OMRON

GD Series

IO-Link Master Unit (Multi-Network Compatible)

User's Manual

Ethernet & Modbus/TCP Edition

GD-ILM□□□-MLP

IO-Link Master Unit

Z516-E1-02

NOTE

- All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form, or by any means, mechanical, electronic, photocopying, recording, or otherwise, without the prior written permission of OMRON.
- No patent liability is assumed with respect to the use of the information contained herein.
  Moreover, because OMRON is constantly striving to improve its high-quality products, the information contained in this manual is subject to change without notice.
- Every precaution has been taken in the preparation of this manual. Nevertheless, OMRON assumes no responsibility for errors or omissions.
  Neither is any liability assumed for damages resulting from the use of the information contained in this publication.

Trademarks

- Microsoft, Windows, Excel, Visual Basic, and Microsoft Edge are either registered trademarks or trademarks of Microsoft Corporation in the United States and other countries.
- ODVA, CIP, CompoNet, DeviceNet, and EtherNet/IP are trademarks of ODVA.
- Modbus is a registered trademark of Schneider Electric USA Inc.
- CC-Link IE Field Basic is a registered trademark of Mitsubishi Electric Corporation and a trademark controlled by the CC-Link Partner Association.

Other company names and product names in this document are the trademarks or registered trademarks of their respective companies.

Copyrights

- Microsoft product screen shots used with permission from Microsoft.

Introduction

Thank you for purchasing the IO-Link Master Unit GD-ILM16C-MLP and GD-ILM16E-MLP.
This manual contains the information required when using *Ethernet & Modbus/TCP* as the host network.
Please read this manual and make sure you understand the functionality and performance of the product before you attempt to build a system. After reading the manual, keep it in a safe and accessible location for further reference.

- Trademarks and registered trademarks appearing in this manual are the property of their respective owners.

Intended Audience

This manual is intended for the following personnel, who must also have knowledge of electrical systems (an electrical engineer or the equivalent).

- Personnel in charge of introducing FA systems.
- Personnel in charge of designing FA systems.
- Personnel in charge of installing and maintaining FA systems.
- Personnel in charge of managing FA systems and facilities.

Applicable Products

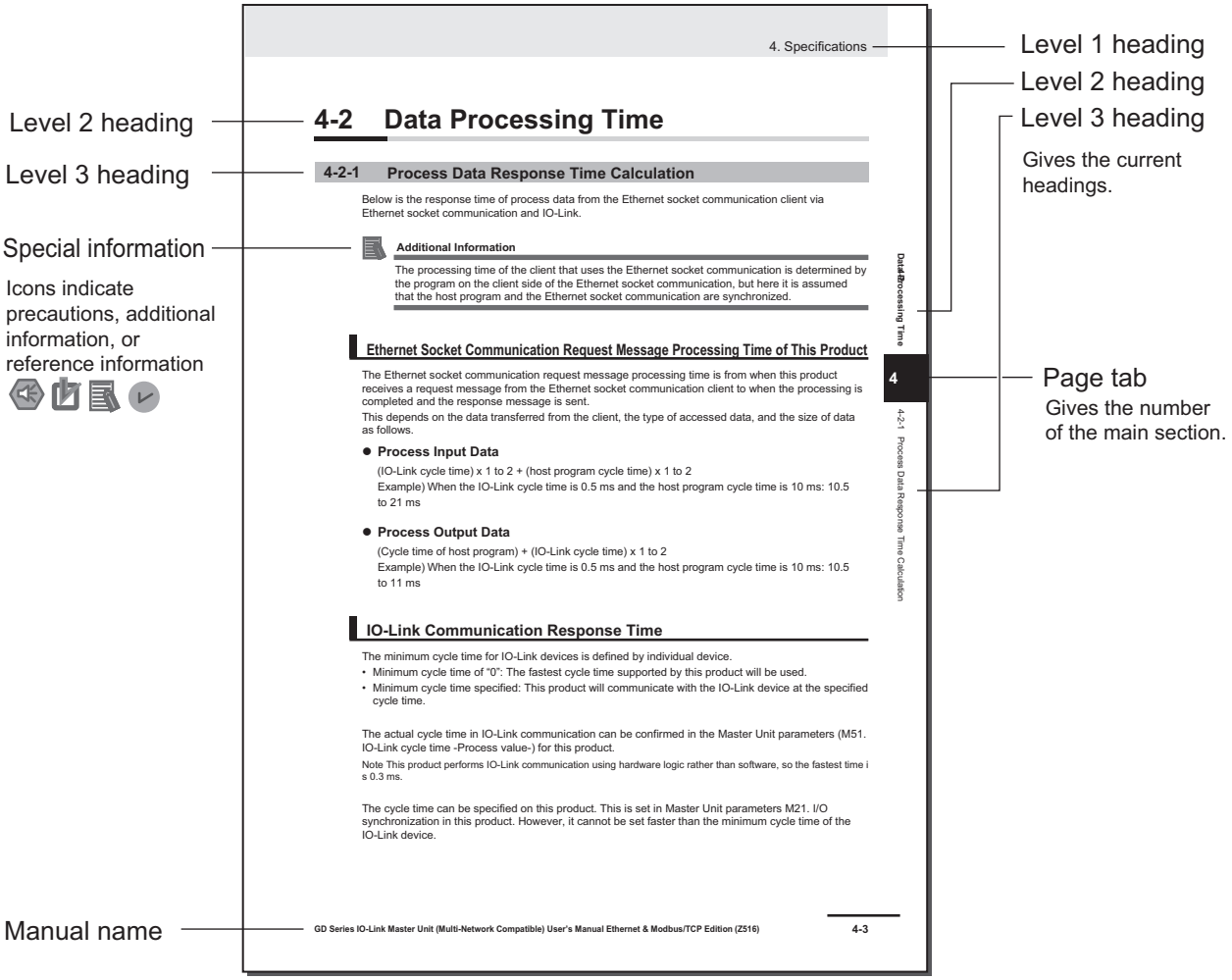
This manual covers the following product.

- GD-series IO-Link Master Unit
  - GD-ILM16C-MLP
  - GD-ILM16E-MLP

Manual Structure

Page Structure

The following page structure is used in this manual.



This illustration is provided only as a sample. It may not literally appear in this manual.

Special Information

Special information in this manual is classified as follows:

Important

This summarizes particularly important points about its performance, including the things to be observed during operation and the advice on usage.

Additional Information

Additional information to read as required.

This information is provided to increase understanding or make operation easier.

Version Information

Information on differences in specifications and functionality for products with different unit versions and for different versions of the Support Software is given.

Notations Used in this Manual

These are the notations used in this manual.

Precautions for Correct Use

This indicates particularly important points to observe during operation.

Additional Information

This information is useful for operation.

Sections in this Manual

CONTENTS

Section 1 Overview

Section 2 Initial Setting of This Product with Regard to Ethernet & Modbus/TCP

Section 3 Communication

Section 4 Specifications

Section 5 Troubleshooting

Appendices

Index

Terms and Conditions Agreement

Warranty, Limitations of Liability

Warranties

- **Exclusive Warranty**

  Omron's exclusive warranty is that the Products will be free from defects in materials and workmanship for a period of twelve months from the date of sale by Omron (or such other period expressed in writing by Omron). Omron disclaims all other warranties, express or implied.

- **Limitations**

  OMRON MAKES NO WARRANTY OR REPRESENTATION, EXPRESS OR IMPLIED, ABOUT NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE OF THE PRODUCTS. BUYER ACKNOWLEDGES THAT IT ALONE HAS DETERMINED THAT THE PRODUCTS WILL SUITABLY MEET THE REQUIREMENTS OF THEIR INTENDED USE.

  Omron further disclaims all warranties and responsibility of any type for claims or expenses based on infringement by the Products or otherwise of any intellectual property right.

- **Buyer Remedy**

  Omron's sole obligation hereunder shall be, at Omron's election, to (i) replace (in the form originally shipped with Buyer responsible for labor charges for removal or replacement thereof) the non-complying Product, (ii) repair the non-complying Product, or (iii) repay or credit Buyer an amount equal to the purchase price of the non-complying Product; provided that in no event shall Omron be responsible for warranty, repair, indemnity or any other claims or expenses regarding the Products unless Omron's analysis confirms that the Products were properly handled, stored, installed and maintained and not subject to contamination, abuse, misuse or inappropriate modification. Return of any Products by Buyer must be approved in writing by Omron before shipment. Omron Companies shall not be liable for the suitability or unsuitability or the results from the use of Products in combination with any electrical or electronic components, circuits, system assemblies or any other materials or substances or environments. Any advice, recommendations or information given orally or in writing, are not to be construed as an amendment or addition to the above warranty.

  See https://www.omron.com/global/ or contact your Omron representative for published information.

Limitation on Liability; Etc

OMRON COMPANIES SHALL NOT BE LIABLE FOR SPECIAL, INDIRECT, INCIDENTAL, OR CONSEQUENTIAL DAMAGES, LOSS OF PROFITS OR PRODUCTION OR COMMERCIAL LOSS IN ANY

WAY CONNECTED WITH THE PRODUCTS, WHETHER SUCH CLAIM IS BASED IN CONTRACT, WARRANTY, NEGLIGENCE OR STRICT LIABILITY.

Further, in no event shall liability of Omron Companies exceed the individual price of the Product on which liability is asserted.

Application Considerations

Suitability of Use

Omron Companies shall not be responsible for conformity with any standards, codes or regulations which apply to the combination of the Product in the Buyer's application or use of the Product. At Buyer's request, Omron will provide applicable third party certification documents identifying ratings and limitations of use which apply to the Product. This information by itself is not sufficient for a complete determination of the suitability of the Product in combination with the end product, machine, system, or other application or use. Buyer shall be solely responsible for determining appropriateness of the particular Product with respect to Buyer's application, product or system. Buyer shall take application responsibility in all cases.

NEVER USE THE PRODUCT FOR AN APPLICATION INVOLVING SERIOUS RISK TO LIFE OR PROPERTY OR IN LARGE QUANTITIES WITHOUT ENSURING THAT THE SYSTEM AS A WHOLE HAS BEEN DESIGNED TO ADDRESS THE RISKS, AND THAT THE OMRON PRODUCT(S) IS PROPERLY RATED AND INSTALLED FOR THE INTENDED USE WITHIN THE OVERALL EQUIPMENT OR SYSTEM.

Programmable Products

Omron Companies shall not be responsible for the user's programming of a programmable Product, or any consequence thereof.

Disclaimers

Performance Data

Data presented in Omron Company websites, catalogs and other materials is provided as a guide for the user in determining suitability and does not constitute a warranty. It may represent the result of Omron's test conditions, and the user must correlate it to actual application requirements. Actual performance is subject to the Omron's Warranty and Limitations of Liability.

Change in Specifications

Product specifications and accessories may be changed at any time based on improvements and other reasons. It is our practice to change part numbers when published ratings or features are changed, or when significant construction changes are made. However, some specifications of the Product may

be changed without any notice. When in doubt, special part numbers may be assigned to fix or establish key specifications for your application. Please consult with your Omron's representative at any time to confirm actual specifications of purchased Product.

Errors and Omissions

Information presented by Omron Companies has been checked and is believed to be accurate; however, no responsibility is assumed for clerical, typographical or proofreading errors or omissions.

Statement of security responsibilities for assumed use cases and against threats

OMRON SHALL NOT BE RESPONSIBLE AND/OR LIABLE FOR ANY LOSS, DAMAGE, OR EXPENSES DIRECTLY OR INDIRECTLY RESULTING FROM THE INFECTION OF OMRON PRODUCTS, ANY SOFTWARE INSTALLED THEREON OR ANY COMPUTER EQUIPMENT, COMPUTER PROGRAMS, NETWORKS, DATABASES OR OTHER PROPRIETARY MATERIAL CONNECTED THERETO BY DISTRIBUTED DENIAL OF SERVICE ATTACK, COMPUTER VIRUSES, OTHER TECHNOLOGICALLY HARMFUL MATERIAL AND/OR UNAUTHORIZED ACCESS.

It shall be the users sole responsibility to determine and use adequate measures and checkpoints to satisfy the users particular requirements for (i) antivirus protection, (ii) data input and output, (iii) maintaining a means for reconstruction of lost data, (iv) preventing Omron Products and/or software installed thereon from being infected with computer viruses and (v) protecting Omron Products from unauthorized access.

Safety Precautions

Refer to the *GD Series IO-Link Master Unit (Multi-Network Compatible) User's Manual Common Edition (Cat. No. Z512)* (hereinafter abbreviated as the *Common Edition*) for safety precautions.

Precautions for Safe Use

Refer to the *GD Series IO-Link Master Unit (Multi-Network Compatible) User's Manual Common Edition (Cat. No. Z512)* (hereinafter abbreviated as the *Common Edition*) for precautions for safe use.

Precautions for Correct Use

Refer to the *GD Series IO-Link Master Unit (Multi-Network Compatible) User's Manual Common Edition (Cat. No. Z512)* (hereinafter abbreviated as the *Common Edition*) for precautions for use.

Regulations and Standards

Refer to the *GD Series IO-Link Master Unit (Multi-Network Compatible) User's Manual Common Edition (Cat. No. Z512)* (hereinafter abbreviated to as the *Common Edition*) for regulations and standards.

Checking the Included Items

Refer to the *GD Series IO-Link Master Unit (Multi-Network Compatible) User's Manual Common Edition (Cat. No. Z512)* (hereinafter abbreviated as the *Common Edition*) for included items.

Related Manuals

Manuals related to this manual are as follows. Reference them as needed.

Manual number	Manual name	Details
Z512	GD-series IO-Link Master Unit (Multi-Network Compatible) User's Manual Common Edition GD-ILM□□□-MLP (Referred to herein as the *Common Edition*.)	Describes this product's common functions and performance as well as operation. Be sure to read these documents.
Z518	GD Series Wave Inspire HUB Operation Manual	Provides necessary information for operating OMRON Support Software (Wave Inspire HUB) for the GD Series.

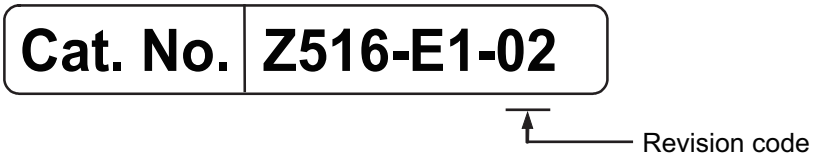
Terminology List

This explains the terminology used in this document.

Terminology	Description
Modbus/TCP	It is a widely-used standardized protocol for industrial Ethernet that implements Modbus as a control protocol in the application layer on the TCP/IP protocol.
Modbus/TCP master	PLC and HMI of the master station of Modbus/TCP communication, etc.
Modbus/TCP communication	Request messages are issued from the Modbus/TCP master, and Holding registers within this product are read and written. When this happens, depending on the Holding register number, this product executes ISDU communication (of IO-Link) with the IO-Link device.
Function code	It is a code that indicates the request type requested by the Modbus/TCP master. The function codes supported by this product are as follows. 03 h: Read Multiple Holding Registers 06 h: Write Single Holding Register 10 h: Write Multiple Holding Registers 17 h: Read/Write Multiple Holding Registers
Holding register	A memory that allows reading and writing. An address (referred to as *register number*) is assigned to each word (16 bits).
Register number	This is the number of the holding register specified on the software such as PLC, HMI and OPC server with a built-in Modbus communication driver. In this product, the register numbers are 400001 to 465536. They are shown in decimal. The *4* in the highest digit of the register number indicates the holding register. The remaining digits are the data register number (decimal number) plus *1*. Therefore, (data register number) = (register number) -400001.
Data register number	This is the register number used in the actual Modbus communication frame. It is specified in hexadecimal on the communication frame.
ISDU Access Library	This is the library area for ISDU communication with IO-Link devices. By registering the index/sub-index number of the IO-Link device to be accessed in this area, it becomes possible to read/write up to 128 types of parameters at once using a simple trigger. It can also be used together with recipe functions such as HMI.
Modbus address monitor registration	Registers the register addresses that are frequently read from the Modbus/TCP master into the library. Up to 125 addresses can be registered, which are output to the output area of the library. By reading the library output area from the Modbus/TCP master, it becomes possible to read up to 125 words at once using a single read command for data from multiple non-consecutive register addresses, achieving improved communication efficiency.
Socket communication	Indicates product unique protocol communication within Ethernet socket communication. This communication designates the IP address and port number (2001 on this product) to send and receive data with UDP/IP or TCP/IP.

Revision History

A manual revision code appears as a suffix to the catalog number on the front and back covers of the manual.



Revision code	Date	Revised content
01	February 2026	Original production
02	April 2026	Corrected mistakes.

1

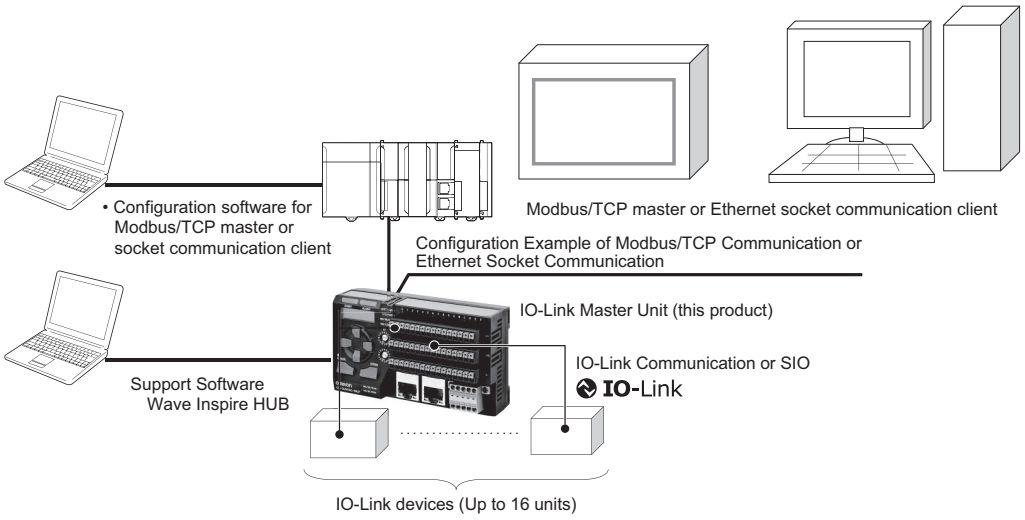
Overview

This section provides an overview of the use of *Ethernet & Modbus/TCP* as the host network.

1-1 Product Overview

When the network type is set to *Ethernet & Modbus/TCP*, this product can read and write to the holding registers in this product from the Modbus/TCP master. It also supports sending and receiving data from clients that use Ethernet socket communication.
In either case, the user can read and write the IO-Link process data of this product, digital I/O, settings of this product, and settings of IO-Link devices via Modbus/TCP or Ethernet socket communication.

Configuration Example of Modbus/TCP Communication or Ethernet Socket Communication and IO-Link System



Features

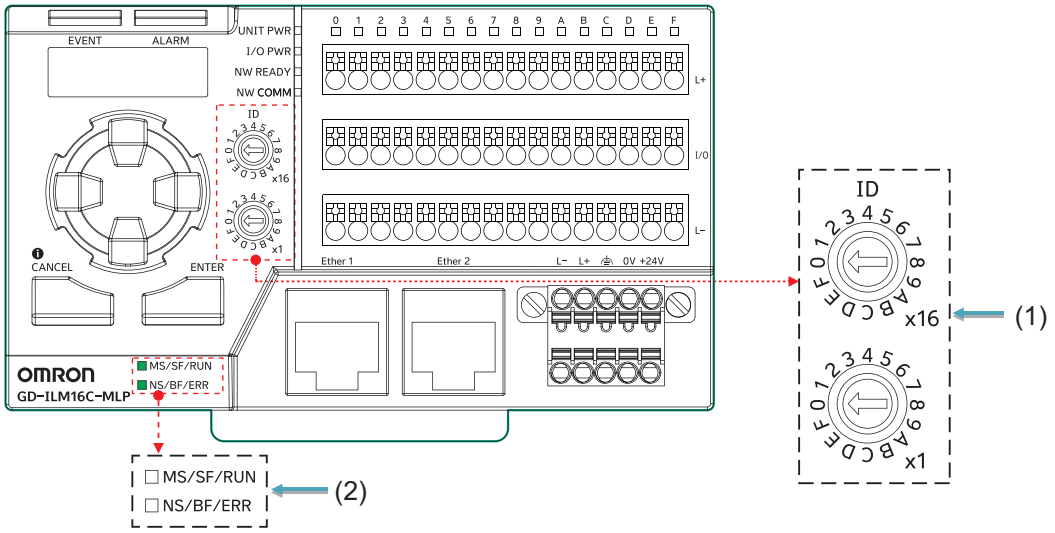
- The host network type can be switched among the following depending on the setting.
  a) EtherNet/IP (default value)
  b) Ethernet & Modbus/TCP
  c) CC-Link IE Field Basic
- The Modbus/TCP communication and socket communication (TCP/UDP) can be used with the *Ethernet & Modbus/TCP* network type.
- Enables efficient access to the parameters (ISDU communication) of the IO-Link device.
  By registering the index/sub-index number of the IO-Link device to be accessed in the library inside of this product, it becomes possible to read/write up to 128 types of parameters at once using a simple trigger. It can also be used together with recipe functions such as HMI.
- Enables efficient communication with the Modbus/TCP master.
  By registering the register address numbers that are frequently read from the Modbus/TCP master in the library inside this product, it becomes possible to read up to 125 words at once using a single read command for data from multiple non-consecutive register addresses.
- If the CC-Link IE Field Basic is selected as the network type, it becomes possible to use both the CC-Link IE Field Basic and Modbus/TCP communication at the same time.

The I/O data used for control can be accessed via cyclic communication of the PLC's CC-Link IE Field Basic, and the parameters of the IO-Link device can be accessed directly via Modbus/TCP communication of the HMI.
It supports complex apps without a PLC ladder program.

1-2 Part Names and Functions

1-2-1 Part Names and Functions

The part names and functions after assembling the terminal block, when using *Ethernet & Modbus/TCP* as the host network, are as follows.
Refer to the Common Edition for other part names and functions.



Note The above figure is for GD-ILM16C-MLP.
The same applies to GD-ILM16E-MLP.

1. **IP address lowest digit setting switch**
   Sets the lowest digit of the IP address value (the “n” in 192.168.250.n). The top value is multiplied by 16 and added to the bottom value.
2. **MS/SF/RUN LED, NS/BF/ERR LED**
   Displays the host network status.
   For Ethernet & Modbus/TCP, it is as follows.

   a) MS/SF/RUN
   Off: Network chip is not active or power is off.
   Blinking: Network chip is working.
   Lit: Network chip is ready.

   b) NS/BF/ERR
   Off: Ethernet cable is not connected or power is off.
   Blinking: Ethernet cable is connected, but communication has not been established.
   Lit: Ethernet communication has been established. This may also indicate that communication is ready.

1-3 Overview of Specifications and Functions for Ethernet & Modbus/TCP

This section outlines the specifications and functions when *Ethernet & Modbus/TCP* is selected as the network type.

1-3-1 For Modbus/TCP Communication

Specifications

Item	Details
Corresponding protocols	Modbus/TCP, Modbus/UDP
Corresponding function code	03 h, 06 h, 10 h, 17 h*1
Corresponding Modbus data area	Holding register Register number: 400001 to 465536 (data address: 0 to 65535)
Master/server	Operates as a Modbus server device
Port number	502 (fixed) Both Modbus/TCP and Modbus/UDP are supported by the same port number.
Ethernet	IEEE 802.3u, 10BASE-T/100BASE-TX automatic identification
No. of connectable units	Number of units that can be connected from one Modbus/TCP master: It depends on the specifications of the Modbus/TCP master.
Number of simultaneous connections	Modbus/TCP: 2 Modbus/UDP: 4

*1.

Function code (FC)	Details		Number of words that can be accessed with one function code
03 h	Read holding registers	Read Multiple Holding Registers	1 to 125 words
06 h	Write single register	Write Single Holding Register	1 word
10 h	Write multiple registers	Write Multiple Holding Registers	1 to 123 words
17 h	Read/Write multiple registers	Read/Write Multiple Holding Registers	Read: 1 to 125 words
			Write: 1 to 121 words

Functional Overview

Functions	Details
Inputting/outputting IO-Link process data	Specify the corresponding register number in unsigned integer and perform reading/writing. **Note** The maximum input/output of IO-Link process data is 32 bytes (16 words) per port for both input and output. It directly accesses process data in IO-Link devices via Modbus/TCP communication. It has no relation with the Master Unit parameter M40/41 (*M40. Process input data words allocation* and *M41. Process output data words allocation*).
Digital input/output	
Allocated status data confirmation, allocated operation data instruction	
Settings of this product	

Functions	Details
Efficient access to the parameters (ISDU communication) of the IO-Link device	By registering the index/sub-index number of the IO-Link device to be accessed in the library inside of this product, it becomes possible to read/write up to 128 types of parameters at once using a simple trigger.
Efficient communication with the Modbus/TCP master	By registering the register address numbers that are frequently read from the Modbus/TCP master in the library inside this product, it becomes possible to read up to 125 words at once using a single read command for data from multiple non-consecutive register addresses.
Remote control of this product	Specify the corresponding register number in unsigned integer and perform writing/reading.
Network time function	Specify the corresponding register number in unsigned integer and perform reading/writing.

1-3-2 For Ethernet Socket Communication

Specifications

Item	Details
Supported socket communication	TCP/IP, UDP/IP
Server/client	Server
Socket communication port number	2001 (decimal number) Fixed (common to both TCP/IP and UDP/IP)
Number of simultaneous connections	TCP/IP: 2 pcs UDP/IP: 4 pcs
IP address	Fixing method
Subnet mask	255.255.255.0
Default gateway	None or 0.0.0.0

Functional Overview

Functions	Details
Inputting/outputting IO-Link process data	Reads process input data, digital input and allocated status data, and writes process output data, digital I/O and allocated operation data with a data access request.
Digital input/output	
Inputting allocated status data, outputting allocated operation data	Reads up to 32 words in total for all ports to be used, 1 word for digital input, and 1 word for allocated status data in a batch. Writes up to 32 words in total for all ports to be used, 1 word for digital output, and 1 word for allocated operation data in a batch. The allocation of process data for each port is based on the Master Unit parameter M40/41 (*M40. Process input data words allocation* and *M41. Process output data words allocation*) of this product.
IO-Link device setting	Specify the index number or sub-index number and execute the parameter access by writing with a data access request.
Settings of this product	Specify the Master Unit parameter number and execute the parameter access by writing with a data access request.

1-4 Inputting/Outputting IO-Link Devices and Process Data

This section explains how to input/output process data of IO-Link devices connected to the product.

The input/output methods are as below.

Method	Description		Reference
1) Input/output via cyclic communication*1	Modbus/TCP communication	Specify the corresponding register number (00089 to 00802) in unsigned integer and perform writing or reading. • Input: Up to 32 bytes/port • Output: Up to 32 bytes/port	*3-2 Modbus/TCP Communication* on page 3-4
		Input (monitor) only: Discontinuous register numbers of this product can be registered as a library in advance and read them in a batch (registered register number library function: depends on the corresponding register numbers 03872 to 04123).	
	Ethernet Socket Communication	Reads process input data, digital input or allocated status data, and writes process output data, digital I/O or allocated operation data with a data access request • Input: Total of 32 words (64 bytes) over all ports • Output: Total of 32 words (64 bytes) over all ports	*3-3 Ethernet Socket Communication* on page 3-33
2) Confirmation with front panel controls	Confirmation is done with product front panel operation.		Common Edition *Section 5 Front Panel Operations*

*1. For Modbus/TCP communication, it may be possible to perform cyclic communication (automatic communication) depending on the master.

1-5 Setting/Monitoring Product Master Unit Parameters

This section describes how to set/monitor product Master Unit parameters.

Methods for setting/monitoring are as shown below.

Method	Description		Reference
1) Setting with front panel controls	Setting is done with product front panel operation.		Common Edition *Section 5 Front Panel Operations*
2) Setting/monitoring via acyclic communication	Modbus/TCP communication	Specify the corresponding register number (05500 to 06565) in unsigned integer and perform writing or reading.	*3-2 Modbus/TCP Communication* on page 3-4
		Monitor only: Discontinuous register numbers of this product can be registered as a library in advance and read them in a batch.	
	Ethernet Socket Communication	Execute parameter access (read/write), specify the Master Unit parameter of this product by unit specification, and specify the Master Unit parameter number or target number with a data access request.	*3-3 Ethernet Socket Communication* on page 3-33

1-6 IO-Link Device Service Data Setting/ Monitoring

This section explains how to set and monitor service data inside IO-Link devices connected to the product.

Methods for setting/monitoring are as shown below.

Method	Description		Reference
1) Setting with front panel controls	Setting is done with product front panel operation.		Common Edition *Section 5 Front Panel Operations*
2) Setting/monitoring via acyclic communication	Modbus/TCP communication	Specify the corresponding register number (00603 to 03871) in unsigned integer and perform writing or reading (ISDU access library function).	*3-2 Modbus/TCP Communication* on page 3-4
		Monitor only: Discontinuous register numbers of this product can be registered as a library in advance and read them in a batch.	
	Ethernet Socket Communication	Execute parameter access (read/write), specify the port number by unit specification, specify the index number / sub-index number with a data access request.	*3-3 Ethernet Socket Communication* on page 3-33

1-7 Basic Procedures

The process from installing and setting this product through beginning Modbus/TCP and IO-Link communication is indicated below.
Decide in advance whether to use Modbus/TCP or Ethernet socket communication.

1-7-1 For Modbus/TCP Communication

Procedure	Details	References
Prior confirmation	• Prepare an Ethernet connection cable and Switching Hub	Checking the Included Items Manual of the Modbus/TCP master being used
Hardware installation and wiring	Setting the lowest digit (n) of the IP address (192.168.250.n) of this product with the rotary switch on the front of this product	*2-1 Setting the Last Digit of the IP Address of This Product* on page 2-2
Initial Settings for Front Panel Controls	Set *Network Type* to *Ethernet & Modbus/TCP*	*2-3 Initial Settings for Front Panel Controls* on page 2-5
Operation to communicate with the Modbus/TCP master	Confirming that *Network Type* setting is *Ethernet & Modbus/TCP*	*3-1-1 Confirming the Network Type Setting* on page 3-2
	Setting the Modbus/TCP master	• *3-1 Initial Operation Required for This Product to Perform Modbus/TCP Communication or Ethernet Socket Communication* on page 3-2 • Manual of the Modbus/TCP master being used
	In the Modbus/TCP master, specify the register number of this product in unsigned integer to create a Modbus/TCP communication program or set for automatic input/output by the communication driver	
This product's Parameter settings	Using front panel controls	Common Edition
	For a communication program from Modbus/TCP master or automatic input/output by communication driver	Manual of the Modbus/TCP master being used
IO-Link device parameter setting	Using front panel controls	Common Edition
	For a communication program from Modbus/TCP master or automatic input/output by communication driver	Manual of the Modbus/TCP master being used
Starting communication	Start system (power ON)	
	Start Modbus/TCP communication (Including PING test) Starting IO-Link communication	Manual of the Modbus/TCP master being used
Checking operation	Verifying the display of the Modbus/TCP master and this product or IO-Link device	• Manual of the Modbus/TCP master being used • Common Edition • Each IO-Link device manual
	Verifying read/write data between this product and an IO-Link device	Each IO-Link device manual
	Verifying data read/write between Modbus/TCP master and this product	Manual of the Modbus/TCP master being used
Troubleshooting	Verifying the display of the Modbus/TCP master and this product or IO-Link device	*Section 5 Troubleshooting* on page 5-1

1-7-2 For Ethernet Socket Communication

Procedure	Details	References
Prior confirmation	• Prepare an Ethernet connection cable and Switching Hub	Checking the Included Items Client manual of the Ethernet socket communication
Hardware installation and wiring	Setting the lowest digit (n) of the IP address (192.168.250.n) of this product with the rotary switch on the front of this product	*2-1 Setting the Last Digit of the IP Address of This Product* on page 2-2
Initial Settings for Front Panel Controls		Common Edition
Operation to communicate with the Modbus/TCP master	Confirming that *Network Type* setting is *Ethernet & Modbus/TCP*	*3-1-1 Confirming the Network Type Setting* on page 3-2
	Client settings of Ethernet socket communication	• *3-1 Initial Operation Required for This Product to Perform Modbus/TCP Communication or Ethernet Socket Communication* on page 3-2 • Client manual of the Ethernet socket communication
	Creating communication programs (open processing, sending/receiving data, closing) on the client side via Ethernet socket communication	
This product's Parameter settings	Using front panel controls	Common Edition
	When using a communication program from the Ethernet client	Client manual of the Ethernet socket communication
IO-Link device parameter setting	Using front panel controls	Common Edition
	When using a communication program from the Ethernet client	Client manual of the Ethernet socket communication
Starting communication	Start system (power ON)	
	Starting Ethernet socket communication (Including PING test) Starting IO-Link communication	Client manual of the Ethernet socket communication
Checking operation	Verifying the display of the Ethernet client and this product / IO-Link device	• Client manual of the Ethernet socket communication • Common Edition • Each IO-Link device manual
	Verifying read/write data between this product and an IO-Link device	Each IO-Link device manual
	Verifying data read/write between the Ethernet client and this product	Client manual of the Ethernet socket communication
Troubleshooting	Verifying the display of the Ethernet client and this product / IO-Link device	*Section 5 Troubleshooting* on page 5-1

Additional Information

Product Master Unit parameters and connected IO-Link device settings can be changed by either of the following methods.
The applications of each are as follows.

Method	Main application	References
1) Unit front operation	Startup time or Maintenance	• *2-3 Initial Settings for Front Panel Controls* on page 2-5
2) Communication from Modbus/TCP master or Ethernet client	Controlled operation	• *3-2 Modbus/TCP Communication* on page 3-4 • *3-3 Ethernet Socket Communication* on page 3-33 • *A-1 List of Product Master Unit Parameters* on page A-2 • *A-2 Socket Communication Program Example* on page A-27

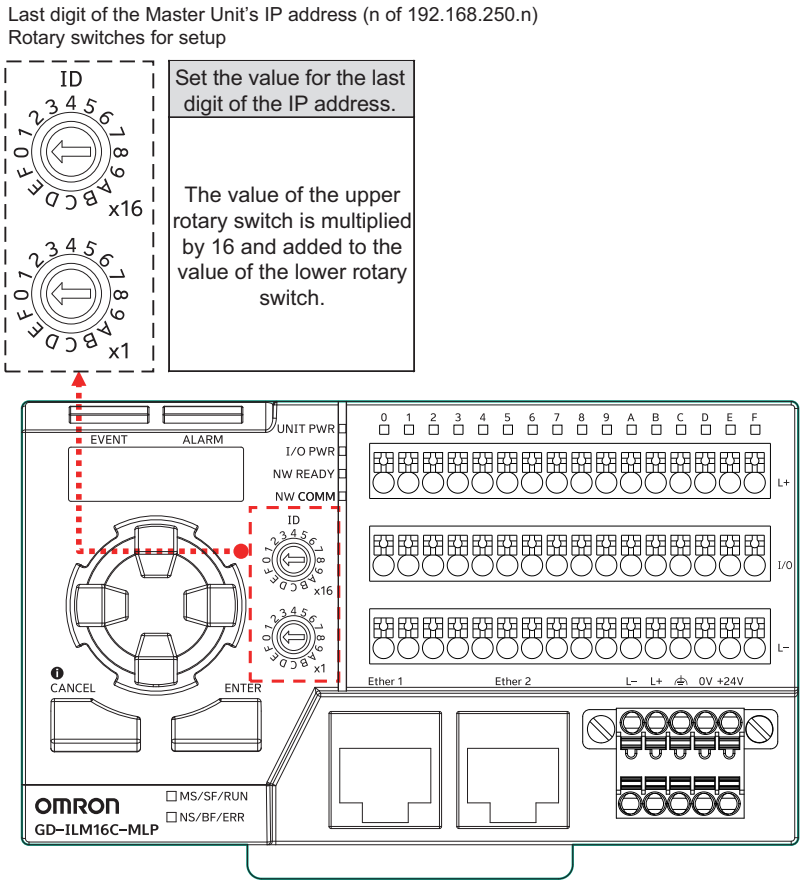
2

Initial Setting of This Product with Regard to Ethernet & Modbus/TCP

Describes initial setting for this product's IP address setting, communication connection, and front panel operation.

2-1 Setting the Last Digit of the IP Address of This Product

Before turning on the power, use the front rotary switch to set the lowest digit (n of 192.168.250.n) of the IP address of this product as a node for host network Modbus/TCP or Ethernet socket communication.



Note The above figure is for GD-ILM16C-MLP. The same applies to GD-ILM16E-MLP.

The top value is multiplied by 16 and added to the bottom value.
If the value is 250, set the upper switch to 15 and the lower switch to 10.

The IP address is initially set to 192.168.250.n (n is the value of the front rotary switch).
To change it from 192.168.250.n, set the Master Unit parameter *M82. IP address* by operating the front panel controls. Turning the front rotary switch while the power is on automatically switches to *M82. IP address*.

The default gateway value is 0.0.0.0.
When a change is required, set the *M80. Default gateway* Master Unit parameter with product front panel controls.
The subnet mask default value is 255.255.255.0.
When a change is required, set the *M81. Subnet mask* Master Unit parameter with product front panel controls.

When changing the IP address, default gateway, or subnet mask with product front panel controls, refer to *5-1-1 Overview of Display Operations* and *5-1-4 Master Unit Parameter List with Product Front Panel Operations* of the Common Edition.

Note that static IP is the only way to set the product IP address (the IP address cannot be acquired via BOOTP or DHCP).

2-2 Ethernet Connection

Shown here is the Ethernet connection configuration when Modbus/TCP communication or Ethernet socket communication is used as the host network.

2-2-1 Connector and Cable

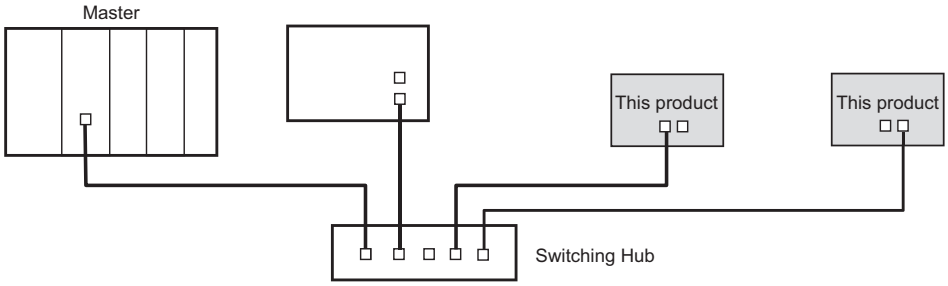
The Ethernet cable may be connected to either of the RJ45 connectors at left or right of the product.
For the cable, use an Ethernet cable that complies with the standard (100BASE-TX, 10BASE-T).
For details, refer to the manual of the client of Modbus/TCP master or Ethernet socket communication.

2-2-2 Ethernet Connection Status

Topologies capable of Modbus/TCP or Ethernet socket communication are star or line connections.

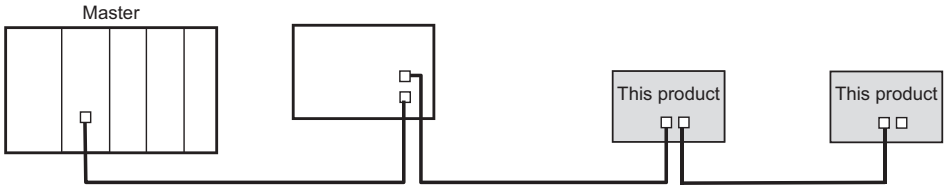
● Star Configuration

The star configuration network enables connection of various devices from a Switching Hub.
Connection to either Ether1 or Ether2 of the product is possible.



● Linear Bus Configuration

A linear bus configuration type network is a connection format in which devices are connected by daisy chain.
It doesn't require a Switching Hub and shortens the required length of LAN cables.
The upstream device (toward the master) and downstream device can be connected to either Ether1 or Ether2 on this product.



For details, refer to the manual of the client of Modbus/TCP master or Ethernet socket communication.

2-3 Initial Settings for Front Panel Controls

Shown here is the initial setting method via product front panel operation when using *Modbus/TCP* as the host network.

2-3-1 Powering Up the Modbus/TCP Master

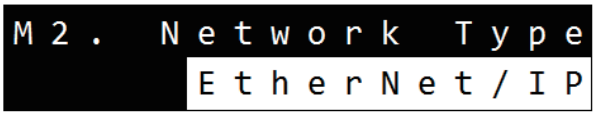
When available, apply power to the Modbus/TCP master station in advance.
This makes it possible to check whether the master is connected correctly.

2-3-2 Language Selection

After turning the product on for the first time, select the language setting.

2-3-3 Network Type Setting

Continuing, the window automatically switches to the network type setting screen shown below.



The factory default network type setting is *EtherNet/IP*.
When setting the host industrial network to *Ethernet & Modbus/TCP*, change the network type setting here to *Ethernet & Modbus/TCP*.
Regarding *M2. Network types*, refer to *5-1-4 Master Unit Parameter List with product Front Panel Operations* of the Common Edition.
Press the ↑ button three times. *Ethernet & Modbus/TCP* will be displayed as below (scroll display).

Press the **ENTER** button. Wait for about twenty seconds. There will be an automatic reboot.

After the reboot, the process data display window will appear (if language selection is complete).

The following process data display is an example of the start window when turning the power on after selecting the language.

Port number

Refer to *5-1 Front Panel Operations* of the Common Edition for further information about front panel operation.

3

Communication

This section describes the initial settings for Modbus/TCP communication or Ethernet socket communication, the configuration of the data to be handled, and communication examples.

3-1 Initial Operation Required for This Product to Perform Modbus/TCP Communication or Ethernet Socket Communication

This section describes the operations required on the Modbus/TCP master or Ethernet client for this product to perform Modbus/TCP communication or Ethernet socket communication, and the operations required for IO-Link communication with the IO-Link device.

For details, refer to the Modbus/TCP master manual or the Ethernet client manual.
Contact the manufacturer of the Modbus/TCP master regarding the Modbus/TCP configuration software.

3-1-1 Confirming the Network Type Setting

Confirm that the Master Unit parameter *M2. Network type* is set to *Ethernet & Modbus/TCP*.
It is set in the initial settings using the front panel. Refer to *2-3-3 Network Type Setting* on page 2-5 for further information.

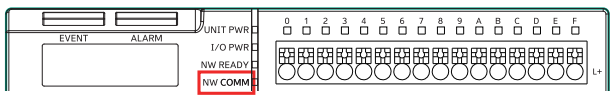
Regarding *M2. Network types* of Master Unit parameters, refer to *5-1-4 Master Unit Parameter List with Product Front Panel Operations* of the Common Edition.

3-1-2 Modbus/TCP Communication Settings Between Modbus/TCP Master and This Product

The required settings differ depending on the Modbus/TCP master. Refer to the manual of the Modbus/TCP master.

Additional Information

- When writing setting details to the Modbus/TCP master, it must be rebooted.
- Changes in the IP address, subnet mask or default gateway of this product will be updated after the power is turned back on.
- If the NW COMM LED on the front panel does not light up in blue-green (indicating normal communication) after the settings are complete, recheck the settings made on the Modbus/TCP master and the settings of this product (IP address, subnet mask, default gateway).



- If communication is not possible even after all the confirmation above, refer to the manuals for each Modbus/TCP master.

3-1-3 Ethernet Socket Communication Settings Between Ethernet Client and This Product

The required settings differ depending on the Ethernet client. Refer to the manual of the Ethernet socket communication client.

3-2 Modbus/TCP Communication

This section shows the overview and detailed list of the Holding registers in the product when accessing it from the Modbus/TCP master via Modbus/TCP communication.
Use the function code (03 h, 06 h, 10 h or 17 h) in the Modbus/TCP request message to access the Holding register of this product according to the function.

3-2-1 Overview of Register Number

Holding register (decimal)	Category	Details
00000 … 00067	Operation of this product	This area is for operating this product from UI devices such as HMI (display) and SCADA. It consists of reading the character strings displayed on the product display, pressing buttons (up, down, left, right, Enter, Cancel), switching the product menu and locking the operation.
00068 … 00070	This product's time	This is the time that this product counts internally. Set the time to display when an error or event occurs. You can also check the Modbus / TCP communication cycle by reading and comparing the time of this product's time multiple times.
00071 … 00088	Status/operation data	It consists of details related to errors, events and resetting various statuses.
00089 … 00802	I/O	I/O input/output area. Input: Process data 16, IO-Link process data 32 bytes/channel Output: Process data 16, IO-Link process data 32 bytes/channel
00603 … 03871	ISDU Access Library	This is the library area for ISDU communication with IO-Link devices. By registering the index/sub-index number of the IO-Link device to be accessed in this area, it becomes possible to read/write up to 128 types of parameters at once using a simple trigger. It can also be used together with recipe functions such as HMI.
03872 … 04123	Modbus address monitor registration	Registers the register addresses that are frequently read from the Modbus/TCP master into the library. Up to 125 addresses can be registered, which are output to the output area of the library. By reading the library output area from the Modbus/TCP master, it becomes possible to read up to 125 words at once using a single read command for data from multiple non-consecutive register addresses, achieving improved communication efficiency. The library output area is read-only.
04124 … 04126	Backing up	This area backs up the value of a specific register address to the non-volatile memory area of this product. • Target ISDU Access Library (0070 to 03871) Modbus address monitor registration (3872 to 3997) When the power is turned back on, the value at the time of backup trigger will be restored.
04127 … 04190	External Device Access Status	The values of the remote registers RWr and RWw of the CC-Link IE Field Basic master station are monitored. **Note** When **CC-Link IEF Basic** is selected from the product menu **M2. Network**, two protocols of CC-Link IEF Basic and Modbus/TCP can be used simultaneously.
05500 … 06565	Master Unit Parameters	This is the parameter area of this product.
07000 … 07031	User area	This is the area that can be freely used by the user.
07032 … 08311	Basic Information of Connected Device	This area stores information (vendor name, device model, user tag) of the IO-Link device connected to this product.

Additional Information

Usage Holding register range by purpose

Purpose	Holding register range (decimal)
Accessing I/O (IO-Link process data).	Process input: 00090 to 00345 Process output: 00347 to 00602
Accessing I/O (digital).	Digital input: 00089 Digital output: 00346
Checking the error and event status.	Status data: 00071 to 00072 Error: 00073 to 00075 Event: 00076 to 00087
Resetting the status of errors, events, etc.	00088
Accessing the parameters (ISDU) of the IO-Link device connected to this product.	00603 to 03871
Changing the parameters of this product.	05500 to 06565
Improving the communication efficiency between the Modbus/TCP master and this product.	03872 to 04123
Retaining the parameters of the ISDU access library or Modbus address monitor registration even after this product is turned off.	04124 to 04126
Remotely controlling the screen of this product from a UI device such as HMI/SCADA.	00000 to 00067
Setting the time to display when an error or event occurs.	00068 to 00070
Checking the Modbus/TCP communication cycle.	
Checking the vendor name, device model or user tag of the IO-Link device connected to this product.	07032 to 08311
Setting an operation authority interlock address on this product when accessing this product from multiple Modbus/TCP masters.	07000 to 07031

3-2-2 Detailed List of Register Numbers

The details of the Holding register of this product specified via Modbus/TCP communication are described below in decimal.
On the communication frame, the Holding register number is specified in hexadecimal.

Operation Of This Product

Category	Data name	Holding register (decimal)	R/W	Bit															
				15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Character string displayed on the display	1st line	00000 … 00031	R	Up to 32 words															
	2nd line	00032 … 00063		Up to 32 words															
Button operation		00064	R/W	→: 0001 h, ←: 0002 h, ↑: 0004 h, ↓: 0008 h, CANCEL: 0010 h, ENTER: 0020 h															

Category	Data name	Holding register (decimal)	R/W	Bit 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Switching the displayed screen	Specify the screen to switch to	00065	R/W	Channel 0 to F: 0 h to F h				• Screen type code Process data: 1 h Master Unit parameter: 2 h Device parameter: 3 h Event/error display: 4 h				Parameter number 1 to 99 (01 h to 63 h)							
	Displayed screen	00066	R	Channel 0 to F: 0 h to F h				• Screen type code Process data: 1 h Master Unit parameter: 2 h Device parameter: 3 h *S* or later: 4 h Event/error display: 5 h				Parameter number 1 to 99 (01 h to 63 h)							
Operation lock	Interlock for each operation	00067	R/W	Reserved												All button operation lock	Event/error clear operation lock	Device parameter write lock	Master Unit parameter write lock

This Product's Time

Category	Data name	Holding register (decimal)	R/W	Bit 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
This Product's Time	Inside this product Count time*1	00068	R/W	Time in milliseconds or less (unit: 15.625 μs)															
		00069	R/W	Low order of UNIX time (unsigned integer)															
		00070		High order of UNIX time (unsigned integer)															

*1. It does not reflect the UNIX time counted in the host network in real time.
Since the time is not retained, you need to set the time when the power is turned on again.

Status/Operation Data

Category	Data name	Holding register (decimal)	R/W	Bit 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Status flag*1		00071	R	Error flag	IO-Link ready flag	Reserved		Latest error channel				Event flag	I/O power supply flag	Output overcurrent flag	Reserved	Latest event channel			
		00072		Applicable channels 0 to F															
Errors		00073	R	Latest error code															
		00074	R	Index number that caused the latest error															
		00075	R	Reserved								Sub-index number that caused the latest error							
Events	• Low-order Event 0 to 5 flag • High-order Event 0	00076	R	Type of event 0		Meaning of event 0		Source of event 0	Event 0 Cause			Details available	Reserved	Event 5 occurred	Event 4 occurred	Event 3 occurred	Event 2 occurred	Event 1 occurred	Event 0 occurred
	Event 0	00077		Event code															
	Event 1	00078	R	Type		Meaning		Source	Cause			Reserved							
		00079		Event code															
	Event 2	00080	R	Type		Meaning		Source	Cause			Reserved							
		00081		Event code															
	Event 3	00082	R	Type		Meaning		Source	Cause			Reserved							
		00083		Event code															
	Event 4	00084	R	Type		Meaning		Source	Cause			Reserved							
		00085		Event code															
	Event 5	00086	R	Type		Meaning		Source	Cause			Reserved							
		00087		Event code															
Status operation, reset		00088	R/W	Reserved												Count	Clear the latest error*2	Clear the latest event*2	Find Me

*1. For the details of each data, refer to *Reading status data of this product* in *4-1 Product Functions* of the Common Edition.
If the host network type is *Ethernet & Modbus/TCP*, the *synchronization establishment flag* does not exist.

*2. For the details of each data, refer to *Writing operation data of this product* in *4-1 Product Functions* of the Common Edition.

I/O (Digital)

Category	Data name	Holding register (decimal)	R/W	Bit															
				15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit input	Channels 0 to F	00089	R	CH F	CH E	CH D	CH C	CH B	CH A	CH 9	CH 8	CH 7	CH 6	CH 5	CH 4	CH 3	CH 2	CH 1	CH 0
Bit output		00346	R/W	CH F	CH E	CH D	CH C	CH B	CH A	CH 9	CH 8	CH 7	CH 6	CH 5	CH 4	CH 3	CH 2	CH 1	CH 0

I/O (Process Data)

Category	Data name	Holding register (decimal)	R/W	Bit															
				15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Process data input	CH 0	00090 … 00105	R	Process input data received from channel 0 Up to 32 bytes (16 words)															
	CH 1	00106 … 00121	R	Process input data received from channel 1 Up to 32 bytes (16 words)															
	CH 2	00122 … 00137	R	Process input data received from channel 2 Up to 32 bytes (16 words)															
	CH 3	00138 … 00153	R	Process input data received from channel 3 Up to 32 bytes (16 words)															
	CH 4	00154 … 00169	R	Process input data received from channel 4 Up to 32 bytes (16 words)															
	CH 5	00170 … 00185	R	Process input data received from channel 5 Up to 32 bytes (16 words)															
	CH 6	00186 … 00201	R	Process input data received from channel 6 Up to 32 bytes (16 words)															
	CH 7	00202 … 00217	R	Process input data received from channel 7 Up to 32 bytes (16 words)															
	CH 8	00218 … 00233	R	Process input data received from channel 8 Up to 32 bytes (16 words)															
	CH 9	00234 … 00249	R	Process input data received from channel 9 Up to 32 bytes (16 words)															
	CH A	00250 … 00265	R	Process input data received from channel A Up to 32 bytes (16 words)															
	CH B	00266 … 00281	R	Process input data received from channel B Up to 32 bytes (16 words)															

Category	Data name	Holding register (decimal)	R/W	Bit 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CH C	00282 … 00297	R	Process input data received from channel C Up to 32 bytes (16 words)															
	CH D	00298 … 00313	R	Process input data received from channel D Up to 32 bytes (16 words)															
	CH E	00314 … 00329	R	Process input data received from channel E Up to 32 bytes (16 words)															
	CH F	00330 … 00345	R	Process input data received from channel F Up to 32 bytes (16 words)															
Process data output	CH 0	00347 … 00362	R/W	Process output data sent to channel 0 Up to 32 bytes (16 words)															
	CH 1	00363 … 00378	R/W	Process output data sent to channel 1 Up to 32 bytes (16 words)															
	CH 2	00379 … 00394	R/W	Process output data sent to channel 2 Up to 32 bytes (16 words)															
	CH 3	00395 … 00410	R/W	Process output data sent to channel 3 Up to 32 bytes (16 words)															
	CH 4	00411 … 00426	R/W	Process output data sent to channel 4 Up to 32 bytes (16 words)															
	CH 5	00427 … 00442	R/W	Process output data sent to channel 5 Up to 32 bytes (16 words)															
	CH 6	00443 … 00458	R/W	Process output data sent to channel 6 Up to 32 bytes (16 words)															
	CH 7	00459 … 00474	R/W	Process output data sent to channel 7 Up to 32 bytes (16 words)															
	CH 8	00475 … 00490	R/W	Process output data sent to channel 8 Up to 32 bytes (16 words)															
	CH 9	00491 … 00506	R/W	Process output data sent to channel 9 Up to 32 bytes (16 words)															
	CH A	00507 … 00522	R/W	Process output data sent to channel A Up to 32 bytes (16 words)															
	CH B	00523 … 00538	R/W	Process output data sent to channel B Up to 32 bytes (16 words)															
	CH C	00539 … 00554	R/W	Process output data sent to channel C Up to 32 bytes (16 words)															
	CH D	00555 … 00570	R/W	Process output data sent to channel D Up to 32 bytes (16 words)															

Category	Data name	Holding register (decimal)	R/W	Bit 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CH E	00571 … 00586	R/W	Process output data sent to channel E Up to 32 bytes (16 words)															
	CH F	00587 … 00602	R/W	Process output data sent to channel F Up to 32 bytes (16 words)															

ISDU Access Library

Category	Data name	Holding register (decimal)	R/W	Bit 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Instruction	Execution target ISDU library specification*2	00603	R/W	LB015	LB014	LB013	LB012	LB011	LB010	LB009	LB008	LB007	LB006	LB005	LB004	LB003	LB002	LB001	LB000
		00604		LB031	LB030	LB029	LB028	LB027	LB026	LB025	LB024	LB023	LB022	LB021	LB020	LB019	LB018	LB017	LB016
		00605		LB047	LB046	LB045	LB044	LB043	LB042	LB041	LB040	LB039	LB038	LB037	LB036	LB035	LB034	LB033	LB032
		00606		LB063	LB062	LB061	LB060	LB059	LB058	LB057	LB056	LB055	LB054	LB053	LB052	LB051	LB050	LB049	LB048
		00607		LB079	LB078	LB077	LB076	LB075	LB074	LB073	LB072	LB071	LB070	LB069	LB068	LB067	LB066	LB065	LB064
		00608		LB095	LB094	LB093	LB092	LB091	LB090	LB089	LB088	LB087	LB086	LB085	LB084	LB083	LB082	LB081	LB080
		00609		LB111	LB110	LB109	LB108	LB107	LB106	LB105	LB104	LB103	LB102	LB101	LB100	LB099	LB098	LB097	LB096
		00610		LB127	LB126	LB125	LB124	LB123	LB122	LB121	LB120	LB119	LB118	LB117	LB116	LB115	LB114	LB113	LB112
	Read/write execution trigger	00611	R/W	0001 h: Read execution 0002 h: Write execution 0000 h: ISDU access library result (00612 to 00636) all clear execution The moment when each of the above values is written will be the execution trigger.															
Result	Read/write execution reception	00612	R	0001 h: Read execution reception 0002 h: Write execution reception 0000 h: ISDU access library result (00612 to 00636) all clear execution reception															
	Execution status	00613	R	Reserved												Suspended due to an unexpected error	All completed (including errors)	All completed (all normal)	Executing
	Error code	00614	R	0000 h: No error Other values: Latest error code															

Category	Data name	Holding register (decimal)	R/W	Bit 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Statistics	00615	R	Total number selected as execution target from LB000 to LB127: 0000 h to 0080 h															
		00616	R	Total number of normal completions from LB000 to LB127: 0000 h to 0080 h															
		00617	R	Total number of error completions from LB000 to LB127: 0000 h to 0080 h															
		00618	R	Total number of incompletions from LB000 to LB127: 0000 h to 0080 h															
		00619	R	Completion progress rate: 0.0% to 100.0% (0000 h to 03E8 h)															
	Normal completion flag Corresponds to ISDU library number	00620	R	LB015	LB014	LB013	LB012	LB011	LB010	LB009	LB008	LB007	LB006	LB005	LB004	LB003	LB002	LB001	LB000
		00621		LB031	LB030	LB029	LB028	LB027	LB026	LB025	LB024	LB023	LB022	LB021	LB020	LB019	LB018	LB017	LB016
		00622		LB047	LB046	LB045	LB044	LB043	LB042	LB041	LB040	LB039	LB038	LB037	LB036	LB035	LB034	LB033	LB032
		00623		LB063	LB062	LB061	LB060	LB059	LB058	LB057	LB056	LB055	LB054	LB053	LB052	LB051	LB050	LB049	LB048
		00624		LB079	LB078	LB077	LB076	LB075	LB074	LB073	LB072	LB071	LB070	LB069	LB068	LB067	LB066	LB065	LB064
		00625		LB095	LB094	LB093	LB092	LB091	LB090	LB089	LB088	LB087	LB086	LB085	LB084	LB083	LB082	LB081	LB080
		00626		LB111	LB110	LB109	LB108	LB107	LB106	LB105	LB104	LB103	LB102	LB101	LB100	LB099	LB098	LB097	LB096
		00627		LB127	LB126	LB125	LB124	LB123	LB122	LB121	LB120	LB119	LB118	LB117	LB116	LB115	LB114	LB113	LB112
	Reserved	00628		Reserved															
	Error completion flag Corresponds to ISDU library number	00629	R	LB015	LB014	LB013	LB012	LB011	LB010	LB009	LB008	LB007	LB006	LB005	LB004	LB003	LB002	LB001	LB000
		00630		LB031	LB030	LB029	LB028	LB027	LB026	LB025	LB024	LB023	LB022	LB021	LB020	LB019	LB018	LB017	LB016
		00631		LB047	LB046	LB045	LB044	LB043	LB042	LB041	LB040	LB039	LB038	LB037	LB036	LB035	LB034	LB033	LB032
		00632		LB063	LB062	LB061	LB060	LB059	LB058	LB057	LB056	LB055	LB054	LB053	LB052	LB051	LB050	LB049	LB048
		00633		LB079	LB078	LB077	LB076	LB075	LB074	LB073	LB072	LB071	LB070	LB069	LB068	LB067	LB066	LB065	LB064
		00634		LB095	LB094	LB093	LB092	LB091	LB090	LB089	LB088	LB087	LB086	LB085	LB084	LB083	LB082	LB081	LB080
		00635		LB111	LB110	LB109	LB108	LB107	LB106	LB105	LB104	LB103	LB102	LB101	LB100	LB099	LB098	LB097	LB096
		00636		LB127	LB126	LB125	LB124	LB123	LB122	LB121	LB120	LB119	LB118	LB117	LB116	LB115	LB114	LB113	LB112
Reserved		00637 … 00699		Reserved															

Category	Data name	Holding register (decimal)	R/W	Bit 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ISDU library Read/write execution target index registration details	LB 000	00700	R/W*1	Target port specification: Channel 0 to F: 0000 h to 000F h															
		00701	R/W*1	Index number specification															
		00702	R/W*1	Sub-index number specification															
		00703	R/W*1	Specify the request data length specification and size in bytes. **Note** When 0000 h is specified, reading is executed according to the number of bytes of the actual IO-Link device. (Variable) Writing is not executed.															
		00705 00706	R/W*1	User memo Any value can be entered. It can be used for specifying indirect register numbers of control devices and making memos.															
		00707	—	Reserved															
		00708 … 00723	R/W*1	Request data (up to 32 bytes) Reading: Read data is stored. Writing: The user stores the write data in this area in advance.															
	LB 001	00724 … 00747	R/W*1	Same as above (24 words)															
	…	…	…	…															
	LB 126	03724 … 03747	R/W*1	Same as above (24 words)															

Category	Data name	Holding register (decimal)	R/W	Bit 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	LB 127 Request data extended version	03748	R/W*1	Target specification: channel 0 to F h (0000 to 000F h), this product (FF00 h)															
		03749	R/W*1	Index number specification															
		03750	R/W*1	Sub-index number specification															
		03751	R/W*1	Request data length specification															
		03753	R/W*1	User memo															
		03754																	
		03755	Reserved																
		03756 … 03871	R/W*1	Request data (Up to 232 bytes)															

*1. The parameter when "4342 h" (backup execution) is written to the execution trigger (register address: 4124) related to the backup is backed up to the non-volatile memory of this product. The value is automatically read when this product is turned on again.

*2. ON: Select the corresponding ISDU library number as the access execution target.
OFF: The corresponding ISDU library number is not targeted for accessing.
It is executed sequentially from the youngest ISDU library number.

Modbus Address Monitor Registration

Category	Data name	Holding register (decimal)	R/W	Bit 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Modbus address monitor registration (instruction)	n+0	03872	R/W*1	Specify the Holding register address number to which the value is read from the Modbus master. Up to 125 addresses can be registered.															
	n+1	03873																	
	n+2	03874																	
	n+3 … n+124	03875 … 03996																	
	Number of monitor words	03997	R/W*1	Specify how many words of registered address are enabled from n+0. 0001 h to 007D h: 1 to 125 words 0000 h: Function is not used															

Category	Data name	Holding register (decimal)	R/W	Bit 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Modbus address monitor registration (output)	Number of monitor words (Reception)	03998	R	Indicates that the specified number of monitor words was received. 0001 h to 007D h: 1 to 125 words 0000 h: Function is not used															
	n+0	03999	R	Values of the Holding register specified by n+0 to 124 of the Modbus address monitor registration (command) is stored. These can be read from the Modbus/TCP master with one command for up to 125 words starting from 03999. This is a useful function for improving the communication efficiency from the Modbus/TCP master.															
	n+1 … n+124	04000 … 04123	R																

*1. The parameter when "4342 h" (backup execution) is written to the execution trigger (register address: 4124) related to the backup is backed up to the non-volatile memory of this product. The value is automatically read when this product is turned on again.

Backing Up

Category	Data name	Holding register (decimal)	R/W	Bit 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Backing up	Execution trigger	04124	R/W	Backs up the parameters of the ISDU access library and Modbus address monitor registration to the non-volatile memory of this product. 4342 h: Backup execution 4C43 h: Erase all backups 0000 h: Execution status flag clear execution **Note** The backup target is the parameters of the following register numbers. • 00700 to 03871: ISDU access specification ISDU library number LB 000 to 127 • 03872 to 03997: Modbus address monitor registration (command) n+0 to n+124, number of monitor words															
	Execution reception	04125	R	4342 h: Execute backup reception 4C43 h: Backup erasure reception 0000 h: Execution status flag clear execution reception Other than the above values, echo back will not be performed and the last value will be maintained.															
	Execution status	04126	R	Reserved													Completed W/ error(s)	Completed normally	Executing

Additional Information

The maximum number of writing of the non-volatile memory of this product is at least 100,000. In order to protect the memory element, do not periodically write to the parameter of this product through backups, etc.

External Device Access Status

Category	Data name	Holding register (decimal)	R/W	Bit 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
External device access status monitor	Rwr CC-Link IE Field Basic	04127	R	Details of the communication of remote register RWr n+0 of the CC-Link IE Field Basic master station															
		04128 … 04158		Same as above n+1 … Same as above n+31															
	RWw CC-Link IE Field Basic	04159	R	Details of the communication of remote register RWw n+0 of the CC-Link IE Field Basic master station															
		04160 … 04190		Same as above n+1 … Same as above n+31															
	Reserved	04191 … 04999	Reserved																

Master Unit Parameters of this product

Data name	Target channel	Holding register (decimal)	R/W	Parameter (monitor)	
M10. I/O setting	CH 0 … CH F	05500 … 05515	RW	0	IO-Link mode
				1	PNP input mode: Internal pull-down resistance is enabled.
				2	NPN input mode: Internal pull-up resistance is enabled.
				3	PNP output mode
				4	NPN output mode
				5 (default value)	Not used
M11. Input filter	CH 0 … CH F	05516 … 05531	RW	0 (default value)	None
				1	0.1 ms
				2	1 ms
				3	5 ms
				4	10 ms
				5	20 ms
M12. Input hold time	CH 0 … CH F	05532 … 05547	RW	0 (default value)	None
				1	1 ms
				2	15 ms
				3	100 ms
M13. IO-Link communication and network error handling	CH 0 … CH F	05548 … 05563	RW	0 (default value)	Clear
				1	Input hold
				2	Output hold
				3	All Hold

Data name	Target channel	Holding register (decimal)	R/W	Parameter (monitor)	
M21. I/O synchronization	CH 0 … CH F	05564 … 05579	RW	0 (default value)	Asynchronous IO-Link communication uses individual devices' fastest times. The digital I/O status is also continually updated.
				2	I/O is synchronized at 0.4 ms cycle. It is targeted for the IO-Link process data and digital I/O. Synchronization is performed between multiple channels for which the same cycle is set.
				3	As above, synchronization at 0.8 ms cycle
				4	As above, synchronization at 1.6 ms cycle
				5	As above, synchronization at 3.2 ms cycle
				6	As above, synchronization at 6.4 ms cycle
				7 to 1003	Specify the IO-Link communication cycle time at 0.4 ms to 100.0 ms.
M30. Device verification	CH 0 … CH F	05580 … 05595	RW	0 (default value)	None **Note** Even for *None*, if storage data is already saved in the product, when turning on the power (unit and I/O power), the IO-Link device type ID and storage data type ID will be verified. An error (FF23 h) will occur if the type ID does not match. The revision ID is not verified.
				1	Verification of type ID (vendor ID and device ID) If the registered type ID and connected device type ID are different, an error (FFFE h: type ID verification error) is generated and the process data with the relevant IO-Link device is treated as invalid. As well, if IO-Link communication is not established within 10 seconds of turning on the I/O power, an error (FFFB h: IO-Link device not connected) is generated. At the same time, the revision ID is also verified.
				2	Type ID + serial number verification Even if the type IDs (vendor ID and device ID) match, if the serial number is different, an error (FFFC h: serial number verify error) is generated, and the process data with the relevant IO-Link device is invalidated. Other operations are the same as Parameter 1.
				3	Type ID + device model name verification Even if the type IDs (vendor ID and device ID) match, if the device model name is different, an error (FFF4 h: model name verification error) is generated, and the process data with the relevant IO-Link device is invalidated. Other operations are the same as Parameter 1.

Data name	Target channel	Holding register (decimal)	R/W	Parameter (monitor)	
M31. Automatic device parameter backup	CH 0 … CH F	05596 … 05611	RW	0 (default value)	None
				1	Auto backup When IO-Link device Parameters are changed, they are automatically backed up in this product. If IO-Link device setting values are changed from this product, the backup operation will automatically start 10 seconds after the last change. **Note** When a device with a different vendor ID or device ID is connected, backup is executed at every startup, so reset *Device verification* (Parameter number: M30) to correct the verification error (FFFE h) as soon as possible.
				2	Auto restoration Automatically restore (download parameters from this product to the IO-Link device) if the IO-Link device parameters differ from the data stored on this product during startup. In this case, note that even if the IO-Link device parameter is changed, it will be overwritten at the next startup with the data saved in this product.
				3	Perform auto backup and auto restoration as above together The storage data stored on this product will always match the IO-Link device parameters. In other words, IO-Link device parameters are backed up to this product whenever they are changed. If any parameters are different from the IO-Link device during startup, the parameters stored in this product will be restored.
M33. Conditions for applying IODD data	CH 0 … CH F	05612 … 05627	RW	0	Device Prioritized
				1 (default value)	IODD
M40. Process input data words allocation	CH 0 … CH F	05628 … 05643	RW		This setting does not have effect on Modbus protocol communication.
M41. Process output data words allocation	CH 0 … CH F	05644 … 05659	RW		This setting does not have effect on Modbus protocol communication.
M43. Process data LSB/MSB	CH 0 … CH F	05660 … 05675	RW	0 (default value)	Little endian LSB
				1	Big endian MSB
Reserved		05676 … 05799	Reserved		
M8. User tag name of this IO-Link Master Unit	This product	05800 … 05815	RW	Up to 32 characters	Set the product user tag name up to 32 characters.
Reserved		05816 … 05829	Reserved		
M29. Device ID	CH 0 … CH F	05830 … 05893	RW	0 h to FFFFFF h (Default value: 0 h)	This is the device ID used for verification with IO-Link devices. Refer to *M29. Device ID* in *5-1-4 Master Unit Parameter List with Product Front Panel Operations* of the Common Edition for details.

Data name	Target channel	Holding register (decimal)	R/W	Parameter (monitor)	
Reserved		05866 … 05899	Reserved		
M4. Display brightness	This product	05900	RW	1 to 20 (Default value: 7)	Display brightness Values multiplied by 5 are equivalent to % display.
M5. Language	This product	05901	RW	0 (default value)	English
				1	Japanese
				2	German
				3	Chinese (Simplified)
				4	French
				5	Spanish
				6	Portuguese
				7	Italian
				9	Korean
				10	Chinese (Traditional)
M6. Time zone -UTC-	This product	05902	RW	-96 to 96 (Default value: 0)	Set the time zone (in 15-minute increments) for the displayed network time (parameter number: M65). If the network time set from the host is based on UTC, setting the time zone as 9 x 4 = 36 will display the Japan standard time (JST).
M44. Time stamp	This product	05904	RW	This setting does not have effect on Modbus protocol communication.	
M67. Network timeout	This product	05905	R/W	0 to 30000	When the network type is *Ethernet & Modbus/TCP*, if the Ethernet communication is interrupted and the specified time elapses, a network communication stop error (FFFA h) occurs. The specified time can be specified from 0 to 30000 (300 seconds) in 10 ms increments. If it set to the default value of 0, even if Ethernet communication is interrupted, it is considered that the data has been updated internally at a cycle of about 10 ms and no error occurs. Even in such a case, an error will occur if the Ethernet cable is disconnected.
Reserved		05907 … 06099	Reserved		
M51. IO-Link cycle time -Process value-	CH 0 … CH F	06100 … 06115	R	1 to 1000	0.1 ms increments
M52. IO-Link transmission rate	CH 0 … CH F	06116 … 06131	R	0	Not communicating
				1	COM1
				2	COM2
				3	COM3
M53. IO-Link communication errors	CH 0 … CH F	06132 … 06147	R	0 to 255	To clear data, write 1 in offset +0 of parameter 114 h.
M54. IO-Link ISDU checksum errors	CH 0 … CH F	06148 … 06163	R	0 to 255	To clear data, write 1 in offset +0 of parameter 114 h.
M55. IO-Link transmission rate error	CH 0 … CH F	06164 … 06179	R	-1000 to 1000	IO-Link transmission rate error 0.1% unit, signed

Data name	Target channel	Holding register (decimal)	R/W	Parameter (monitor)		
M56. IO-Link signal width error	CH 0 … CH F	06180 … 06195	R	-1000 to 1000	IO-Link transmission signal width error 0.1% unit, signed	
M57. IO-Link Communication mode	CH 0 … CH F	06196 … 06211	R	0 to 7	0: NOCOM 1: STARTUP 2: PREOPERATE 3: OPERATE 5: STARTUP (Rev.1.0) 6: PREOPERATE (Rev.1.0) 7: OPERATE (Rev.1.0)	Refer to *M57. IO-Link communication mode* in *5-1-4 Master Unit Parameter List with Product Front Panel Operations* of the Common Edition for details.
Reserved		06212 … 06249	Reserved			
M2. Network type	This product	06250	RW	Specify the host industrial network type.		
				2 (default value)	EtherNet/IP	
				5	Ethernet & Modbus/TCP	
				6	CC-Link IE Field Basic	
M63. I/O power supply voltage	This product	06251	R	0 to 308	Voltage supplied to product I/O power supply (unit: 0.1 V)	
M64. Internal temperature	This product	06252	R	-2739 to 10457	Product internal temperature (unit: 0.1°C)	
M66. Network communication cycle	This product	06253	R	0 to 15000	It is irrelevant when the network type is *Ethernet & Modbus/TCP*.	
Reserved		06254 … 06299	Reserved			
M60. System program version of IO-Link Master Unit	This product	06300, 06301	R	P.PPNNLLFF	Display the data version of the program, etc. P.PP: Firmware, NN: Network chip, LL: Logic, FF: Font 4 bytes in decimal.	
M61. Total operation hours of IO-Link Master Unit	This product	06302, 06303	R	0 to 2097151 (239 years)	Product cumulative operating time information (unit: hours)	
M62. Display driven hours	This product	06304, 06305	R	0 to 262143 (30 years)	Product display operating time information (unit: hours)	
M80. Default gateway	This product	06306, 06307	R/W	0.0.0.0 to 255.255.255.255	Change if otherwise specified. If not, leave the default value. The default value is 0.0.0.0.	
M81. Subnet mask	This product	06308, 06309	R/W	0.0.0.0 to 255.255.255.255	Change if otherwise specified. If not, leave the default value. The default value is 255.255.255.0	
M82. IP address	This product	06310, 06311	R/W	0.0.0.0 to 255.255.255.255	Set the IP address of this product to any value. This is set only when changing the default value of 192.168.250.xxx (xxx is the value set by the rotary switch). Turning the front rotary switch while the power is on automatically switches to *M82. IP address* setting window	

Data name	Target channel	Holding register (decimal)	R/W	Parameter (monitor)	
Reserved		06312 … 06319	Reserved		
M58. Storage backup time	CH 0 … CH F	06320 … 06463	R	00:00:00 00/01/01 to 23:59:59 99/12/31	The storage data backup time is displayed as below. Time and date (ex.: 18:59:00 20/02/29)
Reserved		06464 … 06499	Reserved		
M1. IO-Link Master Unit parameters reset	This product	06500	RW	0 (default value)	None
				1	Initialize I/O assignment for all channels in IO-Link mode. (Storage data, display language, network type, and network No. will not be initialized)
				2	As above, initialize I/O assignment for all channels in PNP input mode.
				3	As above, initialize I/O assignment for all channels in NPN input mode.
				4	As above, initialize I/O assignment for all channels in PNP output mode.
				5	As above, initialize I/O assignment for all channels in NPN output mode.
				6	As above, initialize I/O assignment for all channels as unused.
				7	Delete the installed IODD data.
				8	Initialize including network setting and language as well. I/O assignment is unused. IODD data is not deleted.
M9. I/O batch setting	This product	06501	RW	0 (default value)	None
				1	Change I/O assignment settings for all channels to IO-Link mode.
				2	Change I/O assignment for all channels to PNP input mode.
				3	Change I/O assignment for all channels to NPN input mode.
				4	Change I/O assignment for all channels to PNP output mode.
				5	Change I/O assignment for all channels to NPN output mode.
				6	Change I/O assignment for all channels to unused.
Reserved		06502 … 06549	Reserved		
M32. Device parameter backup/ restore	CH 0 … CH F	06550 … 06565 (corresponds to channel number)	RW	0 (default value)	None
				1	Perform backup (uploading parameters from the device to this product) manually.
				2	Perform restoration (downloading parameters from the product to the device) manually.
				3	Delete the backup data stored in this product manually.

Attached Table 1: Correlation Table between the Channel of This Product Connected to Phases A, B, and Z of the Encoder and the Parameter (Hexadecimal Number) of This Register

Value	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
-	Connecting the encoder to the following channel															
Phase A	-	0	1	2	3	4	5	6	7	8	9	A	B	C	D	-
Phase B	-	1	2	3	4	5	6	7	8	9	A	B	C	D	E	-
Phase Z	-	2	3	4	5	6	7	8	9	A	B	C	D	E	F	-

Value	10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D	1E	1F
-	Connecting the encoder to the following channel															
Phase A	-	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E
Phase B	-	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F

Value	20	21	22	23	24	25	26	27	28	29	2A	2B	2C	2D	2E	2F
-	Connecting the encoder to the following channel															
Phase A	-	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E

Value	30
-	CH
Phase A	F

User Area

Category	Data name	Holding register (decimal)	R/W	Bit 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
User free area	Address that can be freely used by the user	07000 … 07031	R/W	Up to 32 words															

Basic Information of Connected Device

Category	Data name	Holding register (decimal)	R/W	Bit 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Connected IO-Link device Vendor name	CH 0 Vendor name	07032 … 07063	R	Character string (Up to 32 words)															
	…	…		…															
	CH F Vendor name	07512 … 07543		Character string (Up to 32 words)															

Category	Data name	Holding register (decimal)	R/W	Bit 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Connected IO-Link device Device model	CH 0 Device model	07544 … 07575	R	Character string (Up to 32 words)															
	…	…		…															
	CH F Device model	08024 … 08055		Character string (Up to 32 words)															
Connected IO-Link device User tag	CH 0 User tag	08056 … 08071	R	Character string (Up to 32 words)															
	…	…		…															
	CH F User tag	08296 … 08311		Character string (Up to 32 words)															

3-2-3 Modbus TCP/UDP Communication Example

This section shows the specification of register numbers in the example of reading/writing service data of the IO-Link device and the example of batch monitoring the process data via Modbus TCP/UDP communication.

For the communication frame that contains the function code, refer to *3-2-4 Example of Communication Frame for Each Function Code (FC)* on page 3-23.

Read/Write Service Data to and from the IO-Link Device

Example: Reading/writing the service data of the following two IO-Link devices.

- Reading the device status of the photoelectric sensor E3AS-HL on port 0
- Turning OFF the laser emission of the photoelectric sensor E3AS-HL on port 1 (writing)

- The index number for the device status of the photoelectric sensor E3AS-HL is 24 h, and the sub-index number is 0 h
- The index number for laser emission OFF of the photoelectric sensor E3AS-HL is 40 h, the sub-index number is 0 h, and the write data is A1 h

Therefore, the following values are registered in the index access library

Execution ISDU library	Holding register top n (decimal)	n+0	n+1	n+2	n+3	n+4	n+5 to n+7	n+8 to n+23
		Target channel	Index number specification	Sub-index number specification	Request data length specification (byte unit)	Extension code	User memo/ reservation	Request data
LB000	700	0000 h	0024 h	0000 h	0002 h	0000 h	0 fixed	Read data is stored
LB001	724	0001 h	0040 h	0000 h	0002 h	0000 h	0 fixed	00A1 h

Then, the values to the following register numbers are written.

- To execute LB000, set Bit 0 of 00603 to 1 (ON) and write 001 h (read execution) to 00611.

• To execute LB001, set Bit 1 of 00603 to 1 (ON) and write 002 h (write execution) to 00611.

Check whether it completed normally with Bits 0 to 1 of 00620.
Check whether it completed with an error by checking Bits 0 to 1 of 00629.

Batch Monitor the Process Data

Example: Reading the following process data of this product from the Modbus/TCP master in a batch.
• 2 bytes process data input for port 0 IO-Link device (1 word)
• 4 bytes process data input for port 1 IO-Link device (2 word)

Therefore, set the register number and batch read size of the process data input in the registered register number library as follows.

Register number (word unit, decimal)	Registered register number	Parameter (registered register number: hexadecimal)	Details of the registered register number
03872	0	00090 (005A h)	2 bytes process data input for port 0 IO-Link device
03873	1	00106 (006A h)	Top 2 bytes process data input for port 1 IO-Link device
03874	2	00107 (006B h)	Last 2 bytes process data input for port 1 IO-Link device
03997	Batch read word count	0005 h	Specify batch reading of 3 words in total

Then, specify the following register number and word count size from the Modbus/TCP master, and execute batch read.
• Top register number: 03999
• Batch read words count size: The value of 03998 (0005 h)

3-2-4 Example of Communication Frame for Each Function Code (FC)

This section shows the transmission frame and reception frame for each of the following function codes that are supported by this product in Modbus TCP/UDP communication.

Corresponding Modbus function code:

Function code	Details		Number of words that can be accessed
03 h	Read holding registers	Read holding registers	1 to 125 words
06 h	Write single register	Write single register	1 word
10 h	Write multiple registers	Write multiple registers	1 to 123 words
17 h	Read/Write multiple registers	Read/Write multiple registers	Read: 1 to 125 words Write: 1 to 121 words

FC 03 h (Read Multiple Holding Registers)

This command reads the value of the Modbus register number of this product from the Modbus/TCP master.

Example: The following is where 4 words (87 to 90) are read together starting with register number 87 (decimal).

● Transmission Frame

Byte order	Item	Data length	High/low-order byte	Value	Details
+0	Transfer ID	2 bytes	High-order	00 h	It is used for pairing the transmission from the Modbus/TCP master and the response from this product. The value can remain fixed at 0000 h.
+1			Low-order	00 h	
+2	Protocol ID	2 bytes	High-order	00 h	Specify the protocol type. For the Modbus protocol, specify 0000 h.
+3			Low-order	00 h	
+4	Total number of bytes	2 bytes	High-order	00 h	Specify the message byte length from the byte order +6 to +11. For this multi-read command transmission, the value is fixed at 0006 h.
+5			Low-order	06 h	
+6	Unit ID	1 byte	-	01 h	This is for routing with other lines. It can be any value as it is not used in this product.
+7	Function code	1 byte	-	03 h	03 h is the Modbus function code that reads the value of this product.
+8	Read top register number (Hexadecimal specification)	2 bytes	High-order	00 h	Specify the top register number of this product to be read in unsigned integer. For register number 0087, specify 0057 h. Refer to *3-2-2 Detailed List of Register Numbers* on page 3-5 for details on the register number.
+9			Low-order	57 h	
+10	Readout word count	2 bytes	High-order	00 h	Specify how many words are read from the register number specified by the read top register number. To read 4 words, specify 0004 h. Since the maximum number of reading points is 125 words, 007D h is the maximum specified value.
+11			Low-order	04 h	

● Reception Frame (Normal)

Byte order	Item	Data length	High/low-order byte	Value	Details
+0	Transfer ID	2 bytes	High-order	00 h	The Modbus/TCP master returns a copy of the transfer ID value specified during transmission from this product.
+1			Low-order	00 h	
+2	Protocol ID	2 bytes	High-order	00 h	The Modbus/TCP master returns a copy of the protocol ID value specified during transmission from this product.
+3			Low-order	00 h	
+4	Total number of bytes	2 bytes	High-order	00 h	Returns the message byte length from the byte order 7 to the end. The value changes depending on the data read word count. If 4 words are read, the value will be 000B h.
+5			Low-order	0B h	
+6	Unit ID	1 byte	-	01 h	The Modbus/TCP master returns a copy of the unit ID value specified during transmission from this product.
+7	Function code	1 byte	-	03 h	Returns the executed Modbus function code 03 h.

Byte order	Item	Data length	High/low-order byte	Value	Details
+8	Data byte count	1 byte	-	08 h	Byte length of the read data. If 4 words are read, 08 h will be returned. As the maximum number of read words is 125 words (250 bytes), the maximum value will be 00FA h.
+9	Read data n+0	2 bytes	High-order	04 h	The read-out register number value of this product. 1st word Example: When the value is an integer 1234: 04D2 h
+10			Low-order	D2 h	
+11	Same as above n +1	2 bytes	High-order	16 h	The read-out register number value of this product. 2nd word Example: When the value is an integer 5678: 162E h
+12			Low-order	2E h	
+13	Same as above n +2	2 bytes	High-order	03 h	The read-out register number value of this product. 3rd word Example: When the value is an integer 1000: 03E8 h
+14			Low-order	E8 h	
+15	Same as above n +3	2 bytes	High-order	FC h	The read-out register number value of this product. 4th word Example: When the value is a signed integer -1000: FC18 h
+16			Low-order	18 h	

● Reception Frame (with Error)

Byte order	Item	Data length	High/low-order byte	Value	Details
+0	Transfer ID	2 bytes	High-order	00 h	The Modbus/TCP master returns a copy of the transfer ID value specified during transmission from this product.
+1			Low-order	00 h	
+2	Protocol ID	2 bytes	High-order	00 h	The Modbus/TCP master returns a copy of the protocol ID value specified during transmission from this product.
+3			Low-order	00 h	
+4	Total number of bytes	2 bytes	High-order	00 h	Returns the message byte length from the byte order +6 to +8.
+5			Low-order	03 h	
+6	Unit ID	1 byte	-	01 h	The Modbus/TCP master returns a copy of the unit ID value specified during transmission from this product.
+7	Function code	1 byte	-	83 h	It returns the value of the executed Modbus function code 03 plus 80 h (highest bit in the byte), which indicates an error.
+8	Error code	1 byte	-	** h	The generated error code is returned. (01 h function code error, 02 h register number specification error, 03 h data error, 04 h server device error)

FC 06 h (Write Single Holding Register)

This command writes the value from the Modbus/TCP master to one word of the Modbus register number of this product.

Example: The following is when writing a value to register number 345 (decimal).

● Transmission Frame

Byte order	Item	Data length	High/low-order byte	Value	Details
+0	Transfer ID	2 bytes	High-order	00 h	It is used for pairing the transmission from the Modbus/TCP master and the response from this product. The value can remain fixed at 0000 h.
+1			Low-order	00 h	
+2	Protocol ID	2 bytes	High-order	00 h	Specify the protocol type. For the Modbus protocol, specify 0000 h.
+3			Low-order	00 h	
+4	Total number of bytes	2 bytes	High-order	00 h	Specify the message byte length from the byte order +6 to +11. For this single write command transmission, the value will be fixed at 0006 h.
+5			Low-order	06 h	
+6	Unit ID	1 byte	-	01 h	This is for routing with other lines. It can be any value as it is not used in this product.
+7	Function code	1 byte	-	06 h	06 h is the Modbus function code that writes the values to the register number 1 word of this product.
+8	Write target register number (Hexadecimal specification)	2 bytes	High-order	01 h	Specify the register number of this product to write in unsigned integer. For register number 0345, specify 0159 h. Refer to *3-2-2 Detailed List of Register Numbers* on page 3-5 for details on the register number.
+9			Low-order	59 h	
+10	Write value	2 bytes	High-order	03 h	Specify the value to be written. Example: When writing an integer 1000: 03E8 h
+11			Low-order	E8 h	

● Reception Frame (Normal)

Byte order	Item	Data length	High/low-order byte	Value	Details
+0	Transfer ID	2 bytes	High-order	00 h	The Modbus/TCP master returns a copy of the transfer ID value specified during transmission from this product.
+1			Low-order	00 h	
+2	Protocol ID	2 bytes	High-order	00 h	The Modbus/TCP master returns a copy of the protocol ID value specified during transmission from this product.
+3			Low-order	00 h	
+4	Total number of bytes	2 bytes	High-order	00 h	Returns the message byte length from the byte order +6 to +11.
+5			Low-order	06 h	
+6	Unit ID	1 byte	-	01 h	The Modbus/TCP master returns a copy of the unit ID value specified during transmission from this product.
+7	Function code	1 byte	-	06 h	Returns the executed Modbus function code 06 h.
+8	Write target register number (Hexadecimal specification)	2 bytes	High-order	01 h	The written register number is returned.
+9			Low-order	59 h	

Byte order	Item	Data length	High/low-order byte	Value	Details
+10	Write value	2 bytes	High-order	03 h	The written value is returned.
+11			Low-order	E8 h	

● Reception Frame (with Error)

Byte order	Item	Data length	High/low-order byte	Value	Details
+0	Transfer ID	2 bytes	High-order	00 h	The Modbus/TCP master returns a copy of the transfer ID value specified during transmission from this product.
+1			Low-order	00 h	
+2	Protocol ID	2 bytes	High-order	00 h	The Modbus/TCP master returns a copy of the protocol ID value specified during transmission from this product.
+3			Low-order	00 h	
+4	Total number of bytes	2 bytes	High-order	00 h	Returns the message byte length from the byte order +6 to +8.
+5			Low-order	03 h	
+6	Unit ID	1 byte	-	01 h	The Modbus/TCP master returns a copy of the unit ID value specified during transmission from this product.
+7	Function code	1 byte	-	86 h	It returns the value of the executed Modbus function code 06 h plus 80 h (highest bit in the byte), which indicates an error.
+8	Error code	1 byte	-	** h	The generated error code is returned. (01 h function code error, 02 h register number specification error, 03 h data error, 04 h server device error)

FC 10 h (Write Multiple Holding Registers)

This command writes the value from the Modbus/TCP master to multiple words of the Modbus register number of this product.

Example: The following is where the values are written collectively for 4 words (345 to 348) with the register number 345 (decimal) at the top.

● Transmission Frame

Byte order	Item	Data length	High/low-order byte	Value	Details
+0	Transfer ID	2 bytes	High-order	00 h	It is used for pairing the transmission from the Modbus/TCP master and the response from this product. The value can remain fixed at 0000 h.
+1			Low-order	00 h	
+2	Protocol ID	2 bytes	High-order	00 h	Specify the protocol type. For the Modbus protocol, specify 0000 h.
+3			Low-order	00 h	

Byte order	Item	Data length	High/low-order byte	Value	Details
+4	Total number of bytes	2 bytes	High-order	00 h	Specify the message byte length from the byte order +6 to the end. The value changes depending on the number of data write word count. When writing 4 words, it will be 000F h.
+5			Low-order	0F h	
+6	Unit ID	1 byte	-	01	This is for routing with other lines. It can be any value as it is not used in this product.
+7	Function code	1 byte	-	10	10 h is the Modbus function code that writes the value to multiple words of the register number of this product.
+8	Write top register number (Hexadecimal specification)	2 bytes	High-order	01	Specify the register number of this product to write in unsigned integer. For register number 0345, specify 0159 h. Refer to *3-2-2 Detailed List of Register Numbers* on page 3-5 for details on the register number.
+9			Low-order	59	
+10	Number of written words (Hexadecimal specification)	2 bytes	High-order	00 h	Specify how many words are written from the register number specified by the write top register number. When writing 4 words, specify 0004 h. As the maximum number of written words is 123, the maximum specified value will be 007B h.
+11			Low-order	04 h	
+12	Same number of bytes as above	1 byte	-	08 h	When writing 4 words, specify 08 h in 8 bytes.
+13	Write data n+0	2 bytes	High-order	04 h	1st word of the written value. Example: When the value is an integer 1234: 04D2 h
+14			Low-order	D2 h	
+15	Same as above n +1	2 bytes	High-order	16 h	2nd word of the written value. Example: When the value is an integer 5678: 162E h
+16			Low-order	2E h	
+17	Same as above n +2	2 bytes	High-order	03 h	3rd word of the written value. Example: When the value is an integer 1000: 03E8 h
+18			Low-order	E8 h	
+19	Same as above n +3	2 bytes	High-order	FC h	4th word of the written value. Example: When the value is a signed integer -1000: FC18 h
+20			Low-order	18 h	

● Reception Frame (Normal)

Byte order	Item	Data length	High/low-order byte	Value	Details
+0	Transfer ID	2 bytes	High-order	00 h	The Modbus/TCP master returns a copy of the transfer ID value specified during transmission from this product.
+1			Low-order	00 h	
+2	Protocol ID	2 bytes	High-order	00 h	The Modbus/TCP master returns a copy of the protocol ID value specified during transmission from this product.
+3			Low-order	00 h	
+4	Total number of bytes	2 bytes	High-order	00 h	Returns the message byte length from the byte order +6 to +11.
+5			Low-order	06 h	

Byte order	Item	Data length	High/low-order byte	Value	Details
+6	Unit ID	1 byte	-	01 h	The Modbus/TCP master returns a copy of the unit ID value specified during transmission from this product.
+7	Function code	1 byte	-	10 h	Returns the executed Modbus function code 10 h.
+8	Write top register number (Hexadecimal specification)	2 bytes	High-order	01 h	The written top register number is returned.
+9			Low-order	59 h	
+10	Number of written words (Hexadecimal specification)	2 bytes	High-order	00 h	The number of written words is returned.
+11			Low-order	04 h	

● Reception Frame (with Error)

Byte order	Item	Data length	High/low-order byte	Value	Details
+0	Transfer ID	2 bytes	High-order	00 h	The Modbus/TCP master returns a copy of the transfer ID value specified during transmission from this product.
+1			Low-order	00 h	
+2	Protocol ID	2 bytes	High-order	00 h	The Modbus/TCP master returns a copy of the protocol ID value specified during transmission from this product.
+3			Low-order	00 h	
+4	Total number of bytes	2 bytes	High-order	00 h	Returns the message byte length from the byte order +6 to +8.
+5			Low-order	03 h	
+6	Unit ID	1 byte	-	01 h	The Modbus/TCP master returns a copy of the unit ID value specified during transmission from this product.
+7	Function code	1 byte	-	90 h	The executed Modbus function code 10 h is returned with the value obtained by adding 80 h (highest bit in the byte), which indicates an error.
+8	Error code	1 byte	-	**	The generated error code is returned. (01 h function code error, 02 h register number specification error, 03 h data error, 04 h server device error)

FC 17 h (Read/Write Multiple Holding Registers)

This command reads/writes the value from the Modbus/TCP master to multiple words of the Modbus register number of this product.

Example: The following is where 4 words (87 to 90) are read together starting with register number 87 (decimal), and the values are written collectively for 4 words (345 to 348) with the register number 345 (decimal) at the top.

● Transmission Frame

Byte order	Item	Data length	High/low-order byte	Value	Details
+0	Transfer ID	2 bytes	High-order	00 h	It is used for pairing the transmission from the Modbus/TCP master and the response from this product. The value can remain fixed at 0000 h.
+1			Low-order	00 h	
+2	Protocol ID	2 bytes	High-order	00 h	Specify the protocol type. For the Modbus protocol, specify 0000 h.
+3			Low-order	00 h	
+4	Total number of bytes	2 bytes	High-order	00 h	Specify the message byte length of the byte order +6 or less. The value changes depending on the number of data write word count. When writing 4 words, it will be 0013 h.
+5			Low-order	13 h	
+6	Unit ID	1 byte	-	01 h	This is for routing with other lines. It can be any value as it is not used in this product.
+7	Function code	1 byte	-	17 h	17 is the Modbus function code that reads/writes the value to multiple words of the register number of this product.
+8	Read top register number (Hexadecimal specification)	2 bytes	High-order	00 h	Specify the top register number of this product to be read in unsigned integer. For register number 0087, specify 0057 h. Refer to *3-2-2 Detailed List of Register Numbers* on page 3-5 for details on the register number.
+9			Low-order	57 h	
+10	Readout word count	2 bytes	High-order	00 h	Specify how many words are read from the register number specified by the read top register number. To read 4 words, specify 0004 h. Since the maximum number of reading points is 125 words, 007D h is the maximum specified value.
+11			Low-order	04 h	
+12	Write top register number (Hexadecimal specification)	2 bytes	High-order	01 h	Specify the register number of this product to write in unsigned integer. For register number 0345, specify 0159 h. Refer to *3-2-2 Detailed List of Register Numbers* on page 3-5 for details on the register number.
+13			Low-order	59 h	
+14	Number of written words (Hexadecimal specification)	2 bytes	High-order	00 h	Specify how many words are written from the register number specified by the write top register number. When writing 4 words, specify 0004 h. As the maximum number of written words is 121, the maximum specified value will be 0079 h.
+15			Low-order	04 h	
+16	Same number of bytes as above	1 byte	-	08 h	When writing 4 words, specify 08 in 8 bytes.
+17	Write data n+0	2 bytes	High-order	04 h	1st word of the written value. Example: When the value is an integer 1234: 04D2 h
+18			Low-order	D2 h	
+19	Same as above n +1	2 bytes	High-order	16 h	2nd word of the written value. Example: When the value is an integer 5678: 162E h
+20			Low-order	2E h	
+21	Same as above n +2	2 bytes	High-order	03 h	3rd word of the written value. Example: When the value is an integer 1000: 03E8 h
+22			Low-order	E8 h	
+23	Same as above n +3	2 bytes	High-order	FC h	4th word of the written value. Example: When the value is a signed integer -1000: FC18 h
+24			Low-order	18 h	

Note The maximum write data is n+120.

● Reception Frame (Normal)

Byte order	Item	Data length	High/low-order byte	Value	Details
+0	Transfer ID	2 bytes	High-order	00 h	The Modbus/TCP master returns a copy of the transfer ID value specified during transmission from this product.
+1			Low-order	00 h	
+2	Protocol ID	2 bytes	High-order	00 h	The Modbus/TCP master returns a copy of the protocol ID value specified during transmission from this product.
+3			Low-order	00 h	
+4	Total number of bytes	2 bytes	High-order	00 h	Returns the message byte length from the byte order +6 to the end. The value changes depending on the data read word count. If 4 words are read, the value will be 000B h.
+5			Low-order	0B h	
+6	Unit ID	1 byte	-	01 h	The Modbus/TCP master returns a copy of the unit ID value specified during transmission from this product.
+7	Function code	1 byte	-	17 h	Returns the executed Modbus function code 17 h.
+8	Read data byte count	1 byte	-	08 h	Byte length of the read data. If 4 words are read, 08 h will be returned.
+9	Read data n+0	2 bytes	High-order	04 h	The read-out register number value of this product. 1st word Example: When the value is an integer 1234: 04D2 h
+10			Low-order	D2 h	
+11	Same as above n +1	2 bytes	High-order	16 h	The read-out register number value of this product. 2nd word Example: When the value is an integer 5678: 162E h
+12			Low-order	2E h	
+13	Same as above n +2	2 bytes	High-order	03 h	The read-out register number value of this product. 3rd word Example: When the value is an integer 1000: 03E8 h
+14			Low-order	E8 h	
+15	Same as above n +3	2 bytes	High-order	FC h	The read-out register number value of this product. 4th word Example: When the value is a signed integer -1000: FC18 h
+16			Low-order	18 h	

● Reception Frame (with Error)

Byte order	Item	Data length	High/low-order byte	Value	Details
+0	Transfer ID	2 bytes	High-order	00 h	The Modbus/TCP master returns a copy of the transfer ID value specified during transmission from this product.
+1			Low-order	00 h	
+2	Protocol ID	2 bytes	High-order	00 h	The Modbus/TCP master returns a copy of the protocol ID value specified during transmission from this product.
+3			Low-order	00 h	

Byte order	Item	Data length	High/low-order byte	Value	Details
+4	Total number of bytes	2 bytes	High-order	00 h	Returns the message byte length from the byte order +6 to +8.
+5			Low-order	03 h	
+6	Unit ID	1 byte	-	01 h	The Modbus/TCP master returns a copy of the unit ID value specified during transmission from this product.
+7	Function code	1 byte	-	97 h	It returns the value of the executed Modbus function code 17 h plus 80 h (highest bit in the byte), which indicates an error.
+8	Error code	1 byte	-	** h	The generated error code is returned. (01 h function code error, 02 h register number specification error, 03 h data error, 04 h server device error)

3-3 Ethernet Socket Communication

This section shows the data structure when sending/receiving data from a client to this product via socket communication.

3-3-1 Transmission Data and Reception Data

After opening the port, the client sends or receives the following data to or from the product. The port is closed when there no longer is the need to send or receive data.

Transmitted Data

Byte count	Meaning of data	Value of data	Description	
+0	Command	53 h	Data access request	
Unit specification +1	Transfer data specification	Bit 0	Readout of process input data/digital input/allocation status data*1 Reads up to 32 words*3 in total for all ports to be used + 1 word for digital input + 2 word for allocated status data in a batch.	
		Bit 1	Write of process output data/digital I/O/allocated operation data*2 Writes up to 32 words*3 in total for all ports to be used + 1 word for digital output + 2 word for allocated operation data in a batch.	
		Bit 2	Parameter access	When reading the parameter, bit 2 = 1 and bit 3 = 0. When writing the parameter, bit 2 = 1 and bit 3 = 1.
		Bit 3	Executes parameter access by writing (1) or reading (0)	
		Bit 4	Background access mode: Allows the data to be read/written from a slow IO-Link device without interrupting the socket communication from others. When Bit 2 is 1, the normal response code will be 62 h (read) and 42 h (write). When Bit 2 is 0, the response of the last command sent will be checked.	
		Bit 5	Forced stop of background access	
		Bit 7 to 6	Reserved (0)	
+2	Specify unit	FF h	Product Master Unit parameters	
+3	Index number or Master Unit parameter number Low-order byte		IO-Link device index number (decimal low-order) of the parameter access, or parameter number (low-order) for the Master Unit parameter of this product	
+4	Index number or Master Unit parameter number High-order byte		IO-Link device index number (decimal high-order) of the parameter access, or parameter number (high-order) for the Master Unit parameter of this product	
+5	Sub-index number or Master Unit parameter target number		The IO-Link device sub-index number of the parameter access, or the target number for the Master Unit parameter of this product.	
+6	Write data		The write data continues as digital output data (2 bytes), allocated operation data (2 bytes), process output data for all ports (64 bytes), parameter bytes to be read *4 (2 bytes) and parameter to be written (1 to 500 bytes). If process output data is not specified, it will start with the parameter.	
+7	Same as above			
(Same below)				

*1. **Allocated status data**
The allocated status data consists of the following bits.

Bit	Data name
15	Error flag
14	IO-Link ready flag
8 to 11	Latest error port
7	Event flag

Bit	Data name
6	I/O power supply flag
5	Output overcurrent flag
0 to 3	Latest event port

For the details of each data, refer to *Reading status data of this product* in *4-1 Product Functions* of the Common Edition.
If the host network type is *Ethernet & Modbus/TCP*, the *synchronization establishment flag* does not exist.

*2. **Allocated operation data**
The allocated operation data consists of the following bits.

Bit	Data name
15	Clear the latest error
7	Clear the latest event
6	Reset encoder counter

For the details of each data, refer to *Writing operation data of this product* in *4-1 Product Functions* of the Common Edition.

*3. **With socket communication, process data assigned to this product are read or written in a batch for all ports according to the following settings in the Master Unit parameter of this product.**
M40. Number of process input data words allocation
M41. Number of process output data words allocation
Therefore, the allocation for each port depends on the Master Unit parameter M40/41 of this product.

*4. **If decimal 0 is specified for the number of bytes of the parameter to be read, the response for the length of the actual parameter will be returned. In such a case, the maximum number of bytes that can be read will be 500 bytes. If the transfer data specification is decimal 10, it is not necessary to send the parameter bytes to be read.**

Received Data

Byte order	Meaning of data	Value of data	Description
+0	Response code	41 h	Parameter write normal response
		4E h	Response without data specification
		50 h	Process output data write normal response
		61 h	Parameter read normal response
		70 h	Process input data read header
		62 h	Parameter read data is being prepared in the background access mode
		42 h	Parameter writing is being processed in the background access mode
		51 h	Background access mode canceled
		45 h	Parameter write error response
		46 h	Parameter write error response (extension error code)
		65 h	Parameter read error response
		66 h	Parameter read error response (extension error code)

Byte order	Meaning of data	Value of data	Description
+1	Error code	00 h	Normal
		02 h	Received socket data length is too short
		03 h	Time-out occurred in parameter access
		04 h	Unexpected response is returned in the parameter access
		05 h	The unit-specified value is out of range
		06 h	The number of parameter bytes to be read is too long (over 500 bytes)
		07 h	An access request was made again during data access. **Note** After making a background access request, set the transfer data specification of transmission data +1 to 10 to complete reading or writing, or set it to 20 to cancel.
		08 h	Background access has never been performed
	Extension error code		
+2	Read data or the next response code		The write data continues as digital input data (2 bytes), allocated status data (2 bytes), process input data for all ports (64 bytes) and read data (1 to 400 bytes). If process input data is not specified, it will start with the parameter. Example: Normal response when transfer data specification is 07 70 h, 00 h, process input data (68 bytes), 50 h, 00 h, 61 h, 00 h, parameter read data
+3	Same as above		
(Same below)			

3-3-2 Example of Data Transmission/Reception

Below are some examples of sending and receiving data to and from this product via socket communication.

Example: Exchanging Process Input/Output Data

Process input data (including allocated status data) for all ports of this product are read and process output data (including allocated operation data) for all ports are written.
The allocation of process input data for each port depends on the settings of the Master Unit parameter M40 (*Number of process input data words allocation*) of this product. The allocation of process output data for each port depends on the settings of the Master Unit parameter M41 (*Number of process output data words allocation*) of this product.

● Transmission Data String (Process I/O Data Exchange)

Byte order	Value of data	Description
+0	53 h	Command
+1	03 h	Bit 1 = Digital output data and write of process output data of all ports (including allocated operation data) Bit 0 = Reads the digital input data and process input data for all ports (including allocated status data)
+2	00 h	Unit specification (dummy)
+3	00 h	Low order of index number (dummy)
+4	00 h	High order of index number (dummy)
+5	00 h	Subindex number (dummy)
+6	15 h	Digital output data output on/off control Ports 0 to 7: Supports bits 0 to 7 (Example: Ports 0, 2 and 4 will be on at 15 h)
+7	30 h	Digital output data output on/off control Ports 8 to F: Supports bits 8 to 15 (Example: Ports C and D will be on at 30 h)
+8	00 h	Process output data allocated operation data low order

Byte order	Value of data	Description
+9	00 h	Process output data allocated operation data high order
+10	00 h	Process output data allocation +0 word low order
+11	00 h	Process output data allocation +0 word high order
+12	00 h	Process output data allocation +1 word low order
+13	00 h	Process output data allocation +1 word high order
+14	00 h	Process output data allocation +2 word low order
+15	00 h	Process output data allocation +2 word high order
…	…	…
+72	00 h	Process output data allocation +31 word low order
+73	00 h	Process output data allocation +31 word high order

● Response Data String

Byte order	Value of data	Description
+0	70 h	Process input data read header
+1	00 h	Normal response code
+2	A2 h	Digital input Data input on/off state Ports 0 to 7: Supports bits 0 to 7 (Example: Ports 1, 5 and 7 will be on at A2 h)
+3	18 h	Digital input Data input on/off state Ports 8 to F: Supports bits 8 to 15 (Example: Ports B and C will be on at 18 h)
+4	00 h	Process input data allocation status data low order
+5	00 h	Process input data allocation status data high order
+6	00 h	Process input data allocation +0 word low order
+7	00 h	Process input data allocation +0 word high order
+8	00 h	Process input data allocation +1 word low order
+9	00 h	Process input data allocation +1 word high order
+10	00 h	Process input data allocation +2 word low order
+11	00 h	Process input data allocation +2 word high order
…	…	…
+68	00 h	Process input data allocation +31 word low order
+69	00 h	Process input data allocation +31 word high order
+70	50 h	Process output data write normal response
+71	00 h	Normal response code

Example: Exchanging Process Input/Output Data and Reading Master Unit Parameter

Process input data (including allocated status data) for all ports of this product are read and process output data (including allocated operation data) for all ports are written. At the same time, the Master Unit parameter number 8 of this product is read.

● Transmission Data String (Exchanging Process Input/Output Data And Reading Master Unit Parameter)

Byte order	Value of data	Description
+0	53 h	Command

Byte order	Value of data	Description
+1	07 h	Bit 3 = Read specification (0), Bit 2 = Parameter access, Bit 1 = Digital output data and write of process output data of all ports (including allocated operation data) Bit 0 = Reads the digital input data and process input data for all ports (including allocated status data)
+2	FF h	Unit specification = Master Unit parameter of this product
+3	08 h	Low order of the Master Unit parameter number (decimal 8)
+4	00 h	High order of the Master Unit parameter number (decimal 8)
+5	00 h	Subindex number (decimal 0)
+6	15 h	Digital output data output on/off control Ports 0 to 7: Supports bits 0 to 7 (Example: Ports 0, 2 and 4 will be on at 15 h)
+7	30 h	Digital output data output on/off control Ports 8 to F: Supports bits 8 to 15 (Example: Ports C and D will be on at 30 h)
+8	00 h	Process output data allocated operation data low order
+9	00 h	Process output data allocated operation data high order
+10	00 h	Process output data allocation +0 word low order
+11	00 h	Process output data allocation +0 word high order
+12	00 h	Process output data allocation +1 word low order
+13	00 h	Process output data allocation +1 word high order
+14	00 h	Process output data allocation +2 word low order
+15	00 h	Process output data allocation +2 word high order
…	…	…
+72	00 h	Process output data allocation +31 word low order
+73	00 h	Process output data allocation +31 word high order
+74	00 h	Low order of the number of bytes (decimal 0) of the parameter to be read **Note** For decimal 0, only the actual number of data is read.
+75	00 h	High order of the number of bytes (decimal 0) of the parameter to be read

● Response Data String

Byte order	Value of data	Description
+0	70 h	Process input data read header
+1	00 h	Normal response code
+2	A2 h	Digital input Data input on/off state Ports 0 to 7: Supports bits 0 to 7 (Example: Ports 1, 5 and 7 will be on at A2 h)
+3	18 h	Digital input Data input on/off state Ports 8 to F: Supports bits 8 to 15 (Example: Ports B and C will be on at 18 h)
+4	00 h	Process input data allocation status data low order
+5	00 h	Process input data allocation status data high order
+6	00 h	Process input data allocation +0 word low order
+7	00 h	Process input data allocation +0 word high order
+8	00 h	Process input data allocation +1 word low order
+9	00 h	Process input data allocation +1 word high order
+10	00 h	Process input data allocation +2 word low order
+11	00 h	Process input data allocation +2 word high order
…	…	…
+68	00 h	Process input data allocation +31 word low order
+69	00 h	Process input data allocation +31 word high order
+70	50 h	Process output data write normal response
+71	00 h	Normal response code
+72	61 h	Parameter read completed
+73	00 h	Normal response code

Byte order	Value of data	Description
+74	30 h	Read data (1st byte of character string "012345")
+75	31 h	Read data (2nd byte of character string "012345")
+76	32 h	Read data (3rd byte of character string "012345")
+77	33 h	Read data (4th byte of character string "012345")
+78	34 h	Read data (5th byte of character string "012345")
+79	35 h	Read data (6th byte of character string "012345")

3-3-3 Flow in Background Access

Example: IO-Link Device Parameter Read Request

Reads the index number decimal 1000 of the IO-Link device on port 0 in the background.

● Transmission Data String (IO-Link Device Parameter Read Request)

Byte order	Value of data	Description
+0	53 h	Command
+1	14 h	Bit 4 = Background access, Bit 3 = Read specification (0), Bit 2 = Parameter access
+2	80 h	Unit specification (IO-Link device on port 0)
+3	E8 h	Lower index number (decimal 1000)
+4	03 h	Upper index number (decimal 1000)
+5	01 h	Subindex number (decimal 1)
+6	00 h	Low order of the number of bytes of the parameter to be read **Note** For decimal 0, only the actual number of data is read.
+7	00 h	High order of the number of bytes (decimal 0) of the parameter to be read

● Response Data String

The following parts will not occur if there is no conflict.

Byte order	Value of data	Description
+0	65 h	Parameter read error response
+1	07 h	Background access is being used elsewhere

● Transmission Data String (Resend)

The following parts will not occur if there is no conflict.

Byte order	Value of data	Description
+0	53 h	Command
+1	14 h	Bit 4 = Background access, Bit 2 = Parameter access
+2	80 h	Unit specification (IO-Link device on port 0)
+3	E8 h	Lower index number (decimal 1000)
+4	03 h	Upper index number (decimal 1000)
+5	01 h	Subindex number (decimal 1)
+6	00 h	Low order of the number of bytes of the parameter to be read **Note** For decimal 0, only the actual number of data is read.
+7	00 h	High order of the number of bytes (decimal 0) of the parameter to be read

● Response Data String

Byte order	Value of data	Description
+0	62 h	Read data is being prepared
+1	00 h	Normal response code

● Transmission Data (Status Check)

Byte order	Value of data	Description
+0	53 h	Command
+1	10 h	Bit 4 = Background access
+2	80 h	Unit specification (IO-Link device on port 0) **Note** Sends the same data as when a read request is made.
+3	E8 h	Lower index number (decimal 1000)
+4	03 h	Upper index number (decimal 1000)
+5	01 h	Subindex number (decimal 1)

● Response Data String

Byte order	Value of data	Description
+0	61 h	Reading completed
+1	00 h	Normal response code
+2	D0 h	Low order of read data (decimal 2000)
+3	07 h	High order of read data (decimal 2000)

Example: IO-Link Device Parameter Write Request

Writes the index number decimal 100 of the IO-Link device on port 3 to decimal 500 in background.

● Transmission Data String (Parameter Write Request)

Byte order	Value of data	Description
+0	53 h	Command
+1	1C h	Bit 4 = Background access, Bit 3 = Write specification (1), Bit 2 = Parameter access
+2	83 h	Unit specification (IO-Link device on port 3)
+3	64 h	Lower index number (decimal 100)
+4	00 h	Upper index number (decimal 100)
+5	00 h	Subindex number (decimal 0)
+6	F4 h	Low order of write data (decimal 500)
+7	01 h	High order of write data (decimal 500)

● Response Data String

The following parts will not occur if there is no conflict.

Byte order	Value of data	Description
+0	45 h	Parameter write error response
+1	07 h	Background access is being used elsewhere

● Transmission Data String (Resend)

The following parts will not occur if there is no conflict.

Byte order	Value of data	Description
+0	53 h	Command
+1	1C h	Bit 4 = Background access, Bit 3 = Write specification (1), Bit 2 = Parameter access
+2	83 h	Unit specification (IO-Link device on port 3)
+3	64 h	Lower index number (decimal 100)
+4	00 h	Upper index number (decimal 100)
+5	00 h	Subindex number (decimal 0)
+6	F4 h	Low order of write data (decimal 500)
+7	01 h	High order of write data (decimal 500)

● Response Data String

Byte order	Value of data	Description
+0	42 h	Writing in progress
+1	00 h	Normal response code

● Transmission Data (Status Check)

Byte order	Value of data	Description
+0	53 h	Command
+1	10 h	Bit 4 = Background access
+2	83 h	Unit specification (IO-Link device on port 3) **Note** Sends the same data as when a read request is made.
+3	64 h	Lower index number (decimal 100)
+4	00 h	Upper index number (decimal 100)
+5	00 h	Subindex number (decimal 0)

● Response Data String

Byte order	Value of data	Description
+0	41 h	Writing completed
+1	00 h	Normal response code

4

Specifications

Describes the specifications for *Ethernet & Modbus/TCP* network types with this product.

4-1 Specifications

4-1-1 Communication Specifications

Item		Specifications
		GD-ILM16□-MLP
Host network communication protocol		Modbus TCP/UDP Communication or Ethernet socket communication
Conforming standard		IEEE802.3u
Transmission speed		10 Mbps (10BASE-T), 100 Mbps (100BASE-TX)
Cable		Twisted pair cable (STP) Category 5, 5e and above
Ethernet connection type		Star configuration, linear bus configuration
Modbus communication	Master (client) /Server	Operates as a Modbus server
	Transport layer protocol	TCP or UDP
	Port number	502 (common to TCP/IP and UDP/IP)
	Corresponding function code	3, 6, 16, 23
	Register number	400001 to 465536
	Distance between nodes	Within 100m
	IP address setting	Static IP address only
	Number of connectable units	Number of units that can be connected from one Modbus/TCP master: It depends on the specifications of the Modbus/TCP master.
	Number of simultaneous connections	Modbus/TCP: 2 Modbus/UDP: 4
Socket communication	Transport layer protocol	TCP or UDP
	Port number	2001 (common to TCP/IP and UDP/IP)
	IP address setting	Static IP address only
	Number of simultaneous connections	TCP/IP: 2 UDP/IP: 4

4-2 Data Processing Time

4-2-1 Process Data Response Time Calculation

Below is the response time of process data from the Ethernet socket communication client via Ethernet socket communication and IO-Link.

Additional Information

The processing time of the client that uses the Ethernet socket communication is determined by the program on the client side of the Ethernet socket communication, but here it is assumed that the host program and the Ethernet socket communication are synchronized.

Ethernet Socket Communication Request Message Processing Time of This Product

The Ethernet socket communication request message processing time is from when this product receives a request message from the Ethernet socket communication client to when the processing is completed and the response message is sent.
This depends on the data transferred from the client, the type of accessed data, and the size of data as follows.

- **Process Input Data**

  (IO-Link cycle time) x 1 to 2 + (host program cycle time) x 1 to 2
  Example) When the IO-Link cycle time is 0.5 ms and the host program cycle time is 10 ms: 10.5 to 21 ms

- **Process Output Data**

  (Cycle time of host program) + (IO-Link cycle time) x 1 to 2
  Example) When the IO-Link cycle time is 0.5 ms and the host program cycle time is 10 ms: 10.5 to 11 ms

IO-Link Communication Response Time

The minimum cycle time for IO-Link devices is defined by individual device.
- Minimum cycle time of “0”: The fastest cycle time supported by this product will be used.
- Minimum cycle time specified: This product will communicate with the IO-Link device at the specified cycle time.

The actual cycle time of IO-Link communication can be confirmed in the *Master Unit parameters (M51. IO-Link cycle time -Process value-)* for this product.

Note This product performs IO-Link communication using hardware logic rather than software, so the fastest time is 0.3 ms.

The cycle time can be specified on this product. This is set in *Master Unit parameters M21. I/O synchronization* in this product. However, it cannot be set faster than the minimum cycle time of the IO-Link device.

Example of Cycle Time for IO-Link

Conditions: Process input data bytes: 2, process output data bytes: 0, bytes, on-request data bytes: 1
Minimum 0.35 ms: Add the command, checksum and reserve bytes, and then multiply by 0.05 ms.
COM3: 0.4 ms (time under 0.1 ms rounded up)
COM2: 2.4 ms
COM1: 19.2 ms

Synchronization Function Between IO-Link Communication and Digital I/O

For Ethernet socket communication, if the Master Unit parameter *M21. I/O synchronization* is set to *Synchronous timer 0.8/1.6/3.2/6.4 ms* in this product, the internal timer of this product will be used to synchronize IO-Link communication or digital I/O transfer between multiple ports set to the same setting value at a cycle of 0.8/1.6/3.2/6.4 ms. (this can also be confirmed in Master Unit parameter *M51. IO-Link cycle time -Process value-*).
However, in order to suppress noise generated by the communication signal. IO-Link communication delays each port 0.56 µs instead of sending at the exact same time.

Internal Data Transfer Processing Time for This Product

IO-Link device process input data is first rearranged through software in this product, and then transferred to the network processing chip.
The processing time depends on the number of process data bytes for the IO-Link devices on all ports. It will be transferred at a speed of approximately 0.1 to 0.4 ms.

4-2-2 I/O Response Time Example

In the following example, the output response time is calculated for a system where 16 OMRON E3AS-HL photoelectric sensors (IO-Link-compatible) are connected to this product using IO-Link communication.
Note that the minimum cycle time for E3AS-HL photoelectric sensors (IO-Link-compatible) is 1.2 ms.

Condition: Ethernet socket communication cycle of 10 ms
E3AS-HL photoelectric sensor process data: Averaged received light level (bit 15 - 1) + output (bit 0) (default value)
The time required for the output from 16 E3AS-HL sensors to be input to the host PLC through this product and processed by the program is as follows.
(IO-Link device cycle time 1.2 ms) x 1 to 2 + (host program cycle time 10 ms) x 1 to 2 = 11.2 to 22.4 ms

5

Troubleshooting

Describes methods of troubleshooting related to Modbus/TCP communication.

5-1 Troubleshooting

5-1-1 Troubleshooting Based on LEDs

Front LEDs (on this product)		Status	Details	Procedure
NS	MS			
Lit red	Flashing red	Duplicate IP addresses within the network		Make sure the IP address is not a duplicate of any other.
Off	Flashing green	Ethernet cable is not connected		Confirm that the Ethernet cable is connected.
		IP address is not set correctly		Confirm the product IP address setting.
		Network starting up (about 7 seconds)		If the subnet mask is not 255.255.255.0 and the default gateway is not 0.0.0.0, change to these settings.
Flashing green	Lit green	Connection not established	Incorrect device file	Check if the device file of this product is used on the master side.
			IP address setting is invalid	• Check whether the device IP address is correctly set on the master side. • Make sure the network part is the same in the master and product IP addresses. Example) When the subnet mask is 255.255.255.0: Master: 192.168.0.2 This product: 192.168.0.1 The underlined part is the network.

Front LEDs (on this product)	Status	Procedure
Ether1 or Ether2		
Off	LAN cable is disconnected	Check whether the Ether1 or Ether2 LAN cables are disconnected.

Front LEDs (on this product)		Status	Details	Procedure
NW READY	NW COMM			
Lit blue-green	-	Internal IC startup of Modbus/TCP	The Modbus/TCP network chip is starting up.	-
Off	-		Network chip for Modbus/TCP is not operating.	If the NW READY LED does not stay ON (blue-green) even though the power is ON, check the UNIT PWR LED. If the NW READY LED is unlit even though the UNIT PWR LED is lit, the power voltage may be extremely low, or the network type switching may have failed, preventing startup.
Flashing blue-green	-	Network chip firmware transferring	The Modbus/TCP network chip firmware is undergoing internal transfer.	-

Front LEDs (on this product)		Status	Details	Procedure
NW READY	NW COMM			
Lit blue-green	Lit blue-green	Modbus/TCP communication status	Communication with the Modbus/TCP master station is normal.	-
Lit blue-green	Flashing blue-green		Communication disconnection	• Confirm the Modbus/TCP master status (for details, refer to the manual of the Modbus/TCP master station in use). • Confirm the status of the Modbus/TCP cable. • Confirm the status of the Switching Hub between the Modbus/TCP master and this product. • Confirm the following if no problems are found. a) IP address setting b) Subnet mask setting c) Default gateway setting After changing the IP address, subnet mask, or default gateway setting, restart the product power.
-	Off		Not communicating	• Confirm the following. a) Communication wiring with the Modbus/TCP master b) Product front panel lowest digit setting IP address c) Modbus/TCP master status • Reset the CPU module of the host PLC.

5-1-2 Troubleshooting Based on Symptoms

Phenomenon	Front LEDs (on this product)	Cyclic communication flag	Error code (Hexadecimal)	Probable cause	Procedure
Data sent to/received from an IO-Link device via IO-Link communication cannot be read/written properly by Modbus/TCP	ALARM LED flashing red	Error flag ON	FFFA	Modbus/TCP communication stops	Confirm the status of the host network (Modbus/TCP).

5-1-3 Error Code List

Code (hexadecimal)	Message	Conditions	Procedure
1000	No Service generated in ISDU communication	This occurs when the start code of the ISDU communication response used to access the setting value for the IO-Link device is “0” (No Service).	A setting value that is not supported by the IO-Link device is being accessed in this case. Confirm what you are trying to access (index number, etc.). This ISDU handling error may occur when the power supply to the IO-Link device is unstable. Use a power supply with sufficient capacity.
1001	IO-Link communication has stopped	This occurs when communication is established with an IO-Link device but then is disconnected. This error does not occur if the I/O power supply is shutdown. The error is also automatically cleared if IO-Link communication is restored.	Check the wiring between the IO-Link device and this product. Check the I/O power supply. Confirm whether the issue is resolved after changing the connection to another port for this product or replacing the IO-Link device and cable.

Code (hexadecimal)	Message	Conditions	Procedure
1100	Timeout generated in ISDU communication	When using ISDU communication to access an IO-Link device setting value, no ISDU communication response is received even after five seconds have passed.	Confirm what you are trying to access on the IO-Link device (index number, write data, etc.).
5600	Checksum error generated in ISDU communication	When using ISDU communication to access an IO-Link device setting value, a mismatch occurs when calculating the checksum of the ISDU communication response.	This could be caused by noise between the IO-Link device and this product. Resolve this through such means as using a separate conduit for the power line, or maintaining distance between the C/Q wires of other IO-Link devices (do not bundle wires together). Confirm that the cable between the IO-Link device and this product is not too long (over 20 m).
5700	Unregulated ISDU communication data length	This occurs when the data length of the ISDU communication response is either “0” or too long, when using ISDU communication to access an IO-Link device setting value.	
6001	Revision ID verification error	The revision ID registered in this product does not match the revision ID of the connected IO-Link device. Process data is not transferred and setting values are not accessed.	Change the setting value for device verification (setting value number: M30) to *None*.
8033	Setting value is too long	This occurs when the data length is too long, when using ISDU communication to access an IO-Link device setting value.	Write data using the data length specified for the IO-Link device.
FF23	Storage data does not match the connected device vendor ID or device ID	This occurs when the value of the vendor ID or device ID of the connected IO-Link device differs from the stored storage data, when device verification (setting value number: M30) is set to *None* and storage data exists (however, this only occurs when power is turned ON).	If an IO-Link device with a vendor ID or device ID that differs from the storage data is connected, connect the correct IO-Link device. If it is safe to delete the storage data stored on this product by port, write with device parameter backup/restore (setting value number: M32) set to *Delete*.
FF24	Storage buffer overload	This occurs when setting value data is too long and cannot be stored, when backing up setting values from an IO-Link device. The data length stored during backup will be *16 bytes + index 18 (model name) length + number of setting values to backup × 4 + total data length of setting values to backup*. This can be stored as long as it is 4,032 bytes or less.	Backup cannot be performed for the connected IO-Link device because the storage data is too large.
FF25	Storage data access was refused	This occurs when access to storage data is locked on the IO-Link device.	If this is required, release the lock setting (index number 12) on the IO-Link device.
FFEA	Duplicate IP addresses	A module with a duplicate IP address was connected.	Confirm the IP address of the connected module.
FFEB	Timeout generated in conflict with ISDU communication	While attempting to perform ISDU communication on the same IO-Link port, ISDU communication was being used elsewhere and communication could not be performed within a time 330 times the cycle time.	Storage takes some time, so wait a short while and try ISDU communication again (access by index number to IO-Link device).
FFEC	EEPROM write protection signal abnormality	The write protection signal is always permitted for the EEPROM used for saving setting values, etc.	This will not cause any immediate issues with operation. However, this indicates a hardware error and the hardware will need to be replaced.
FFED	Failure in EEPROM writing	Setting value, storage data and operation time writing failed.	There is something wrong with the connection to the EEPROM, or the EEPROM has reached its maximum number of writes. It can be rewritten 1,000,000 times.
FFEE	IO-Link trace has stopped automatically	This is a notification indicating that IO-Link tracing has stopped due to an error occurring or the buffer being full.	Read the IO-Link trace data and confirm communication information.

Code (hexadecimal)	Message	Conditions	Procedure
FFEF	Storage was interrupted	An error response was received from the IO-Link device while backing up to or restoring from storage, and the storage stopped operating. Backed up data will not be saved. The data being restored may have partially been transferred to the IO-Link device.	Perform the storage operation (backup or restore) again. If this occurs again, it may be due to noise. If so, resolve this through such means as using a separate conduit for the power line, or maintaining distance between the C/Q wires of other IO-Link devices (do not bundle wires together). Confirm that the cable between the IO-Link device and this product is not too long (over 20 m). If this still occurs, it may be caused by the firmware of the IO-Link device or this product. If so, update or replace the firmware of the IO-Link device or this product.
FFF0	Invalid data in setting value information	There is an invalid character in the setting value information provided by the IO-Link device or in installed IODD data information, or a value exceeding the permitted value was specified.	This is caused by the firmware of the IO-Link device or this product. Update or replace the firmware of the IO-Link device or this product.
FFF1	Writing firmware data is abnormal.	An attempt to write invalid data was made when updating the firmware.	There is something wrong with the data for the firmware being written. Reacquire the file and try again.
FFF3	The revision ID of the IO-Link device to verify is not registered in this product.	This occurs when the revision ID registered to this product is “00 h”, when device verification (setting value number: M30) is set to a value other than *None*.	Change the setting value for device verification (setting value number: M30) to *None*. Or, register the revision ID of the IO-Link device to verify.
FFF4	IO-Link device model name is different	This occurs when the model name (index number 18 [product name] character string) of the registered IO-Link device differs from the model name of the device that is actually connected, when device verification (setting value number: M30) is set to *Type name*. Cyclic communication is not performed with the applicable IO-Link device.	Change the setting value for device verification (setting value number: M30) to a value other than *Type name*. Or, connect the correct IO-Link device.
FFF5	Unsupported setting value version	The version of the setting value data restored to setting value memory is new, and it may not be possible to recognize some of it.	Update the firmware of this product.
FFF6	Internal temperature is too high	This occurs when the temperature of the main CPU exceeds 85°C. The error occurs every 10 minutes.	Lower the operating temperature of this product, install a cooling fan, or lower the output load current (for example, by using a separate relay).
FFF7	EEPROM write frequency is too high	The setting value write count occurs under the following conditions. • The count is cleared if nothing is written for 450 seconds. • When writing 2 times with a frequency of 1 in less than 1 seconds. • When writing 20 times with a frequency of 1 in less than 10 seconds. • When writing 200 times with a frequency of once in less than 110 seconds.	Confirm whether setting value write operations are being performed frequently from the host PLC. Confirm whether IO-Link device setting values are frequently rewritten, with automatic device parameter backup (setting value number: M31) set to *Backup* or *Both*.
FFF8	Software version does not match	There is version incompatibility with the main firmware, host network communication firmware or IO-Link communication logic, and some functions may not operate normally.	Update the FPGA data of this product and the network chip firmware.
FFF9	Network chip is not operating	This occurs when there is no communication between the main CPU and the chip that is performing host network processing.	The network chip may have failed, or power may have turned OFF while the network chip firmware was being updated.

Code (hexa-deci-mal)	Message	Conditions	Procedure
FFFA	Network communication has stopped	This occurs when host EtherNet/IP communication is established but then disconnected. The error will be automatically cleared when the status is restored.	Check whether the Ethernet cable is disconnected, the host PLC has been reset, or the power has turned OFF. Confirm the host PLC parameters. This product may stop being recognized on the network if host PLC network settings are changed.
FFFB	IO-Link device is not connected	This occurs when the IO-Link device is not connected under the following conditions. • The IO-Link device is not connected when running storage functions (manual or automatic backup/ restore of IO-Link device) • Ten seconds elapse without the IO-Link device connecting after the I/O power supply is turned ON, when device verification (setting value number: M30) is set to a value other than *None* • The IO-Link device is not connected when confirming device information or reading/writing a setting value number from the device setting values window • The user switched to the device setting value window when connected to a device that does not support ISDU communication	Connect the IO-Link device properly. Or, change the setting value for device verification (setting value number: M30) to *None*. Or, set the I/O assignment settings (setting value number: M10) to a value other than *IO-Link* for any ports not connected to an IO-Link device.
FFFC	Serial number verification error	This occurs when the registered serial number differs from the serial number of the connected IO-Link device, when device verification (setting value number: M30) is set to *Serial number*. Cyclic communication is not performed with the applicable IO-Link device.	Change the setting value for device verification (setting value number: M30) to a value other than *Serial number*. Or, connect the correct IO-Link device.
FFFD	No backup data	This occurs when storage data is not saved in this product, when restoring storage data to an IO-Link device.	A backup must be performed in order to perform a restore. Select *Backup* in device parameter backup/restore (setting value number: M32) and perform a backup.
FFFE	IO-Link device type ID is different	This occurs when the registered vendor ID or device ID differs from the value of the connected IO-Link device, when device verification (setting value number: M30) is set to a value other than *None*. Cyclic communication is not performed with the applicable IO-Link device.	Change the setting value for device verification (setting value number: M30) to *None*. Or, connect the correct IO-Link device.
FFFF	Type ID of the device to restore is different	This occurs when the vendor ID or device ID differs when storage data is restored (manual restore or automatic restore) to an IO-Link device.	Connect the correct IO-Link device.

Note 1. Errors will be ignored if the same error code occurs within one second on the same port.

Note 2. Up to 20 entries will be stored across all ports in the buffer used to store error information. Error information will begin being discarded beginning with the oldest entry when the number of entries exceeds 20.

A

Appendices

The following is listed.

- Master Unit parameter of this product when reading and writing by Modbus/TCP communication or socket communication
- Socket communication program example

A-1 List of Product Master Unit Parameters

This section lists the Master Unit parameters of this product that can be set or monitored via socket communication.

- For Modbus/TCP communication, register numbers 05500 to 06565 are written or read.
  Refer to *Master Unit Parameters of this product* on page 3-15 in *3-2-2 Detailed List of Register Numbers* on page 3-5 in *3-2 Modbus/TCP Communication* on page 3-4 for further information.

For socket communication, bit 2/3 (parameter access / write or read) with transfer data specification, FF h (Master Unit parameter of this unit) with unit specification, Master Unit parameter number, and target number are specified and written/read.

To set or monitor with the front panel controls, refer to *5-1-4 Master Unit Parameter List with Product Front Panel Operations* of the Common Edition.

The Master Unit parameters are classified as below.

- User settings
- Data for access from PLC

Note that this product's Master Unit parameters are handled as little endian (beginning from the lower byte).

A-1-1 User Settings

For socket communication, specify the Master Unit parameter number and target number in the table below.

Note In the right column of the table below, items that can be accessed via product front panel operation, Modbus/TCP communication or socket communication are marked with l.

Master Unit parameter numbers DEC (HEX)	Target number 0 to 15: Corresponds to channel number, 0: This product	Data name	R/W	Format	Length bytes	Value		Front panel controls (Master Unit parameter numbers)	Socket communication
1(1 h)	0	IO-Link Master Unit parameters reset	RW	UINT	1	0 (default value)	None	●(M1)	●
						1	Initialize I/O assignment for all ports in IO-Link mode. (Storage data, display language, network type, and network No. will not be initialized)		
						2	As above, initialize I/O assignment for all ports in PNP input mode.		
						3	As above, initialize I/O assignment for all ports in NPN input mode.		
						4	As above, initialize I/O assignment for all ports in PNP output mode.		
						5	As above, initialize I/O assignment for all ports in NPN output mode.		
						6	As above, initialize I/O assignment for all ports as unused.		
						7	Delete the installed IODD data.		
						8	Initialize including network setting and language as well. I/O assignment is unused. IODD data is not deleted.		
2(2 h)	0	Network Network type	RW	UINT	1	Specifies the host industrial network type.		●(M2)	●
						2 (default value)	EtherNet/IP		
						5	Ethernet & Modbus/TCP		
						6	CC-Link IE Field Basic		
4(4 h)	0	Display brightness	RW	UINT	1	1 to 20(default value: 7)	Display brightness Values multiplied by 5 are equivalent to % display.	●(M4)	●

Master Unit parameter numbers DEC (HEX)	Target number 0 to 15: Corresponds to channel number, 0: This product	Data name	R/W	Format	Length bytes	Value		Front panel controls (Master Unit parameter numbers)	Socket communication
5(5 h)	0	Language	RW	UINT	1	0 (default value)	English	●(M5)	●
						1	Japanese		
						2	German		
						3	Chinese (Simplified)		
						4	French		
						5	Spanish		
						6	Portuguese		
						7	Italian		
						9	Korean		
						10	Chinese (Traditional)		
6(6 h)	0	Time zone -UTC-	RW	INT	1	-96 to 96(default value: 0)	Set the time zone (in 15-minute increments) for the displayed network time (parameter number: M65). If the network time set from the host is based on UTC, setting the time zone as 9 x 4 = 36 will display the Japan standard time (JST).	●(M6)	●
8(8 h)	0	User tag name of this IO-Link Master Unit	RW	STRING	1 to 32	Up to 32 characters	Set the product user tag name up to 32 characters.	●(M8)	●

Master Unit parameter numbers DEC (HEX)	Target number 0 to 15: Corresponds to channel number, 0: This product	Data name	R/W	Format	Length bytes	Value		Front panel controls (Master Unit parameter numbers)	Socket communication
9(9 h)	0	I/O batch setting	RW	UINT	1	0 (default value)	None	●(M9)	●
						1	Change I/O assignment settings for all ports to IO-Link mode.		
						2	Change I/O assignment for all ports to PNP input mode.		
						3	Change I/O assignment for all ports to NPN input mode.		
						4	Change I/O assignment for all ports to PNP output mode.		
						5	Change I/O assignment for all ports to NPN output mode.		
						6	Change I/O assignment for all ports to unused.		
10(A h)	0 to 15	I/O assignment settings	RW	UINT	1	0	IO-Link Mode	●(M10)	●
						1	PNP input mode: Internal pull-down resistance is enabled.		
						2	NPN input mode: Internal pull-up resistance is enabled.		
						3	PNP output mode		
						4	NPN output mode		
						5 (default value)	Not used		
11(B h)	0 to 15	Input filter	RW	UINT	1	0 (default value)	None	●(M11)	●
						1	0.1 ms		
						2	1 ms		
						3	5 ms		
						4	10 ms		
						5	20 ms		
12(C h)	0 to 15	Input hold time	RW	UINT	1	0 (default value)	None	●(M12)	●
						1	1 ms		
						2	15 ms		
						3	100 ms		

Master Unit parameter numbers DEC (HEX)	Target number 0 to 15: Corresponds to channel number, 0: This product	Data name	R/W	Format	Length bytes	Value		Front panel controls (Master Unit parameter numbers)	Socket communication
13(Dh)	0 to 15	Handling of IO-Link communication and network error generation	RW	UINT	1	0 (default value)	Clear	●(M13)	●
						1	Input hold		
						2	Output hold		
						3	All Hold		
16(10h)	0 to 15	Bit assignment of Process input data	RW	INT	1	-1	Turns ON RX0□ for the corresponding port when the IO-Link device's process input data is valid.		●
						0 to 254	Specifies which bit from the first bit of the process input data defined as boolean (bit) data in the IO-Link device's process input data information to use as the input ON/OFF information. Data formats other than Boolean, such as Integer, cannot be specified. If the IO-Link device does not provide an index (0x0E) for the process input data format, you must install the IODD for the corresponding IO-Link device.		●

Master Unit parameter numbers DEC (HEX)	Target number 0 to 15: Corresponds to channel number, 0: This product	Data name	R/W	Format	Length bytes	Value		Front panel controls (Master Unit parameter numbers)	Socket communication
17(11 h)	0 to 15	Input port to output	RW	UINT	1	0	Turns ON or OFF the PNP/NPN output based on the bit information of the process output data.		●
						1 to 16	Turns ON or OFF the PNP/NPN output based on the input bit information for “the specified port number + 1”. Use this when the IO-Link device’s process input data bit information needs to be output as-is to the outside.		●
18(12 h)	0 to 15	Pulse input	RW	UINT	1	None	Does not use the counter function.	●(M18)	●
						ABZ	Uses all of phase A, phase B, and phase Z of the incremental encoder. The rotation count is incremented or decremented at the rising edge of the Z phase, with the count value cleared. The count value is incremented at both the rising edge and falling edge of the input pulse.		
						AB	Uses only phase A and phase B of the incremental encoder. The count value is incremented at both the rising edge and falling edge of the input pulse.		
						Single	The count value is incremented at both the rising edge and falling edge of the input pulse.		
						Frequency	Measures the frequency of the single-phase pulse input in units of 1 Hz every second.		
						High response	This is used when a high-speed response is required. It measures the frequency of the AB-phase or single-phase pulse input at the rising edge of each input cycle.		

Master Unit parameter numbers DEC (HEX)	Target number 0 to 15: Corresponds to channel number, 0: This product	Data name	R/W	Format	Length bytes	Value		Front panel controls (Master Unit parameter numbers)	Socket communication
19 (13 h)	0 to 15	Input inversion	RW	UINT	1	None	Passes the input status of the specified port as-is to the inside.	●(M19)	●
						Reverse	Passes the input status of the specified port to the inside after reversing it.		
21(15 h)	0 to 15	I/O synchronization	RW	UINT	2	0 (default value)	Asynchronous IO-Link communication uses individual devices' fastest times. The digital I/O status is also continually updated.	●(M21)	●
						2	Using the product's internal timer, at an 0.4 ms cycle, synchronize IO-Link communication or digital I/O transfer between multiple ports with the same setting (I/O synchronization).		
						3	As above, synchronization at 0.8 ms cycle		
						4	As above, synchronization at 1.6 ms cycle		
						5	As above, synchronization at 3.2 ms cycle		
						6	As above, synchronization at 6.4 ms cycle		
						7 to 1003	Specify the IO-Link communication cycle time at 0.4 ms to 100.0 ms.		

Master Unit parameter numbers DEC (HEX)	Target number 0 to 15: Corresponds to channel number, 0: This product	Data name	R/W	Format	Length bytes	Value		Front panel controls (Master Unit parameter numbers)	Socket communication
22(16 h)	0 to 15	Lowest frequency	RW	UINT	1	27.2 Hz	Measures the lowest frequency with a cycle of 16 rising edges of the pulse input. This ensures stable measured values with improved frequency measurement resolution, although the measurable lowest frequency is 27.2 Hz.	●(M22)	●
						13.6 Hz	Measures the lowest frequency with a cycle of 8 rising edges of the pulse input.		
						6.8 Hz	Measures the lowest frequency with a cycle of 4 rising edges of the pulse input.		
						3.4 Hz	Measures the lowest frequency with a cycle of 2 rising edges of the pulse input.		
						1.7 Hz	Measures the lowest frequency with a cycle of 1 rising edges of the pulse input.		
29(1D h)	0 to 15	Device ID	RW	UINT	3	0x0 - 0xFFFFFF(default value: 0x 0)	This is the device ID used for verification with IO-Link devices. Refer to *M29. Device ID* in *5-1-4 Master Unit Parameter List with Product Front Panel Operations* of the Common Edition for details.	●(M29)	●
						0 (default value)	None **Note** Even for *None*, if storage data is already saved in the product, when turning on the power (unit and I/O power), the IO-Link device type ID and storage data type ID will be verified. An error (FF23 h) will occur if the type ID does not match. The revision ID is not verified.	●(M30)	●

Master Unit parameter numbers DEC (HEX)	Target number 0 to 15: Corresponds to channel number, 0: This product	Data name	R/W	Format	Length bytes	Value		Front panel controls (Master Unit parameter numbers)	Socket communication
30(1Eh)	0 to 15	Device verification	RW	UINT	1	1	Type ID (vendor ID and device ID) verification If the registered type ID and connected device type ID are different, an error (FFFE h: type ID verification error) is generated and the process data with the relevant IO-Link device is treated as invalid. As well, if IO-Link communication is not established within 10 seconds of turning on the I/O power, an error (FFFB h: IO-Link device not connected) is generated. At the same time, the revision ID is also verified.		
						2	Type ID and serial number verification Even if the type IDs (vendor ID and device ID) match, if the serial number is different, an error (FFFC h: serial number verify error) is generated, and the process data with the relevant IO-Link device is invalidated. Other operations are the same as Parameter 1.		
						3	Type ID and device model name verification Even if the type IDs (vendor ID and device ID) match, if the device model name is different, an error (FFF4 h: model name verification error) is generated, and the process data with the relevant IO-Link device is invalidated. Other operations are the same as Parameter 1.		
						0 (default value)	None	●(M31)	●

Master Unit parameter numbers DEC (HEX)	Target number 0 to 15: Corresponds to channel number, 0: This product	Data name	R/W	Format	Length bytes	Value		Front panel controls (Master Unit parameter numbers)	Socket communication
31(1Fh)	0 to 15	Automatic device parameter backup	RW	UINT	1	1	Auto backup When IO-Link device Parameters are changed, they are automatically backed up in this product. If IO-Link device setting values are changed from this product, the backup operation will automatically start 10 seconds after the last change. **Note** When a device with a different vendor ID or device ID is connected, backup is executed at every startup, so reset *Device verification* (Parameter number: M30) to correct the verification error (FFFE h) as soon as possible.		
						2	Auto restoration Automatically restore (download parameters from this product to the IO-Link device) if the IO-Link device parameters differ from the data stored on this product during startup. In this case, note that even if the IO-Link device parameter is changed, it will be overwritten at the next startup with the data saved in this product.		
						3	Perform auto backup and auto restoration as above together The storage data stored on this product will always match the IO-Link device parameters. In other words, IO-Link device parameters are backed up to this product whenever they are changed. If any parameters are different from the IO-Link device during startup, the parameters stored in this product will be restored.		

Master Unit parameter numbers DEC (HEX)	Target number 0 to 15: Corresponds to channel number, 0: This product	Data name	R/W	Format	Length bytes	Value		Front panel controls (Master Unit parameter numbers)	Socket communication
32(20 h)	0 to 15	Device parameter backup/restore	RW	UINT	1	0 (default value)	None	●(M32)	●
						1	Execute backup (uploading Parameters from devices to this product) manually		
						2	Execute restoration (downloading Parameters from the product to devices) manually		
						3	Delete backup data saved in the product manually		
33(21 h)	0 to 15	Conditions for applying IODD data	RW	UINT	1	0	Device	●(M33)	●
						1 (default value)	IODD		
40(28 h)	0 to 15	Process input data words allocation	RW	UINT	1	0 to 16(default value: 2)	Process input data words allocated to the input cycle communication area If the parameter is 0 words, it will be assigned in a packed manner.	●(M40)	●

Master Unit parameter numbers DEC (HEX)	Target number 0 to 15: Corresponds to channel number, 0: This product	Data name	R/W	Format	Length bytes	Value		Front panel controls (Master Unit parameter numbers)	Socket communication
41(29h)	0 to 15	Process output data words allocation	RW	UINT	1	0 to 16(default value: 2)	Process output data words allocated to the output cycle communication area If the parameter is 0 words, it will be assigned in a packed manner.	●(M41)	●
42(2Ah)	0	Process data words auto allocation	RW	UINT	1	0 (default value)	None	●(M42)	●
						1	Auto allocation		
43(2Bh)	0 to 15	Process data LSB/MSB	RW	UINT	1	0 (default value)	Little endian LSB	●(M43)	●
						1	Big endian MSB		
						2	Little endian MSB		
						3	Swap bytes		

Master Unit parameter numbers DEC (HEX)	Target number 0 to 15: Corresponds to channel number, 0: This product	Data name	R/W	Format	Length bytes	Value			Front panel controls (Master Unit parameter numbers)	Socket communication
44(2Ch)	0	Time stamp	RW	UINT	1	0 (default value)	No	Refer to *M44. Timestamp* in *5-1-4 Master Unit Parameter List with Product Front Panel Operations* of the Common Edition for details.	●(M44)	●
						1	Serial			
						2	Common Era BCD			
						3	Serial + Parity			
						4	Common Era + Parity			
51(33h)	0 to 15	IO-Link cycle time -Process value-	R	UINT	2	1 to 1000	0.1 ms increments		●(M51)	●
52(34h)	0 to 15	IO-Link transmission rate	R	UINT	1	0	Not communicating		●(M52)	●
						1	COM1			
						2	COM2			
						3	COM3			

Master Unit parameter numbers DEC (HEX)	Target number 0 to 15: Corresponds to channel number, 0: This product	Data name	R/W	Format	Length bytes	Value		Front panel controls (Master Unit parameter numbers)	Socket communication
53(35 h)	0 to 15	IO-Link communication error	R	UINT	1	0 to 255	To clear data, write 1 in offset +0 of parameter 114 h.	●(M53)	●
54(36 h)	0 to 15	IO-Link ISDU checksum errors	R	UINT	1	0 to 255	To clear data, write 1 in offset +0 of parameter 114 h.	●(M54)	●
55(37 h)	0 to 15	IO-Link transmission rate error	R	INT	2	-1000 to 1000	IO-Link transmission rate error (0.1% unit, signed)	●(M55)	●
56(38 h)	0 to 15	IO-Link signal width error	R	INT	2	-1000 to 1000	IO-Link transmission signal width error (0.1% unit, signed)	●(M56)	●

Master Unit parameter numbers DEC (HEX)	Target number 0 to 15: Corresponds to channel number, 0: This product	Data name	R/W	Format	Length bytes	Value		Front panel controls (Master Unit parameter numbers)	Socket communication
57(39 h)	0 to 15	IO-Link communication mode	R	UINT	1	0 to 7	0: NOCOM 1: STARTUP 2: PREOPERATE 3: OPERATE 5: STARTUP (Rev.1.0) 6: PREOPERATE (Rev.1.0) 7: OPERATE (Rev.1.0) Refer to *M57. Communication mode* in *5-1-4 Master Unit parameter list with product front panel operation* of the Common Edition for details.	●(M57)	●
58(3A h)	0 to 15	Storage backup time	R	STRING	17	00:00:00 00/01/01 to 23:59:59 99/12/31	The storage data backup time is displayed as below. Example: December 31, 2022, 18:59:00 **18:59:00 22/12/31**	●(M58)	●
60(3C h)	0	System program version of IO-Link Master Unit	R	UINT	4	P.PPNNLLFF	Display the data version of the program, etc. P.PP: Firmware, NN: Network chip, LL: Logic, FF: Font 4 bytes in decimal.	●(M60)	●

Master Unit parameter numbers DEC (HEX)	Target number 0 to 15: Corresponds to channel number, 0: This product	Data name	R/W	Format	Length bytes	Value		Front panel controls (Master Unit parameter numbers)	Socket communication
61(3Dh)	0	Total operation hours of IO-Link Master Unit	R	UINT	4	0 to 2097151 (239 years)	Product cumulative operating time information (unit: hours)	●(M61)	●
62(3Eh)	0	Display driven hours	R	UINT	4	0 to 262143 (30 years)	Product display operating time information (unit: hours)	●(M62)	●
63(3Fh)	0	I/O power supply voltage	R	UINT	2	0 to 308	Voltage supplied to product I/O power supply (unit: 0.1 V)	●(M63)	●
64(40h)	0	Internal temperature	R	UINT	2	-2739 to 10457	Product internal temperature (unit: 0.1°C)	●(M64)	●
65(41h)	0	Network time	R	STRING	22	00:00:00 00/01/01 to 23:59:59 99/12/31	Time and date Example: December 31, 2022, 18:59:00.0000 **18:59:00.0000 22/12/31**	●(M65)	●

Master Unit parameter numbers DEC (HEX)	Target number 0 to 15: Corresponds to channel number, 0: This product	Data name	R/W	Format	Length bytes	Value		Front panel controls (Master Unit parameter numbers)	Socket communication
66(42 h)	0	Network communication cycle	R	UINT	2	0 to 15000	It is irrelevant when the network type is *Ethernet & Modbus/TCP*.	●(M66)	●
67(43 h)	0	Network timeout	R/W	UINT	2	0 to 30000	When the network type is *Ethernet & Modbus/TCP*, if the Ethernet communication is interrupted and the specified time elapses, a network communication stop error (FFFA h) occurs. The specified time can be specified from 0 to 30000 (300 seconds) in 10 ms increments. If it set to the default value of 0, even if Ethernet communication is interrupted, it is considered that the data has been updated internally at a cycle of about 10 ms and no error occurs. Even in such a case, an error will occur if the Ethernet cable is disconnected.	●(M67)	●
80(50 h)	0	Default gateway	R/W	UINT	4	0.0.0.0 to 255.255.255.255	Change if otherwise specified. If not, leave the default value. The default value is 0.0.0.0.	●(M80)	●
81(51 h)	0	Subnet mask	R/W	UINT	4	0.0.0.0 to 255.255.255.255	Change if otherwise specified. If not, leave the default value. The default value is 255.255.255.0	●(M81)	●

Master Unit parameter numbers DEC (HEX)	Target number 0 to 15: Corresponds to channel number, 0: This product	Data name	R/W	Format	Length bytes	Value		Front panel controls (Master Unit parameter numbers)	Socket communication
82(52 h)	0	IP address	R/W	UINT	4	0.0.0.0 to 255.255.255.255	Set the IP address of this product to any value. This is set only when changing the default value of 192.168.250.xxx (xxx is the value set by the rotary switch). Turning the front rotary switch while the power is on automatically switches to *M82. IP address* setting window	●(M82)	●

A-1-2 Data For Access from PLC

For socket communication, specify the Master Unit parameter number and target number in the table below.

Note In the right column of the table below, items that can be accessed via product front panel operation or socket communication are marked with ●

Master Unit parameter numbers	Target number	Data name	R/W	Off-set	Value	Front panel controls	Socket communication
100 h	0 to 15(corresponds to port number)	Latest error code readout by port	R	+0	Error code lower		●
				+1	Error code upper		
				+2	Lower designated index number causing error		
				+3	Upper designated index number causing error		
				+4	Designated index number causing error		
				+5 to 26	Error time and date text string (in order of generation) See the form below. HH:MM:SS YY:MN:DD (HH: hour, MM: minute, SS: second, YY: year last two digits, MN: month, DD: day) Ex.: Generated at 18:59:00 on February 19, 2020: 18:59:00 20/02/19 **Note** When 100-µs unit data is added, HH:MM:SS:ssss YY:MN:DD (ssss: seconds in 100 µs increments).		
101 h	0 to 15(corresponds to port number)	Event data readout by port	R	+0	Event flag		●
				+1	Event byte order 1 type *1		
				+2	Event code upper		
				+3	Event code lower		
				+4	0		
				+5	Event byte order 2 type *1		
				+6	Event code upper		
				+7	Event code lower		
				+8	0		
				+9	Event byte order 3 type *1		
				+10	Event code upper		
				+11	Event code lower		
				+12	0		
				+13	Event byte order 4 type *1		
				+14	Event code upper		
				+15	Event code lower		
				+16	0		
				+17	Event byte order 5 type *1		
				+18	Event code upper		
				+19	Event code lower		
				+20	0		
				+21	Event byte order 6 type *1		
				+22	Event code upper		
				+23	Event code lower		
				+24	0		

Master Unit parameter numbers	Target number	Data name	R/W	Off-set	Value	Front panel controls	Socket communication
				+25 to 150	The text string for generation times of event byte orders 1 to 6 will be displayed (in order) after the event byte order. Ex. 1: Event byte order 1 only generated at 19:15:32 on June 24, 2020: (1) 19:15:32 20/06/24 Ex. 2: Event byte orders 1 to 6 all generated at different times: (1) 19:15:32 20/06/24 (2) 19:10:18 20/6/24 (3) 18:25:32 20/06/24 (4) 19:05:48 20/6/24 (5) 19:15:32 20/06/24 (6) 19:10:18 20/6/02		
102 h	0	Parameter readout time from IO-Link devices (binary data)	R	+0	15.265 µs unit data lower	-	●
				+1	15.265 µs unit data upper		
				+2	Set 0 as January 1, 1970, 00:00:00, counting data every second, least-significant		
				+3	As above 2nd byte		
				+4	As above 3rd byte		
				+5	Same as above, most-significant		
				+6	Lower parity information with readout Parameters as word units with exclusive disjunction (XOR) at 35 AC h		
				+7	Same as above, upper		
	1	Parameter readout time from IO-Link devices (CE BCD data)	R	+0	15.265 µs unit data lower		
				+1	15.265 µs unit data upper		
				+2	second		
				+3	minute		
				+4	hour		
				+5	day		
				+6	month		
				+7	year		
				+8	Lower parity information with readout Parameters as word units with exclusive disjunction (XOR) at 0 x35AC		
				+9	Same as above, upper		
	2	Parameter readout time from IO-Link devices (text string data)	R	-	A 27-character text string will be displayed. Ex.: 2020/2/29 18:59:0.1234 parity 8B61 h → 18:59:00.1234 20/02/29 8B61		
103 h	0 (This product)	Event/error message readout	R	-	Executing readout switches the display to the event/error display window.		●
				+0	Applicable port (0 to 15)		
				+1	Designated index number causing error/event qualifier		
				+2	Lower index number in error		
				+3	Upper index number in error		
				+4	Error code lower		
				+5	Error code upper		
				+6	Message text string		
				+7	(continues)		

Master Unit parameter numbers	Target number	Data name	R/W	Off-set	Value		Front panel controls	Socket communication
104 h	0 (This product)	Button operation	R	+0	Returns the status of the currently pressed button. Value is the same as below.		-	●
			W	-	Writes the decimal below. 1: Right button operation 2: Left button operation 4: ↑ button operation 8: ↓button operation 16: **CANCEL** button operation 32: **ENTER** button operation	Remotely operates the product buttons. Operation is the same as pressing once, not holding. **Note** Even if buttons are remotely operated here, the display off timer will not be cleared.		
105 h	0	Event/error clear	R	-	Executes operation equivalent to pressing the ENTER, Down, and Up buttons on the event/error menu. Executes with readout			●
	1	Go to next event						
	2	Return to previous event						
106 h	0 (This product)	Operation lock	RW	-	Bit 0: Master Unit parameter write lock	**Note** Start up while pressing the **CANCEL** button to release the lock temporarily.	-	●
					Bit 1: Device Parameter write lock			
					Bit 2: /Event/error clear operation lock			
					Bit 3: All button operation lock			
107 h	0	Readout display text string (1st row)	R	+0 to +499	Text string on display Character codes: Japanese: Shift-JIS Simplified Chinese: GB2312 Traditional Chinese: Big-5 Korean: EUC-kr		-	●
	1	Readout display text string (2nd row)	R					
108 h	1	Right button operation response	R	-	Operation response when remotely operating product buttons using the specified target number is stored under the following decimals. Responses are as below. 1: Referencing Parameters 10 to 19: Digit position of Parameters being edited (equivalent to digits 1 to 10) 100 up: Digit position in text string being edited Values up to 255 are 1-byte responses; values from 256 up are 2-byte responses. Even if buttons are remotely operated here, the display off timer will not be cleared. When specifying over 192 values, only 192 will be received. The readout values at that point are port numbers.		-	●
	2	Left button operation response						
	4	Up button operation response						
	8	Down button operation response						
	16	**CANCEL** button operation response						
	32	**ENTER** button operation response						
	192	Process data information update operation response						
109 h	0 (This product)	Readout of port numbers displayed	R	+0	When the IO-Link device numbers specified in writing are out of range, an out-of-range error will be generated without switching. The readout will return the current port numbers.		-	●

Master Unit parameter numbers	Target number	Data name	R/W	Off-set	Value	Front panel controls	Socket communication
10A h	0 (This product)	Find Me requests	RW	+0	0: Cancel a Find Me request to the product. 1: Send a Find Me request to the product. During Find Me, the event/error display will blink and the display will show *I am here*. Press any button to clear Find Me.	-	●
10D h	0 to 15(corresponds to port number)	Model name thumb value registration for verification	R	+0 to +1	Write the model name as a text string to calculate and store the thumb value. The readout value will be a 16-bit thumb value.	-	●
			W	+0 to +63			
10E h	0 to 15(corresponds to port number)	Vendor ID and device ID registered for verification	RW	+0 to +1	Registered IO-Link device vendor ID	-	●
				+2 to +4	Registered IO-Link device ID		
				+5	0		
				+6	Registered IO-Link device revision ID Value 10 h indicates that the revision ID is 1.0. Value 11 h indicates that the revision ID is 1.1.		
10F h	0 to 15(corresponds to port number)	Write serial number to be registered	W	+0 to +15	Writes the serial number used for verification with IO-Link devices.	-	●
110 h	0 to 15(corresponds to port number)	Readout of vendor ID/device ID of storage data backed up in this product and registered vendor ID/ device ID and serial number	R	+0 to +1	Vendor ID in product backup data **Note** In the case of a vendor ID mismatch, it will be invalid and 0 will be read out.	-	●
				+2 to +4	Device ID in product backup data **Note** In the case of a device ID mismatch, it will be invalid and 0 will be read out.		
				+5	0		
				+6 to +7	Storage backup execution times since power on		
				+8 to +9	Storage restoration execution times since power on		
				+10 to +26	Text string of date and time of backup		
111 h	0 (This product)	Diagnostic information	R	+0	Main memory (EEPROM) checksum error count	-	●
				+1	Sub memory (flash) checksum error count		
				+2	Always 0.		
				+3	Bit 0 to 5: Result of pressing switch alone in past		
				+4 to 5	Display device drive power supply voltage (0.1 V unit)		
				+6	Noncyclic communication timeout count		
				+7	Main memory (EEPROM) write frequency. Updated every 450 seconds, with an alarm generated at over 200.		

Master Unit parameter numbers	Target number	Data name	R/W	Off-set	Value	Front panel controls	Socket communication
				+8	Maximum value of the internal temperature measured inside the product. (unit: °C)		
				+9	Parameter version number (0)		
				+10	Network communication error count		
				+11	Storage function state number (normally 0, changes when storage-related functions operate)		
				+12 to +13	Interlock status (Always 0)		
112 h	0 (This product)	Output overcurrent status	R	+0 to +1	Current output overcurrent status. Bits 0 to 15 are equivalent to I/O ports 0 to 9 and A to F.	-	●
				+2 to +17	Output overcurrent count for ports 0 to F. Counted up to 255.		
			W	+0	Write 1 to clear all ports' output overcurrent count. Write 2 to 17 to clear ports 0 to F separately.		
113 h	0 to 15(corresponds to port number)	Read vendor ID and device ID of IO-Link device currently connected	R	+0 to +1	Vendor ID	-	●
				+2 to +4	Device ID		
				+5	Always 0.		
				+6	Revision ID		
				+7	Always 0.		
				+8 to +9	Function ID		

Master Unit parameter numbers	Target number	Data name	R/W	Off-set	Value	Front panel controls	Socket communication
114 h	0 to 15(corresponds to port number)	IO-Link device diagnostic information	R	+0	Number of IO-Link communication errors.	-	●
				+1	IO-Link ISDU sum error count		
				+2 to +3	IO-Link transmission rate error (0.1% unit, signed)		
				+4 to +5	IO-Link transmission signal width error (0.1% unit, signed)		
				+6 to +7	Actual cycle time for IO-Link communication (0.1 ms increments)		
				+8	IO-Link transmission rate (0: not communicating, 1: COM1, 2: COM2, 3: COM3)		
				+9	Process input data byte count		
				+10	Process output data byte count		
				+11	IO-Link communication status (0: NOCOM, 1: STARTUP, 2: PREOPERATE, 3: OPERATE, 4 to 7: Same as 0 to 3 with IO-Link Revision 1.0)		
				+12	IO-Link device on-request data byte count		
				+13	Process input data invalid (0: valid, 1: invalid)		
	0	Clear IO-Link communication error count	W	+0	Write 1 to clear all ports' IO-Link communication error count. Write 2 to 17 to clear ports 0 to F separately.		
	1	Clear IO-Link ISDU checksum error count		+0	Write 1 to clear all ports' ISDU checksum error count. Write 2 to 17 to clear ports 0 to F separately.		
1FE h	0 to 31	Readout of IO-Link communication trace data	R	+0 to +127	From the position where the port number was specified, 128 bytes of trace send/receive data will return as a hexadecimal text string. When it is read out, the trace ends. The specified port number 31 is the most recent trace data, 30 is the previous one, and 0 is the oldest data. A ">" is inserted at the start of the send data and a ":" at the start of the receive data.	-	●
	0	Start of IO-Link communication trace	W	+0	Bits 3 to 0: IO-Link device port number Saves the specified IO-Link device send/receive data to buffer memory (4096 bytes). Bit 5: Set to 1 to stop the trace when an IO-Link communication error is generated. Bit 6: Set to 1 to stop the trace when an ISDU communication negative response is generated. Bit 7: Set to 1 to stop the trace when the buffer is full.	-	●
				+1	Bit 0: Set to 1 to add a line break instead of ">" at the start of send data.		
1FF h	0 (This product)	Readout of MAC register number and serial number	R	+0 to +5	6-byte MAC register number data	-	●
				+6	I/O voltage measurement calibrated value		
				+7	Internal temperature measurement calibrated value		
				+8 to 23	Product serial number (16 characters)		

*1. Details of event type (from IO-Link specifications)

Bit 7 to 6 Event occurrence type 0: Reservation, 1: Single, 2: Occurrence, 3: Resolution
Bit 5 to 4 Event type 0: Reservation, 1: Notification, 2: Warning, 3: Error
Bit 3 Event source 0: Device, 1: Master
Bit 2 to 0 Event cause 0: Unknown, 1 to 3: Reservation, 4: Application, 5 to 7: Reservation

A-2 Socket Communication Program Example

Here is an example of socket communication program.

Note When performing socket communication from a PC to this product, specify a fixed IP address and a subnet mask of 255.255.255.0. The default gateway is blank (or 0.0.0.0).
If it is necessary to make the upper 3 digits of the IP address of the PC the same as the upper 3 digits of the IP address of this product, add the IP address as a port using the **Detail Setting** button in *Properties* of *Internet Protocol Version 4 (TCP/IPv4)*.


```
// Sample program for GD Series Ethernet/TCP Visual Studio C++
//
// GD_sample.cpp: Defines the entry point for console application.
//

#include "stdafx.h"

// Added include file
#include &lt<winsock2.h&gt>
#include &lt<ws2tcpip.h&gt>

// Add ws2_32.lib; to Configuration properties -> Linker -> Input -> Additional dep
endent files.
// For the project character set, use the multi-byte character set instead of Unico
de.
```

Various Definitions

```
// Various definitions
//#define TCPIP                                     // Unspecified: UDP, specified:
TCP/IP

#define PDIN_LEN      68                            // Process input data length
#define PDOUT_LEN     68                            // Process output data length
#define PARAM_LEN     400                           // Maximum data length of parame
ters
#define RESPCODE_LEN  2                             // Response code length

// Definition of the unit to access
#define UNIT_MAIN     0xFF                          // Specify the GD-ILM16C-MLP uni
t
#define UNIT_IOL_CH0  0x80                          // IO-Link device on port 0
#define UNIT_IOL_CH1  0x81                          // IO-Link device on port 1
```

```
#define UNIT_IOL_CH2  0x82                          // IO-Link device on port 2
#define UNIT_IOL_CH3  0x83                          // IO-Link device on port 3
#define UNIT_IOL_CH4  0x84                          // IO-Link device on port 4
#define UNIT_IOL_CH5  0x85                          // IO-Link device on port 5
#define UNIT_IOL_CH6  0x86                          // IO-Link device on port 6
#define UNIT_IOL_CH7  0x87                          // IO-Link device on port 7
#define UNIT_IOL_CH8  0x88                          // IO-Link device on port 8
#define UNIT_IOL_CH9  0x89                          // IO-Link device on port 9
#define UNIT_IOL_CHA  0x8A                          // IO-Link device on port A
#define UNIT_IOL_CHB  0x8B                          // IO-Link device on port B
#define UNIT_IOL_CHC  0x8C                          // IO-Link device on port C
#define UNIT_IOL_CHD  0x8D                          // IO-Link device on port D
#define UNIT_IOL_CHE  0x8E                          // IO-Link device on port E
#define UNIT_IOL_CHF  0x8F                          // IO-Link device on port F

// Definition of transfer data
#define PDIN_DATA     1                             // Read process input data
#define PDOUT_DATA    2                             // Write process output data
#define PARAM_READ    4                             // Read parameter
#define PARAM_WRITE   (PARAM_READ + 8)              // Write parameter
#define BACK_STATUS   0x10                          // Background access mode
#define BACK_READ     (PARAM_READ + BACK_STATUS)    // Background reading of paramet
er
#define BACK_WRITE    (PARAM_WRITE + BACK_STATUS)   // Background writing of paramet
er

// Response code
#define RESP_PWRITE   0x41                          // Write parameter
#define RESP_NODATA   0x4E                          // No data
#define RESP_PDOUT    0x50                          // Write process output data
#define RESP_PREAD    0x61                          // Read parameter
#define RESP_PDIN     0x70                          // Read process input data
#define RESP_PBUSY    0x62                          // Parameter read busy

// Ethernet/TCP protocol definition
typedef struct{
  BYTE command;                       // Command
  BYTE xferData;                      // Transfer data specification
  BYTE unit;                          // Unit specification
  BYTE indexL;                        // Lower index number
  BYTE indexH;                        // Upper index number
  BYTE subindex;                      // Subindex number
  BYTE readLenL;                      // Lower byte length to read
  BYTE readLenH;                      // Upper byte length to read
} txDataNoWrite;                      // No write

typedef struct {
```

```
  BYTE command;                    // Command
  BYTE xferData;                   // Transfer data specification
  BYTE unit;                       // Unit specification
  BYTE indexL;                     // Lower index number
  BYTE indexH;                     // Upper index number
  BYTE subindex;                   // Subindex number
  BYTE pdOutData[PDOUT_LEN];       // Process output data
  BYTE readLenL;                   // Lower byte length to read
  BYTE readLenH;                   // Upper byte length to read
} txDataPDout;                     // Write process output data only

typedef struct {
  BYTE command;                    // Command
  BYTE xferData;                   // Transfer data specification
  BYTE unit;                       // Unit specification
  BYTE indexL;                     // Lower index number
  BYTE indexH;                     // Upper index number
  BYTE subindex;                   // Subindex number
  BYTE writeData[PARAM_LEN];       // Parameter write data
} txDataPwrite;                    // Write parameter only

typedef struct {
  BYTE command;                    // Command
  BYTE xferData;                   // Transfer data specification
  BYTE unit;                       // Unit specification
  BYTE indexL;                     // Lower index number
  BYTE indexH;                     // Upper index number
  BYTE subindex;                   // Subindex number
  BYTE pdOutData[PDOUT_LEN];       // Parameter write data
  BYTE writeData[PARAM_LEN];       // Parameter write data
} txDataPDoutPwrite;               // Write process output data and parameter

// Send buffer union
union txBlock {
  txDataPDoutPwrite pdOutPwrite;   // Write process output data and parameter
  txDataPwrite pWrite;             // Write parameter
  txDataPDout pdOut;               // Write process output data
  txDataNoWrite noWrite;           // Read only
};

typedef struct {
  BYTE respCode;                   // Response code (RESP_NODATA, RESP_PDOUT, RE
SP_PWRITE)
  BYTE errCode;                    // Error code (extension error code)
} rxDataNoRead;                    // No read

typedef struct {
```

```
  BYTE respCode;                    // Response code (RESP_PDIN)
  BYTE errCode;                     // Error code (extension error code)
  BYTE pdInData[PDIN_LEN];          // Process input data
} rxDataPDin;                       // Read process input data

typedef struct {
  BYTE respCode;                    // Response code (RESP_PREAD)
  BYTE errCode;                     // Error code (extension error code)
  BYTE readData[PARAM_LEN];         // Parameter read data
} rxDataPread;                      // Read parameter only

typedef struct {
  rxDataPDin pdIn;                  // Response to process read input data
  rxDataPread pRead;                // Response to read parameter
} rxDataPDinPread;

typedef struct {
  rxDataPDin pdIn;                  // Response to process read input data
  rxDataNoRead pWrite;              // Response to write parameter
} rxDataPDinPwrite;

typedef struct {
  rxDataNoRead pdOut;               // Response to write process output data
  rxDataPread pRead;                // Response to read parameter
} rxDataPDoutPread;

typedef struct {
  rxDataNoRead pdOut;               // Response to write process output data
  rxDataNoRead pWrite;              // Response to write parameter
} rxDataPDoutPwrite;

typedef struct {
  rxDataPDin pdIn;                  // Response to process read input data
  rxDataNoRead pdOut;               // Response to write process output data
} rxDataPDinPDout;

typedef struct {
  rxDataPDin pdIn;                  // Response to process read input data
  rxDataNoRead pdOut;               // Response to write process output data
  rxDataPread pRead;                // Response to read parameter
} rxDataPDinPDoutPread;

typedef struct {
  rxDataPDin pdIn;                  // Response to process read input data
  rxDataNoRead pdOut;               // Response to write process output data
  rxDataNoRead pWrite;              // Response to write parameter
} rxDataPDinPDoutPwrite;
```

```
// Receive buffer union
union rxBlock {
  rxDataPDinPDoutPwrite pdInOutWrite; // Read process input data, write process out
put data, and write parameter
  rxDataPDinPDoutPread pdInOutPread;  // Read process input data, write process out
put data and read parameter
  rxDataPDinPDout pdInOut;            // Read process input data read and write pro
cess output data
  rxDataPDoutPwrite pdOutPwrite;      // Write process output data and parameter
  rxDataPDoutPread pdOutPread;        // Write process output data and read paramet
er
  rxDataPDinPwrite pdInPwrite;        // Read process output data and write paramet
er
  rxDataPDinPread pdInPread;          // Read process input data and parameter
  rxDataPread pRead;                  // Read parameter only
  rxDataPDin pdIn;                    // Read process input data only
  rxDataNoRead noRead;                // Write process output data or parameter onl
y
};

// Global variable
struct sockaddr_in dstAddr;

//-------------------------------------
//  Socket send/receive
//  dstSocket : Connection socket ID
//  txBuffer  : Send buffer
//  rxBuffer  : Receive buffer
//  txLength  : Send data length
//  rxLength  : Receive data length
//  return    : Receive data length
//         -1 : Send error
//         -2 : Receive error
//-------------------------------------
int sendRecSocket(int dstSocket, char *txBuffer, char *rxBuffer, int txLength, int
rxLength)
{
  // Socket send
#ifdef TCPIP
  if (send(dstSocket, txBuffer, txLength, 0) == 0) return -1;
#else
  if (sendto(dstSocket, txBuffer, txLength, 0, (struct sockaddr *)&dstAddr, sizeof(
dstAddr)) == 0) return -1;
#endif

  // Socket receive
```

```
  int recByte;
  recByte = recv(dstSocket, rxBuffer, rxLength, 0);
  if (recByte &lt; 2) return -2;

return recByte;
}

//-------------------------------------
//  Console display of receive data
//  message  : Response type string
//  recByte  : Receive data length
//  respCode : Response code
//  respExp  : Expected response code
//  errCode  : Error code
//  buffer   : Displayed buffer
//  length   : Displayed data length
//  return   : None
//-------------------------------------
void dispRxData(char *message, int recByte, BYTE respCode, BYTE respExp, BYTE errCo
de, BYTE *buffer, int length)
{
  int i;

  if (recByte == -1) {
    printf("There was an error with socket send. ¥n");
  }
  else if (recByte == -2) {
    printf("There was an error with socket receive. ¥n");
  }
  else if (respCode != respExp || errCode != 0) {
    printf("%s response code (0x% 02X) is abnormal. ¥n", message, respCode);
    printf("The error code is 0x%02X. ¥n", errCode);
  }
  else {
    printf("%s response code (0x% 02X) is normal. ¥n", message, respCode);
    if (length &gt; 0) {
      // Displays the received data in hexadecimal
      printf("Read data:");
      for (i = 0; i &lt; length; i++) {
        printf("%02X ", buffer[i]);
      }
      printf("¥n");
    }
  }
}
```

Main Processing

```
//---------------
//  Main function
//---------------
int main()
{
   // Socket initialization
   WSADATA wsdata;
   if (WSAStartup(MAKEWORD(2, 0), &wsdata)) {
      printf("Socket initialization failed¥n");
      return 2;
   }

   // Set the destination IP address
   char destination[32];
   BYTE ipAdrs3rd = 1;          // 3rd byte
   BYTE ipAdrs4th = 30;         // 4th byte
   sprintf_s(destination, 32, "192.168.%d.%d¥0", ipAdrs3rd, ipAdrs4th);

   //  Specify the port number (fixed to 2001 for Ethernet/TCP of GD)
   int port = 2001;

   // Set the socket structure
   memset(&dstAddr, 0, sizeof(dstAddr));
   dstAddr.sin_port = htons(port);
   dstAddr.sin_family = AF_INET;
   InetPton(AF_INET, destination, &dstAddr.sin_addr.s_addr);

   // Socket generation
   int dstSocket;
#ifdef TCPIP
   // TCP/IP connection
   dstSocket = socket(AF_INET, SOCK_STREAM, 0);

   // Connect to socket
   if (connect(dstSocket, (struct sockaddr *) &dstAddr, sizeof(dstAddr))) {
      printf("Could not connect to%s ¥n", destination);
      return 1;
   }
   printf("Connected to%s ¥n", destination);
#else
   // UDP connection
   dstSocket = socket(AF_INET, SOCK_DGRAM, 0);
#endif
```

```
   // Secure send/receive buffer
   union txBlock txBuffer = { 0 };
   union rxBlock rxBuffer = { 0 };
   int recByte;
```

Sample Program 1: Exchanging Process Input Data and Process Output Data

```
//------------------------------------------------------------
// Sample 1: Exchanging process input data and process output data
//------------------------------------------------------------
// Structure according to the transfer data specification
// Send: As it is process output data, txBuffer.pdOut
// Receive: As it is a response of process input data + process output data, rxBuff
er.pdInOut
printf("sample 1¥n");
txBuffer.pdOut.command = 0x53;
txBuffer.pdOut.xferData = PDIN_DATA | PDOUT_DATA;

// Process output data: Turn on the output of port 2 and port F
txBuffer.pdOut.pdOutData[0] = 0x04;
txBuffer.pdOut.pdOutData[1] = 0x80;

// Socket send/receive
recByte = sendRecSocket(
   dstSocket,    // Socket ID
   (char *)&txBuffer,                          // Send buffer
   (char *)&rxBuffer,                          // Receive buffer
   sizeof(txDataPDout),                        // Send data length
   sizeof(rxDataPDinPDout)                     // Receive data length
);
dispRxData(
   "Read process input data",                  // Displayed character string
   recByte,                                    // Receive data length
   rxBuffer.pdInOut.pdIn.respCode,             // Response code of read process in
put data
   RESP_PDIN,                                  // Expected response code
   rxBuffer.pdInOut.pdIn.errCode,              // Error code of read process input
 data
   (BYTE *)&rxBuffer.pdInOut.pdIn.pdInData,    // Display data address
   PDIN_LEN                                    // Displayed data length
);
dispRxData(
   "Write process output data",                // Displayed character string
   recByte,                                    // Receive data length
   rxBuffer.pdInOut.pdOut.respCode,            // Response code of write process o
```

```
utput data
   RESP_PDOUT,                                  // Expected response code
   rxBuffer.pdInOut.pdOut.errCode,              // Error code of write process outp
ut data
   (BYTE *)&rxBuffer.pdInOut.pdOut.errCode,     // Display data address (dummy)
   0                                            // Displayed data length
);
```

Sample Program 2: Reading Parameter

```
//-------------------------------
// Sample 2: Reading parameter
//-------------------------------
// Structure according to the transfer data specification
// Send: There is no data to write, so txBuffer.noWrite
// Receive: Parameter read data, so rxBuffer.pRead
printf("sample 2¥n");
txBuffer.noWrite.command = 0x53;
txBuffer.noWrite.xferData = PARAM_READ;
txBuffer.noWrite.unit = 0xFF;            // Targets the parameters of the GD-ILM16C
-MLP unit
txBuffer.noWrite.indexH = 0x00;
txBuffer.noWrite.indexL = 0x08;          // Read index number = 8
txBuffer.noWrite.subindex = 0x00;
txBuffer.noWrite.readLenL = 0;           // If 0 is specified as the data length to
 be read, the response will be the actual data length
txBuffer.noWrite.readLenH = 0;

// Socket send/receive
recByte = sendRecSocket(
   dstSocket,                            // Socket ID
   (char *)&txBuffer,                    // Send buffer
   (char *)&rxBuffer,                    // Receive buffer
   sizeof(txDataNoWrite),                // Send data length
   sizeof(rxDataPread)                   // Receive data length
);
dispRxData(
   "Read parameter",                     // Displayed character string
   recByte,                              // Receive data length
   rxBuffer.pRead.respCode,              // Response code of read parameter
   RESP_PREAD,                           // Expected response code
   rxBuffer.pRead.errCode,               // Error code of read default value
   (BYTE *)&rxBuffer.pRead.readData,     // Display data address
   recByte - RESPCODE_LEN                // Displayed data length (delete one respo
nse code)
);
```

Sample Program 3: Exchanging Process Input Data and Process Output Data and Reading Parameters

```
//------------------------------------------------------------------------------
// Sample 3: Exchanging process input data and process output data and reading para
meters
//------------------------------------------------------------------------------
// Structure according to the transfer data specification
// Send: As it is process output data, txBuffer.pdOut
// Receive: Process input data + Process output data response + Parameter read data
, so rxBuffer.pdInOutPread
printf("sample 3¥n");
txBuffer.pdOut.command = 0x53;
txBuffer.pdOut.xferData = PDIN_DATA | PDOUT_DATA | PARAM_READ;

// Process output data: Turn on the output of ports 6 and 8
txBuffer.pdOut.pdOutData[0] = 0x40;
txBuffer.pdOut.pdOutData[1] = 0x01;
txBuffer.pdOut.unit = 0xFF;                          // Targets the parameters of
the GD unit
txBuffer.pdOut.indexH = 0x00;
txBuffer.pdOut.indexL = 0x41;                        // Read index number = 65
txBuffer.pdOut.subindex = 0x00;
txBuffer.pdOut.readLenL = 0;                         // If 0 is specified as the d
ata length to be read, the response will be the actual
txBuffer.pdOut.readLenH = 0;

// Socket send/receive
recByte = sendRecSocket(
   dstSocket,                                        // Socket ID
   (char *)&txBuffer,                                // Send buffer
   (char *)&rxBuffer,                                // Receive buffer
   sizeof(txDataPDout),                              // Send data length
   sizeof(rxDataPDinPDoutPread)                      // Receive data length
);
dispRxData(
   "Read process input data",                        // Displayed character string
   recByte,                                          // Receive data length
   rxBuffer.pdInOutPread.pdIn.respCode,              // Response code for read pro
cess input data
   RESP_PDIN,                                        // Expected response code
   rxBuffer.pdInOutPread.pdIn.errCode,               // Error code of read process
 input data
   (BYTE *)&rxBuffer.pdInOutPread.pdIn.pdInData,     // Display data address
   PDIN_LEN                                          // Displayed data length
);
```

```
dispRxData(
   “Write process output data”,                          // Displayed character string
   recByte,                                               // Receive data length
   rxBuffer.pdInOutPread.pdOut.respCode,                  // Response code of write pro
cess output data
   RESP_PDOUT,                                            // Expected response code
   rxBuffer.pdInOutPread.pdOut.errCode,                   // Error code of write proces
s output data
   (BYTE *)&rxBuffer.pdInOutPread.pdOut.errCode,          // Display data address (dumm
y)
   0                                                      // Displayed data length
);
dispRxData(
   “Read parameter”,                                      // Displayed character string
   recByte,                                               // Receive data length
   rxBuffer.pdInOutPread.pRead.respCode,                  // Response code of read para
meters
   RESP_PREAD,                                            // Expected response code
   rxBuffer.pdInOutPread.pRead.errCode,                   // Error code of read paramet
ers
   (BYTE *)&rxBuffer.pdInOutPread.pRead.readData,         // Display data address
   recByte - PDIN_LEN - (RESPCODE_LEN * 3)                // Displayed data length (del
ete process input data and three response codes)
);
```

Sample Program 4: Exchanging Process Output Data and Writing Parameters

```
//-----------------------------------------------------------
// Sample 4: Exchanging process output data and writing parameters
//-----------------------------------------------------------
// Structure according to the transfer data specification
// Send: Process output data + parameter write data, so txBuffer.pdOutPwrite
// Receive: Process output data response + parameter write response, so rxBuffer.pd
OutPwrite
printf(“sample 4¥n”);
txBuffer.pdOutPwrite.command = 0x53;
txBuffer.pdOutPwrite.xferData = PDOUT_DATA | PARAM_WRITE;

// Process output data: Turn on the output of ports 6 and 8
txBuffer.pdOutPwrite.pdOutData[0] = 0x40;
txBuffer.pdOutPwrite.pdOutData[1] = 0x01;

txBuffer.pdOutPwrite.unit = 0xFF;                  // Targets the parameters of the G
D unit
txBuffer.pdOutPwrite.indexH = 0x00;
```

```
txBuffer.pdOutPwrite.indexL = 0x08;              // Write index number = 8
txBuffer.pdOutPwrite.subindex = 0x00;
txBuffer.pdOutPwrite.writeData[0] = 'a';         // Data to be written
txBuffer.pdOutPwrite.writeData[1] = 'b';
txBuffer.pdOutPwrite.writeData[2] = 'c';
txBuffer.pdOutPwrite.writeData[3] = 'd';
txBuffer.pdOutPwrite.writeData[4] = 'e';
int pWriteLen = 5;                               // Parameter write byte length

// Socket send/receive
recByte = sendRecSocket(
   dstSocket,    // Socket ID
   (char *)&txBuffer,                            // Send buffer
   (char *)&rxBuffer,                            // Receive buffer
   sizeof(txDataPDout) + pWriteLen,              // Send data length
   sizeof(rxDataPDoutPwrite)                     // Receive data length
);
dispRxData(
   "Write process output data",                  // Displayed character string
   recByte,                                      // Receive data length
   rxBuffer.pdOutPwrite.pdOut.respCode,          // Response code of write process
output data
   RESP_PDOUT,                                   // Expected response code
   rxBuffer.pdOutPwrite.pdOut.errCode,           // Error code of write process out
put data
   (BYTE *)&rxBuffer.pdOutPwrite.pdOut.errCode,  // Display data address (dummy)
   0                                             // Displayed data length
);
dispRxData(
   "Write parameter",                            // Displayed character string
   recByte,                                      // Receive data length
   rxBuffer.pdOutPwrite.pWrite.respCode,         // Response code of write paramete
r
   RESP_PWRITE,                                  // Expected response code
   rxBuffer.pdOutPwrite.pWrite.errCode,          // Error code of write parameter
   (BYTE *)&rxBuffer.pdOutPwrite.pWrite.errCode, // Display data address (dummy)
   0                                             // Displayed data length
);
```

Sample Program 5: Exchanging Process Input Data and Process Output Data and Reading Parameters in the Background

```
//-------------------------------------------------------------------------------
------------
// Sample 5: Exchanging process input data and process output data and reading para
meters in the background
```

```
//-------------------------------------------------------------------------------
------------
// Structure according to the transfer data specification
// Send: As it is process output data, txBuffer.pdOut
// Receive: Process input data + Process output data response + Parameter read data
, so rxBuffer.pdInOutPread
printf("sample 5¥n");
txBuffer.pdOut.command = 0x53;
txBuffer.pdOut.xferData = PDIN_DATA | PDOUT_DATA | BACK_READ;

// Process output data: Turn on the output of ports 6 and 8
txBuffer.pdOut.pdOutData[0] = 0x40;
txBuffer.pdOut.pdOutData[1] = 0x01;
txBuffer.pdOut.unit = 0xFF;    // Targets the parameters of the GD unit
txBuffer.pdOut.indexH = 0x00;
txBuffer.pdOut.indexL = 0x41;                          // Read index number = 65
txBuffer.pdOut.subindex = 0x00;
txBuffer.pdOut.readLenL = 0;                           // If 0 is specified as the
data length to be read, the response will be the actual data length
txBuffer.pdOut.readLenH = 0;

do {
  // Socket send/receive
  recByte = sendRecSocket(
    dstSocket,                                         // Socket ID
    (char *)&txBuffer,                                 // Send buffer
    (char *)&rxBuffer,                                 // Receive buffer
    sizeof(txDataPDout),                               // Send data length
    sizeof(rxDataPDinPDoutPread)                       // Receive data length
  );
  dispRxData(
    "Read process input data",                         // Displayed character strin
g
    recByte,                                           // Receive data length
    rxBuffer.pdInOutPread.pdIn.respCode,               // Response code for read pr
ocess input data
    RESP_PDIN,                                         // Expected response code
    rxBuffer.pdInOutPread.pdIn.errCode,                // Error code of read proces
s input data
    (BYTE *)&rxBuffer.pdInOutPread.pdIn.pdInData,      // Display data address
    PDIN_LEN    // Displayed data length
  );
  dispRxData(
    "Write process output data",                       // Displayed character strin
g
    recByte,                                           // Receive data length
    rxBuffer.pdInOutPread.pdOut.respCode,              // Response code of write pr
```

```
ocess output data
    RESP_PDOUT,                                          // Expected response code
    rxBuffer.pdInOutPread.pdOut.errCode,                 // Error code of write proce
ss output data
    (BYTE *)&rxBuffer.pdInOutPread.pdOut.errCode,        // Display data address (dum
my)
    0    // Displayed data length
  );
  dispRxData(
    "Read parameter",                                    // Displayed character strin
g
    recByte,                                             // Receive data length
    rxBuffer.pdInOutPread.pRead.respCode,                // Response code of read par
ameters
    RESP_PREAD,                                          // Expected response code
    rxBuffer.pdInOutPread.pRead.errCode,                 // Error code of read parame
ters
    (BYTE *)&rxBuffer.pdInOutPread.pRead.readData,       // Display data address
    recByte - PDIN_LEN - (RESPCODE_LEN * 3)              // Displayed data length (de
lete process input data and three response codes)
  );

  // Change to background access status check
  txBuffer.pdOut.xferData = PDIN_DATA | PDOUT_DATA | BACK_STATUS;

} while (rxBuffer.pdInOutPread.pRead.respCode == RESP_PBUSY);

// Close socket
closesocket(dstSocket);
WSACleanup();

return 0;
}
```

I

Index

Index

U

OMRON Corporation **Industrial Automation Company**

Kyoto, JAPAN **Contact : www.ia.omron.com**

Regional Headquarters

OMRON EUROPE B.V.
Wegalaan 67-69, 2132 JD Hoofddorp
The Netherlands
Tel: (31) 2356-81-300 Fax: (31) 2356-81-388

OMRON ELECTRONICS LLC
2895 Greenspoint Parkway, Suite 200
Hoffman Estates, IL 60169 U.S.A.
Tel: (1) 847-843-7900 Fax: (1) 847-843-7787

OMRON ASIA PACIFIC PTE. LTD.
438B Alexandra Road, #08-01/02 Alexandra
Technopark, Singapore 119968
Tel: (65) 6835-3011 Fax: (65) 6835-3011

OMRON (CHINA) CO., LTD.
Room 2211, Bank of China Tower,
200 Yin Cheng Zhong Road,
PuDong New Area, Shanghai, 200120, China
Tel: (86) 21-6023-0333 Fax: (86) 21-5037-2388

Authorized Distributor:

©OMRON Corporation 2026 All Rights Reserved.
In the interest of product improvement,
specifications are subject to change without notice.

Cat. No. Z516-E1-02 0426