Applicable Modules:

<table>
<thead>
<tr>
<th>Model Code</th>
<th>Model Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>F3LE12-0T</td>
<td>Ethernet Interface Module</td>
</tr>
</tbody>
</table>
Applicable Product:

- **Range-free Multi-controller FA-M3**
  
  Model code : F3LE12-0T  
  Model name : Ethernet Interface Module  

The document number and document model code for this manual are given below.  
Refer to the document number in all communications; also refer to the document number or the document model code when purchasing additional copies of this manual.

  
<table>
<thead>
<tr>
<th>Document No.</th>
<th>IM 34M6H24-04E</th>
</tr>
</thead>
<tbody>
<tr>
<td>Document Model Code</td>
<td>DOCIM</td>
</tr>
</tbody>
</table>
Important

About This Manual

- This Manual should be passed on to the end user.
- Before using the controller, read this manual thoroughly to have a clear understanding of the controller.
- This manual explains the functions of this product, but there is no guarantee that they will suit the particular purpose of the user.
- Under absolutely no circumstances may the contents of this manual be transcribed or copied, in part or in whole, without permission.
- The contents of this manual are subject to change without prior notice.
- Every effort has been made to ensure accuracy in the preparation of this manual. However, should any errors or omissions come to the attention of the user, please contact the nearest Yokogawa Electric representative or sales office.

Safety Precautions when Using/Maintaining the Product

- The following safety symbols are used on the product as well as in this manual.

**Danger.** This symbol on the product indicates that the operator must follow the instructions laid out in this instruction manual to avoid the risk of personnel injuries, fatalities, or damage to the instrument. Where indicated by this symbol, the manual describes what special care the operator must exercise to prevent electrical shock or other dangers that may result in injury or the loss of life.

**Protective Ground Terminal.** Before using the instrument, be sure to ground this terminal.

**Function Ground Terminal.** Before using the instrument, be sure to ground this terminal.

**Alternating current.** Indicates alternating current.

**Direct current.** Indicates direct current.
The following symbols are used only in the instruction manual.

⚠️ **WARNING**

Indicates a “Warning”.
Draws attention to information essential to prevent hardware damage, software damage or system failure.

⚠️ **CAUTION**

Indicates a “Caution”
Draws attention to information essential to the understanding of operation and functions.

**TIP**

Indicates a “TIP”
Gives information that complements the present topic.

**SEE ALSO**

Indicates a “SEE ALSO” reference.
Identifies a source to which to refer.

- For the protection and safe use of the product and the system controlled by it, be sure to follow the instructions and precautions on safety stated in this manual whenever handling the product. Take special note that if you handle the product in a manner other than prescribed in these instructions, the protection feature of the product may be damaged or impaired. In such cases, Yokogawa cannot guarantee the quality, performance, function and safety of the product.

- When installing protection and/or safety circuits such as lightning protection devices and equipment for the product and control system as well as designing or installing separate protection and/or safety circuits for fool-proof design and fail-safe design of processes and lines using the product and the system controlled by it, the user should implement it using devices and equipment, additional to this product.

- If component parts or consumable are to be replaced, be sure to use parts specified by the company.

- This product is not designed or manufactured to be used in critical applications which directly affect or threaten human lives and safety — such as nuclear power equipment, devices using radioactivity, railway facilities, aviation equipment, air navigation facilities, aviation facilities or medical equipment. If so used, it is the user’s responsibility to include in the system additional equipment and devices that ensure personnel safety.

- Do not attempt to modify the product.

### Exemption from Responsibility

- Yokogawa Electric Corporation (hereinafter simply referred to as Yokogawa Electric) makes no warranties regarding the product except those stated in the WARRANTY that is provided separately.

- Yokogawa Electric assumes no liability to any party for any loss or damage, direct or indirect, caused by the use or any unpredictable defect of the product.
- Yokogawa Electric makes no other warranties expressed or implied except as provided in its warranty clause for software supplied by the company.
- Use the software with one computer only. You must purchase another copy of the software for use with each additional computer.
- Copying the software for any purposes other than backup is strictly prohibited.
- Store the original media, such as floppy disks, that contain the software in a safe place.
- Reverse engineering, such as decompiling of the software, is strictly prohibited.
- No portion of the software supplied by Yokogawa Electric may be transferred, exchanged, or sublet or leased for use by any third party without prior permission by Yokogawa Electric.
General Requirements for Using the FA-M3 Controller

● Avoid installing the FA-M3 controller in the following locations:
  - Where the instrument will be exposed to direct sunlight, or where the operating
temperature exceeds the range 0°C to 55°C (32°F to 131°F).
  - Where the relative humidity is outside the range 10 to 90%, or where sudden
temperature changes may occur and cause condensation.
  - Where corrosive or flammable gases are present.
  - Where the instrument will be exposed to direct mechanical vibration or shock.
  - Where the instrument may be exposed to extreme levels of radioactivity.

● Use the correct types of wire for external wiring:
  - Use copper wire with temperature ratings greater than 75°C.

● Securely tighten screws:
  - Securely tighten module mounting screws and terminal screws to avoid problems
such as faulty operation.
  - Tighten terminal block screws with the correct tightening torque as given in this
manual.

● Securely lock connecting cables:
  - Securely lock the connectors of cables, and check them thoroughly before turning
on the power.

● Interlock with emergency-stop circuitry using external relays:
  - Equipment incorporating the FA-M3 controller must be furnished with emergency-
stop circuitry that uses external relays. This circuitry should be set up to interlock
correctly with controller status (stop/run).

● Ground for low impedance:
  - For safety reasons, connect the [FG] grounding terminal to a Japanese Industrial
Standards (JIS) Class D (earlier called Class 3) Ground*1. For compliance to CE
Marking, use braided or other wires that can ensure low impedance even at high
frequencies for grounding.

*1 Japanese Industrial Standard (JIS) Class D Ground means grounding resistance of 100 Ω max.

● Configure and route cables with noise control considerations:
  - Perform installation and wiring that segregates system parts that may likely become
noise sources and system parts that are susceptible to noise. Segregation can be
achieved by measures such as segregating by distance, installing a filter or
segregating the grounding system.

● Configure for CE Marking Conformance:
  - For compliance to CE Marking, perform installation and cable routing according to
the description on compliance to CE Marking in the “Hardware Manual”
(IM34M6C11-01E).
Keep spare parts on hand:
- Stock up on maintenance parts including spare modules, in advance.

Discharge static electricity before operating the system:
- Because static charge can accumulate in dry conditions, first touch grounded metal to discharge any static electricity before touching the system.

Never use solvents such as paint thinner for cleaning:
- Gently clean the surfaces of the FA-M3 controller with a cloth that has been soaked in water or a neutral detergent and wringed.
- Do not use volatile solvents such as benzine or paint thinner or chemicals for cleaning, as they may cause deformity, discoloration, or malfunctioning.

Avoid storing the FA-M3 controller in places with high temperature or humidity:
- Since the CPU module has a built-in battery, avoid storage in places with high temperature or humidity.
- Since the service life of the battery is drastically reduced by exposure to high temperatures, take special care (storage temperature should be from -20°C to 75°C).
- There is a built-in lithium battery in a CPU module and temperature control module which serves as backup power supply for programs, device information and configuration information. The service life of this battery is more than 10 years in standby mode at room temperature. Take note that the service life of the battery may be shortened when installed or stored at locations of extreme low or high temperatures. Therefore, we recommend that modules with built-in batteries be stored at room temperature.

Always turn off the power before installing or removing modules:
- Failing to turn off the power supply when installing or removing modules, may result in damage.

Do not touch components in the module:
- In some modules you can remove the right-side cover and install ROM packs or change switch settings. While doing this, do not touch any components on the printed-circuit board, otherwise components may be damaged and modules may fail to work.

Do not use unused terminals:
- Do not connect wires to unused terminals on a terminal block or in a connector. Doing so may adversely affect the functions of the module.
Waste Electrical and Electronic Equipment

Waste Electrical and Electronic Equipment (WEEE), Directive 2002/96/EC
(This directive is only valid in the EU.)

This product complies with the WEEE Directive (2002/96/EC) marking requirement. The following marking indicates that you must not discard this electrical/electronic product in domestic household waste.

Product Category
With reference to the equipment types in the WEEE directive Annex 1, this product is classified as a “Monitoring and Control instrumentation” product.
Do not dispose in domestic household waste.
When disposing products in the EU, contact your local Yokogawa Europe B. V. office.
Introduction

■ Overview of the Manual

This manual describes the specifications, operations, and communications protocol of the F3LE12-0T Ethernet interface module, which is to be inserted in an I/O slot of the FA-M3 Range-free multi-controller.

■ Related Instruction Manuals

● For communications commands and responses:
  - Personal Computer Link Commands (IM34M6P41-01E)
  In addition, refer to the respective user manuals of equipment connected to the same network as required.

● For the functions of Sequence CPU modules:
  - Sequence CPU Instruction Manual – Functions (for F3SP28-3N/3S, F3SP38-6N/6S, F3SP53-4H/4S, F3SP58-6H/6S, F3SP59-7S)(IM34M6P13-01E)
  - Sequence CPU Instruction Manual – Functions (for F3SP21, F3SP25, and F3SP35) (IM34M6P12-02E)

● For the instructions of Sequence CPU modules:
  - Sequence CPU Instruction Manual – Instructions (IM34M6P12-03E)

● For information on creating ladder programs:
  - FA-M3 Programming Tool WideField2 (IM34M6Q15-01E)
  - FA-M3 Programming Tool WideField (IM34M6Q14-01E)

● For the FA-M3 specifications and configurations*, installation and wiring, test run, maintenance, and module installation restrictions:
  *1: Refer to the relevant product manuals for specifications except for power supply modules, base modules, input/output modules, cables and terminal units.
  - Hardware Manual (IM34M6C11-01E)
Copyrights and Trademarks

■ Copyrights

Copyrights of the programs and online manual included in this CD-ROM belong to Yokogawa Electric Corporation.

This online manual may be printed but PDF security settings have been made to prevent alteration of its contents.

This online manual may only be printed and used for the sole purpose of operating this product. When using a printed copy of the online manual, pay attention to possible inconsistencies with the latest version of the online manual. Ensure that the edition agrees with the latest CD-ROM version.

Copying, passing, selling or distribution (including transferring over computer networks) of the contents of the online manual, in part or in whole, to any third party, is strictly prohibited. Registering or recording onto videotapes and other media is also prohibited without expressed permission of Yokogawa Electric Corporation.

■ Trademarks

The trade and company names that are referred to in this document are either trademarks or registered trademarks of their respective companies.
CONTENTS

Applicable Product.................................................................i
Important ....................................................................................ii
Introduction ..................................................................................viii
Copyrights and Trademarks .........................................................ix

1. Overview ................................................................................. 1-1
   1.1 Features ................................................................................ 1-1
   1.2 Software Configuration ....................................................... 1-2

2. System Configuration .......................................................... 2-1

3. Specifications ......................................................................... 3-1
   3.1 Components and Functions .................................................. 3-1
       ■ Model and Suffix Codes ......................................................... 3-1
       ■ Compatible CPU Modules ................................................. 3-1
       ■ General Specifications ..................................................... 3-1
   3.2 External Dimensions .......................................................... 3-3
   3.3 I/O Relays and Registers ...................................................... 3-4
       ■ I/O Relays ........................................................................... 3-4
       ■ Registers ............................................................................ 3-5

4. Setting Up the Module ......................................................... 4-1
   4.1 Attaching/Detaching the Module ......................................... 4-1
   4.2 Setup before Operation ....................................................... 4-3
        4.2.1 Setup Procedure .......................................................... 4-3
        4.2.2 Setting the Switches ..................................................... 4-4
        4.2.3 External Wiring .......................................................... 4-6
        4.2.4 Loopback Test ............................................................ 4-9

5. Higher-level Link Service ....................................................... 5-1
   5.1 Overview ............................................................................. 5-1
   5.2 Specifications ...................................................................... 5-2
       5.2.1 Functional Specifications .............................................. 5-2
       5.2.2 Communications Protocol .......................................... 5-3
   5.3 Network Data Frame Format ................................................. 5-5
       5.3.1 Communications Data Format ..................................... 5-5
       5.3.2 Higher-level Link Service Data Frame ......................... 5-5
5.4 **Response Exit Code** ................................................................. 5-11
   5.4.1 Exit Code ................................................................................ 5-11
   5.4.2 Detailed Error Code ................................................................ 5-11

5.5 **Specifying Devices** ............................................................... 5-12
   5.5.1 Specifying a Device in ASCII Format .................................... 5-12
   5.5.2 Specifying a Device in Binary Format .................................... 5-13
   5.5.3 List of Supported Devices ..................................................... 5-14

6. **Remote Programming Service** ............................................... 6-1
   6.1 Overview .................................................................................... 6-1
   6.2 Functional Specifications ........................................................... 6-1
   6.3 Setup ........................................................................................ 6-1
   6.4 Reference .................................................................................. 6-1

7. **Device Monitor Service** ............................................................ 7-1
   7.1 Overview .................................................................................... 7-1
   7.2 Specifications ............................................................................ 7-2
   7.2.1 Functional Specifications ........................................................ 7-2
   7.2.2 Communications Protocol ....................................................... 7-2
   7.3 **Network Data Frame Format** ................................................ 7-3
   7.3.1 Communications Data Format ................................................. 7-3
   7.3.2 Device Monitor Service Data Frame ....................................... 7-3
   7.4 **Response Exit Code** ............................................................... 7-6
   7.4.1 Exit Code ............................................................................... 7-6
   7.4.2 Detailed Error Code ............................................................... 7-6
   7.5 Specifying Devices ...................................................................... 7-7

8. **Messaging Service** ................................................................ 8-1
   8.1 Overview .................................................................................... 8-1
   8.2 Specifications ............................................................................ 8-1
   8.2.1 Functional Specifications ........................................................ 8-1
   8.2.2 Send/Receive Message Format ............................................... 8-1
   8.3 Socket Handles .......................................................................... 8-2
   8.4 Messaging .................................................................................. 8-3
   8.5 Messaging Procedure (UDP/IP) ................................................... 8-6

9. **Network Setup** ....................................................................... 9-1
   9.1 Network Environment Setup ...................................................... 9-1

10. **Troubleshooting** .................................................................. 10-1
    10.1 Hardware or Human Error? ....................................................... 10-1
    10.2 When RDY LED Is Not Lit ....................................................... 10-2
    10.3 When ERR LED Is Lit ............................................................. 10-3

Index ............................................................................................. Index-1

Revision Information ..................................................................... i
1. Overview

The F3LE12-0T Ethernet interface module can be connected to IEEE802.3-compliant networks (100BASE-TX or 10BASE-T).

Including the module into a network enables data exchange between nodes, such as personal computers and workstations, on a network, using the TCP/IP or UDP/IP protocol.

1.1 Features

This section describes the features of the module.

- **Accessing the Sequence CPU from Remote Nodes (higher-level link service)**
  - Enables reading from and writing to all FA-M3 CPU module devices.
    Reading from and writing to the devices of the CPU module in bits or words.
  - Provides remote run and stop control.
    A user program on the sequence CPU can be started or stopped from a remote node.
  - Supports loading and saving of programs.
    Enables a user program to be replaced from a remote node by reading and writing from the remote node.
  - Enables retrieval of information on the sequence CPU.
    A user can read various information about the sequence program (program name, size, block name etc.) and error log information.
  - Dispenses with the need to create communications programs.
    Data communications is always initiated by a remote node. Responses to requests are automatically returned by this module. There is no need to create or modify any special-purpose sequence program for data communications.
  - Selectable data formats
    A user can select either ASCII or binary code for data communications.

- **Remote Control (remote programming service)**
  - Supports communications via an Ethernet (TCP/IP) using the FA-M3 programming tool WideField2.
    A user can upload or download programs, monitor devices or perform online editing from a PC on the network using the FA-M3 programming tool WideField2.
  - Dispenses with the need to create a user program for remote programming.
    Using the FA-M3 programming tool WideField2 removes any need to create or modify special-purpose user programs for data communications.

- **Monitoring the Devices of Sequence CPU (device monitor service)**
  - Allows reading from all devices of a sequence CPU module.
    Reads a maximum of 512 words of data at a time from the devices of a CPU module in word units.
    There is no need to create any special-purpose user program for data communications as all communication requests are made by a remote node and the module responds by providing system services.
Messaging (message communications service)

- Messaging using a user program
  Allows communications with other nodes through UDP/IP using sockets. A maximum of 1024 bytes of data can be transmitted per socket.

1.2 Software Configuration

The Ethernet Interface Module conforms to the ISO reference model and provides standard or proprietary protocols for each layer.

**Figure 1.1 Software Configuration**

<table>
<thead>
<tr>
<th>Application layer</th>
<th>Interface to sequence CPU</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transport layer</td>
<td>TCP, UDP</td>
</tr>
<tr>
<td></td>
<td>IP, ICMP, ARP</td>
</tr>
<tr>
<td>Network layer</td>
<td>Ethernet</td>
</tr>
<tr>
<td>Data link layer</td>
<td>10BASE-T/100BASE-TX</td>
</tr>
<tr>
<td>Physical layer</td>
<td></td>
</tr>
</tbody>
</table>

| IP     | Internet Protocol |
| TCP    | Transmission Control Protocol |
| UDP    | User Datagram Protocol |
| ICMP   | Internet Control Message Protocol |
| ARP    | Address Resolution Protocol |

The Ethernet interface module incorporates some parts of the software provided by the Regents of University of California.
2. System Configuration

This section describes the instruments required to configure an Ethernet network system.

■ Connection Using 10BASE-T

10BASE-T is an Ethernet connection method using twisted-pair cables with transmission rate of 10Mbps.

In a 10BASE-T network, PCs are connected to a hub using a star topology.

![Diagram of 10BASE-T network](image)

**CAUTION**

Use hubs and twisted-pair cables, which conform to the Ethernet (10BASE-T) specifications.
Up to 4 segments are allowed for cascade connections to the hub.
The maximum length allowed for the twisted-pair cables is 100 m.
Connection Using 100BASE-TX

100BASE-TX is another Ethernet connection method using twisted-pair cables with transmission rate of 100Mbps.

In a 100BASE-TX network, PCs are connected to a hub using a star topology.

CAUTION

Use hubs and twisted-pair cables, which conform to the Ethernet (100BASE-TX) specifications.
Up to 2 segments are allowed for cascade connections to the hub.
The maximum length allowed for the twisted-pair cables is 100 m.
3. Specifications

■ Model and Suffix Codes

<table>
<thead>
<tr>
<th>Model</th>
<th>Suffix Code</th>
<th>Style Code</th>
<th>Option Code</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>F3LE12</td>
<td>-0T</td>
<td>...</td>
<td>...</td>
<td>10 Mbps/100 Mbps</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>10BASE-T/100BASE-TX</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Messaging</td>
</tr>
</tbody>
</table>

■ Compatible CPU Modules

The following table shows the CPU modules that are compatible with the F3LE12-0T module.

<table>
<thead>
<tr>
<th></th>
<th>F3SP05</th>
<th>F3SP08</th>
<th>F3SP21</th>
<th>F3SP25</th>
<th>F3SP28</th>
<th>F3SP35</th>
<th>F3SP38</th>
<th>F3SP53</th>
<th>F3SP58</th>
<th>F3SP59</th>
</tr>
</thead>
<tbody>
<tr>
<td>F3LE12-0T</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
</tr>
</tbody>
</table>

■ General Specifications

<table>
<thead>
<tr>
<th>Item</th>
<th>Specifications</th>
<th>100BASE-TX</th>
<th>10BASE-T</th>
</tr>
</thead>
<tbody>
<tr>
<td>Access control</td>
<td>CSMA/CD</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Transmission speed</td>
<td>100 Mbps</td>
<td>10 Mbps</td>
<td></td>
</tr>
<tr>
<td>Transmission method</td>
<td>Baseband</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Maximum distance between nodes</td>
<td>100 m&lt;sup&gt;1&lt;/sup&gt;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Largest connection configuration</td>
<td>2 cascade segments</td>
<td>4 cascade segments</td>
<td></td>
</tr>
<tr>
<td>Protocol</td>
<td>TCP, UDP, IP, ICMP, and ARP</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Maximum number of installed modules</td>
<td>Total number of modules with similar functions&lt;sup&gt;3&lt;/sup&gt;</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>For F3SP05, F3SP08, or F3SP21 CPU module; 2 max.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>For F3SP25, F3SP28, F3SP35, F3SP38, F3SP53, F3SP58, or F3SP59 CPU module; 6 max.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Internal current consumption</td>
<td>500 mA</td>
<td></td>
<td></td>
</tr>
<tr>
<td>External power supply</td>
<td>—</td>
<td></td>
<td></td>
</tr>
<tr>
<td>External dimensions</td>
<td>28.98 (W) x 100 (H) x 83.2 (D) mm&lt;sup&gt;2&lt;/sup&gt;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Weight</td>
<td>130 g</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<sup>1</sup>: Distance between a hub and the module
<sup>2</sup>: Excluding protrusions
<sup>3</sup>: The module may not be installed in a subunit.

3.1 Components and Functions

■ External Views and Components

![External View and Components](image-url)
Functions of Components

● LED indicators

The LED display indicates the operating status of the module.

<table>
<thead>
<tr>
<th>Name (color)</th>
<th>Description</th>
<th>Lit</th>
<th>Flashing</th>
<th>Not Lit</th>
</tr>
</thead>
<tbody>
<tr>
<td>RDY (green)</td>
<td>Internal circuitry status</td>
<td>Normal</td>
<td></td>
<td>Error</td>
</tr>
<tr>
<td>LNK (green)</td>
<td>Network status</td>
<td>Connected</td>
<td></td>
<td>Not connected</td>
</tr>
<tr>
<td>COM (green)</td>
<td>Data transmission status</td>
<td>Carrier detected</td>
<td></td>
<td>No carrier detected</td>
</tr>
<tr>
<td>ERR (red)</td>
<td>Error</td>
<td>Error</td>
<td></td>
<td>Normal</td>
</tr>
</tbody>
</table>

10BASE-T/100BASE-TX connector

This connector is used for connecting to a 10BASE-T/100BASE-TX (UTP/STP) cable. Through automatic negotiation, the module automatically switches between 10BASE-T and 100BASE-TX according to the connected hub.

● SHIELD terminal

The SHIELD terminal is connected to the SHIELD wire of the 10BASE-T/100BASE-TX cable.

It is isolated from the FG terminal of the power supply module.

● FG terminal

The FG terminal is connected to the FG terminal of the power supply module.

● IP address switches

This is a set of 16-position rotary switches for setting the IP address.

Example: Setting the IP address to 192.168.250.210

```
     C     D
     0     2
     A
```

<table>
<thead>
<tr>
<th>Hexadecimal</th>
<th>Decimal</th>
</tr>
</thead>
<tbody>
<tr>
<td>C0</td>
<td>192</td>
</tr>
<tr>
<td>A8</td>
<td>168</td>
</tr>
<tr>
<td>FA</td>
<td>250</td>
</tr>
<tr>
<td>D2</td>
<td>210</td>
</tr>
</tbody>
</table>

CAUTION

Do not touch the components inside the module when setting the IP address switches. Doing so may damage the components and lead to module failure.
● Operation condition switch

This is a DIP switch for setting various operation conditions.

<table>
<thead>
<tr>
<th>Element</th>
<th>Function</th>
<th>OFF</th>
<th>ON</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Data format</td>
<td>ASCII</td>
<td>Binary</td>
</tr>
<tr>
<td>2</td>
<td>Write protection</td>
<td>Disabled</td>
<td>Enabled</td>
</tr>
<tr>
<td>3</td>
<td>Reserved</td>
<td>Always OFF</td>
<td>Not allowed</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>Operation mode</td>
<td>Normal operation</td>
<td>Test operation</td>
</tr>
</tbody>
</table>

⚠️ CAUTION

Do not touch the components inside the module when setting the operation condition switch. Doing so may damage the components and lead to module failure.

● MAC Address

ETHERNET
MODEL : F3LE12
SUFFIX : -0T
STYLE : S1
REV : 00:00
SUPPLY : -
INPUT : -
OUTPUT : -
MAC ID : 000064
DATE : 
NO. :

Ethernet address
This is a 12-digit hexadecimal number to be assigned uniquely to each module.

3.2 External Dimensions

Unit: mm
## 3.3 I/O Relays and Registers

### I/O Relays

<table>
<thead>
<tr>
<th>Input Relay</th>
<th>Description</th>
<th>Output Relay</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>X0口01</td>
<td>Socket creation completed</td>
<td>Y0口33</td>
<td>Socket creation request</td>
</tr>
<tr>
<td>X0口02</td>
<td>Send completed</td>
<td>Y0口34</td>
<td>Send request</td>
</tr>
<tr>
<td>X0口03</td>
<td>Clear to receive</td>
<td>Y0口35</td>
<td>Receive completed</td>
</tr>
<tr>
<td>X0口04</td>
<td>Socket creation completed</td>
<td>Y0口36</td>
<td>Socket creation request</td>
</tr>
<tr>
<td>X0口05</td>
<td>Send completed</td>
<td>Y0口37</td>
<td>Send request</td>
</tr>
<tr>
<td>X0口06</td>
<td>Clear to receive</td>
<td>Y0口38</td>
<td>Receive completed</td>
</tr>
<tr>
<td>X0口07</td>
<td>Socket creation completed</td>
<td>Y0口39</td>
<td>Socket creation request</td>
</tr>
<tr>
<td>X0口08</td>
<td>Send completed</td>
<td>Y0口40</td>
<td>Send request</td>
</tr>
<tr>
<td>X0口09</td>
<td>Clear to receive</td>
<td>Y0口41</td>
<td>Receive completed</td>
</tr>
<tr>
<td>X0口10</td>
<td>Socket creation completed</td>
<td>Y0口42</td>
<td>Socket creation request</td>
</tr>
<tr>
<td>X0口11</td>
<td>Send completed</td>
<td>Y0口43</td>
<td>Send request</td>
</tr>
<tr>
<td>X0口12</td>
<td>Clear to receive</td>
<td>Y0口44</td>
<td>Receive completed</td>
</tr>
<tr>
<td>X0口13</td>
<td>Socket creation completed</td>
<td>Y0口45</td>
<td>Socket creation request</td>
</tr>
<tr>
<td>X0口14</td>
<td>Send completed</td>
<td>Y0口46</td>
<td>Send request</td>
</tr>
<tr>
<td>X0口15</td>
<td>Clear to receive</td>
<td>Y0口47</td>
<td>Receive completed</td>
</tr>
<tr>
<td>X0口16</td>
<td>Socket creation completed</td>
<td>Y0口48</td>
<td>Socket creation request</td>
</tr>
<tr>
<td>X0口17</td>
<td>Send completed</td>
<td>Y0口49</td>
<td>Send request</td>
</tr>
<tr>
<td>X0口18</td>
<td>Clear to receive</td>
<td>Y0口50</td>
<td>Receive completed</td>
</tr>
<tr>
<td>X0口19</td>
<td>Socket creation completed</td>
<td>Y0口51</td>
<td>Socket creation request</td>
</tr>
<tr>
<td>X0口20</td>
<td>Send completed</td>
<td>Y0口52</td>
<td>Send request</td>
</tr>
<tr>
<td>X0口21</td>
<td>Clear to receive</td>
<td>Y0口53</td>
<td>Receive completed</td>
</tr>
<tr>
<td>X0口22</td>
<td>Socket creation completed</td>
<td>Y0口54</td>
<td>Socket creation request</td>
</tr>
<tr>
<td>X0口23</td>
<td>Send completed</td>
<td>Y0口55</td>
<td>Send request</td>
</tr>
<tr>
<td>X0口24</td>
<td>Clear to receive</td>
<td>Y0口56</td>
<td>Receive completed</td>
</tr>
<tr>
<td>X0口25</td>
<td>Socket creation completed</td>
<td>Y0口57</td>
<td>Socket creation request</td>
</tr>
<tr>
<td>X0口26</td>
<td>Send completed</td>
<td>Y0口58</td>
<td>Send request</td>
</tr>
<tr>
<td>X0口27</td>
<td>Clear to receive</td>
<td>Y0口59</td>
<td>Receive completed</td>
</tr>
<tr>
<td>X0口28</td>
<td>Socket creation completed</td>
<td>Y0口60</td>
<td>Socket creation request</td>
</tr>
<tr>
<td>X0口29</td>
<td>Network environment setup completed</td>
<td>Y0口61</td>
<td>Network environment setup request</td>
</tr>
<tr>
<td>X0口30</td>
<td>Reserved</td>
<td>Y0口62</td>
<td>Reserved</td>
</tr>
<tr>
<td>X0口31</td>
<td>Reserved</td>
<td>Y0口63</td>
<td>Reserved</td>
</tr>
<tr>
<td>X0口32</td>
<td>Reserved</td>
<td>Y0口64</td>
<td>Reserved</td>
</tr>
</tbody>
</table>

CAUTION

Do not attempt to write to reserved areas. Doing so may lead to an error.
## Registers

<table>
<thead>
<tr>
<th>Data Position No.</th>
<th>Description</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>MAC address</td>
<td>Module MAC address</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>Data position No. [0][3][2][1]</td>
</tr>
<tr>
<td>3</td>
<td>IP address</td>
<td>Module IP address</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>Data position No. [5][4]</td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6 to 16</td>
<td>Reserved</td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>Operation condition switch 1</td>
<td>0→OFF</td>
</tr>
<tr>
<td>18</td>
<td>Operation condition switch 2</td>
<td>1→ON</td>
</tr>
<tr>
<td>19</td>
<td>Operation condition switch 3</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>Operation condition switch 4</td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>Operation condition switch 5</td>
<td></td>
</tr>
<tr>
<td>22</td>
<td>Operation condition switch 6</td>
<td></td>
</tr>
<tr>
<td>23</td>
<td>Operation condition switch 7</td>
<td></td>
</tr>
<tr>
<td>24</td>
<td>Operation condition switch 8</td>
<td></td>
</tr>
<tr>
<td>25 to 960</td>
<td>Reserved</td>
<td></td>
</tr>
<tr>
<td>961</td>
<td>Bank register</td>
<td>0 to 31</td>
</tr>
<tr>
<td>962 to 1024</td>
<td>Reserved</td>
<td></td>
</tr>
</tbody>
</table>

### Bank register

Data positions 1025-2048 are accessed through bank switching by writing an appropriate value to the bank register. The bank register is write-only and cannot be read.

### Bank 0

#### Network Environment

<table>
<thead>
<tr>
<th>Data Position No.</th>
<th>Description</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1025</td>
<td>Subnet mask</td>
<td>Subnet mask [<a href="http://www.xxx">www.xxx</a>][yyy.zzz]</td>
</tr>
<tr>
<td>1026</td>
<td></td>
<td>Data position No. 1026 1025</td>
</tr>
<tr>
<td>1027</td>
<td>Default gateway IP address</td>
<td>Default gateway IP address [<a href="http://www.xxx">www.xxx</a>][yyy.zzz]</td>
</tr>
<tr>
<td>1028</td>
<td></td>
<td>Data position No. 1028 1027</td>
</tr>
<tr>
<td>1029 to 1160</td>
<td>Reserved</td>
<td></td>
</tr>
<tr>
<td>1161</td>
<td>Network environment setup completed status</td>
<td></td>
</tr>
<tr>
<td>1162 to 1536</td>
<td>Reserved</td>
<td></td>
</tr>
</tbody>
</table>

#### Socket Parameter Setup Area (1/2)

<table>
<thead>
<tr>
<th>Data Position No.</th>
<th>Description</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1537</td>
<td>Socket type</td>
<td></td>
</tr>
<tr>
<td>1538</td>
<td>Port number</td>
<td></td>
</tr>
<tr>
<td>1539 to 1543</td>
<td>Reserved</td>
<td></td>
</tr>
<tr>
<td>1544</td>
<td>Socket creation completed status</td>
<td></td>
</tr>
<tr>
<td>1545 to 1568</td>
<td>Reserved</td>
<td></td>
</tr>
<tr>
<td>Data Position No.</td>
<td>Description</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------</td>
<td>-------------</td>
<td>-------------</td>
</tr>
<tr>
<td>1569</td>
<td>Socket type</td>
<td></td>
</tr>
<tr>
<td>1570</td>
<td>Port number</td>
<td></td>
</tr>
<tr>
<td>1571 to 1575</td>
<td>Socket handle 2</td>
<td>Reserved</td>
</tr>
<tr>
<td>1576</td>
<td>Socket creation completed status</td>
<td></td>
</tr>
<tr>
<td>1577 to 1600</td>
<td>Reserved</td>
<td></td>
</tr>
<tr>
<td>1601</td>
<td>Socket type</td>
<td></td>
</tr>
<tr>
<td>1602</td>
<td>Port number</td>
<td></td>
</tr>
<tr>
<td>1603 to 1607</td>
<td>Socket handle 3</td>
<td>Reserved</td>
</tr>
<tr>
<td>1608</td>
<td>Socket creation completed status</td>
<td></td>
</tr>
<tr>
<td>1609 to 1632</td>
<td>Reserved</td>
<td></td>
</tr>
<tr>
<td>1633</td>
<td>Socket type</td>
<td></td>
</tr>
<tr>
<td>1634</td>
<td>Port number</td>
<td></td>
</tr>
<tr>
<td>1635 to 1639</td>
<td>Socket handle 4</td>
<td>Reserved</td>
</tr>
<tr>
<td>1640</td>
<td>Socket creation completed status</td>
<td></td>
</tr>
<tr>
<td>1641 to 1664</td>
<td>Reserved</td>
<td></td>
</tr>
<tr>
<td>1665</td>
<td>Socket type</td>
<td></td>
</tr>
<tr>
<td>1666</td>
<td>Port number</td>
<td></td>
</tr>
<tr>
<td>1667 to 1671</td>
<td>Socket handle 5</td>
<td>Reserved</td>
</tr>
<tr>
<td>1672</td>
<td>Socket creation completed status</td>
<td></td>
</tr>
<tr>
<td>1673 to 1696</td>
<td>Reserved</td>
<td></td>
</tr>
<tr>
<td>1697</td>
<td>Socket type</td>
<td></td>
</tr>
<tr>
<td>1698</td>
<td>Port number</td>
<td></td>
</tr>
<tr>
<td>1699 to 1703</td>
<td>Socket handle 6</td>
<td>Reserved</td>
</tr>
<tr>
<td>1704</td>
<td>Socket creation completed status</td>
<td></td>
</tr>
<tr>
<td>1705 to 1728</td>
<td>Reserved</td>
<td></td>
</tr>
<tr>
<td>1729</td>
<td>Socket type</td>
<td></td>
</tr>
<tr>
<td>1730</td>
<td>Port number</td>
<td></td>
</tr>
<tr>
<td>1731 to 1735</td>
<td>Socket handle 7</td>
<td>Reserved</td>
</tr>
<tr>
<td>1736</td>
<td>Socket creation completed status</td>
<td></td>
</tr>
<tr>
<td>1737 to 1760</td>
<td>Reserved</td>
<td></td>
</tr>
<tr>
<td>1761</td>
<td>Socket type</td>
<td></td>
</tr>
<tr>
<td>1762</td>
<td>Port number</td>
<td></td>
</tr>
<tr>
<td>1763 to 1767</td>
<td>Socket handle 8</td>
<td>Reserved</td>
</tr>
<tr>
<td>1768</td>
<td>Socket creation completed status</td>
<td></td>
</tr>
<tr>
<td>1769 to 1792</td>
<td>Reserved</td>
<td></td>
</tr>
<tr>
<td>1793 to 2048</td>
<td>Reserved</td>
<td></td>
</tr>
</tbody>
</table>
● Banks 1-8

Handle m Send Control Block (where m = bank number)

<table>
<thead>
<tr>
<th>Data Position No.</th>
<th>Description</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1025 to 1536</td>
<td>Send data</td>
<td></td>
</tr>
<tr>
<td>1537 to 1792</td>
<td>Reserved</td>
<td></td>
</tr>
<tr>
<td>1793</td>
<td>Destination IP address</td>
<td>Destination IP address <a href="http://www.xxx.yyy.zzz">www.xxx.yyy.zzz</a></td>
</tr>
<tr>
<td>1794</td>
<td></td>
<td>Data position No. 1794 1793</td>
</tr>
<tr>
<td>1795</td>
<td>Destination port number</td>
<td></td>
</tr>
<tr>
<td>1796</td>
<td>Send message size</td>
<td></td>
</tr>
<tr>
<td>1797 to 1799</td>
<td>Reserved</td>
<td></td>
</tr>
<tr>
<td>1800</td>
<td>Send completed status</td>
<td></td>
</tr>
<tr>
<td>1801 to 2048</td>
<td>Reserved</td>
<td></td>
</tr>
</tbody>
</table>

Do not attempt to write data into reserved areas. Otherwise, the module may not work correctly.

● Banks 16-23

Handle n Receive Control Block (where n = bank number - 15)

<table>
<thead>
<tr>
<th>Data Position No.</th>
<th>Description</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1025 to 1536</td>
<td>Received data</td>
<td></td>
</tr>
<tr>
<td>1537 to 1792</td>
<td>Reserved</td>
<td></td>
</tr>
<tr>
<td>1793</td>
<td>Source IP address</td>
<td>Source IP address <a href="http://www.xxx.yyy.zzz">www.xxx.yyy.zzz</a></td>
</tr>
<tr>
<td>1794</td>
<td></td>
<td>Data position No. 1794 1793</td>
</tr>
<tr>
<td>1795</td>
<td>Source port number</td>
<td></td>
</tr>
<tr>
<td>1796</td>
<td>Receive message size</td>
<td></td>
</tr>
<tr>
<td>1797 to 2048</td>
<td>Reserved</td>
<td></td>
</tr>
</tbody>
</table>

⚠️ CAUTION

Do not attempt to write data into reserved areas. Otherwise, the module may not work correctly.
4. Setting Up the Module

4.1 Attaching/Detaching the Module

■ Attaching the Module

Figure 4.1 shows how to attach this module to the base module. First hook the anchor slot at the bottom of the module to be attached onto the anchor pin on the bottom of the base module. Push the top of the module toward the base module until the anchor/release button clicks into place.

⚠️ CAUTION

Always switch off the power before attaching or detaching the module.

Do not bend the connector on the rear of the module by force during the above operation. If the module is pushed with improper force, the connector may bend, causing an error.

■ Detaching the Module

To remove this module from the base module, reverse the above operation. Press the anchor/release button on the top of this module to unlock it and tilt the module away from the base module.
Attaching the Module in Intense Vibration Environments

If the module is used in intense vibration environments, fasten the module with a screw. Use screws of type listed in the table below.

Insert these screws into the screw holes on top of the module and tighten them with a Phillips screwdriver.

<table>
<thead>
<tr>
<th>Required Screw</th>
</tr>
</thead>
<tbody>
<tr>
<td>Binding head machine screw M4 of 12 to 15 mm long</td>
</tr>
<tr>
<td>(washer screw of 14-15 mm long)</td>
</tr>
</tbody>
</table>

Figure 4.2  Securing Module Using Screws

CAUTION

If you fasten the module with a screw, be careful not to tighten the screw excessively.
4.2 Setup before Operation

4.2.1 Setup Procedure

The following figure shows a procedure flowchart for setting up an FA-M3 system using the Ethernet Interface Module.

```
    Startup
       ↓
Establish the overall system configuration.
       ↓
Set up Ethernet Interface Module.  
   Set up switches.  
   Set to test mode.  
       ↓
Cabling and connecting
       ↓
Power on
       ↓
Loopback test
       ↓
Power off
       ↓
Set up Ethernet Interface Module.  
   Set up switches.  
   Set to test mode.  
       ↓
Power On
       ↓
Start operation
```

See Section 4.2.2 "Setting the Switches"

See Section 4.2.4 "Loopback Test"

See Section 4.2.2 "Setting the Switches"

Figure 4.3 Setup Flowchart
4.2.2 Setting the Switches

This section describes the switches to be set before attaching this module to the base module of FA-M3. Figure 4.4 shows the names and locations of the switches.

**IP Address Switches**

Use the hexadecimal rotary switches located on the side of the module (inside the cover) to set the IP address of the module.

The IP address is set in hexadecimal values as shown in Figure 4.5.

The factory setting is 0.0.0.0.

![Diagram of IP address switches](image)

**Figure 4.4 Locations of the Switches**

**Figure 4.5 Setting Up an IP Address**

<table>
<thead>
<tr>
<th>Hexadecimal</th>
<th>Decimal</th>
</tr>
</thead>
<tbody>
<tr>
<td>C0 A8 FA D2</td>
<td>192 168 250 210</td>
</tr>
</tbody>
</table>
# Operation condition Switch

The operation conditions of the module can be set using the DIP switches located on the side of the module (inside the cover).

| Table 4.1   Operation Condition Switch Setup |
|-------------------------|-------------------------|-------------------------|-------------------------|
| **Element** | **Function** | **OFF** | **ON** |
| 1 | Data format | ASCII | Binary |
| 2 | Write protection | Not protected | Protected |
| 3 |  |  |  |
| 4 |  |  |  |
| 5 |  |  |  |
| 6 |  |  |  |
| 7 |  |  |  |
| 8 | Operation mode | Normal | Test |

- **Data format**

  Sets the data format for higher-level link service (port number: 12289/$3001), which can be requested by a remote node.

  - OFF : ASCII format
  - ON : Binary format

  Sets the port number for device monitor service, which may be requested by a remote node.

  - OFF : 12308 ($3014)
  - ON : 12292 ($3004)

- **Write protection**

  Enables or disables the writing of data into the sequence CPU module or the downloading of programs for higher-level link service, which can be requested by a remote node.

  - OFF : Disabled (Not protected)
  - ON : Enabled (Protected)

  This setting is not applicable if the higher-level link service is not used.

- **Operation mode**

  Sets the operation mode of the module when it is switched on.

  - OFF : Normal mode
  - ON : Test mode

  In test mode, the module performs a loopback test. The RDY LED turns on when the loopback test terminates normally. The ERR LED turns on if the test fails.

  In test mode, all functions of the module are disabled after a loopback test. To use the module normally, set this switch element to OFF (normal mode) before switching on the module.

---

⚠️ **CAUTION**

Do not perform a loopback test with the sequence CPU mounted.
4.2.3 External Wiring

- Connecting Communications Cable

- 10BASE-T/100BASE-TX Cable Connection

> Twist-pair cable connection

- Connect the twist-pair cable to connector for 10BASE-T/100BASE-TX

Figure 4.6 10BASE-T/100BASE-TX Cable Connection Flowchart

CAUTION

Adequate safety measures should be adopted when laying the Ethernet cables. We recommend that you commission a professional contractor to lay the Ethernet cables.
Shielding

When using a shielded twisted-pair cable (STP), ground the shield of the cable as required.

The shield of the cable is connected to the shell of the 10BASE-T/100BASE-TX connector, and the shell is connected to the shield terminal of the module when the connector is inserted.

There are two ways to ground the shield terminal of the module.

To enhance the shielding effect, perform grounding according to the procedure (for CE marking compliance) described on the next page.

- **For connection to the aluminum chassis of the base module and the FG terminal of the power supply module through the module:**
  
  Connect the shield terminal and the FG terminal of the module together using a connector.

- **For direct connection (not through the module):**
  
  Connect the shield terminal of the module directly to the inner plate of a low-impedance panel or the GND terminal inside a panel enclosure.

  Use a wire of AWG28-16 for grounding. Tighten the screw on the grounding connector to a torque of 0.25 N-m to secure the grounding wire.

![Figure 4.7  Grounding the Shield of the Cable](image)

- **Fixing the grounding connector to the module**
  
  Insert the grounding connector and tighten the fixing screws to a torque of 0.25 N-m as shown below.

![Figure 4.8  Fixing the Grounding Connector to the Module](image)
The internal wiring of the module is shown below.

![Internal Wiring Diagram](image)

Note: Figure shows front view of connectors.

Figure 4.9 Internal Wiring of the Module

- **CE marking conformance**
  - Use a shielded cable for conforming equipment incorporating the Ethernet Interface Module to CE Marking. Remove part of the cable insulation to expose the shield, and ground and secure the shield with an FG clamp.

![CE Marking Diagram](image)
4.2.4 Loopback Test

The Ethernet Interface Module can perform self-diagnostics (loopback test) according to the setting of the operation condition switch. In the loopback test, the module sends test data to the network, and when the data is returned, it checks if the sent data and the returned data are the same.

● Test method

1. Turn on switch element 8 of the operation condition switch on the right face of the module.

2. Connect the module to the network.
3. Switch on the module.
4. Test starts (the RDY LED starts flashing).

● Test result

- If the received data is correct, the RDY LED turns on.
- If the received data is incorrect, the ERR LED turns on.

At the end of the loopback test, all functions of the module remain disabled. After the test, set the operation mode to normal mode by turning off switch element 8, and then switching off and switching on the module.

⚠️ CAUTION

Do not perform the loopback test with the sequence CPU mounted.
5. **Higher-level Link Service**

5.1 **Overview**

The higher-level link service of the Ethernet Interface Module (F3LE12-0T) allows a user to monitor and set up the FA-M3 sequence CPU module, start and stop its program, and load and save a program by sending commands to the F3LE12-0T module from a remote node. All these are achieved without requiring the user to write a communications program on the FA-M3 side.

The higher-level link service uses interactive communications control in which commands and responses are exchanged using TCP/IP or UDP/IP (through a socket interface).

A command is an instruction issued to the Ethernet Interface Module from a remote node. A response is a reply from the module to the remote node in response to a command.

The module does not transmit commands to remote nodes. Remote nodes have the right to initiate a transmission. When a remote node sends a command to the module, the module responds by returning a response to the command.

For details on the commands and responses used in the higher-level link service, see “Personal Computer Link Command Instruction Manual” (IM34M6P41-01E).
5.2 Specifications

5.2.1 Functional Specifications

Table 5.1 Higher-level Link Service Specifications

<table>
<thead>
<tr>
<th>Item</th>
<th>Specifications</th>
</tr>
</thead>
<tbody>
<tr>
<td>Communications protocol</td>
<td>TCP/IP</td>
</tr>
<tr>
<td></td>
<td>UDP/IP</td>
</tr>
<tr>
<td>Data format</td>
<td>ASCII/binary</td>
</tr>
<tr>
<td>Maximum number of connections</td>
<td>9*1</td>
</tr>
<tr>
<td>Port number</td>
<td>12289 ($3001)/12291 ($3003)</td>
</tr>
<tr>
<td>Write protection</td>
<td>Protected</td>
</tr>
</tbody>
</table>

*1: Up to 9 connections are available for higher-level link service and device monitor service combined.

■ Data Format and Port Number

The available data formats are ASCII and binary.

The data format for a port number of 12289 ($3001) can be specified using element 1 of the operation condition switch located on the side of the module as shown below.

Table 5.2 Port Number and Data Format

<table>
<thead>
<tr>
<th>Element 1</th>
<th>Port Number</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>12289 ($3001)</td>
</tr>
<tr>
<td>OFF</td>
<td>ASCII</td>
</tr>
<tr>
<td>ON</td>
<td>Binary</td>
</tr>
</tbody>
</table>

■ Write Protection

Write protection allows you to prohibit the execution of certain commands.

To enable or disable write protection, use element 2 of the operation condition switch located on the side of the module. The following table lists the commands that are disabled when write protection is enabled.

Table 5.3 Commands that are Disabled by Write Protection

- Device Word Write, Device Bit Write
- Device Word Write of Same Data, Device Bit Write of Same Data
- Device Word Random Write, Device Bit Random Write
- Special Module Word Write/Special Module Long-word Write
- Load program
- Write date and time
5.2.2 Communications Protocol

Communications Using TCP/IP Protocol

A remote node and the module communicate with each other through a socket interface as follows:

![Diagram of TCP/IP Communications between Remote Node and the Module]

The software of the module automatically starts running when the module is switched on.
Communications Using UDP/IP Protocol

A remote node and the module communicate with each other through a socket interface as follows:

![Diagram showing socket operation](image)

**Figure 5.2  UDP/IP Communications between Remote Node and the Module**

The software of the module automatically starts running when the module is switched on.
5.3 Network Data Frame Format

5.3.1 Communications Data Format
The format of the data exchanged between a remote node and the module is as follows:

<table>
<thead>
<tr>
<th>SFD</th>
<th>Ethernet Header</th>
<th>IP Header</th>
<th>TCP/UDP Header</th>
<th>Higher-level link service data</th>
<th>FCS</th>
</tr>
</thead>
</table>

Figure 5.3  Higher-level Link Service Data Format

5.3.2 Higher-level Link Service Data Frame
You can set the data format for data exchanged via the higher-level link service to either ASCII or binary. The selected data format applies to the entire higher-level link data frame.
The respective formats for the higher-level link data frame are shown below.

- **Data Frame in ASCII Format**

  **Command**

  ![](FA5311.VSD)

  **Response**

  ![](FA5321.VSD)

  Figure 5.4  ASCII Format Data Frame
Subheader

2 bytes

- Command/response flag
  Differentiates a command and a response using an ASCII character as follows:
  “0” ($30): Command
  “1” ($31): Response

- CPU number
  Specifies which sequence CPU module is addressed using an ASCII character as follows:
  “1” ($31): Sequence CPU module in slot 1
  “2” ($32): Sequence CPU module in slot 2
  “3” ($33): Sequence CPU module in slot 3
  “4” ($34): Sequence CPU module in slot 4

Command

Specifies the type of a request from a remote node.
For details on the commands, see IM34M6P41-01E, "Personal Computer Link Command Instruction Manual."

3 bytes

Command parameter

Contains a device name, a number, data, or nothing depending on the command. For details on the commands, see IM34M6P41-01E, "Personal Computer Link Command Instruction Manual."
● Exit code
Indicates the result of command execution. For details on the exit code, see Section 5.4, "Response Exit Code."

Normal

2 bytes

\[
\begin{array}{c}
\text{"O"} \\
\text{\$4F}
\end{array}
\quad \quad \quad \quad \quad
\begin{array}{c}
\text{"K"} \\
\text{\$4B}
\end{array}
\]

Error

4 bytes

\[
\begin{array}{c}
\text{"E"} \\
\text{\$45}
\end{array}
\quad \quad \quad \quad \quad
\begin{array}{c}
\text{"R"} \\
\text{\$52}
\end{array}
\quad \quad \quad \quad \quad
\begin{array}{c}
\text{"xx"} \\
\text{\$xx}
\end{array}
\quad \quad \quad \quad \quad
\begin{array}{c}
\text{"xx"} \\
\text{\$xx}
\end{array}
\]

● Response parameter

If normal (exit code = "OK")
Contains returned information or nothing depending on the command that has initiated this response. For details on responses, see IM34M6P41-01E, "Personal Computer Link Command Instruction Manual."

If in error (exit code = "ER\[""")

5 bytes

- Detailed error code
Contains a detailed error code only when the exit code is either “ER03”, “ER04”, “ER05”, “ER08”, or “ER52.” For further information on detailed error codes, see Section 5.4, "Response Exit Code."

- Command
Contains the command that has caused this response.
• End characters

Each data frame in ASCII format must end with the end characters \( \text{CR} \text{LF} (\$0D\$0A) \).
When preparing a command, you must end it with these end characters. The module automatically produces a response complete with these end characters.

■ Data Frame in Binary Format

**Command**

<table>
<thead>
<tr>
<th>Sub-header</th>
<th>CPU No.</th>
<th>Size</th>
<th>Command parameter</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 byte</td>
<td>1 byte</td>
<td>2 bytes</td>
<td>Size</td>
</tr>
</tbody>
</table>

**Response**

<table>
<thead>
<tr>
<th>Sub-header</th>
<th>Exit code</th>
<th>Size</th>
<th>Response parameter</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 byte</td>
<td>1 byte</td>
<td>2 bytes</td>
<td>Size</td>
</tr>
</tbody>
</table>

Figure 5.5 Binary Format Data Frame

• Subheader

- **Command/response flag**
  Differentiates command and response using one bit as follows:
  0: Command
  1: Response
- **Command/responses type**
  Specifies the type of a request from a remote node.

For details on the commands, see IM34M6P41-01E, "Personal Computer Link Command Instruction Manual."
- **CPU number**

  1 byte

  Specifies which sequence CPU module is addressed as follows:
  
  $01$: Sequence CPU module in slot 1  
  $02$: Sequence CPU module in slot 2  
  $03$: Sequence CPU module in slot 3  
  $04$: Sequence CPU module in slot 4

- **Exit code**

  Indicates the result of command execution. For details on the exit code, see Section 5.4, "Response Exit Code."

  1 byte

  Exit code: If $00$, ended normally. If other than $00$, ended in error.

- **Size**

  2 bytes

  Indicates the size in bytes of a command or response parameter. Size 0 means no command or response parameter.

- **Command parameter**

  Contains a device name, a number, data, or even nothing all depending on the command. For details on the commands, see IM34M6P41-01E, "Personal Computer Link Command Instruction Manual."
- **Response parameter**

  **If normal (exit code = $00)**
  Contains returned information or nothing depending on the command that has initiated this response. For details on responses, see IM34M6P41-01E, "Personal Computer Link Command Instruction Manual."

  **If in error (exit code = other than $00)**
  
  - **Detailed error code (one byte long)**
    Contains a detailed error code only when the exit code is either $03, $04, $05, $08, or $52. For further information on detailed error codes, see Section 5.4, "Response Exit Code."
  
  - **Command (one byte long)**
    Contains the command that has caused this response.
5.4 Response Exit Code

5.4.1 Exit Code

Table 5.4 lists the exit codes that may be included in a response.

<table>
<thead>
<tr>
<th>ASCII</th>
<th>Binary</th>
<th>Description</th>
<th>Possible Causes</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;OK&quot;</td>
<td>$00</td>
<td>Normal exit</td>
<td></td>
</tr>
<tr>
<td>&quot;ER01&quot;</td>
<td>$01</td>
<td>Invalid CPU number</td>
<td>The specified CPU number is not within the range of 1 to 4.</td>
</tr>
<tr>
<td>&quot;ER02&quot;</td>
<td>$02</td>
<td>Command error</td>
<td>The specified command does not exist or the command cannot be executed.</td>
</tr>
<tr>
<td>&quot;ER03&quot;</td>
<td>$03</td>
<td>Invalid device</td>
<td>The device does not exist.</td>
</tr>
<tr>
<td>&quot;ER04&quot;</td>
<td>$04</td>
<td>Setting value out of range</td>
<td>A bit setting is neither 0 nor 1.</td>
</tr>
<tr>
<td>&quot;ER05&quot;</td>
<td>$05</td>
<td>Number of data items out of range</td>
<td>The number of bits or words specified exceeded the specifications range. Or, the number of parameters was different from the specified number of data items or devices.</td>
</tr>
<tr>
<td>&quot;ER06&quot;</td>
<td>$06</td>
<td>Monitor error</td>
<td>An attempt was made to run a monitor with no monitor specified.</td>
</tr>
<tr>
<td>&quot;ER08&quot;</td>
<td>$08</td>
<td>Parameter error</td>
<td>An invalid parameter, other than the above cases, is specified.</td>
</tr>
<tr>
<td>&quot;ER51&quot;</td>
<td>$51</td>
<td>Sequence CPU error</td>
<td>The sequence CPU module fails to respond within a specified time span (timeout).</td>
</tr>
<tr>
<td>&quot;ER52&quot;</td>
<td>$52</td>
<td>Sequence CPU processing error</td>
<td>An error was detected during CPU execution.</td>
</tr>
</tbody>
</table>

5.4.2 Detailed Error Code

If the exit code in a response is other than "OK" for ASCII format or $00 for binary format, the response parameter contains a detailed error code.

The detailed error code indicated is valid only if the exit code is "ER03," "ER04," "ER05," "ER08," or "ER52" for ASCII format; or $03, $04, $05, $08, or $52 for binary format. Otherwise, the value has no meaning.

Table 5.5 Detailed Error Codes

<table>
<thead>
<tr>
<th>Exit Code</th>
<th>Description</th>
<th>Detailed Error Code and Its Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;ER03&quot;</td>
<td>Invalid device</td>
<td>The detailed error codes are listed below. For ASCII format, the code is represented as a hexadecimal string.</td>
</tr>
<tr>
<td>&quot;ER04&quot;</td>
<td>Setting value out of range</td>
<td></td>
</tr>
<tr>
<td>&quot;ER05&quot;</td>
<td>Number of data items out of range</td>
<td></td>
</tr>
<tr>
<td>&quot;ER08&quot;</td>
<td>Parameter error</td>
<td></td>
</tr>
<tr>
<td>&quot;ER52&quot;</td>
<td>Sequence CPU processing error</td>
<td></td>
</tr>
</tbody>
</table>

Note: "□" denotes an indeterminate number or character.
5.5 Specifying Devices

This section describes how to address a device of a sequence CPU module in a command.

5.5.1 Specifying a Device in ASCII Format

Specify a device using a six-character name string as shown in the table below.

<table>
<thead>
<tr>
<th>Device Type</th>
<th>How to Specify</th>
<th>Device Type</th>
<th>How to Specify</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input relay</td>
<td>&quot;X        &quot;</td>
<td>Data register</td>
<td>&quot;D        &quot;</td>
</tr>
<tr>
<td>Output relay</td>
<td>&quot;Y        &quot;</td>
<td>File register</td>
<td>&quot;B        &quot;</td>
</tr>
<tr>
<td>Internal relay</td>
<td>&quot;I        &quot;</td>
<td>Shared register</td>
<td>&quot;R        &quot;</td>
</tr>
<tr>
<td>Shared relay</td>
<td>&quot;E        &quot;</td>
<td>Index register</td>
<td>&quot;V        &quot;</td>
</tr>
<tr>
<td>Special relay</td>
<td>&quot;M        &quot;</td>
<td>Special register</td>
<td>&quot;Z        &quot;</td>
</tr>
<tr>
<td>Timer relay</td>
<td>&quot;T        &quot;</td>
<td>Timer preset value</td>
<td>&quot;TS       &quot;</td>
</tr>
<tr>
<td>Counter relay</td>
<td>&quot;C        &quot;</td>
<td>Timer current value</td>
<td>&quot;TC       &quot;</td>
</tr>
<tr>
<td>Link relay</td>
<td>&quot;L        &quot;</td>
<td>Counter preset value</td>
<td>&quot;CP       &quot;</td>
</tr>
</tbody>
</table>

Q: Device number

Example: Specify data register 123 (D0123).

```
"D" $44 "0" $30 "1" $32 "2" $33
```

Figure 5.6 Example for Specifying a Device (in ASCII format)
5.5.2 Specifying a Device in Binary Format

Specify a device by its device attribute and device number as follows:

<table>
<thead>
<tr>
<th>Device Type</th>
<th>Device Attribute</th>
<th>Device Type</th>
<th>Device Attribute</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input relay</td>
<td>X $0018</td>
<td>Data register</td>
<td>D $0004</td>
</tr>
<tr>
<td>Output relay</td>
<td>Y $0019</td>
<td>File register</td>
<td>B $0002</td>
</tr>
<tr>
<td>Internal relay</td>
<td>I $0009</td>
<td>Shared register</td>
<td>R $0012</td>
</tr>
<tr>
<td>Shared relay</td>
<td>E $0005</td>
<td>Index register</td>
<td>V $0016</td>
</tr>
<tr>
<td>Special relay</td>
<td>M $00D</td>
<td>Special register</td>
<td>Z $001A</td>
</tr>
<tr>
<td>Timer relay</td>
<td>T $0014</td>
<td>Link register</td>
<td>W $0017</td>
</tr>
<tr>
<td>Counter relay</td>
<td>C $0003</td>
<td>Timer preset value</td>
<td>$0020</td>
</tr>
<tr>
<td>Link relay</td>
<td>L $00C</td>
<td>Timer current value</td>
<td>$0021</td>
</tr>
</tbody>
</table>

- **Timer current value (for count-up timers)** $0025
- **Counter preset value** $0030
- **Counter current value** $0031
- **Counter current value (for count-up counters)** $0035

### Device Attribute

The table below shows the mapping between device type and device attribute.

### Device Number

Specify the device number using 4 bytes.

Example: Specify data register 123 (D0123).
### 5.5.3 List of Supported Devices

The table below lists the devices of a sequence CPU module that are accessible using commands.

<table>
<thead>
<tr>
<th>Devices</th>
<th>Read Command</th>
<th>Write Command</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>By bit</td>
<td>By word</td>
</tr>
<tr>
<td><strong>Bit devices</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Input relay</td>
<td>√</td>
<td></td>
</tr>
<tr>
<td>Output relay</td>
<td>√</td>
<td></td>
</tr>
<tr>
<td>Internal relay</td>
<td>√</td>
<td></td>
</tr>
<tr>
<td>Shared relay</td>
<td>√</td>
<td></td>
</tr>
<tr>
<td>Special relay</td>
<td>√</td>
<td></td>
</tr>
<tr>
<td>Timer relay</td>
<td>√</td>
<td></td>
</tr>
<tr>
<td>Counter relay</td>
<td>√</td>
<td></td>
</tr>
<tr>
<td>Link relay</td>
<td>√</td>
<td></td>
</tr>
<tr>
<td><strong>Word devices</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Data register</td>
<td>×</td>
<td>√</td>
</tr>
<tr>
<td>File register</td>
<td>×</td>
<td></td>
</tr>
<tr>
<td>Shared register</td>
<td>×</td>
<td>√</td>
</tr>
<tr>
<td>Index register</td>
<td>×</td>
<td></td>
</tr>
<tr>
<td>Special register</td>
<td>×</td>
<td></td>
</tr>
<tr>
<td>Link register</td>
<td>×</td>
<td></td>
</tr>
<tr>
<td>Timer preset value</td>
<td>×</td>
<td>√</td>
</tr>
<tr>
<td>Timer current value</td>
<td>×</td>
<td></td>
</tr>
<tr>
<td>Timer current value</td>
<td>×</td>
<td></td>
</tr>
<tr>
<td>Counter preset value</td>
<td>×</td>
<td></td>
</tr>
<tr>
<td>Counter current value</td>
<td>×</td>
<td></td>
</tr>
<tr>
<td>Counter current value (for count-up counters)</td>
<td>×</td>
<td>√</td>
</tr>
</tbody>
</table>

- √: Supported
- x: Not supported
6. Remote Programming Service

6.1 Overview
The remote programming service allows a user to perform remote programming of a sequence CPU from a remote node (personal computer) on a network using the FA-M3 program development tool WideField or WideField2.

6.2 Functional Specifications

Table 6.1  Remote Programming Service Specifications

<table>
<thead>
<tr>
<th>Item</th>
<th>Specifications</th>
</tr>
</thead>
<tbody>
<tr>
<td>Communications protocol</td>
<td>TCP/IP</td>
</tr>
<tr>
<td>Maximum number of connections</td>
<td>1</td>
</tr>
<tr>
<td>Port No.</td>
<td>12290 ($3002)</td>
</tr>
</tbody>
</table>

6.3 Setup
This service requires no special setup.

6.4 Reference
For details on how to use WideField or WideField2, see the following instruction manuals.

- **Connecting with WideField**
  "FA-M3 Programming Tool WideField Instruction Manual" (IM34M6Q14-01E)

- **Connecting with WideField2**
  "FA-M3 Programming Tool WideField2 Instruction Manual" (IM34M6Q15-01E)
7. Device Monitor Service

7.1 Overview

The device monitor service of the Ethernet Interface Module (F3LE12-0T) allows a user to monitor the devices of the FA-M3 sequence CPU module by sending commands to the F3LE12-0T module from a remote node. All these are achieved without requiring the user to write a communications program on the FA-M3 side.

The device monitor service uses interactive communications control in which commands and responses are exchanged using TCP/IP (through a socket interface).

A command is an instruction issued to the Ethernet Interface Module from a remote node. A response is a reply from the module to the remote node in response to a command. A single command can address up to 512 words of devices across 32 device areas.

The module does not transmit commands to remote nodes. Remote nodes have the right to initiate this service. When a remote node sends a command to the module, the module responds by returning a response to the command.

![Diagram of Device Monitor Service](image)

Figure 7.1 Overview of Device Monitor Service
7.2 Specifications

7.2.1 Functional Specifications

Table 7.1 Device Monitor Service Specifications

<table>
<thead>
<tr>
<th>Item</th>
<th>Specifications</th>
</tr>
</thead>
<tbody>
<tr>
<td>Communications protocol</td>
<td>TCP/IP</td>
</tr>
<tr>
<td>Data format</td>
<td>Binary</td>
</tr>
<tr>
<td>Maximum number of connections</td>
<td>9*</td>
</tr>
<tr>
<td>Port number</td>
<td>12292 ($3004)/12308 ($3014)</td>
</tr>
</tbody>
</table>

*1: Up to 9 connections are available for higher-level link service and device monitor service combined.

Port Number and Operation Condition Switch

You can set the port number to either 12308 ($3014) or 12292 ($3004) using element 1 of the operation condition switch located on the side of the module as shown below.

Table 7.2 Port Number and Operation Condition Switch

<table>
<thead>
<tr>
<th>Element 1</th>
<th>Port Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>OFF</td>
<td>12308 ($3014)</td>
</tr>
<tr>
<td>ON</td>
<td>12292 ($3004)</td>
</tr>
</tbody>
</table>

7.2.2 Communications Protocol

Communications Using TCP/IP Protocol

A remote node and the module communicate with each other through a socket interface as shown below.

![Socket Interface Diagram](image)

Figure 7.2 Communications between Remote Node and the Module

The software of the module automatically starts running when the module is switched on.
7.3 Network Data Frame Format

7.3.1 Communications Data Format

The format of the data exchanged between a remote node and the module is shown below.

![Device Monitor Service Data Format](image)

7.3.2 Device Monitor Service Data Frame

The whole frame of device monitor service data is in binary format as shown below.

- **Data Frame**

  - **Header**
    - 8 bytes

  - **Device Information**
    - 8 × n bytes

  - **Parameter**
    - 2 × m bytes

![Device Monitor Service Data Frame](image)

- **Command/response flag**
  Differentiates command and response using one bit as follows:
  - $00$: Command
  - $80$: Response
- **CPU number**
  For a command, this specifies which sequence CPU module is addressed as follows:
  - $01$: Sequence CPU module in slot 1
  - $02$: Sequence CPU module in slot 2
  - $03$: Sequence CPU module in slot 3
  - $04$: Sequence CPU module in slot 4

  For a response, this contains the content of the command.

- **Data size**
  This is the size in bytes of combined device information and parameter (i.e., the data frame less the header). Device information is 8 bytes long per device area and parameter is 2 bytes long per device.

- **Number of device areas**
  For a command, this specifies the number of device areas to be addressed, where up to 32 device areas can be addressed at the same time. For a response, this contains the content of the command.

- **Reserved/exit code**
  For a command, this has no meaning (system reserved). For a response, this contains the result of command execution.

- **Device information**
  For a command, this defines up to 32 device areas. For a response, this contains what is specified by the command.

- **Number of devices**
  Specifies the number of devices (words) in a device area.

- **Device attribute**
  Indicates the attribute (type) of the devices in a device area.

- **First device number**
  This is the device number of the first device in a device area.
Parameter

A command has no parameter field. A response stores device data in the parameter field. The parameter stores sequentially two bytes of device data for each of the specified number of devices for each device area as defined by device information.

Figure 7.6 Data Frame Structure

Header

Device information

Parameter

Device areas

Lx: Parameter data size in words
m: Data size in bytes for device information and parameter data combined
n: Number of device areas
7.4 Response Exit Code

7.4.1 Exit Code

The exit code as appended to a response has the following meanings:

<table>
<thead>
<tr>
<th>Exit Code</th>
<th>Description</th>
<th>Possible Causes</th>
</tr>
</thead>
<tbody>
<tr>
<td>$0000</td>
<td>Normal exit</td>
<td></td>
</tr>
<tr>
<td>$0100</td>
<td>Invalid CPU number</td>
<td>The specified CPU number is not within the range of 1 to 4.</td>
</tr>
<tr>
<td>$0200</td>
<td>Command error</td>
<td>The specified command does not exist or the command cannot be executed.</td>
</tr>
<tr>
<td>$0500</td>
<td>Number of data items out of range</td>
<td>The specified number of device areas is out of valid range.</td>
</tr>
<tr>
<td>$5100</td>
<td>Sequence CPU error</td>
<td>The sequence CPU module fails to respond within a specified time span (timeout).</td>
</tr>
<tr>
<td>$5200</td>
<td>Sequence CPU processing error</td>
<td>An error was detected during CPU execution.</td>
</tr>
</tbody>
</table>

7.4.2 Detailed Error Code

An exit code other than $0000 ("normal exit") stores a detailed error code in its low-order byte. A detailed error code is valid only for exit code $0800 or $5200 (meaningless for other exit codes).

<table>
<thead>
<tr>
<th>High-order Byte of Exit Code</th>
<th>Description</th>
<th>Detailed Error Code and its Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>$52</td>
<td>Sequence CPU processing error</td>
<td>The detailed error codes are listed below:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$4</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$8</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$9</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$A</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$B</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$C</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$F</td>
</tr>
</tbody>
</table>
7.5 Specifying Devices

The mapping between the device type and the device attribute to be specified in device information for a sequence CPU module is shown below.

Table 7.5 Device Attribute

<table>
<thead>
<tr>
<th>Device Type</th>
<th>Device Attribute</th>
<th>Device Attribute</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input relay</td>
<td>X $0018</td>
<td>Data register</td>
</tr>
<tr>
<td>Output relay</td>
<td>Y $0019</td>
<td>File register</td>
</tr>
<tr>
<td>Internal relay</td>
<td>I $0009</td>
<td>Shared register</td>
</tr>
<tr>
<td>Shared relay</td>
<td>E $0005</td>
<td>Index register</td>
</tr>
<tr>
<td>Special relay</td>
<td>M $000D</td>
<td>Special register</td>
</tr>
<tr>
<td>Timer relay</td>
<td>T $0014</td>
<td>Link register</td>
</tr>
<tr>
<td>Counter relay</td>
<td>C $0003</td>
<td>Timer preset value</td>
</tr>
<tr>
<td>Link relay</td>
<td>L $000C</td>
<td>Timer current value</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Timer current value (for count-up timers)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Counter preset value</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Counter current value</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Counter current value (for count-up counters)</td>
</tr>
</tbody>
</table>

Example: Specifying device information for accessing 200 words of data registers starting from D0123.

Figure 7.7 Example for Specifying Device Information

List of Supported Devices

The table below lists the devices of a sequence CPU module that are accessible through the device monitor service.

Table 7.6 List of Supported Devices

<table>
<thead>
<tr>
<th>Devices</th>
<th>Availability</th>
<th>Devices</th>
<th>Availability</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bit devices</td>
<td></td>
<td>Word devices</td>
<td></td>
</tr>
<tr>
<td>Input relay</td>
<td>!</td>
<td>Data register</td>
<td>√</td>
</tr>
<tr>
<td>Output relay</td>
<td>!</td>
<td>File register</td>
<td>√</td>
</tr>
<tr>
<td>Internal relay</td>
<td>!</td>
<td>Shared register</td>
<td>√</td>
</tr>
<tr>
<td>Shared relay</td>
<td>!</td>
<td>Index register</td>
<td>√</td>
</tr>
<tr>
<td>Special relay</td>
<td>!</td>
<td>Special register</td>
<td>√</td>
</tr>
<tr>
<td>Timer relay</td>
<td>!</td>
<td>Link register</td>
<td>√</td>
</tr>
<tr>
<td>Counter relay</td>
<td>!</td>
<td>Timer preset value</td>
<td>√</td>
</tr>
<tr>
<td>Link relay</td>
<td>!</td>
<td>Timer current value</td>
<td>√</td>
</tr>
<tr>
<td>Timer current value (for count-up timers)</td>
<td>√</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Counter preset value</td>
<td>√</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Counter current value</td>
<td>√</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Counter current value (for count-up counters)</td>
<td>√</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Restrictions

Beware of the following restrictions when specifying devices:

- For bit devices, 16 bits are handled as one word, starting from the specified address.
- One device area may contain up to 64 words of bit devices.
- One device area may contain up to 256 words of word devices.
- One device area may contain only link relays or registers having the same system number.
8. Messaging Service

8.1 Overview

The messaging service of the module allows the sequence CPU module to exchange data (messages) with a remote node.

The messaging service features:
- Generation of up to 8 sockets (datagrams)
- Access from a user program through socket handles
- Independent 1024-byte send buffer and 1024-byte receive buffer for each socket handle
- Asynchronous send/receive using a single socket handle

8.2 Specifications

8.2.1 Functional Specifications

Table 8.1 Messaging Service Specifications

<table>
<thead>
<tr>
<th>Item</th>
<th>Specifications</th>
</tr>
</thead>
<tbody>
<tr>
<td>Communications protocol</td>
<td>UDP/IP</td>
</tr>
<tr>
<td>Maximum number of sockets</td>
<td>8</td>
</tr>
<tr>
<td>Maximum send message size</td>
<td>1024 bytes</td>
</tr>
<tr>
<td>Maximum receive message size</td>
<td>1024 bytes</td>
</tr>
</tbody>
</table>

8.2.2 Send/Receive Message Format

A user program running on the sequence CPU module can access only the data part of a message frame. It cannot access the header part of a message frame.

Figure 8.1 Message Format
### 8.3 Socket Handles

A user program (application) running on a sequence CPU module specifies a socket of the F3LE12-0T module to send or receive data as part of a message, where a socket handle is used to associate the data with the socket.

A socket handle includes port number, message size, destination IP address and other information necessary for processing a send and receive message.

Each socket handle is identified with a unique socket handle number. A user program specifies the I/O relays and send/receive buffers of the F3LE12-0T module to be used for exchanging data during messaging by means of a socket handle. Here, using a socket means specifying a set of I/O relays and buffer addresses (data position Nos.). In other words, logical addresses to be addressed are automatically determined by a socket handle number.

![Figure 8.2 Socket Handle Overview](image)
8.4 Messaging

This section describes how messaging is carried out.

Figure 8.3 Messaging

Figure 8.4 Socket Creation
**Socket Creation**

Messaging requires creation of a socket. The module creates a socket for a socket handle upon request by a user program running on a sequence CPU module. Each socket handle specifies a local port number and UDP (datagram). The port number specified for each socket handle must be unique and different from system-reserved port numbers.

**Send Processing**

The module sends a message upon the request of a user program. A socket must be created before a message can be sent. To send a message, a user program writes the send message size and send data, and then turns on the Send Request relay. The module sends the message according to the specified destination port and specified destination IP address, both of which can be changed between messages. A message can also be broadcasted to a number of remote nodes. When a message is delivered, the module turns on the Send Completed relay. The send message buffer size is 1024 bytes.
Receive Processing

When the module receives a message from a remote node, it stores the data in the receive buffer and turns on the Clear to Receive relay. The sequence CPU module then reads the receive message size, source IP address and source port number, as well as the data as necessary. After reading the message, the sequence CPU module turns on the Receive Completed relay.

Socket Closure

A user program issues a request to close a socket when it no longer needs it. To change the port number associated with the socket handle of an open socket, a user program must first close the socket and then reissue a socket creation request.
8.5 Messaging Procedure (UDP/IP)

- Messaging Procedure Overview

Figure 8.6 Communications Using UDP/IP Protocol
Creating a Socket

A socket must be created and associated to a socket handle before it can be used for communications with a remote node. Up to 8 sockets may be created, one for each of the eight available socket handles.

Socket creation procedure

The procedure for creating a socket is shown below.

- **Setting up socket parameters**

  Set up socket parameters by writing values into the socket parameter setup area of the module using the WRITE instruction. The table below shows the contents of the socket parameter setup area.

  **Table 8.2  Socket Parameter Setup Area**

<table>
<thead>
<tr>
<th>Data Position No.</th>
<th>Handle 1</th>
<th>Handle 2</th>
<th>Handle 3</th>
<th>Handle 4</th>
<th>Handle 5</th>
<th>Handle 6</th>
<th>Handle 7</th>
<th>Handle 8</th>
</tr>
</thead>
<tbody>
<tr>
<td>961</td>
<td>1537</td>
<td>1569</td>
<td>1601</td>
<td>1633</td>
<td>1665</td>
<td>1697</td>
<td>1729</td>
<td>1761</td>
</tr>
<tr>
<td></td>
<td>1538</td>
<td>1570</td>
<td>1602</td>
<td>1634</td>
<td>1666</td>
<td>1698</td>
<td>1730</td>
<td>1762</td>
</tr>
</tbody>
</table>

  - **Bank register**
    
    Data positions 1025-2048 are accessed through bank switching by writing an appropriate value to the bank register. The socket parameter setup area is in bank 0.

  - **Socket type**
    
    The type of the socket to be created is fixed to 0 for UDP (SOCK_DGRAM).
- **Port No.**
  Specify the port number to be bound to the socket.

- **Socket Creation Request relays and Socket Creation Completed relays**

  Turn on the Socket Creation Request relay for a socket handle to create a socket (socket/bind) and bind it to the socket handle. After creating the socket, the module turns on the Socket Creation Completed relay. The Socket Creation Request relays and Socket Creation Completed relays for each socket handle are listed in the table below.

  **Table 8.3  Socket Creation Request and Socket Creation Completed Relays**

<table>
<thead>
<tr>
<th>Relays</th>
<th>Handle 1</th>
<th>Handle 2</th>
<th>Handle 3</th>
<th>Handle 4</th>
<th>Handle 5</th>
<th>Handle 6</th>
<th>Handle 7</th>
<th>Handle 8</th>
</tr>
</thead>
<tbody>
<tr>
<td>Y0□□033</td>
<td>X0□□04</td>
<td>Y0□□06</td>
<td>Y0□□07</td>
<td>X0□□13</td>
<td>X0□□15</td>
<td>X0□□17</td>
<td>X0□□19</td>
<td>X0□□22</td>
</tr>
<tr>
<td>Y0□□01</td>
<td>X0□□04</td>
<td>Y0□□07</td>
<td>X0□□10</td>
<td>X0□□13</td>
<td>X0□□16</td>
<td>X0□□19</td>
<td>X0□□22</td>
<td></td>
</tr>
</tbody>
</table>

  ††: Slot No.

  **- Socket Creation Request relay**

  Turning on the Socket Creation Request relay for a socket handle requests the module to create a socket according to the specified socket parameters and binds it to the socket handle. Always check that the Socket Creation Completed relay is off (indicating that the handle has no associated socket) before turning on the Socket Creation Request relay. Turning off the Socket Creation Request relay for a socket handle releases its associated socket. You must always turn on the Socket Creation Request relay to create a socket for a socket handle before you can use the socket handle to send or receive messages.

  **- Socket Creation Completed relay**

  After completing socket creation, the module turns on this relay and stores the result of socket creation in the Socket Creation Completed Status register. The module turns off this relay if the Socket Creation Request relay is turned off.

  The behaviors of the Socket Creation Request relay output relay and the Socket Creation Completed input relay are shown below.

  ![Socket creation diagram](image)

- **Socket Creation Completed status**

  When the Socket Creation Completed relay is turned on, the result of socket creation (normal/error) is stored in the Socket Creation Completed Status register. The Socket Creation Completed Status register is allocated to each socket handle as follows:

  **Table 8.4  Socket Creation Completed Status Registers**

<table>
<thead>
<tr>
<th>Data Position No.</th>
<th>Handle 1</th>
<th>Handle 2</th>
<th>Handle 3</th>
<th>Handle 4</th>
<th>Handle 5</th>
<th>Handle 6</th>
<th>Handle 7</th>
<th>Handle 8</th>
</tr>
</thead>
<tbody>
<tr>
<td>961</td>
<td>1544</td>
<td>1576</td>
<td>1608</td>
<td>1640</td>
<td>1672</td>
<td>1704</td>
<td>1736</td>
<td>1768</td>
</tr>
</tbody>
</table>

  Socket Creation Completed Status
- **Bank register**
  Data positions 1025-2048 are accessed through bank switching by writing an appropriate value to the bank register. The Socket Creation Completed Status registers are located in bank 0.

- **Socket Creation Completed Status**
  The meaning of the Socket Creation Completed Status is as follows:

<table>
<thead>
<tr>
<th>Socket Creation Completed Status</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Completed normally</td>
</tr>
<tr>
<td>2</td>
<td>Invalid port No.</td>
</tr>
<tr>
<td>3</td>
<td>Invalid protocol</td>
</tr>
<tr>
<td>5</td>
<td>Bind error</td>
</tr>
<tr>
<td>7</td>
<td>Socket creation error</td>
</tr>
<tr>
<td>8</td>
<td>Socket option setup error</td>
</tr>
</tbody>
</table>

**Sample program for socket creation**

Now let us create a socket by setting up socket parameters and using I/O relays. To set up socket parameters, specify a socket type and port number and write the setup data to the module using the WRITE instruction. Then, turn on the Socket Creation Request. When a socket is successfully created, the Socket Creation Completed relay turns on and the result of socket creation is stored in the Socket Creation Completed Status register. Read the Socket Creation Completed status using the READ instruction.

Figure 8.9 shows a sample program for creating a socket with the parameters shown in Table 8.6.

**Table 8.6  Socket Parameters for Sample Program**

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>Socket type</td>
<td>UDP (= 0)</td>
</tr>
<tr>
<td>Port No.</td>
<td>10001</td>
</tr>
</tbody>
</table>

The sample program assumes that /I00001 is its trigger signal; the module is installed in slot 5 of the base module; and the socket handle No. is 1.
Figure 8.8 Sample Program for Socket Creation

- Issue socket creation request
- Check completion of socket creation
- Error processing

Sample program for socket creation

<table>
<thead>
<tr>
<th>Socket type</th>
<th>Port No.</th>
<th>Bank register</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Socket creation request</th>
</tr>
</thead>
<tbody>
<tr>
<td>MOV 0</td>
</tr>
<tr>
<td>MOV 1000</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Bank register</th>
</tr>
</thead>
<tbody>
<tr>
<td>WRITE 0</td>
</tr>
<tr>
<td>WRITE /00000</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Socket creation completed status</th>
</tr>
</thead>
<tbody>
<tr>
<td>SET 100003</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Error processing</th>
</tr>
</thead>
<tbody>
<tr>
<td>READ 5 3041 /200000</td>
</tr>
</tbody>
</table>
Sending

Once a socket is created for a socket handle, you can use the socket handle to send any number of messages. You can send a message to a specific node by specifying a destination IP address/port number or broadcast a message to a number of nodes at the same time.

Turning on the Send Request relay sends the data in the send message area to a remote node according to the specified destination IP address, destination port No. and send message size. When finishing sending, the module turns on the Send Completed relay. A 1024-byte send message area is provided for each socket handle. The result of send processing is stored in the Send Completed Status register.

- **Send procedure**

  The procedure for sending a message is shown below.

  ![Figure 8.9 Procedure for Sending a Message](image)

- **Setting up send parameters**

  Set up destination IP address, destination port No., and send message size by writing values for send parameters to the module using the WRITE instruction. The table below shows the data position numbers for send parameters.
Table 8.7  Send Parameter Setup Area

<table>
<thead>
<tr>
<th>Data Position No.</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>961</td>
<td>Bank register (set to socket handle No.)</td>
</tr>
<tr>
<td>1793</td>
<td>Destination IP address</td>
</tr>
<tr>
<td>1794</td>
<td>Destination port No.</td>
</tr>
<tr>
<td>1796</td>
<td>Send message size</td>
</tr>
</tbody>
</table>

- **Bank register**
  Data positions 1025-2048 are accessed through bank switching using the bank register.
  Set the bank register to the required socket handle No.

- **Destination IP address**
  Specify the IP address of a remote node to which a message is to be sent.

- **Destination port No.**
  Specify an appropriate port No. of a remote node to which a message is to be sent.

- **Send message size**
  Specify the size of data to be sent in bytes.

● **Writing send data to the module**
Write the data to be sent into the send message setup area using the WRITE instruction. The table below shows the contents of the area.

Table 8.8  Send Message Setup Area

<table>
<thead>
<tr>
<th>Data Position No.</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>961</td>
<td>Bank register (set to socket handle No.)</td>
</tr>
<tr>
<td>1025 to 1536</td>
<td>Send data</td>
</tr>
</tbody>
</table>

- **Bank register**
  Data positions 1025-2048 are accessed through bank switching using the bank register.
  Set the bank register to a socket handle number

- **Send data**
  Data in the send message setup area is sent sequentially starting from the first byte.
  As the area is formatted in units of words, if an odd number of bytes of data is to be sent, store the last byte in the high-order byte of the last word.

● **Send Request relays and Send Completed relays**
Turning on the Send Request relay sends a message. After sending the message, the module turns on the Send Completed relay. The table below lists these relays.
Table 8.9 Send Request and Send Completed Relays

<table>
<thead>
<tr>
<th>Handle 1</th>
<th>Handle 2</th>
<th>Handle 3</th>
<th>Handle 4</th>
<th>Handle 5</th>
<th>Handle 6</th>
<th>Handle 7</th>
<th>Handle 8</th>
</tr>
</thead>
<tbody>
<tr>
<td>Y00034</td>
<td>Y00037</td>
<td>Y00040</td>
<td>Y00043</td>
<td>Y00046</td>
<td>Y00049</td>
<td>Y00052</td>
<td>Y00055</td>
</tr>
<tr>
<td>X00002</td>
<td>X00005</td>
<td>X00008</td>
<td>X00011</td>
<td>X00014</td>
<td>X00017</td>
<td>X00020</td>
<td>X00023</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Slot No.</th>
</tr>
</thead>
</table>

- **Send Request relay**
  Turning on this relay requests the module to send a message according to the send parameters. Always check that the Send Completed relay is off before turning on the Send Request relay.

- **Send Completed relay**
  After sending a message, the module turns on this relay regardless of whether sending is successful or not. Turning off the Send Request relay also turns off the Send Completed relay.

The behaviors of the Send Request output relay and the Send Completed input relay are shown below.

![Sending a message diagram]

**Send Completed Status**

When the module turns on the Send Completed relay, it also stores the result of send processing in the Send Completed Status register as follows:

Table 8.10 Send Completed Status Register

<table>
<thead>
<tr>
<th>Data Position</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>961</td>
<td>Bank register (set to socket handle No.)</td>
</tr>
<tr>
<td>1800</td>
<td>Send Completed Status</td>
</tr>
</tbody>
</table>

- **Bank register**
  Data positions 1025-2048 are accessed through bank switching using the bank register.
  Set the bank to a socket handle No.

- **Send Completed status**
  The table below shows the meanings of the Send Completed status.

Table 8.11 Send Completed Status

<table>
<thead>
<tr>
<th>Send Completed status</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Completed normally</td>
</tr>
<tr>
<td>1</td>
<td>Socket not created</td>
</tr>
<tr>
<td>2</td>
<td>Invalid message size</td>
</tr>
<tr>
<td>Others</td>
<td>Sending not allowed</td>
</tr>
</tbody>
</table>
## Sample program for sending a message

Now let us send a message by setting up send data and using I/O relays. To send a message, specify send parameters (destination IP address and port No.), write send data to the module using the WRITE instruction, and turn on the Send Request relay. After sending a message, the module turns on the Send Completed relay and stores the result of send processing in the Send Completed Status register, which can be read using the READ instruction.

Figure 8.10 shows a sample program for sending message according to the parameters shown in Table 8.12.

### Table 8.12 Send Parameters for Sample Program

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>Destination IP address</td>
<td>192.168.1.1</td>
</tr>
<tr>
<td>Destination port No.</td>
<td>10002</td>
</tr>
<tr>
<td>Send data Data stored in the 512-word data positions starting with D1025</td>
<td></td>
</tr>
</tbody>
</table>

The sample program assumes that /I00001 is its trigger signal; the module is installed in slot 5 of the base module; and the socket handle No. is 1.

![Sample program for sending a message](image-url)
Receiving

When the module receives a message from a remote node, it turns on the Clear to Receive relay. Using a socket handle, you can read the data that has been received for a socket. You can also read information on the IP address and port No. of the source node and the receive message size. A 1024-byte received data area is provided for each socket handle.

- **Receive procedure**

  The procedure for receiving a message is shown below.

  ![Procedure for Receiving a Message Diagram]

  - **Reading receive parameters**

    The receive parameters store the source IP address, source port No., and receive message size for a message that the module has received from a remote node. Read these parameters using the READ instruction. The table below shows the data position numbers for receive parameters.
Table 8.13 Receive Parameter Area

<table>
<thead>
<tr>
<th>Data Position No.</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>961</td>
<td>Bank register (set to socket handle No. + 15)</td>
</tr>
<tr>
<td>1793</td>
<td>Source IP address</td>
</tr>
<tr>
<td>1794</td>
<td>Source port No.</td>
</tr>
<tr>
<td>1795</td>
<td>Source port No.</td>
</tr>
<tr>
<td>1796</td>
<td>Receive message size</td>
</tr>
</tbody>
</table>

- **Bank register**
  Data positions 1025-2048 are accessed through bank switching using the bank register.
  Set the bank register to the required socket handle No. plus 15.

- **Source IP address**
  Stores the IP address of a remote node that has sent a message to the module.

- **Source port No.**
  Stores the port No. of a remote node that has sent a message to the module.

- **Receive message size**
  Stores the size of the data received in bytes.

**Reading received data**
Read the received data from the received data area of the module using the READ instruction. The table below shows the contents of the area.

Table 8.14 Received Data Area

<table>
<thead>
<tr>
<th>Data Position No.</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>961</td>
<td>Bank register (set to socket handle No. + 15)</td>
</tr>
<tr>
<td>1025 to 1536</td>
<td>Received data</td>
</tr>
</tbody>
</table>

- **Bank register**
  Data positions 1025-2048 are accessed through bank switching using the bank register.
  Set the bank register to the required socket handle No. plus 15.

- **Received data**
  Received data is stored in the received data area from the first byte sequentially.
  Because the area is formatted in units of words, if an odd number of bytes of data are received, the last byte data is stored in the higher-order byte of a word.

**Clear to Receive relays and Receive Completed relays**
The module turns on the Clear to Receive relay after it has received a message from a remote node. It turns off the relay after a program has read the data and turns on the Receive Completed relay. The table below lists these relays.
Table 8.15 Clear to Receive Relays and Receive Completed Relays

<table>
<thead>
<tr>
<th>Handle 1</th>
<th>Handle 2</th>
<th>Handle 3</th>
<th>Handle 4</th>
<th>Handle 5</th>
<th>Handle 6</th>
<th>Handle 7</th>
<th>Handle 8</th>
</tr>
</thead>
<tbody>
<tr>
<td>X0</td>
<td>X0</td>
<td>X0</td>
<td>X0</td>
<td>X0</td>
<td>X0</td>
<td>X0</td>
<td>X0</td>
</tr>
<tr>
<td>Y0</td>
<td>Y0</td>
<td>Y0</td>
<td>Y0</td>
<td>Y0</td>
<td>Y0</td>
<td>Y0</td>
<td>Y0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Clear to Receive relays</th>
<th>Receive Completed relays</th>
</tr>
</thead>
<tbody>
<tr>
<td>X0††</td>
<td>Y0††</td>
</tr>
</tbody>
</table>

- **Clear to Receive relay**
  When the module receives a message from a remote node, it stores source node information in the receive parameters, stores received data in the received data area, and turns on the Clear to Receive relay. The module cannot read the next message from the socket while the Clear to Receive relay is on.

- **Receive Completed relay**
  When a program finishes reading received data and turns on the Receive Completed relay, the module turns off the Clear to Receive relay. The module cannot read the next message from the socket while the Clear to Receive relay is on.

The behaviors of the Clear to Receive input relay and the Receive Completed output relay are shown below.

- **Sample program for receiving a message**
  When the module receives a message from a remote node, it stores source node information (IP address and port No.) in the receive parameters, stores received data in the received data area, and turns on the Clear to Receive relay. After confirming that the Clear to Receive relay is on and the Receive Completed relay is off, you can read the received data by issuing the READ instruction.

Figure 8.13 shows a sample program for reading received data, assuming that the module is installed in slot 5 of the base module and the socket handle No. is 1.
Sample program for receiving a message

Read receive parameters

![Diagram showing the program flow]

Calculate the size of received data

![Diagram showing the program flow]

Read received data

![Diagram showing the program flow]

Figure 8.13  Sample Program for Receiving a Message

The program stores the receive parameter information and received data in the devices of the sequence CPU module as shown in Table 8.16.

Table 8.16  Data Registers Used by Sample Program

<table>
<thead>
<tr>
<th>Devices</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>/D00001</td>
<td>Source IP address</td>
</tr>
<tr>
<td>/D00002</td>
<td></td>
</tr>
<tr>
<td>/D00003</td>
<td>Source port No.</td>
</tr>
<tr>
<td>/D00004</td>
<td>Receive message size</td>
</tr>
<tr>
<td>/D00513 and onwards</td>
<td>Received data</td>
</tr>
</tbody>
</table>
9. Network Setup

This chapter describes how to set up the network environment of the module using a program on the sequence CPU module.

9.1 Network Environment Setup

Set up the subnet mask and default gateway of the module. If necessary, consult the network administrator in charge of the network to which the module is to be connected.

Network Environment Setup Parameters

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Contents</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Subnet mask</td>
<td>Subnet mask</td>
<td>2-word value</td>
</tr>
<tr>
<td>Default gateway IP address</td>
<td>IP address of the network gateway</td>
<td>2-word value</td>
</tr>
</tbody>
</table>

- **Subnet mask**

  Specify the subnet mask of the network or system to which the module is to be connected. A subnet mask is used together with an IP address to define a network address. Depending on the network environment, it may not be necessary to set up the subnet mask.

- **Default gateway IP address**

  Specify the IP address of the gateway to another network. A gateway is normally a router within the same network or system serving as a connection point to another network. Depending on the network environment, it may not be necessary to set up the default gateway IP address.
Network Environment Setup Procedure

The procedure for setting up the network environment is given below.

- Set up parameters
- Turn on Network Environment Setup Request relay
- Configure the module according to the parameters
- Turn on Network Environment Setup Completed relay
- Read Network Environment Setup Completed status
- Turn off Network Environment Setup Request relay
- Turn off Network Environment Setup Completed relay

Program operation
Module operation

Figure 9.1 Procedure for Setting up Network Environment using a Program

Setting Parameters

Set up the network environment parameters of the module by writing parameter values using the WRITE instruction as follows:

<table>
<thead>
<tr>
<th>Data Position No.</th>
<th>Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>961</td>
<td>Bank register (= 0)</td>
</tr>
<tr>
<td>1025</td>
<td>Subnet mask</td>
</tr>
<tr>
<td>1026</td>
<td>Default gateway IP address</td>
</tr>
<tr>
<td>1027</td>
<td></td>
</tr>
<tr>
<td>1028</td>
<td></td>
</tr>
</tbody>
</table>

- **Bank register**
  
  Data positions 1025-2048 are accessed through bank switching using the bank register.

  Set the bank register to 0 before accessing data positions 1025-1028 (in bank 0).

Network Environment Setup Request relay and Network Environment Setup Completed relay

When the program turns on the Network Environment Setup Request relay, the module starts configuring its network environment according to the specified network environment parameters. The module turns on the Network Environment Setup Completed relay when it finishes configuration.

The values of the network environment parameters are stored in the internal flash memory so no further network environment setup is required even after the module is switched off.
The relay numbers of the Network Environment Setup Request relay and Network Environment Setup Completed relay are given below.

Table 9.3 Network Environment Setup Relays

<table>
<thead>
<tr>
<th>Relay No.</th>
<th>Relay Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Y0</td>
<td>Network Environment Setup Request relay</td>
</tr>
<tr>
<td>X0</td>
<td>Network Environment Setup Completed relay</td>
</tr>
</tbody>
</table>

- **Network Environment Setup Request relay**
  Check that the Network Environment Setup Completed relay is off before turning on the Network Environment Setup Request relay to have the module configure the network environment according to the specified network environment parameters.

- **Network Environment Setup Completed relay**
  The module turns on the Network Environment Setup Completed relay when it has completed the configuration or if it detects an error.

⚠️ **CAUTION**

The internal flash memory may be rewritten up to 100,000 times. Ensure that the power supply is not interrupted while the memory is being rewritten.

- **Network environment setup completed status**
  When the Network Environment Setup Completed relay turns on, the setup result is stored as a code in the Network Environment Setup Completed Status register.

Table 9.4 Network Environment Setup Completed Status

<table>
<thead>
<tr>
<th>Data Position No.</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>961</td>
<td>Bank register (= 0)</td>
</tr>
<tr>
<td>1161</td>
<td>Network Environment Setup Completed Status</td>
</tr>
</tbody>
</table>

- **Network environment setup completed status codes**
  The list of network environment setup completed status codes are described below.

Table 9.5 Network Environment Setup Completed Status Codes

<table>
<thead>
<tr>
<th>Codes</th>
<th>Meaning</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$0000</td>
<td>Normal exit</td>
<td>Network environment setup is successful.</td>
</tr>
<tr>
<td>$0001</td>
<td>Subnet mask error</td>
<td>The specified subnet mask is invalid.</td>
</tr>
<tr>
<td>$0002</td>
<td>Invalid default gateway IP address</td>
<td>The specified default gateway IP address is invalid.</td>
</tr>
</tbody>
</table>

- **Sample program for setting up network environment**
  A program sets up the network environment of the module by writing data into network environment parameters and manipulating network environment relays.

Now let us study a sample program that sets data to the network environment parameters by executing the WRITE instruction, and turns on the Network Environment Setup Request relay to initiate network environment configuration. When the module completes the configuration and turns on the Network Environment Setup Completed relay, the program reads the setup result from the Network Environment Setup Completed status by executing a READ instruction.
Suppose that the module uses the network environment parameters given in Table 9.6. Figure 9.2 shows the sample program.

Table 9.6 Network Environment Parameters for Sample Program

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>Subnet mask</td>
<td>255.255.255.0</td>
</tr>
<tr>
<td>Default gateway IP</td>
<td>192.168.1.254</td>
</tr>
</tbody>
</table>

The sample program assumes that /I00001 is its trigger signal and the module is installed in slot 5 of the base module.
10. Troubleshooting

This chapter describes how to troubleshoot communications failure and other problems.

10.1 Hardware or Human Error?

Check the following to ensure that there is no human error involved before proceeding with troubleshooting:

- Is the module installed correctly?
- Is the operation condition switch of the module set correctly?
- Is the IP address of the module set correctly?
- Is the communications cable connected correctly?
- Is the network configured correctly?
- Is the network equipment set correctly?
- Is the network equipment switched on?
10.2 When RDY LED Is Not Lit

Troubleshoot as described below if RDY LED is not lit.

- **ERR LED is not lit**
  - Is the power supply module supplied with the required voltage? **No** Apply correct voltage
  - Yes Is RDY LED of the power supply module lit? **No** Replace the power supply module
  - Yes Is RDY LED of the other modules lit? **No** Check constraints (current consumption) or replace the power supply or base module
  - Yes Is the module properly mounted to the base module? **No** Push the module down until it clicks into place
  - Yes Is RDY LED lit if the module is mounted in another slot? **No** Replace the module
  - Yes

Replace the module
10.3 When ERR LED Is Lit

Troubleshoot as described below if ERR LED is lit.

- **ERR LED is lit**
  - Are module switches properly set? No → **Set the IP address and operation condition switches properly**
  - **Yes**
    - Is the module properly connected to network? No → **Connect the module to the network properly**
    - **Yes**
      - Operation mode?
        - **Normal mode**
          - Loopback test
            - Normal → **Replace the module**
            - Error → **Fix the network environment**
### INDEX

**A**  
ARP .............................................................. 1-2, 3-1  
ASCII format ...................................................... 4-5, 5-2, 5-5, 5-12  
automatic negotiation .................................................. 3-2

**B**  
bank register ........................................................... 3-5  
binary format ......................................................... 4-5, 5-2, 5-8, 5-13

**C**  
CE marking conformance ........................................... 4-8  
command flag  
  device monitor service ............................................. 7-3  
command parameter  
  device monitor service ............................................. 7-3  
  higher-level link service ........................................... 5-6, 5-9  
command  
  device monitor service ............................................. 7-1  
  higher-level link service ........................................... 5-1, 5-5  
connection  
  using 10BASE-T ....................................................... 2-1  
  using 100BASE-T ....................................................... 2-2  
connector ................................................................. 4-7  
CPU number ............................................................. 5-6, 5-9, 7-4  
current consumption .................................................... 3-1

**D**  
data format ............................................................ 3-3, 4-5  
  device monitor service ............................................. 7-2  
  higher-level link service ........................................... 5-2  
setup ........................................................................... 4-5  
default gateway IP address ............................................. 3-5, 9-1  
device attribute  
  device monitor service ............................................. 7-7  
  higher-level link service ........................................... 5-12  
device information ......................................................... 7-4  
device monitor service ................................................... 1-1, 7-1

**E**  
end characters .......................................................... 5-8  
exit code  
  device monitor service ............................................. 7-4, 7-6  
  higher-level link service ........................................... 5-9, 5-11

**F**  
FG terminal ............................................................ 3-2, 4-7

**H**  
header ........................................................................ 7-3

**I**  
ICMP ........................................................................ 1-2  
IP address ................................................................. 3-5  
default gateway IP address ............................................. 3-5, 9-1  
destination IP address ..................................................... 3-7, 8-12  
IP address switches ......................................................... 3-2, 4-4  
source IP address ........................................................ 3-7, 8-16  
IP .................................................................................. 1-2  
ISO reference mode ....................................................... 1-2

**L**  
LED indicators ............................................................ 3-1, 3-2  
loopback test .................................................................. 4-9

**M**  
MAC address ............................................................. 3-3, 3-5  
maximum number of connections  
  device monitor service ............................................. 7-2  
  higher-level link service ........................................... 5-2  
  remote programming service ........................................ 6-1  
maximum number of installed modules ................................. 3-1  
messaging  
  messaging service ....................................................... 1-2, 8-11  
  receive message procedure ........................................... 8-15  
  receive message size ................................................... 3-7, 8-16  
  send message procedure ............................................. 8-11  
  send message size ...................................................... 3-7, 8-12

**N**  
network environment setup ............................................ 9-1  
network environment parameters ....................................... 9-2  
Network Environment Setup Completed  
  relay ........................................................................... 3-4, 9-2  
Network Environment Setup Completed Status ......................... 3-5, 9-3  
Network Environment Setup Request  
  relay ........................................................................... 3-4, 9-2  
number of connections  
  device monitor service ............................................. 7-2  
  higher-level link service ........................................... 5-2  
  remote programming service ........................................ 6-1

**O**  
operation condition switch ........................................... 3-3, 3-5, 4-5  
operation mode ............................................................ 3-3, 4-5
P
port number
destination port number......................... 3-7, 8-12
device monitor service.............................. 7-2
higher-level link service........................... 5-2
messaging service...................................... 8-4
remote programming..................................... 6-1
source port number................................... 3-7, 8-16
power supply............................................ 3-1

R
receiving
Clear to Receive relay ........................... 3-4, 8-17
higher-level link service........................... 1-1, 5-1
Receive Completed relay ......................... 3-4, 8-17
receive control block................................... 3-7
receive message size............................... 3-7, 8-16
receive parameters........................................ 8-15
receive processing....................................... 8-5
received data area................................... 3-7, 8-16
relays
Clear to Receive relay ........................... 3-4, 8-17
Network Environment Setup Completed relay........ 3-4, 9-2
Network Environment Setup Request relay........... 3-4, 9-2
Receive Completed relay ......................... 3-4, 8-17
Socket Creation Completed relay .................. 3-4, 8-8
Socket Creation Request relay..................... 3-4, 8-8
Send Completed relay............................... 3-4, 8-12
Send Request relay...................................... 3-4, 8-12
remote programming service......................... 1-1, 6-1
response flag
device monitor service............................... 7-3
higher-level link service............................ 5-6, 5-8
response parameters
device monitor service............................... 7-5
higher-level link service............................ 5-7, 5-10
response
device monitor service............................... 7-3
higher-level link service............................ 5-5, 5-8

S
sending
destination IP address............................... 3-7, 8-12
destination port number.............................. 3-7, 8-12
Send Completed relay............................... 3-4, 8-12
Send Completed status.............................. 3-7, 8-13
send control block...................................... 3-7
send message setup area............................. 3-7, 8-12
send message size...................................... 3-7, 8-12
send parameters........................................ 8-11
send processing......................................... 8-4
Send Request relay...................................... 3-4, 8-12
source IP address...................................... 3-7, 8-16
source port number................................. 3-7, 8-16
setup
data format............................................ 4-5
IP address................................................. 4-4
shield terminal......................................... 3-2, 4-7
size
parameter size......................................... 5-9, 7-4
receive message size............................... 3-7, 8-16
send message size...................................... 3-7, 8-12
socket
maximum receive message size..................... 8-1
maximum send message size......................... 8-1
number of device areas............................. 7-4
number of socket handles........................... 8-1
socket closure.......................................... 8-5
Socket Creation Completed relay................... 3-4, 8-8
Socket Creation Completed Status................ 3-5, 8-8
socket creation procedure........................... 8-7
Socket Creation Request relay..................... 3-4, 8-8
socket creation.......................................... 8-4
socket handles......................................... 8-2
socket parameters..................................... 8-7
status
Network Environment Setup Completed Status........ 3-5, 9-3
Send Completed Status.............................. 3-7, 8-13
Socket Creation Completed Status................... 3-5, 8-8
subheader................................................. 5-6, 5-8
subnet mask.............................................. 3-5, 9-1
switch
IP address switches................................. 3-2, 4-4
operation condition switch........................ 3-3, 4-5

T
TCP........................................................... 1-2
transmission method.................................... 3-1
transmission specifications............................. 3-1
transmission speed...................................... 3-1

U
UDP.......................................................... 1-2

W
WideField2.................................................. 6-1
write protection......................................... 3-3, 4-5, 5-2