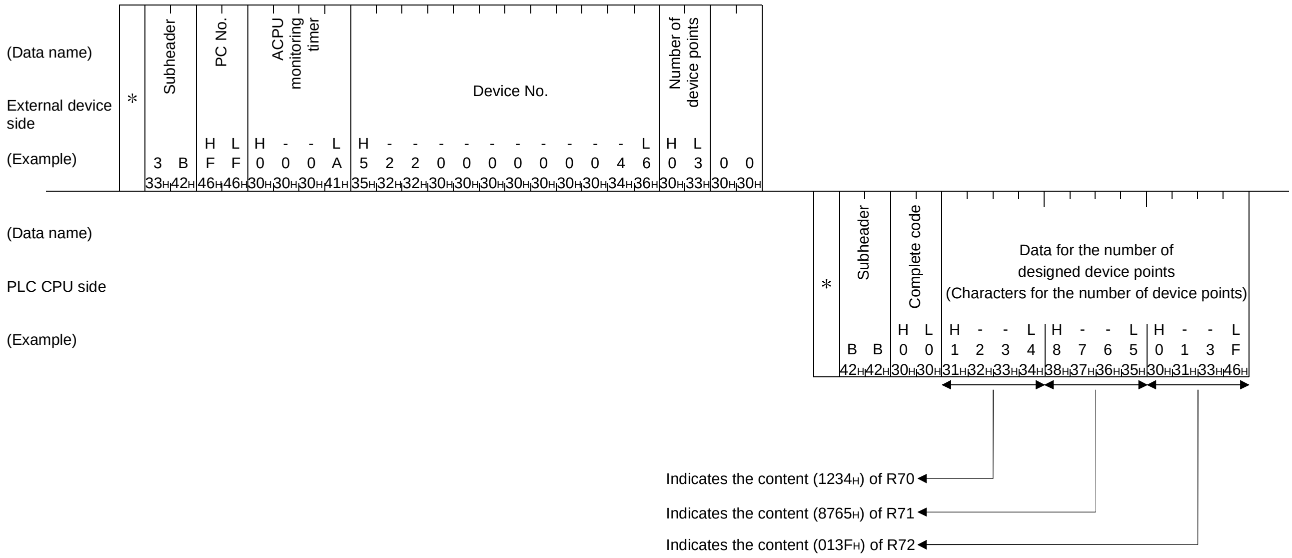
(1) Communication in binary code




REMARK

Use the designation "00H" when the number of device points is 256 points.

## (2) Communication in ASCII code




**REMARK**

Use the designation "00" when the number of device points is 256 points.

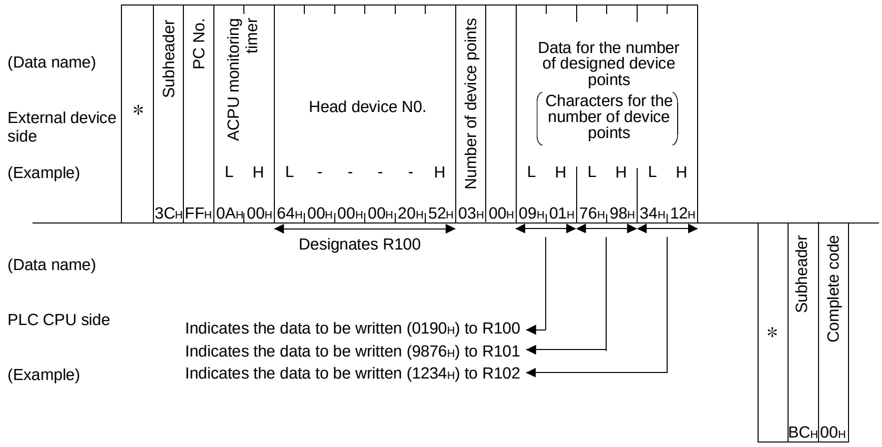
## 6.4.7 Direct writing of the extension file register (command: 3C)

The examples shown in this section explain the command/response format when directly writing to extension file registers.
For more details on the order and contents of data items of the areas marked by "*" shown in the control procedure diagram, see Section 6.1.

[Control procedure]

Writing data to extension file registers R100 to 102 of the PLC CPU on which the Q series E71 is loaded.

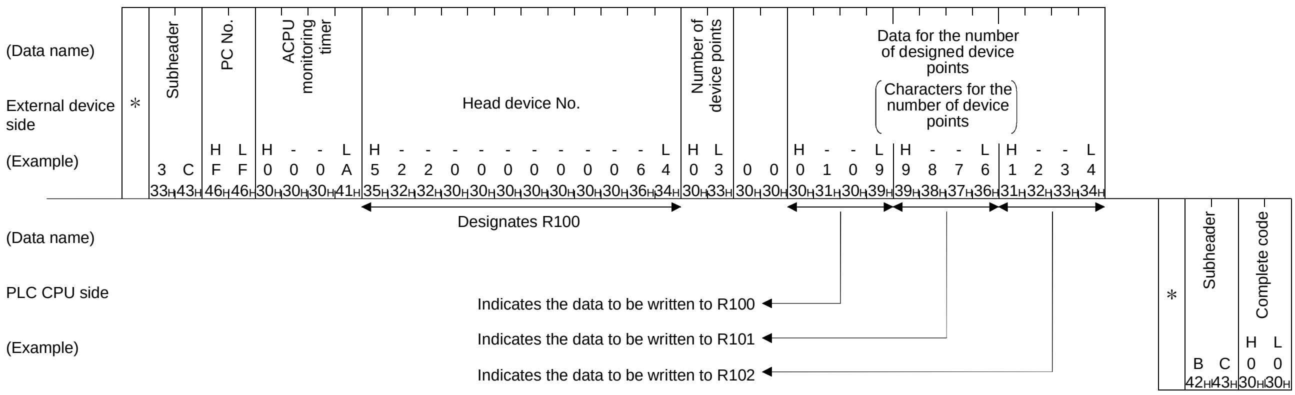
### (1) Communication in binary code




REMARK

Use the designation "00H" when the number of device points is 256 points.

### (2) Communication in ASCII code




REMARK

Use the designation "00" when the number of device points is 256 points.

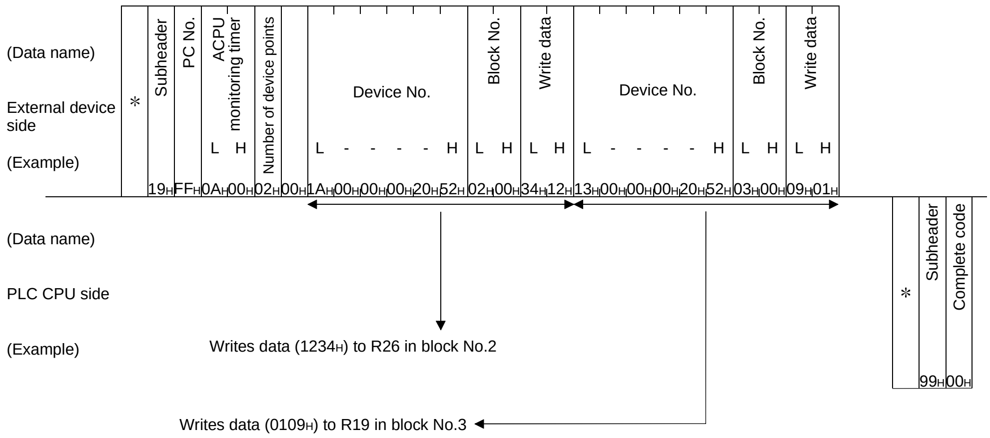
### 6.4.8 Test of extension file register (random write) (command: 19)

The examples shown in this section explain the command/response format when writing data by designating extension file registers at random.
For more details on the order and contents of data items of the areas marked by "∗" shown in the control procedure diagram, see Section 6.1.

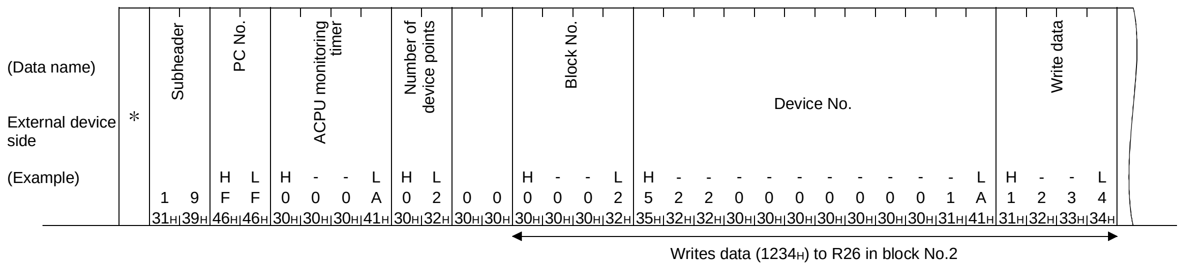
[Control procedure]

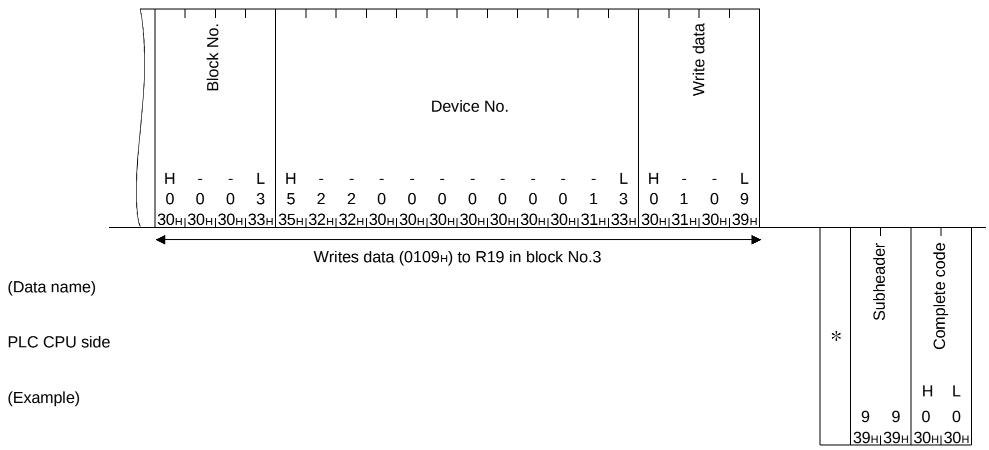
Writing data to R26 in block No.2 and R19 in block No. 3 of the PLC CPU on which the Q series E71 is loaded.

(1) Communication in binary code




(2) Communication in ASCII code



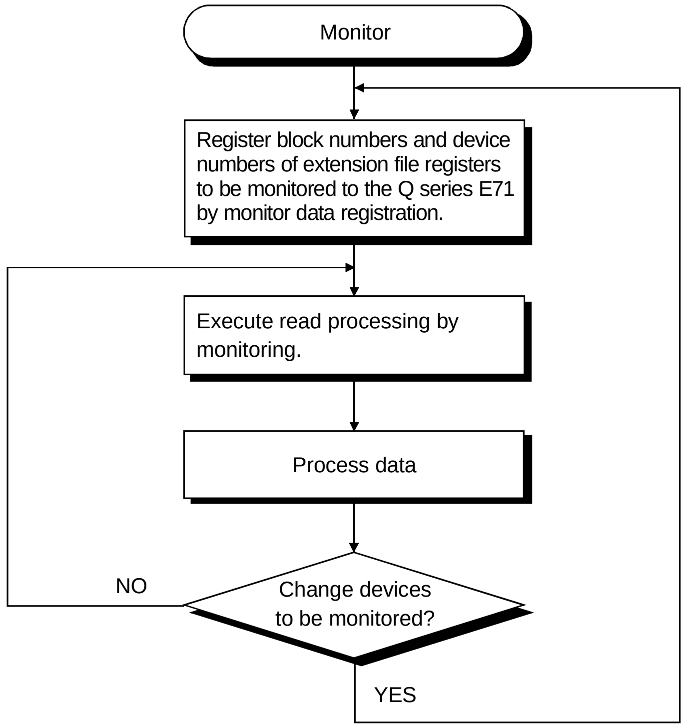



### 6.4.9 Monitoring extension file registers

The contents of extension file registers in the PLC CPU can be monitored from an external device by registering the relevant block numbers and device numbers to the Q series E71 in advance and executing a monitoring command from the external device. When reading using the batch read of extension file register command, the processed device numbers must be consecutive. However, when reading using the monitor command, extension file registers can be monitored by designating the file registers of arbitrary block numbers and device numbers freely

#### (1) Monitoring procedure

The operation procedure for monitoring is shown below.




**POINT**

(1) When a monitoring is performed using the procedure above, the monitor data must always be registered. If the monitoring is performed without registering the monitor data, an error occurs (complete code $57_H$).

(2) The contents of registered monitor data are deleted when power supply is turned off or the PLC CPU is reset.

(3) The monitor data for 1 command registered at the end of the three types of commands, expansion file register, device memory bit unit and device memory word unit can be registered in the Q series E71.
See Section 6.3.8 about monitoring the device memory.

(4) When multiple external devices register monitor data in the PLC CPU device memory of the same station, the registered data is overwritten, and the device memory that was registered last becomes valid.

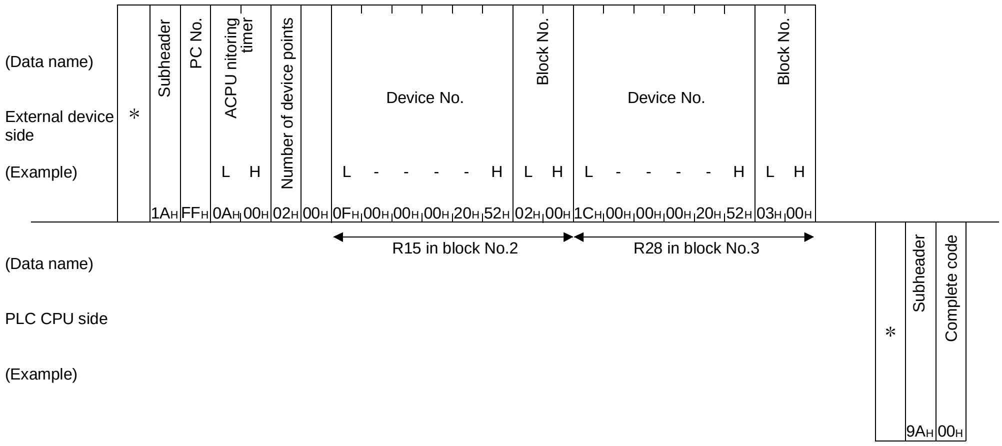
## (2) Monitor data registration of extension file register (command: 1A)

The examples shown in this section explain the command/response format when registering device numbers of extension file registers to be monitored.
For more details on the order and contents of data items of the areas marked by "*" shown in the control procedure diagram, see Section 6.1.

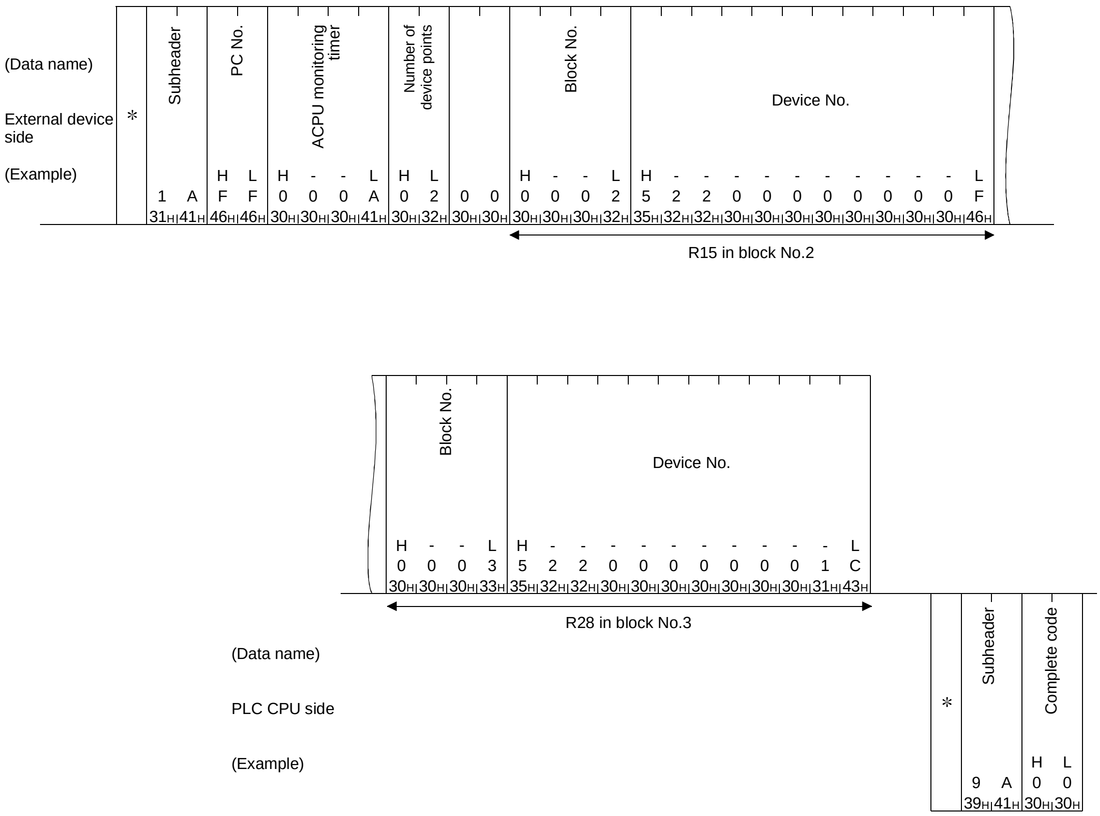
[Control procedure]

Registering R15 in extension file register block No.2 and R28 in block No.3 of the PLC CPU on which the Q series E71 is loaded.

(a) Communication in binary code




(b) Communication in ASCII code



## (3) Monitor (command: 1B)

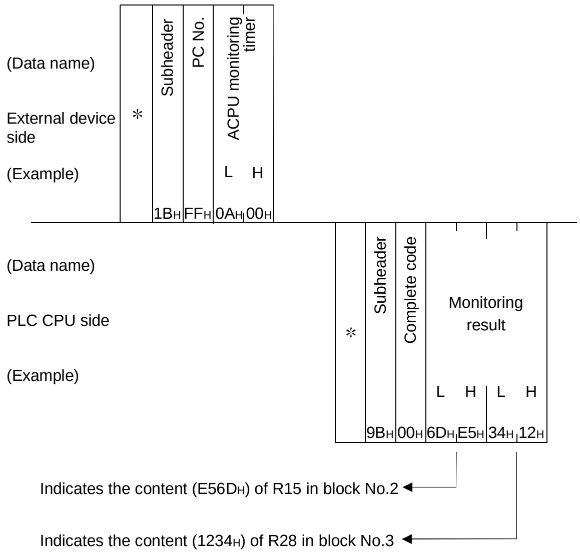
The examples shown in this section explain the command/response format when monitoring extension file registers registered by monitor data registration.

For more details on the order and contents of data items of the areas marked by "✻" shown in the control procedure diagram, see Section 6.1.

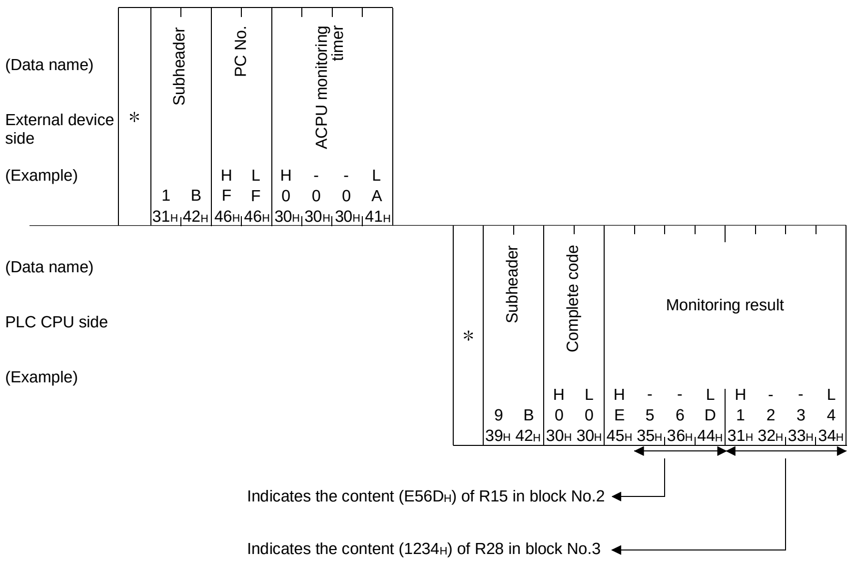
[Control procedure]

Monitoring R15 in block No.2 and R28 in block No.3 of the PLC CPU on which the Q series E71 is loaded.

(a) Communication in binary code




(b) Communication in ASCII code



## 6.5 Reading and Writing in the Buffer Memory of an Intelligent Function Module

The following explains how to designate the control procedure, as well as the contents of setting items and designation examples, when reading data from and writing data to the buffer memory of an intelligent module (special functional module) in an A series PLC CPU station or a MELSECNET/10 (both are other station).
This command accesses the buffer memory of an intelligent function module in byte units.

### 6.5.1 Commands and processing

(1) Commands

| Item | Command/response format | Processing | Number of points processed per communication | PLC CPU status | | |
|---|---|---|---|---|---|---|
| | | | | During STOP | During RUN | |
| | | | | | Write allow setting | Write prohibit setting |
| Batch read | $0E_H$ | Reads from the buffer memory of the intelligent function module. | 256 bytes (128 words) | ○ | ○ | ○ |
| Batch write | $0F_H$ | Writes data to the buffer memory of an intelligent functional module.. | | ○ | ○ | × |

In the PLC CPU status column of the table above, ○ indicates that the corresponding function is executable and × indicate that it is not executable.

(2) Accessible modules and addresses of the buffer memory
The module models in the A/QnA series that can be accessed using this function and the designated head address of the buffer memory are the same as when accessing an intelligent function module while communicating using QnA compatible 3E/3C/4C frames. See Sections 3.5.1 and 3.5.2. (Intelligent function modules of the Q series cannot be accessed.)
This function reads and writes data in byte units.

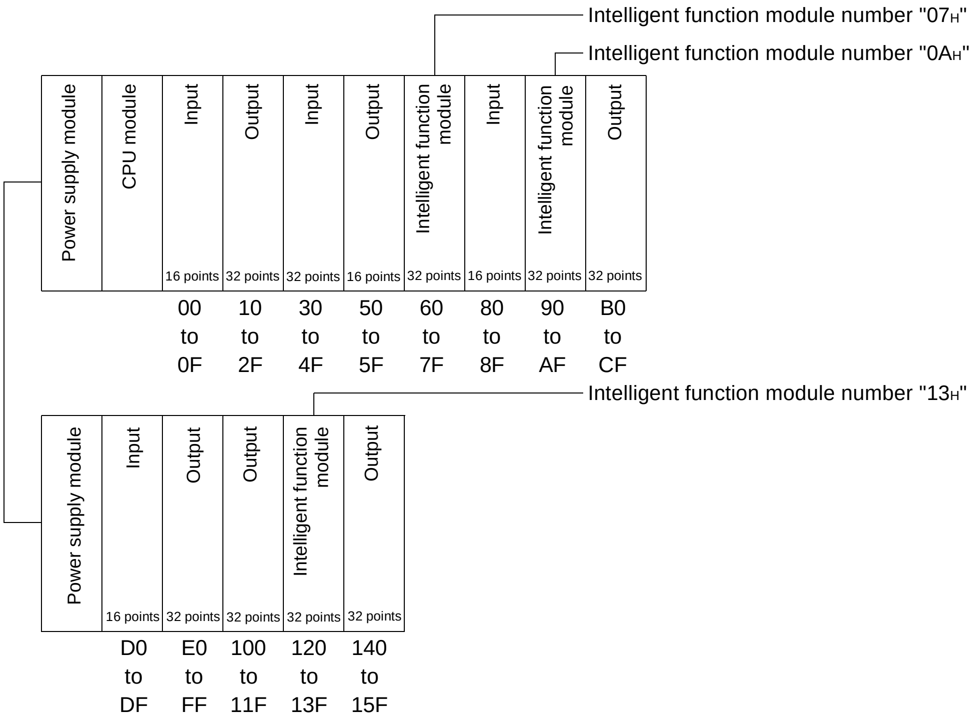
**POINT**

The buffer memory of each intelligent function module has read/write areas, read only areas, write only areas, and areas that may used by the OS but not by the user.
Execute this function according to the explanation in each module's manual.
If the read/write operations are not performed properly, an error may occur in the PLC CPU and each intelligent function module.

### 6.5.2 Understanding the intelligent function module numbers in the control procedure

(1) Module number of an intelligent function module that occupies 1 slot

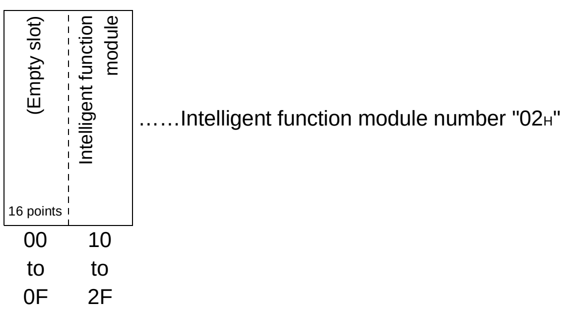
The intelligent function module number designated in the control procedure is the first 2 digits of the last (3-digit) number of the input/output signal (I/O address).



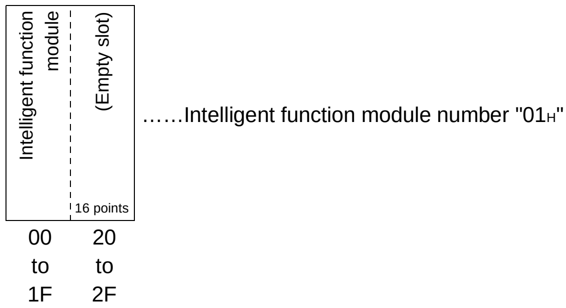

(2) Module number of an intelligent function module that occupies 2 slots

For an intelligent function module that occupies two slots, the number of occupied points for each slot is determined for each module.
The intelligent function module number designated in the control procedure is the first 2 digits of the last (3-digit) number of the input/output signal (I/O address) of the slot on which the module is installed.
For details on how each slot is assigned in each module, see the User's Manual of the applicable intelligent functional module.

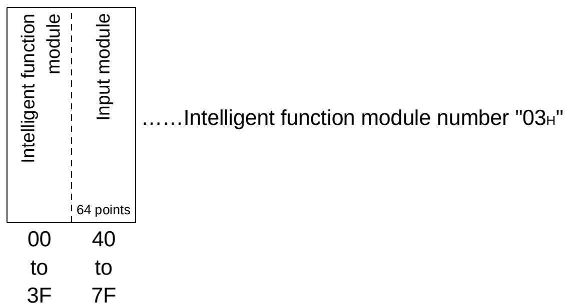
1) In case of a module that assign the first half slots as empty slots (AD72, A84AD, etc.)



2) In case of a module that assign the last half slots as empty slots (A61LS, etc.)

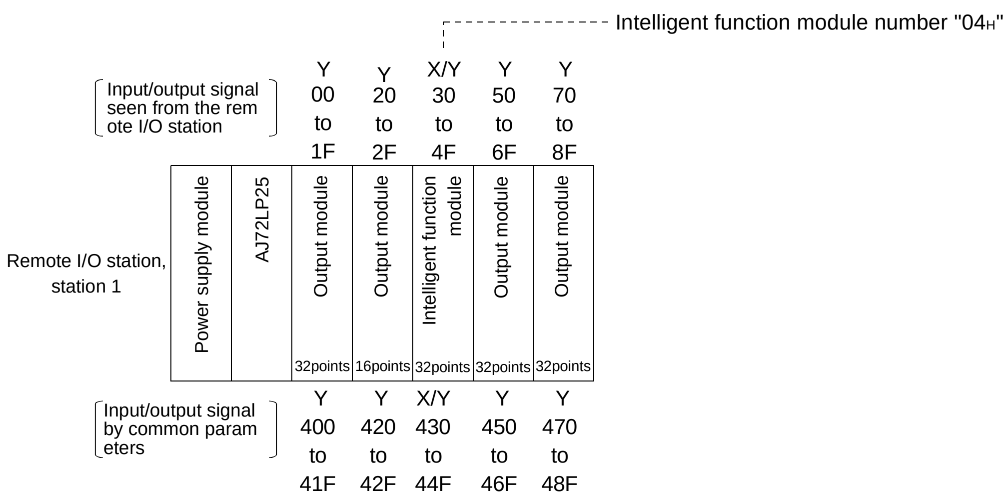



3) In case of a module in which both an intelligent function module and an input/output module are assigned
(In case of the A81CPU)




## (3) Intelligent function module number of the MELSECNET/10 remote I/O station

All intelligent function module numbers of remote I/O station are given by the first 2 digits of the last (3-digit) number of the "input/output signal see from the remote I/O station" shown below.
Designate the intelligent function module number using the "input/output signal see from the remote I/O station" regardless of the common parameters set in the master station of MELSECNET/10 remote I/O net.



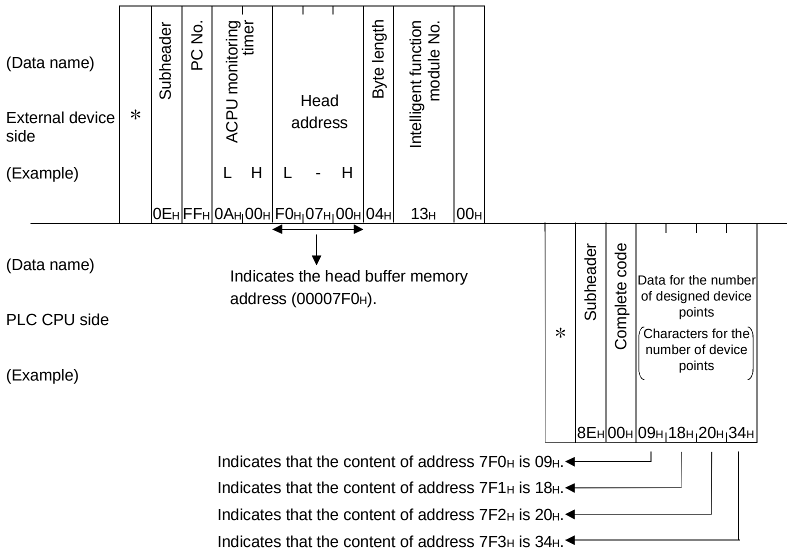
### 6.5.3 Reading from the buffer memory of an intelligent function module (command: 0E)

The examples shown in this section explain the command/response format when reading data from the buffer memory of the intelligent function module.
For more details on the order and contents of data items of the areas marked by "✻" shown in the control procedure diagram, see Section 6.1.

[Control procedure]

Reading the contents of buffer memory address 7F0H to 7F3H of the intelligent function modules (input/output signal: 120 to 13F (module No.13H)) on the same station on which the Q series E71 is loaded.

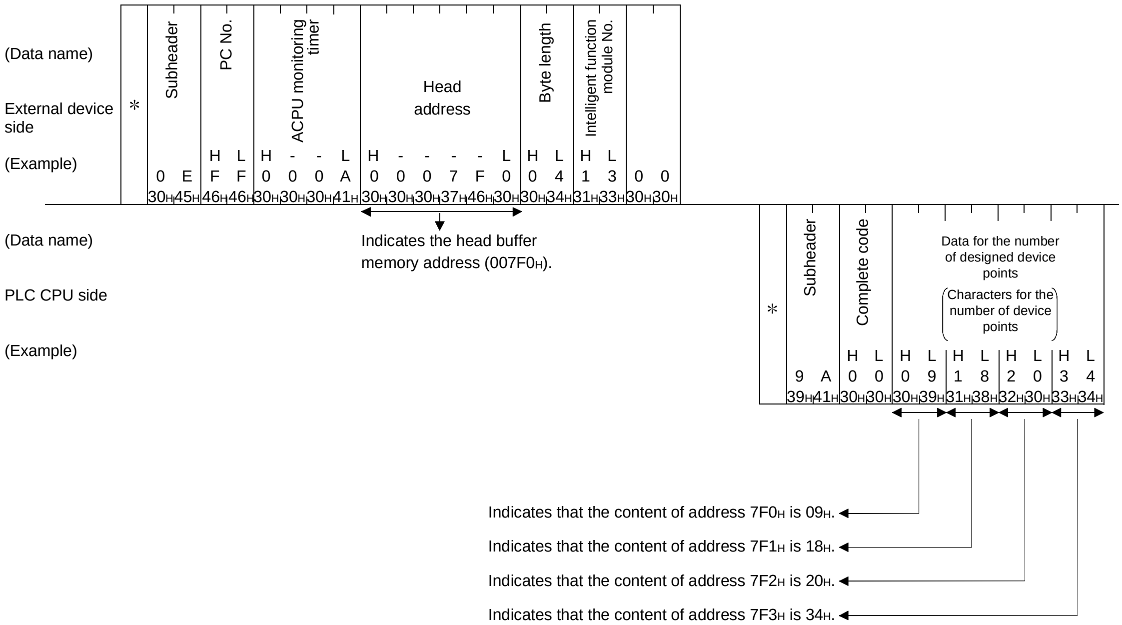
(1) Communication in binary code




**REMARK**

Use the designation "00H" when the byte length is 256 bytes.

(2) Communication in ASCII code




**REMARK**

Use the designation "00" is used when the byte length is 256 bytes.

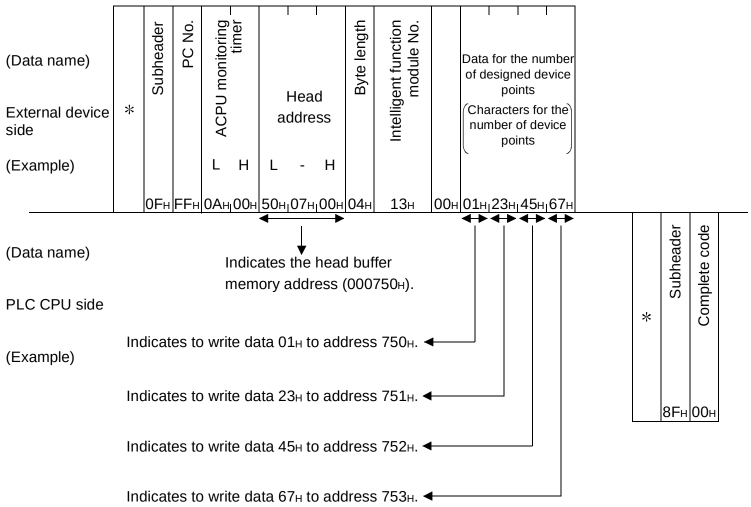
## 6.5.4 Writing to the buffer memory of an intelligent function module (command: 0F)

The examples shown in this section explain the command/response format when writing data to the buffer memory of the intelligent function module.
For more details on the order and contents of data items of the areas marked by "∗" shown in the control procedure diagram, see Section 6.1.

[Control procedure]

Writing data to buffer memory address 750H to 753H of the intelligent function modules (input/output signal: 120 to 13F (module No. 13H)) on the same station on which the Q series E71 is loaded.

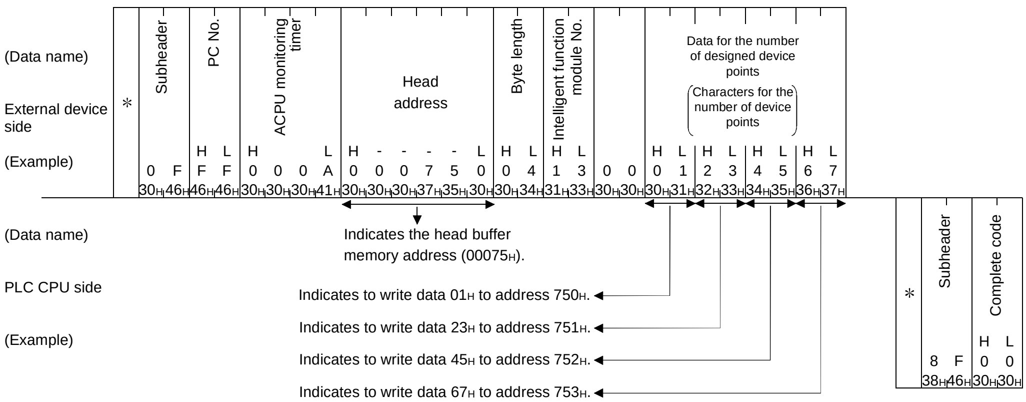
### (1) Communication in binary code




REMARK

Use the designation "00H" when the byte length is 256 bytes.

### (2) Communication in ASCII code




REMARK

Use the designation "00" when the byte length is 256 bytes.

# APPENDIX

## Appendix 1 Reading and Writing by Designation of the Device Memory Extension

The extension designation of the device memory is used by an external device to designate other devices than the ones described in Section 3.3.1 (3) using the commands listed in Section 3.3.1 (1). It is also used to qualify and designate device numbers and network numbers of the device memory to access.
The following section provides an outline of extension designation of the device memory.
In the explanations that follow this section, these designation expressions are indicated as [Designation-1] to [Designation-5].

### [Designation-1]

A designation for accessing a direct link device of MELSECNET/H, MELSECNET/10 (link input, link output, link special relay, etc.; see the table in Appendix 1.2 (2).)

### [Designation-2]

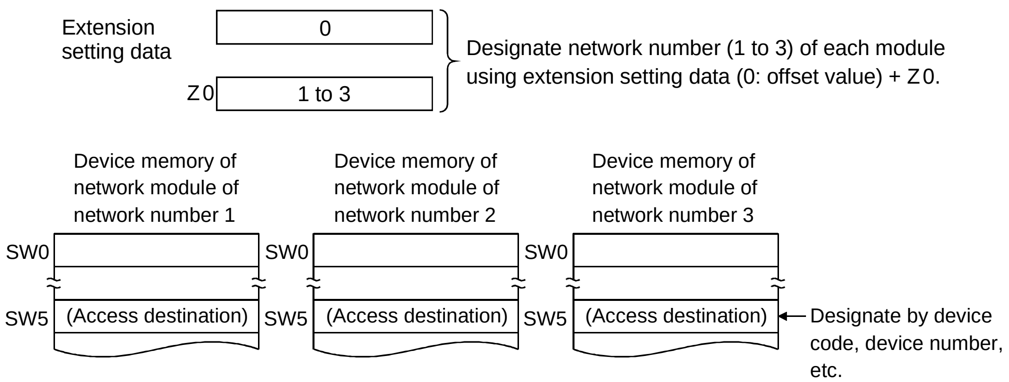
A designation for accessing an intelligent function module device (including a special function module device) of an intelligent function module (including special function modules of the A/QnA series) on a Q/QnACPU station and an MELSECNET/H, MELSECNET/10 remote I/O station (buffer register; see the table in Appendix 1.2 (2).)

### [Designation-3]

A designation of target modules by qualifying extension setting of network number and input/output signals by 1) and 2) below.

1) Designating target modules of [Designation-1] above
Target a network module of (arbitrary network number) + (network number designated by index register)
2) Designating target modules of [Designation-2] above
Target an intelligent function module of (arbitrary input/output signal number) + (input/output signal number designated by index register)

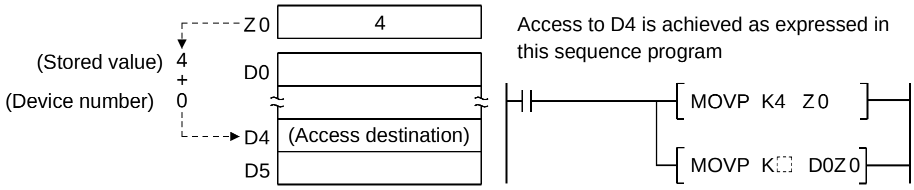
For example, when multiple network modules are loaded on the access station, an external device can access the same device memory (SW5) of each module by registering the extension setting data and index register "Z0" as shown in the following explanation diagram.



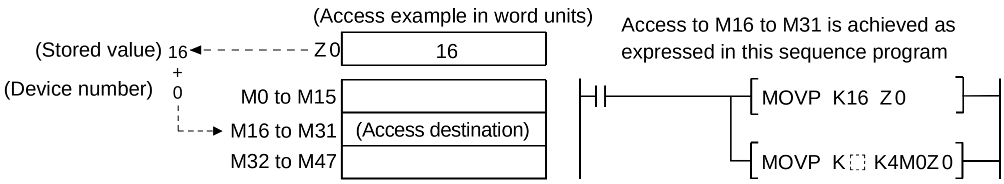
## [Designation-4]

A designation of target device number for devices of [Designation-1] and [Designation-2] above and Section 3.3.1 (3) by qualifying devices with device numbers and index registers.

For example, by designating D0 and Z0, device memory (D4) can be accessed.



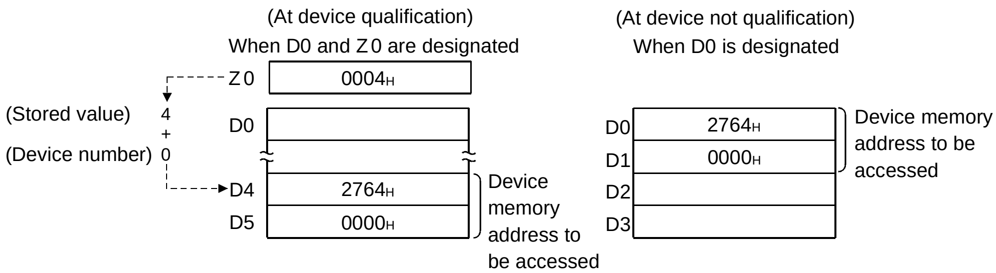

By designating M0 and Z0, device memory (M16 or M16 to 31) can be accessed.




## [Designation-5]

An indirect designation of a word device in which the stored value in the designated word device in turn gives the address of the device memory to be accessed.

In the following case, the device memory address to be accessed, 00002764H, is assigned to the designated word device.




**POINT**

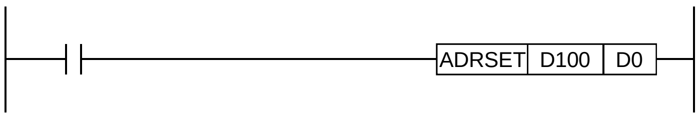
When an external device accesses a device memory by indirect designation, arrange the following with the PLC CPU side before accessing.

(1) Confirm that the address of the memory to which the device memory to be accessed is correctly assigned, using the ADRSET instruction of the PLC CPU.

(2) Store the address confirmed in 1) above in the device memory designated by the indirect designation.

(Example)

An example of a sequence program that stores the address of D100 to D0 and D1 for when an external device accesses D100.
(The external device can access D100 by designating D0 indirectly.)



## Appendix 1.1 Data order and contents in the character areas when the device memory extension is designated

This section explains what is designated in the additional area for extension designation, etc. when designating extension of a device memory that reads and writes data.
See Section 3.3.1 (2) and (3) for explanations of the data items that are the same as when the device memory extension is not designated.

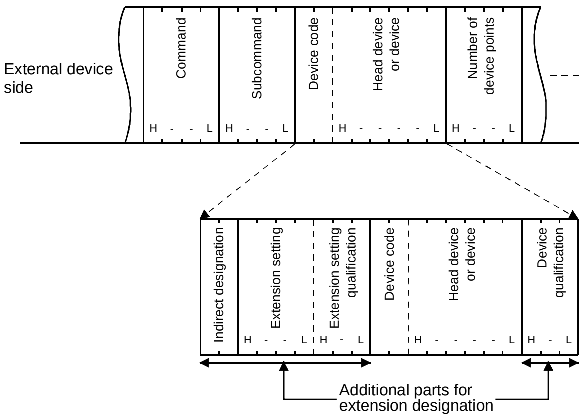
**POINT**

(1) When designating extension, designate the device memory by replacing the device memory designation area used when extension is not designated with the one used when extension is designated.

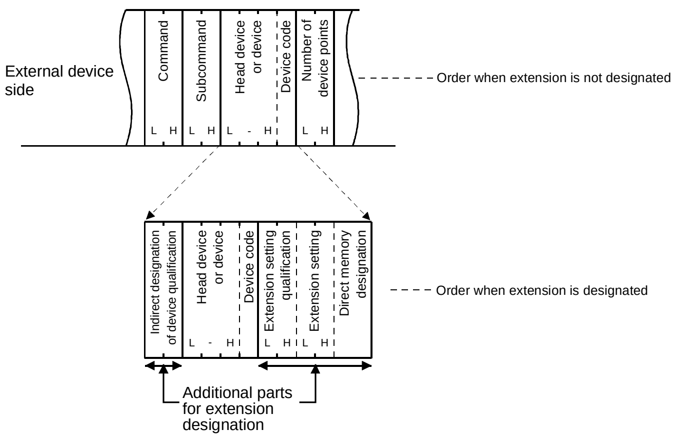
(2) When designating multiple device memory, designate them all to be extended.

(a) Data order of the device memory designation area when designating extension of the device memory.

1) Data order of QnA compatible 3E/3C/4C frames when performing data communication in ASCII code




2) Data order of QnA compatible 3E/3C/4C frames when performing data communication in binary code



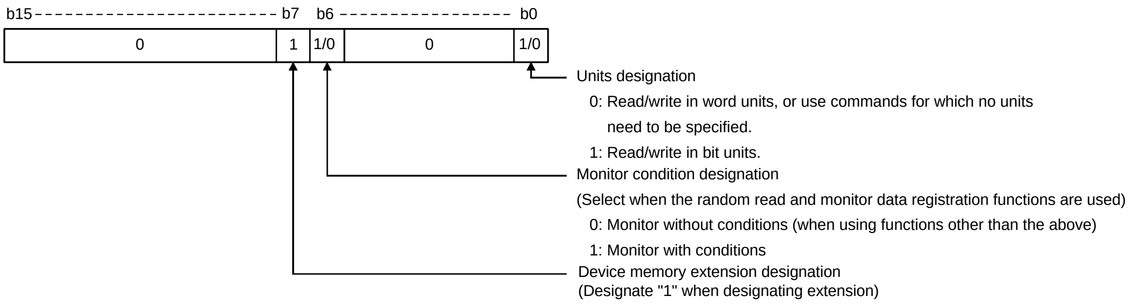
(b) Contents of character areas when designating extension of device memory
The following tables show each value to be designated in the character areas when designating extension of device memory.
(Data communication in ASCII code)

| | | Value designated by an external device | | | | | Number of designated characters |
|---|---|---|---|---|---|---|---|
| | | [Designation-1] | [Designation-2] | [Designation-3] | [Designation-4] | [Designation-5] | |
| Subcommand | Monitor with condition | "0080"/"0081" (See 1)) | | | | | 4 |
| | Monitor without condition | "00C0"/"00C1" (See 1)) | | | | | |
| Indirect designation | | "00" | | | | "0@' | 2 |
| Extension setting | | "J□" (See 2)) | "U□" (See 2)) | Either value to the left/"0000" (See 2)) | | "0000" | 4 |
| Extension setting qualification | | "000" | | "Z□" (See 3)) | Either value to the left (See 3)) | "000" | 3 |
| Device code | | (See Appendix 1.2 (2) (a) and Section 3.3.1(3)) | | | | | 2 |
| Head device or device | | | | | | | 6 |
| Device qualification | | "000" | | | "Z□" (See 4)) | Either value to the left (See 4)) | 3 |

(Data communication in binary code)

| | | Value designated by an external device | | | | | Number of designated bytes |
|---|---|---|---|---|---|---|---|
| | | [Designation-1] | [Designation-2] | [Designation-3] | [Designation-4] | [Designation-5] | |
| Subcommand | Monitor with condition | "0080H"/"0081H" (See 1)) | | | | | 2 |
| | Monitor without condition | "00C0H"/"00C1H" (See 1)) | | | | | |
| Device qualification indirect designation | | 0000H | | | 40□H (See 4)) | 0800H/ □8□H (See 4)) | 2 |
| Head device or device | | (See Appendix 1.2 (2) (a) and Section 3.3.1(3)) | | | | | 3 |
| Device code | | | | | | | 1 |
| Extension setting qualification | | 0000H | | 40□H (See 3)) | Either value to the left (See 3)) | 0000H | 2 |
| Extension setting | | □H (See 2)) | | | | 0000H | 2 |
| Direct memory designation | | F9H | F8H | Either value to the left/00H (See 2)) | | 00H | 1 |

1) Subcommand
This value is for designating units for read/write, type of device to be designated, conditions for reading data, etc.
a) Data communication in ASCII code
The value shown below is converted to 4-digit ASCII code (hexadecimal) and transmitted sequentially beginning from the most significant digit.
b) Data communication in binary code
The 2-byte value shown below is transmitted.
c) The following contents are designated with a subcommand.



d) For details on the monitor condition designation (for random read and monitor data registration) and the read timing of device memory when monitoring with conditions, see Section 3.3.8.

e) When device memory extension is designated, the subcommand should be either $0080_H$, $0081_H$, $00C0_H$, or $00C1_H$ according to the monitor condition and unit designations.

2) Extension designation and direct memory designation

These values are for device memory designation when an external device accesses the device memory listed below.

- Direct link device : [Designation-1]
- Direct device of intelligent functional module : [Designation-2]

a) Data communication in ASCII code

Designate only the extension setting data and send the following value sequentially beginning from the most significant digit.

| Designation value | Target device memory | Remark |
|---|---|---|
| "0000" | (Without extension setting) | – |
| "J □ " | Direct link device | Designate the access destination network number by converting it to 3-digit ASCII code (hexadecimal) and inserting it in the □ field. |
| "U □ " | Intelligent function module device of intelligent function module | Designate head input/output signal (4-digit hexadecimal) of the access destination intelligent function module, inserting the three most significant digits in the □ field. |

d) Data communication in binary code

Designate extension setting and direct memory designation data and send the following value. (The extension setting data should be sent beginning from the low byte (L: bits 0 to 7).)

| Designation value | | Target device memory | Remark |
|---|---|---|---|
| Extension setting | Direct memory designation | | |
| $0000_H$ | $00_H$ | (Without extension setting) | – |
| $□_H$ | $F9_H$ | Direct link device | Designate the access destination network number in the □ field. |
| $□_H$ | $F8_H$ | Intelligent function module device of intelligent function module | Designate head input/output signal (hexadecimal) of the access destination intelligent function module, inserting the three most specificant digits in the □ field. |

3) Extension setting qualification (for [Designation-3])

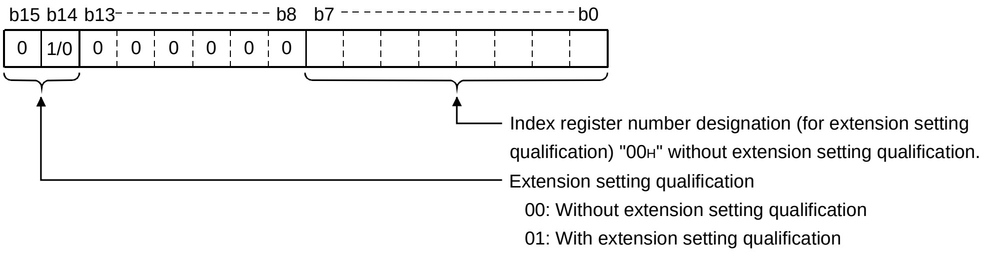
This value is used to designate modules of arbitrary network number and input/output signal, by using the designated value in the extension setting as offset value for the network number and input/output signal, and furthermore using the index register.

a) Data communication in ASCII code

The value shown below is transmitted sequentially beginning from the most significant digit.

| Designation value | Description | Remark |
|---|---|---|
| "000" | (Without extension setting qualification) | – |
| "Z□" | Index register for extension setting qualification | Designate index register number by converting it to 2-digit ASCII code (decimal) and inserting it in the □ field. |

b) When data communicating in binary code, use the value below and send it beginning from the low byte (L: bits 0 to 7).)




c) An index register (Z0 to Z15) can be used for the extension setting qualification.

d) When storing an input/output signal in an index register for extension setting qualification, designate it by the value obtained from the following subtraction.

(value stored to index register for extension setting qualification) =
(most significant 3-digits of head input/output (4-digit) signal of access destination module)
- (extension setting designation value)

4) Device qualification and indirect designation

(Device qualification: for [Designation-4])

This data item is for designating an arbitrary device number of the same device by using the designation value in head device (or device) as the offset value for the device number, and further adding the value in the index register.

(Indirect designation: for [Designation-5])

By this data designation, the value stored in the designated head device (or device) and the value stored in the immediately following device memory define the address of the device memory to be accessed by an external device.

Indirect designation can be used when accessing word devices.

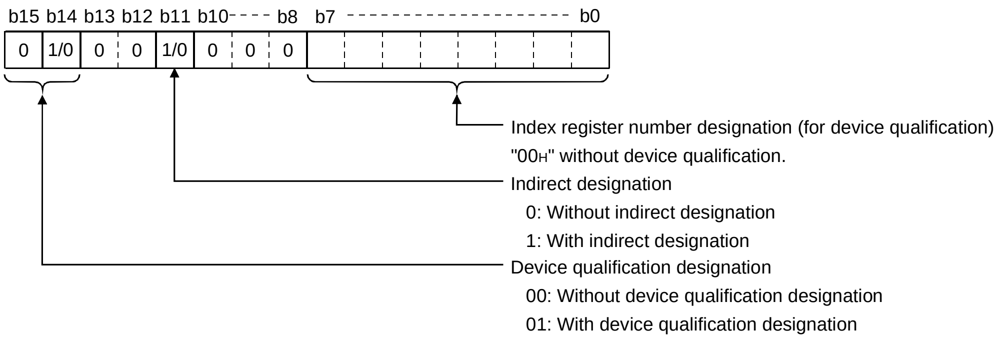
a) Data communication in ASCII code

Designate only the device qualification data and send the following value sequentially beginning from the most significant digit.

| Designation value | Description | Remark |
|---|---|---|
| "000" | (Without device qualification) | – |
| "Z□" | Index register for device qualification | Designate index register number by converting it to 2-digit ASCII code (decimal) and inserting it in the □ field. |

b) Data communication in binary code

Designate device qualification and indirect designation data, use the following value and send it beginning from low byte (L: bits 0 to 7).)




c) The index registers (Z0 to Z15) can be used for extension setting qualification.

## Appendix 1.2 Device memory which can have designated extensions and an example of designation

(a) Device memory which can have extensions designated

The extension designation is possible for the following device memory in stations on which the Q series C24/E71 is loaded and Q/QnACPU station connected by MELSECNET/H, MELSECNET/10.

1) Device memory of the Q/QnACPU (see Section 3.3.1 (3))
2) Link direct devices of network modules and intelligent function module devices of intelligent function modules (see the table below)

| Classification | Device | Device identification | | Extension setting | | Device code | | Device number range | Representation | | Remark |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | Bit | Word | Communication in ASCII code | Communication in binary code | Communication in ASCII code | Communication in binary code | | Hexa-decimal | Hexa-decimal Decimal | |
| Link direct device | Link input | ○ | | J□ | F9$_H$ | X ✻ | 9C$_H$ | 000000 to 001FFF | – | ○ | Designate network number in □ of extension setting. |
| | Link output | ○ | | | | Y ✻ | 9D$_H$ | 000000 to 001FFF | – | ○ | |
| | Link relay | ○ | | | | B ✻ | A0$_H$ | 000000 to 001FFF | – | ○ | |
| | Link special relay | ○ | | | | SB | A1$_H$ | 000000 to 0001FF | – | ○ | |
| | Link register | – | ○ | | | W ✻ | B4$_H$ | 000000 to 001FFF | – | ○ | |
| | Link special register | – | ○ | | | S ✻ | B5$_H$ | 000000 to 0001FF | – | ○ | |
| Intelligent function module devices | Buffer memory register (buffer memory) | – | ○ | U□ | F8$_H$ | G ✻ | AB$_H$ | 000000 to 016383 | ○ | – | Designate input/output signal of the target module in □ of extension setting. |

✻ "00...0" of device code and head device (or device) designations can be designated by blank spaces (code: 20$_H$) as shown in Section 3.3.1 (2) (c) 2) and 3).

(b) Example of device memory extension designation

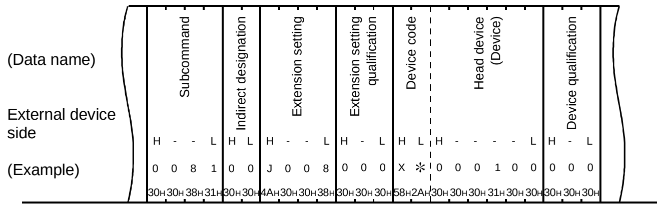
The following diagrams illustrate examples of device memory extension designation (using subcommands without monitor condition).

[Designation-1]

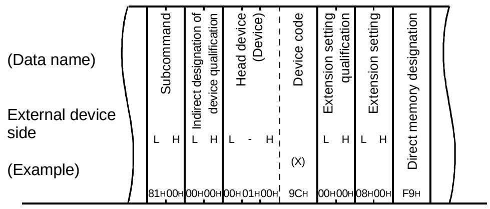
a) Accessing the following device memory

- Target module : Network module with network number 8 (008H)
- Device number : X100: Access in bit units

(Designation method when performing data communication in ASCII code)



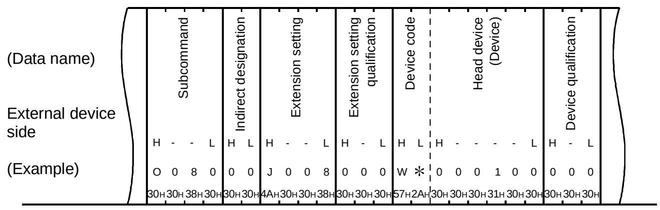

(Designation method when performing data communication in binary code)



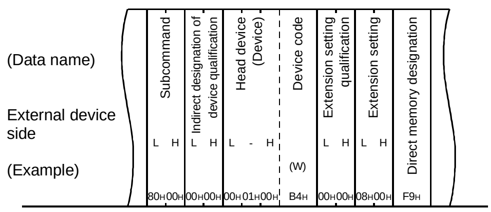

b) Accessing the following device memory

- Target module : Network module with network number 8 (008H)
- Device number : W100: Access in word units

(Designation method when performing data communication in ASCII code)




(Designation method when performing data communication in binary code)

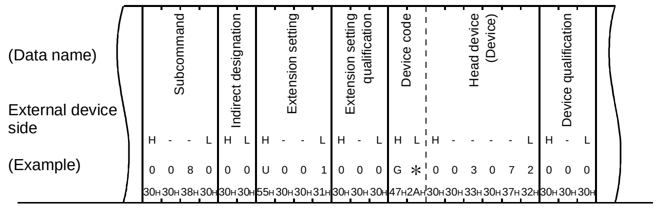


[Designation-2]

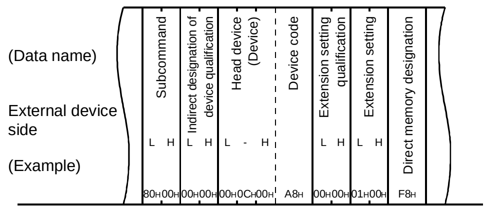
a) Accessing the following buffer memory

- Target module : Intelligent function module with input/output signal 010H
- Device number : 3072 (C00H)

(Designation method when performing data communication in ASCII code)




(Designation method when performing data communication in binary code)

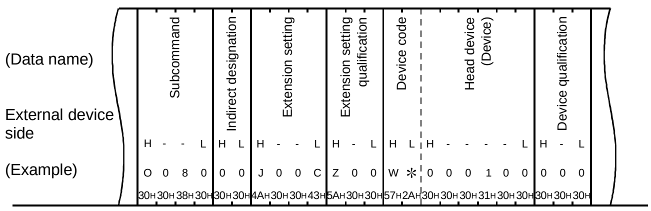


[Designation-3]

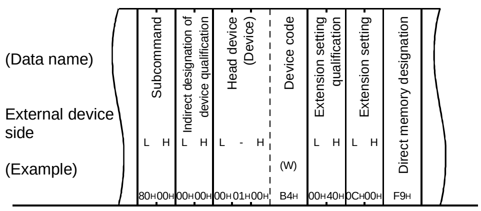
a) Accessing the following device memory

- Target module : Network module whose network number indicated by 12 (0CH) + Z0
- Head address : W100

(Designation method when performing data communication in ASCII code)



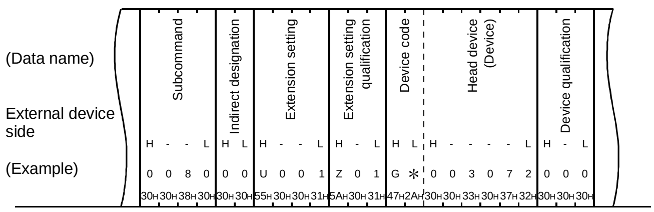

(Designation method when performing data communication in binary code)



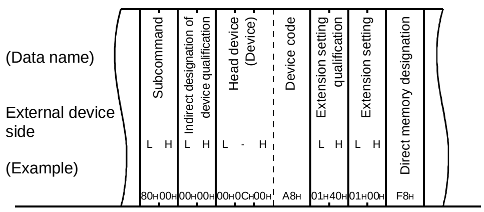

b) Accessing the following buffer memory

- Target module : Intelligent functional module whose head input/output signal is indicated by 010H + Z1
- Head address : 3072 (C00H)

(Designation method when performing data communication in ASCII code)




(Designation method when performing data communication in binary code)

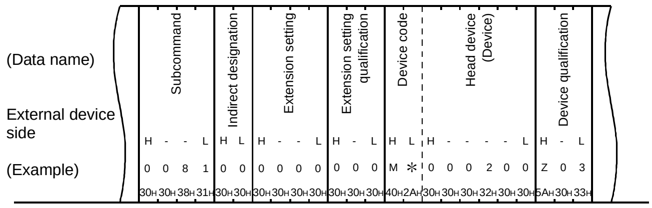


[Designation-4]

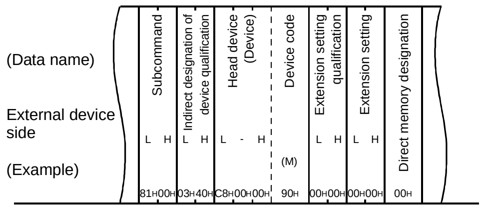
a) Accessing the following device memory

- Device number : Internal relay (M) designated by M200 + Z3 : Access in bit units

(Designation method when performing data communication in ASCII code)



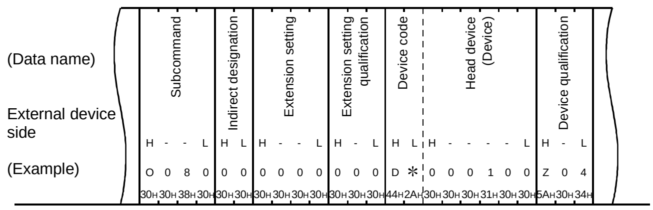

(Designation method when performing data communication in binary code)



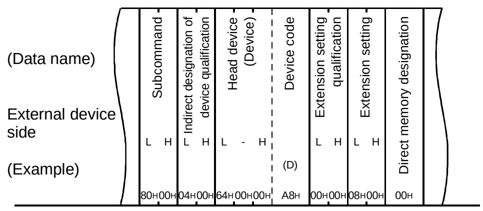

b) Accessing the following device memory

- Device number : Data register (D) designated by D100 + Z4 : Access in word units

(Designation method when performing data communication in ASCII code)



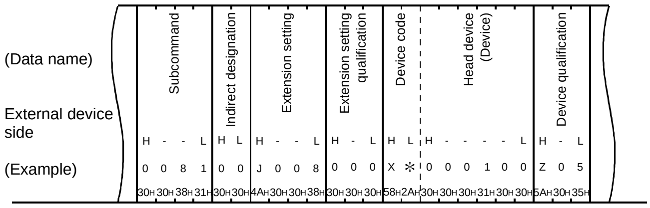

(Designation method when performing data communication in binary code)



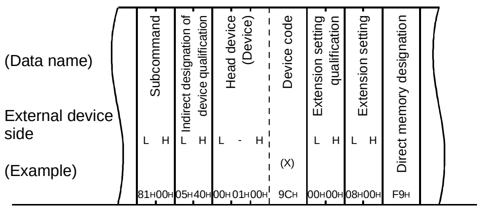
c) Accessing the following device memory

- Target module : Network module with network number 8 (008H)
- Device number : Link input (X) indicated by X100 + Z5 : Access in bit units

(Designation method when performing data communication in ASCII code)



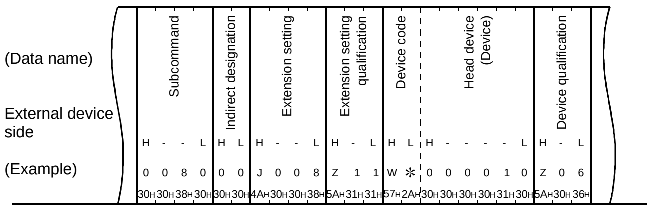

(Designation method when performing data communication in binary code)



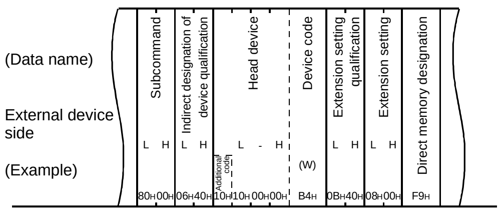

d) Accessing the following device memory

- Target module : Network module whose network number is indicated by 8 (008H) + Z11
- Device number : Link register (W) indicated by W10 + Z6 : Access in word units

(Designation method when performing data communication in ASCII code)



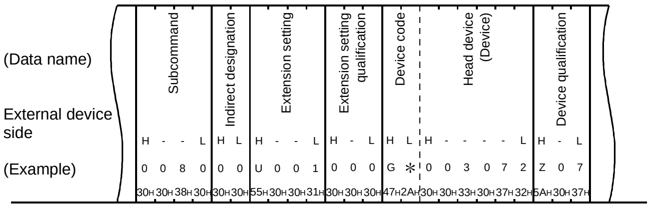

(Designation method when performing data communication in binary code)



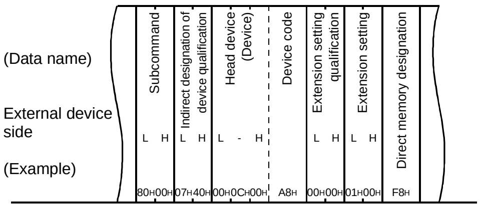
e) Accessing the following buffer memory

- Target module : Intelligent function module with head input/output signal 010H
- Head address : The address indicated by 3072 (C00H) + Z7

(Designation method when performing data communication in ASCII code)



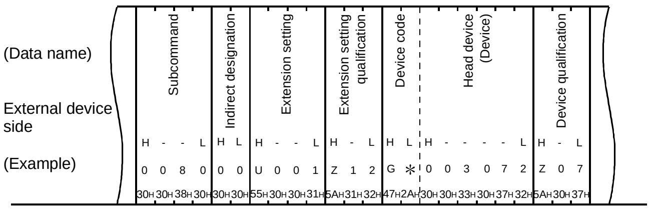

(Designation method when performing data communication in binary code)



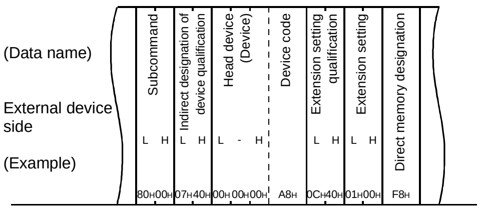

f) Accessing the following buffer memory

- Target module : Intelligent function module whose head input/output signal is indicated by 010H + Z12
- Head address : The address indicated by 3072 (C00H) + Z7

(Designation method when performing data communication in ASCII code)




(Designation method when performing data communication in binary code)

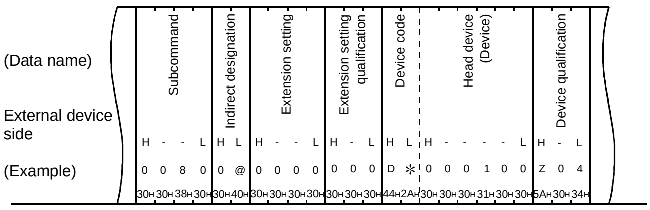


[Designation-5]

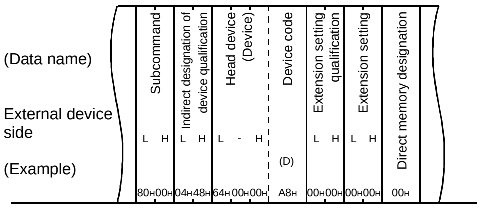
Accessing the following device memory

- Device number : Device memory address stored in data register (D) indicated by D100 + Z4 : Access in word units

(Designation method when performing data communication in ASCII code)




(Designation method when performing data communication in binary code)



## Appendix 1.3 Restrictions when designating the device memory extension

This section explains the restrictions that apply when designating the device memory extension.

(a) Commands that can designate the device memory extension
The following table outlines the device memory extension designation items for each of the commands for reading from/writing to the device memory for QnA compatible 3E/3C/4C frames.

| Function | | Access device | Command | Device memory extension designation item | | | | | Designation expression | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | Indirect designation | Extension setting | Extension setting qualification | Device qualification | Direct memory designation | Designation -1 | Designation -2 | Designation -3 | Designation -4 | Designation -5 |
| Batch read<br>Batch write | Bit units | Bit | 0401<br>1401 | × | ○ | × | × | ○ | ○ | ○ | × | × | × |
| | Word units | Bit | | | | | | | | | | | |
| | | Word | | | | | | | | | | | |
| Random read | Word units | Bit | 0403 | × | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ |
| | | Word | | ○ | | | | | | | | | |
| Test<br>(random write) | Bit units | Bit | 1402 | × | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ |
| | Word units | Bit | | × | | | | | | | | | |
| | | Word | | ○ | | | | | | | | | |
| Monitor data registration | Word units | Bit | 0801 | × | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ |
| | | Word | | ○ | | | | | | | | | |
| Monitor | Word units | – | 0802 | × | × | × | × | × | – | – | – | – | – |
| Multiple block batch read<br>Multiple block batch read | Word units | Bit | 0406<br>1406 | × | ○ | × | × | ○ | ○ | ○ | × | × | × |
| | | Word | | | | | | | | | | | |

Bit: bit device
Word: word device

○: Can be designated ×: Cannot be designated

(b) Composite designations of device memory extension designations
When using the following functions, multiple device names can be designated in a command message. In this case, designate all the devices designated to be extended in the command message when designating the device memory extension.
"With extension designation" and "without extention designation" cannot be used at the same time for device memory.
- Random read function : (command: 0403)
- Test (random write) function : (command: 1402)
- Monitor data registration function : (command: 0801)
- Multiple block batch read/write function : (command: 0406, 1406)

(c) Access to intelligent function modules (including special function modules)
1) It is possible to read and write by designating device memory extension in the intelligent function module buffer memory of an intelligent function modules loaded on the following stations.
   - Q series C24/E71 loaded stations connected to external devices and Q/QnACPU stations linked with such Q series C24 by multidrop connection.
   - Q/QnACPU stations on MELSECNET/H, MELSECNET/10 and corresponding Q/QnA stations linked by multidrop connection with Q series C24 via such Q/QnACPU stations.
2) Follow the detailed explanations in Section 3.5 when reading from and writing to intelligent function modules loaded on stations other than above.

## Appendix 2 Reading from and Writing to the Buffer Memory

This section explains functions used by an external device to read or write data in the buffer memory of an intelligent function module (including special function modules) loaded on a QCPU station on which the Q series C24/E71 is loaded (local station), or on a PLC station without Q series C24/E71 in a network system (other station).
The following table outlines the PLC stations whose intelligent function module buffer memory can be read/written in, the functions used, and their commands.

The following table lists the target stations and modules of Access-1 to Access-4 and the target functions/commands of method of read/write.

| | Read/write target | | | | Method of read/write | | | |
|---|---|---|---|---|---|---|---|---|
| | No. | Target station | | Target module | QnA extension designation (appendix 1) | QnA compatible 3E/3C/4C frames [0601], [1601] | QnA compatible 3E/3C/4C frames [0613], [1613] | Command for A compatible 1E/1C frames [OE], [OF], [TR], [TW] |
| Access-1 (local station) | (1) | External device connection station | QCPU station | Q series C24/E71 | × | × | ○ | × |
| | | | Q compatible remote I/O station | | | | | |
| Access-2 (local station) | (2) | External device connection station | QCPU station | Intelligent function module (Excluding the Q series C24/E71) | ○ | ○ | × | × |
| | | | Q compatible remote I/O station | | × | ○ | × | × |
| Access-3 (other station) | (3) | Station on MELSECNET/H, MELSECNET/10 | Q/QnACPU station | Intelligent function module (Including the Q series C24/E71) | ○ | ○ | × | × |
| | (4) | | QnA compatible remote I/O station | | × | ○ | × | × |
| Access-4 (other station) | (5) | | PLC CPU station other than Q/QnACPU | | × | × | × | × |
| | (6) | | AnU compatible remote I/O station | | × | × | × | × |

○: Can be designated ×: Cannot be designated

### (1) Target station and modules in Access-1 to Access-6 shown in the table

- Access-1 (local station)
  1) The Q series C24/E71 connected to an external device and the Q series C24 linked with such Q series C24 by multidrop connection.
- Access-2 (local station)
  2) An intelligent function module of a station on which the Q series C24/E71 is loaded, connected to an external device and an intelligent function module of the Q series C24 loaded station linked to such Q series C24 by multidrop connection.
- Access-3 (other station)
  3) An intelligent function module loaded on a Q/QnACPU station on MELSECNET/H, MELSECNET/10.
  4) An intelligent function module loaded on a Q/QnA compatible remote I/O station on MELSECNET/H, MELSECNET/10.
- Access-4 (other station)
  5) An intelligent function module loaded on a station other than Q/QnACPU on MELSECNET/10.
  6) An intelligent function module loaded on an AnU compatible remote I/O station on MELSECNET/10.

(2) Target functions/commands of the read/write methods shown in the table

- QnA extension designation: Read/write with the device memory extension designation shown in Appendix-1.
- Commands for QnA compatible 3E, QnA compatible 3C, and QnA compatible 4C frames

| | | |
|---|---|---|
| "0613" and "1613" | : | Read and write using the "0613" and "1613" commands for QnA compatible 3E/3C/4C frames as shown in Section 3.4. |
| "0601" and "1601" | : | Read and write using the "0601" and "1601" commands for QnA compatible 3E/3C/4C frames as shown in Section 3.5. |

- Commands for A compatible 1E/1C frames

| | | |
|---|---|---|
| "0E" and "0F" | : | Read and write using the "0E" and "0F" commands for A compatible 1E frames as shown in Chapter 6. |
| "TR" and "TW" | : | Read and write using the "TR" and "TW" commands for A compatible 1C frames as shown in Chapter 5. |

**POINT**

The buffer memory of each intelligent function module, as well as of the Q series C24/E71, has read/write areas, read only area, write only area, and areas for the OS to which user access is prohibited.
Perform these functions according to the explanations found in the applicable manuals for the modules used.
An error may occur in each intelligent function module if their memory is incorrectly read/written in.

# Appendix 3 Processing Time of the PLC CPU Side While Communicating Using the MC Protocol

This section explains the processing time on the PLC CPU side while communicating using the MC protocol.

## Appendix 3.1 Processing time of the PLC CPU (scan time delay)

In data communication using the MC protocol, the PLC CPU, while in the RUN status, processes requests from an external device for the number of processing points that can be performed for each END processing.
The following table outlines the intervening time to the scan time and the number of scans required for the processing in this case.

### (1) Processing time of the PLC CPU via Q series C24/E71
(In the case of commands for QnA compatible 3E/3C/4C frames)

| Item | | | Command | Sub-command | Number of access points 1)/2) | Intervening time [ms] (Scan time increase) | | | | Number of scans required for processing | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | Q02HCPU | | Q01CPU | | | |
| | | | | | | Number of access points: When 1) | Number of access points: When 2) | Number of access points: When 1) | Number of access points: When 2) | When [Enable during RUN] is set | When [Disable during RUN] is set |
| Device memory | Batch read | Bit units | 0401 | 0001 | 1/7904 | 0.10 | 0.24 | 1.44 | 2.25 | 1 | 1 |
| | | Word units | | 0000 | 1/960 | 0.09 | 0.16 | 1.45 | 2.54 | 1 | 1 |
| | Batch write | Bit units | 1401 | 0001 | 1/7904 | 0.11 | 0.26 | 1.46 | 2.89 | 1 | 2 |
| | | Word units | | 0000 | 1/960 | 0.09 | 0.16 | 1.47 | 2.66 | 1 | 2 |
| | Random read | Word units | 0403 | 0000 | 1/192 | 0.14 | 0.33 | 1.46 | 4.74 | 1 | 1 |
| | Test ＊ Random write | Bit units | 1402 | 0001 | 1/188 | 0.14 | 3.10 | 1.46 | 7.40 | 1 | 2 |
| | | Word units | | 0000 | 1/160 (＊1) | 0.15 | 2.05 | 1.48 | 4.32 | 1 | 2 |
| | Monitor data registration | Word units | 0801 | 0000 | – | 0.00 | 0.00 | 1.32 | 1.32 | 0 | 0 |
| | Monitor | | 0802 | 0000 | 1/192 (＊2) | 0.16 | 0.92 | 1.44 | 5.92 | 1 | 1 |
| | Multiple block batch read | Word units | 0406 | 0000 | 1/960 | 0.11 | 0.35 | 1.44 | 2.52 | 1 | 1 |
| | Multiple block batch read | Word units | 1406 | 0000 | 1/960 | 0.13 | 0.17 | 1.46 | 1.72 | 1 | 2 |
| Intelligent function module | Batch read | | 0601 | 0000 | — | 0.00 | 0.00 | — | — | 0 | 0 |
| | Batch write | | 1601 | 0000 | — | | | | | 0 | 1 |
| PLC CPU | Remote RUN | | 1001 | 0000 | (for 1 station) | | — | | | 1 | 1 |
| | Remote STOP | | 1002 | 0000 | | | | | | 1 | 1 |
| | Remote PAUSE | | 1003 | 0000 | | | | | | 1 | 1 |
| | Remote Latch clear | | 1005 | 0000 | | | | | | 1 | 1 |
| | Remote RESET | | 1006 | 0000 | | | | | | 1 | 1 |
| | Read CPU model name | | 0101 | 0000 | | | | | | 0 | 0 |

| Item | Command | Sub-command | Number of access points 1)/2) | Intervening time [ms] (Scan time increase) Q02HCPU Number of access points: When 1) | Q02HCPU Number of access points: When 2) | Q01CPU Number of access points: When 1) | Q01CPU Number of access points: When 2) | Number of scans required for processing When [Enable during RUN] is set | When [Disable during RUN] is set |
|---|---|---|---|---|---|---|---|---|---|
| Read directory/file information | 1810 | 0000 | 1/30 | 0.00 | 0.00 | — | — | 0 | 0 |
| Search directory/file information | 1811 | 0000 | — | | — | | | 0 | 0 |
| File creation | 1820 | 0000 | 1/— | 0.07 (∗3) | | | | 1 | 2 |
| File delete | 1822 | 0000 | | 0.06 (∗3) | | | | 0 | 1 |
| File copy | 1824 | 0000 | | | | | | 1 | 2 |
| File attribute modification | 1825 | 0000 | | 0.07 | | | | 0 | 1 |
| Date and time of file creation modification | 1826 | 0000 | | | | | | 0 | 1 |
| File open | 1827 | 0000 | | 0.00 (∗3) | | | | 0 | 0 |
| File read | 1828 | 0000 | 1/1920 | 0.07 | 0.07 | | | 1 | 1 |
| File write | 1829 | 0000 | | | | | | 1 | 2 |
| File close | 182A | 0000 | 1/— | 0.00 | — | | | 0 | 0 |
| Global | 1618 | 000n | 1/80 | | 0.00 | | | — | |
| On-demand | 2101 | — | | | | | | | |

∗1 Indicates processing time required when accessing by designating only the number of word access points.
∗2 Indicates processing time required when accessing by designating only the number of double word access points.
∗3 The target is a file whose size is "'0 bytes."

**POINT**

(1) The PLC CPU processes only one of the above items during END processing. When GX Developer, each module, etc. access the relevant PLC CPU at the same time, the processing of each new request has to wait until other requests are processed, thus the number of scans required for the processing increases even more.
(2) If this has an influence on the extension of scan time, access the PLC CPU a small amount at a time, breaking it up over many accesses.

(2) Processing time of the PLC CPU when using the Q00/Q01CPU serial communication function (In the case of command (system 4) for QnA compatible 4C frame)

<table>
<tr><th rowspan="3" colspan="3">Item</th><th rowspan="3">Command</th><th rowspan="3">Subcommand</th><th rowspan="3">Number of access points 1)/2)</th><th colspan="2">Intervening time [ms] (Scan time increase)</th><th colspan="2" rowspan="2">Number of scans required for processing</th></tr>
<tr><th colspan="2">Q01CPU</th></tr>
<tr><th>Number of access points: When 1)</th><th>Number of access points: When 2)</th><th>When [Enable during RUN] is set</th><th>When [Disable during RUN] is set</th></tr>
<tr><td rowspan="10">Device memory</td><td rowspan="2">Batch read</td><td>Bit units</td><td rowspan="2">0401</td><td>0001</td><td>1/3584</td><td>0.8</td><td>13.1</td><td colspan="2">3</td></tr>
<tr><td>Word units</td><td>0000</td><td>1/480</td><td>0.8</td><td>6.2</td><td colspan="2">3</td></tr>
<tr><td rowspan="2">Batch write</td><td>Bit units</td><td rowspan="2">1401</td><td>0001</td><td>1/3584</td><td>0.9</td><td>7.5</td><td colspan="2">3</td></tr>
<tr><td>Word units</td><td>0000</td><td>1/480</td><td>0.8</td><td>9.3</td><td colspan="2">3</td></tr>
<tr><td>Random read</td><td>Word units</td><td>0403</td><td>0000</td><td>1/96</td><td>0.8</td><td>7.5</td><td colspan="2">3</td></tr>
<tr><td rowspan="2">Test<br>* Random write</td><td>Bit units</td><td rowspan="2">1402</td><td>0001</td><td>1/94</td><td>0.8</td><td>7.1</td><td colspan="2">3</td></tr>
<tr><td>Word units</td><td>0000</td><td>1/80<br>(*1)</td><td>0.9</td><td>8.2</td><td colspan="2">3</td></tr>
<tr><td>Monitor data registration</td><td rowspan="2">Word units</td><td>0801</td><td>0000</td><td>1/96<br>(*2)</td><td>0.8</td><td>7.3</td><td colspan="2">3</td></tr>
<tr><td>Monitor</td><td>0802</td><td>0000</td><td>1/96<br>(*2)</td><td>0.7</td><td>2.4</td><td colspan="2">3</td></tr>
</table>

*1 Indicates processing time required when accessing by designating only the number of word access points.
*2 Indicates processing time required when accessing by designating only the number of double word access points.

**POINT**

(1) The PLC CPU processes only one of the above items during END processing. When GX Developer, each module, etc. access the relevant PLC CPU at the same time, the processing of each new request has to wait until other requests are processed, thus the number of scans required for the processing increases even more.

(2) If this has an influence on the extension of scan time, access the PLC CPU a small amount at a time, breaking it up over many accesses.

# MEMO

# INDEX





Index - 1

Ind

[W]





Index - 2

# 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.

## 6. Product application

(1) In using the Mitsubishi MELSEC programmable logic controller, the usage conditions shall be that the application will not lead to a major accident even if any problem or fault should occur in the programmable logic controller device, and that backup and fail-safe functions are systematically provided outside of the device for any problem or fault.

(2) The Mitsubishi programmable logic controller has been designed and manufactured for applications in general industries, etc. Thus, applications in which the public could be affected such as in nuclear power plants and other power plants operated by respective power companies, and applications in which a special quality assurance system is required, such as for Railway companies or Public service purposes shall be excluded from the programmable logic controller applications.
In addition, applications in which human life or property that could be greatly affected, such as in aircraft, medical applications, incineration and fuel devices, manned transportation, equipment for recreation and amusement, and safety devices, shall also be excluded from the programmable logic controller range of applications.
However, in certain cases, some applications may be possible, providing the user consults their local Mitsubishi representative outlining the special requirements of the project, and providing that all parties concerned agree to the special circumstances, solely at the users discretion.

# Q Corresponding MELSEC Communication Protocol

## Reference Manual

| | |
|---|---|
| MODEL | MC-PROTOCOL-R-E |
| MODEL CODE | 13JF89 |
| SH(NA)-080008-F(0406)MEE | |


MITSUBISHI ELECTRIC CORPORATION

HEAD OFFICE : 1-8-12, OFFICE TOWER Z 14F HARUMI CHUO-KU 104-6212,JAPAN
NAGOYA WORKS : 1-14 , YADA-MINAMI 5-CHOME , HIGASHI-KU, NAGOYA , JAPAN