Applicable Product:

FA-M3 Range-free Multi-Controller
Model: F3LD01-0N
Name: DeviceNet Scanner Module

The document number and document model code for this manual are as follows:

Document No.: IM 34M6H28-01E
Document Model Code: DOCIM

Refer to the document number in all communications; also refer to the document number or the document model code when purchasing additional manuals.
Important

About This Manual

(1) This manual should be passed on to the end user.

(2) Before using the module, read this manual completely to get a thorough understanding of the module.

(3) This manual explains the functions contained in this product, but does not warrant that those will suit the particular purpose of the user.

(4) Under absolutely no circumstances may the contents of this manual be transcribed or copied, in part or in whole, without permission.

(5) The contents of this manual are subject to change without prior notice.

(6) Every effort has been made to ensure accuracy in the preparation of this manual. However, should any errors or omissions come to the attention of the user, please contact the nearest Yokogawa Electric representative or sales office.
Safety Precautions when Using/Maintaining the Product

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

CAUTION

This symbol indicates that the operator must follow the instructions laid out in this manual in order to avoid the risk of personnel injuries or fatalities or damage to the instrument. The manual describes what special care the operator must exercise to prevent electrical shock or other dangers that may result in injury or the loss of life.

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

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

Indicates alternating current.
Indicates direct current.

(1) The following symbols are used only in the instruction manual.

WARNING

Indicates that the operator must refer to the instructions in this manual in order to prevent the instrument (hardware) or software from being damaged, or a system failure from occurring.

CAUTION

Draws attention to information essential for understanding the operation and functions.

TIP

Gives information that complements the present topic.

SEE ALSO

Identifies a source to which to refer.

(2) For the protection and safe use of the product and the system controlled by it, be sure to follow the instructions and precautions on safety stated in this manual whenever handling the product. Take special note that if you handle the product in a manner other than prescribed in these instructions, safety cannot be guaranteed.

(3) If separate protection and/or safety circuits for this product or the system which is controlled by this product are to be installed, ensure that such circuits are installed external to the product.

(4) If component parts or consumables are to be replaced, be sure to use parts specified by the company.
(5) Do not attempt to make modifications or additions internal to the product.

■ Force Majeure

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

(2) Yokogawa Electric assumes no liability to any party for any loss or damage, direct or indirect, caused by the user or any unpredictable defect of the product.

■ Software Supplied by the Company

(1) Yokogawa Electric makes no other warranties expressed or implied except as provided in its warranty clause for software supplied by the company.

(2) Use the relevant software with one specified computer only. You must purchase another copy of the software for use with each additional computer.

(3) Copying the software for any purpose other than backup is strictly prohibited.

(4) Store the floppy disks (originals) of this software in a safe place.

(5) Reverse engineering, such as decompiling of the software, is strictly prohibited.

(6) No portion of the software supplied by Yokogawa Electric may be transferred, exchanged, or sublet or leased for use by any third party without prior permission by Yokogawa Electric.
General Requirements for Using FA-M3 Controllers

- Avoid installing FA-M3 controllers in the following locations:
  - Where the instrument will be exposed to direct sunlight, or where the operating temperature is outside the range 0°C to 55°C.
  - Where the relative humidity is outside the range 10 to 90%, or where sudden temperature changes may occur and cause condensation.
  - Where corrosive or inflammable gases are present.
  - Where the instrument will be exposed to direct mechanical vibration or shock.

- Securely tighten screws:
  - Securely tighten module mounting screws and terminal screws to avoid problems such as faulty operation.

- Securely fasten connectors of interconnecting cables:
  - Securely fasten connectors of interconnecting cables, and check them thoroughly before turning on the power.

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

- Ground FA-M3 controllers to an independent Japanese Industrial Standard (JIS) Class 3 Ground:
  - Avoid grounding the FG terminal of the FA-M3 controller to the same ground as high-voltage power lines. Ground the terminal to an independent JIS Class 3 ground (ground resistance up to 100 Ω).

- Observe countermeasures against noise:
  - When assigning inputs/outputs, the user should avoid locating AC-supplied I/O modules in the vicinity of the CPU module.

- Keep spare parts on hand:
  - Stock up on maintenance parts, including spare modules, in advance.

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

- Never use solvents such as paint thinner for cleaning:
  - Gently clean the surfaces of the FA-M3 controllers with a piece of soft cloth soaked in water or a neutral detergent.
  - Do not use solvents such as paint thinner for cleaning, as they may cause deformation, discoloration, or malfunctioning.
● Avoid storing the FA-M3 controllers in places with high temperature or humidity:
  • Since the CPU module has a built-in battery, avoid storing it in places with high tempera
ture or humidity.
  • Since the service life of the battery is drastically reduced by exposure to high tem-
peratures, so take special care (storage temperature can be from -20°F to 75°C).

● Always turn off the power before installing or removing modules:
  • Turn off power to the power supply module when installing or removing modules, other-
wise damage may result.

● When installing ROM packs and changing switch settings:
  • In some modules you can remove the right-side cover and install ROM packs or change
switch settings. While doing this, do not touch any components on the printed-
circuit board, otherwise components may be damaged and modules fail.
Waste Electrical and Electronic Equipment

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

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

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

Overview of the Manual

This is the instruction manual of the DeviceNet Scanner Module of the FA-M3 Multi-controller.

Other Instruction Manuals

Consult the following FA-M3 manuals as necessary when using this module:

- Sequence CPU Instruction Manual – Functions (IM 34M6P12-02E)
- Sequence CPU Instruction Manual – Instructions (IM 34M6P12-03E)
- Sequence CPU Instruction Manual (for F3FP36) (IM 34M6P22-01E)
- Personal Computer Link Command Module Instruction Manual (IM 34M6P41-01E)
- Ladder Diagram Support Program M3 Instruction Manual (IM 34M6Q13-01E)
- BASIC CPU Module and BASIC Programming Tool M3 Instruction Manual (IM 34M6Q22-01E)

Copyright/Trademarks

- DeviceNet is a registered trademark of ODVA (Open DeviceNet Vendor Association, Inc.).
- COMBICON is a registered trademark of Phoenix Contact Co., Ltd.
- All the product names and company names mentioned in this manual are the trademarks or registered trademarks of their respective companies.
- The copyright of the software loaded in this module belongs to SST Inc. (S-S Technologies).

Terminology Used

- This manual uses terminology in the DeviceNet Specification published by ODVA.
CONTENTS

◆ Important ...................................................................................................... ii
◆ Introduction ................................................................................................ viii
1. Overview .................................................................................................. 1-1
2. Overview of DeviceNet .............................................................................. 2-1
   2.1 Network Configuration ........................................................................... 2-1
   2.2 I/O Communications (I/O Slave Message) ........................................... 2-3
   2.3 Message Communications (Explicit Peer-to-Peer Message) .............. 2-4
3. Preparations for Operation ........................................................................ 3-1
4. Module Specifications ............................................................................... 4-1
   4.1 Function Specifications ......................................................................... 4-1
   4.2 Operating Environment ......................................................................... 4-3
   4.3 Model and Suffix Codes ....................................................................... 4-3
   4.4 Components ......................................................................................... 4-3
   4.5 Display Panel ....................................................................................... 4-4
   4.6 Setting Switches ................................................................................... 4-5
      4.6.1 Node Address Setting Switch ..................................................... 4-5
      4.6.2 Operation Mode Setting Switch ................................................. 4-5
      4.6.3 Terminating Resistor Setting Switch ....................................... 4-6
   4.7 External Dimensions ............................................................................ 4-7
5. Attaching and Wiring ................................................................................. 5-1
   5.1 Attaching and Detaching a Module .................................................... 5-1
   5.2 External Wiring .................................................................................... 5-3
6. Module Access ............................................................................................ 6-1
   6.1 I/O Relay .............................................................................................. 6-1
   6.2 I/O Data Register ................................................................................. 6-2
   6.3 Access from the CPU Module ............................................................ 6-4
      6.3.1 Access Using Ladder Sequence ............................................... 6-4
      6.3.2 Access Using BASIC ............................................................... 6-5
7. I/O Communications .................................................................................... 7-1
   7.1 Scan List ............................................................................................... 7-1
   7.2 Module Operation .................................................................................. 7-4
   7.3 Transmission Time Intervals ............................................................... 7-10
   7.4 I/O Response Time .............................................................................. 7-11
1. **Overview**

The F3LD01-0N DeviceNet Scanner Module (hereafter referred to as this module) is a module for connecting to DeviceNet.

**What is a DeviceNet?**

DeviceNet is an open field network and is mainly used in manufacturing equipment or in production lines. In a multi-vendor environment, a wide range of devices, ranging from sensors and actuators to advanced equipment which is compatible with DeviceNet, can be connected.

As the specifications of DeviceNet are open and licensing is not required, many manufacturers, both in Japan and overseas, are developing new devices which, when connected with DeviceNet, allow exchanges of I/O data. Overseas service has also become possible. Further, many types of I/O communications and message communications are supported. Decentralized control with multi-master systems is also possible now.

A maximum of 64 machines (masters + slaves) can be connected to DeviceNet. The maximum data rate is 500 kbps and the maximum trunk line distance is 500 m (when the data rate is 125 kbps). Thus, with these special characteristics of high connectability to many stations, a high data rate and a long distance, DeviceNet can be used in a wide range of applications. Wiring can be done effectively by freely combining the T-branching method and daisy chain method, with no wastage.

Each equipment type is defined by a device profile (specifications which define the operation of the equipment). Equipment is built according to these profiles, thus resulting in the compatibility and interchangeability of the equipment.

![Diagram](F0101.EPS)  
*Figure 1.1 Example of a DeviceNet Connection*
Features of this module

This module has the master functions of DeviceNet. Using this module, you can exchange I/O data periodically with other connected devices and exchange message data like parameters and status.

The following are the main features of this module.

- A maximum of 63 slave machines can be connected.
- Data rate is 125/250/500 kbps.
- Supports an I/O communication function which regulates and controls I/O data of 16,000 points (input 8,000 points; output 8,000 points). Anything from small-scale I/O data to multiple-point analog data and information data can be handled in real time.
- A message communication function is supported. If required, the exchange of message data like parameters and status is also possible. Therefore, the time required for system configuration is reduced, and maintenance as well as calibration becomes easier.
- Operation in a multi-master environment with decentralized control is also possible.
- 16 modules of this kind can be installed in a single FA-M3. It is easy to manage multiple-point I/O data and to add new lines.
- The transmission time interval of I/O data can be set for every 5 msec. Thus, it is compatible even with a slow-responding slave.
2. Overview of DeviceNet

An explanation of the functions which are supported by this module and which are related to the DeviceNet specifications is given here.

SEE ALSO

1) For details on the DeviceNet specifications, see also to the DeviceNet Specifications Manual, published by ODVA (Open Device Net Vendor Association, Inc.). ODVA is a non-profit company, formed by vendors with the objective to promote and manage the DeviceNet specifications.

2) For product information of third-party products required for network configuration, refer to Appendix 2, “Information on Third-party Products.”

2.1 Network Configuration

The network configuration of DeviceNet is given below.

The elements of the network configuration are explained below.

- **Node**

A device (equipment) that is connected to DeviceNet is called a node. A node can be either a master, which collects and distributes data, or a slave, which outputs and inputs data according to the instructions received from the master.

There are no restrictions on the arrangements or maximum number of devices that can be connected to the master of each of the slaves. A maximum of 64 devices can be freely connected (without arrangement restrictions).

An address from 0 to 63, called MACID (Media Access Control Identifier), is assigned to each node. In this manual, these addresses are called node addresses.
• Cable

Five special purpose cables (2 signal lines, 2 power supply lines, and 1 SHIELD cable) are used. There are 2 types of cables, namely, thick cables and thin cables.

• Trunk Line

Terminating resistors are attached at both ends of this cable. Both thick and thin cables can be used. They can also be used together.

For a thin cable, the maximum network length is 100 m. In the case of thick cables, the length limits depend on the data rate (see Table 2.1).

• Drop Line

This is a branch cable from the trunk line and normally, thin cables are used for this purpose.

The length (the distance from the point of branching from the trunk line to the end of the drop line) of a drop-line cable should be 6 m or less. (There are no minimum distance limits.) Generally, the total drop-line length depends on the data rate (see Table 2.1).

Table 2.1 Maximum Cable Length of DeviceNet

<table>
<thead>
<tr>
<th>Data rate</th>
<th>Maximum cable length of trunk line</th>
<th>Length of drop line</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Only thick cables used</td>
<td>Only thin cable used</td>
</tr>
<tr>
<td>125 kbps</td>
<td>500 m</td>
<td>100 m</td>
</tr>
<tr>
<td>250 kbps</td>
<td>250 m</td>
<td></td>
</tr>
<tr>
<td>500 kbps</td>
<td>100 m</td>
<td></td>
</tr>
</tbody>
</table>

• Connection Method

There are 2 methods of wire connection, the T-branching (using the T-branching tap) and multiple-node daisy-chain methods. In a trunk line/ drop line, both methods can be combined.

• Terminating Resistor

Terminating resistors (121Ω) are attached to both ends of a trunk line. This reduces the reflection of signals and stabilizes communication.

In this module, there is a built-in terminating resistor. Therefore, when connecting the trunk line using the daisy chain method, you can turn on the terminating resistor setting switch and use the built-in terminating resistor. In other connection methods, please turn the switch off so that the built-in terminating resistor is not used.

• Network Power Supply

Out of the 5 special-purpose cables, two are for the 24 V DC network power supply. It is necessary to supply power to each node of the network using the cables. If a single network power supply source is used in a network, you can connect the power supply directly to the trunk line. However, if multiple power supply sources are to be used in the network, it is necessary to use special-purpose power supply taps.
2.2 I/O Communications (I/O Slave Message)

This is a communication function which allows the exchange of control information (I/O data) between the master and a slave in real time.

This module supports the following two types of I/O communications protocols of DeviceNet.

Polling .......... Sends polling instructions containing output data from the master to a specified slave; the master then receives a response containing the input data from the slave

(1 : 1 communication).

Bit Strobe ...... Multicasts (broadcasts) Bit Strobe request instructions from the master to multiple slave systems and receives a response with input data from each slave. A communication request that can be transmitted to multiple slaves at a time (1:N communications) will improve network throughput. However, this is possible only in the case of slave systems with input data of less than 8 bytes.

In this module, transmission of I/O communications requests is cyclic. This period (transmission time interval) can be specified for all the slaves in a batch in 5 ms intervals.

SEE ALSO

For information on I/O communications functions other than Polling and Bit Strobe, see also the “DeviceNet Specification” published by ODVA.
2.3 Message Communications (Explicit Peer-to-Peer Message)

This is a function for setting and reading parameters, controlling and managing operations and exchanging information between nodes (1:1) when necessary. In this module, it is possible to send a service request to other nodes (master and slaves) using the explicit messages defined in DeviceNet.

Figure 2.3 Message Communications Concept Diagram
3. Preparations for Operation

The preparations required before starting the operation are shown in Figure 3.1.

- Design DeviceNet system
- Set module switches
- Install the module in the base module
- Wiring of cables
- Slave machine power ON, network power ON
- Master equipment power ON
- Create scan list
- Set transmission time interval
- Test run

- Is it operating normally?
  - NO
  - YES

- Operation

(1) Node address
(2) Data rate
(3) Terminating resistor

- Perform local setup (switch, etc.) of devices other than this module.
- Power supply is connected to the network and all the devices.
- Registers the information, I/O data parameters of devices which will communicate with this module. Also registers the translation time interval.
- Start scanning and verify whether communications with the other devices are correct. Also verify the transmission time interval.

(1) If an error occurs, check the setting, wiring, scan list of this module and other devices and re-start after proper adjustment and setting.
(2) If the measured value of the transmission time interval exceeds the set value, set a value larger than the measured value.

(Note) Reset the module. Restart the power supply to start operation.

Figure 3.1 Operation Procedure Flow
SEE ALSO

(1) For details on the interface of the CPU module and this module related to operation preparations, see also the following sections of this manual:
   - Section 6.1 I/O Relay
   - Section 6.2 I/O Data Register
   - Section 8.2 Interface (Message Communications)
   - Section 9.1 Tool Commands

(2) For overview of DeviceNet, see Section 2, “Overview of DeviceNet,” of this manual.

(3) For information on the DeviceNet features supported in this module, see Section 4.1, “Function Specifications,” of this manual.

(4) For information on I/O functions supported by this module, see Section 7, “I/O Communications,” provided later in this manual.
4. Module Specifications

The specifications of this module are explained herein.

4.1 Function Specifications

The range of DeviceNet functions supported by this module is shown in the following table.

<table>
<thead>
<tr>
<th>DeviceNet functions</th>
<th>Device Communication adaptor</th>
<th>Master/Scanner</th>
</tr>
</thead>
<tbody>
<tr>
<td>Explicit peer-to-peer message</td>
<td>Y</td>
<td>I/O slave message</td>
</tr>
<tr>
<td>I/O peer-to-peer message</td>
<td>N</td>
<td>Bit strobe</td>
</tr>
<tr>
<td>Configuration consistency value</td>
<td>N</td>
<td>Polling</td>
</tr>
<tr>
<td>Fault node recovery</td>
<td>N</td>
<td>Cyclic</td>
</tr>
<tr>
<td>Baud rate</td>
<td>125K, 250K, 500K</td>
<td>Change of state (COS)</td>
</tr>
</tbody>
</table>

(Y: Supported; N: Not supported)

SEE ALSO

For details on unsupported functions, see also the DeviceNet Specification published by ODVA.
The function specifications are shown in Table 4.2.

### Table 4.2 General Specifications

<table>
<thead>
<tr>
<th>Items</th>
<th>Specifications</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interface</td>
<td>Conforms to DeviceNet.</td>
</tr>
<tr>
<td>Data rate</td>
<td>125/250/500 kbps (selected by switch positions)</td>
</tr>
<tr>
<td>Transmission medium</td>
<td>5 special cables (2 for signals, 1 SHIELD, 2 for power supply)</td>
</tr>
<tr>
<td>Transmission distance</td>
<td></td>
</tr>
<tr>
<td>Data rate</td>
<td>Only thick cables</td>
</tr>
<tr>
<td>125 kbps</td>
<td>500 m</td>
</tr>
<tr>
<td>250 kbps</td>
<td>250 m</td>
</tr>
<tr>
<td>500 kbps</td>
<td>100 m</td>
</tr>
<tr>
<td>Transmission distance</td>
<td>Maximum length of trunk line</td>
</tr>
<tr>
<td>125 kbps</td>
<td>500 m</td>
</tr>
<tr>
<td>250 kbps</td>
<td>250 m</td>
</tr>
<tr>
<td>500 kbps</td>
<td>100 m</td>
</tr>
<tr>
<td>Connection method</td>
<td>Multiple-mode daisy chain method, T-branching method</td>
</tr>
<tr>
<td>Number of connected nodes</td>
<td>64 devices (including the master)</td>
</tr>
<tr>
<td>Error detection</td>
<td>CRC error, node address check, scan list check</td>
</tr>
<tr>
<td>Power supply for communication</td>
<td>Voltage: 11 to 25 V DC</td>
</tr>
<tr>
<td></td>
<td>Current consumption: Less than 40 mA (24 V DC) (supplied via a DeviceNet connector)</td>
</tr>
<tr>
<td>Terminating resistor</td>
<td>121Ω (built-in, specified using switches when terminating)</td>
</tr>
<tr>
<td>No. of I/O points</td>
<td>Input 8,000 points, Output 8,000 points, Total 16,000 points (1,000 words)</td>
</tr>
<tr>
<td>Maximum message length</td>
<td>Transmission: 84 bytes, receive: 88 bytes (service data)</td>
</tr>
<tr>
<td>No. of installed modules</td>
<td>16 max.</td>
</tr>
<tr>
<td>Current consumption</td>
<td>200 mA (5 V DC)</td>
</tr>
<tr>
<td>External dimensions</td>
<td>28.9 (W) × 100 (H) × 83.2 (D) mm*1</td>
</tr>
<tr>
<td>Mass</td>
<td>110g</td>
</tr>
</tbody>
</table>

*1 : Dimensions excluding protrusions (see External Dimensions for details)
4.2 Operating Environment

There are no restrictions on the CPU modules that can be used with this module.

4.3 Model and Suffix Codes

<table>
<thead>
<tr>
<th>Model</th>
<th>Suffix Code</th>
<th>Style Code</th>
<th>Option Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>F3LD01</td>
<td>-0N</td>
<td>...</td>
<td>...</td>
<td>DeviceNet 1 port</td>
</tr>
</tbody>
</table>

4.4 Components

- **Front view**

  Indicators
  - RDY: Lit when the internal circuitry is in normal operation.
  - MS: Indicates the module status. Green light is on when normal.
  - NS: Indicates the network status. Green light is on when communications are proper.
  - Node address setting switch: Sets node address (0 to 63) Sets 00 at the factory.
  - Terminating resistor setting switch: Used to terminate modules.
  - DeviceNet connector: Connects the communication cable.

- **Right side view**

  Operation mode setting switch
  - SW1 and 2: Data rate
  - SW3: System reserved
  - SW4: System reserved
  - All switches are set to OFF at the factory.

Figure 4.1 Components
4.5 Display Panel

The display panel of this module displays the status of the module and the network.

- **RDY LED**

  The RDY LED is on when the module is working normally and off when there is no proper power supply or if an error has occurred in the module.
  
  When the power supply is turned on, the LED turns on after a self-diagnosis.

- **MS LED/NS LED**

  MS LED and NS LED conform to the DeviceNet communication protocol. The MS LED indicates the module status and the NS LED indicates the network status. The meaning of each status of the LEDs is displayed in Table 4.3.

<table>
<thead>
<tr>
<th>Table 4.3 Explanation of LED Status</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>LED</strong></td>
</tr>
<tr>
<td>MS</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Red</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>NS</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Red</td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>

*1 Minor fault
The switch setting is out of range; therefore, operation is not possible. Set the switch properly and restart the module.

*2 No power/not on-line
No network power supply. Waits for network power supply.

*3 Link OK, on-line, connected
Module is in the network and scanning.
Scanning is the status in which I/O communications are cyclically executed and a request for message communications is received. Message communications are executed whenever a message communications request is received during scanning.

*4 On-line, not connected
Module is in the network. However, scanning is not performed.
It takes approximately 6 seconds after starting the module for it to come on-line.

*5 Critical link failure
If the node address is duplicated or a bus-off error has occurred, communication becomes impossible and the module stops operating. Remove the node address duplication or improve the network environment and restart the module.
A bus-off error occurs when the error rate in the network is very high.

*6 Connection time-out
When the scan operation mode is specified as "continue," even if the node information mismatches, or if the node is absent, scanning continues with the flashing red NS LED indicating abnormal communication.
When the scan operation mode is specified as "stop," scanning stops on a communication error. (NS LED: flashing in green)
A node information mismatch is said to have occurred if the information obtained from the target node through communication at the beginning of scanning and the information that is already registered are found to be different.
"Node absent" means that no response has been received from the target node.
During scanning, if an FA-M3 system failure occurs, scanning stops regardless of the value of the scan operation mode.
4.6 Setting Switches

An explanation about the switch settings of this module is given in this section.

4.6.1 Node Address Setting Switch

Node addresses can be set using 2 decimal rotary switches which are on the front side of this module.

Set the switch to a value from 0 to 63, before switching the power supply on. If the switch is set to 64 or higher, operation will not be normal.

Settings that are mode after switching the power supply on have no effect.

The factory setting is 0.

![Figure 4.2 Node Address Setting Switch]

4.6.2 Operation Mode Setting Switch

Four DIP switches can be seen when the cover on the right side of the module is removed. The operation mode is set by setting the switches to on or off.

Set the switches before switching the power supply on.

Settings that are made after turning the power supply on have no effect.

The factory setting for all the switches is off.

<table>
<thead>
<tr>
<th>Number</th>
<th>Functions</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Data rate (see the table below.)</td>
</tr>
<tr>
<td>2</td>
<td>System reserved (Make sure it is off)</td>
</tr>
</tbody>
</table>

- Data rate

<table>
<thead>
<tr>
<th>Status of number 1</th>
<th>Status of number 2</th>
<th>Data rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Off</td>
<td>Off</td>
<td>125 kbps</td>
</tr>
<tr>
<td>On</td>
<td>Off</td>
<td>250 kbps</td>
</tr>
<tr>
<td>Off</td>
<td>On</td>
<td>500 kbps</td>
</tr>
<tr>
<td>On</td>
<td>On</td>
<td>Invalid setting</td>
</tr>
</tbody>
</table>

![Figure 4.3 Operation Mode Setting Switches]
4.6.3 Terminating Resistor Setting Switch

Using the knife switch on the front face of the module, the built-in resistors can be used as terminating resistors of a trunk line.

The built-in resistors are used only when connecting to the trunk line using daisy chain method when they are installed at both ends of a trunk line.

External terminating resistors can be used in place of the built-in terminating resistors.

If the built-in terminating resistor is used when this module is not installed in a terminal, the operation may not be correct.

The factory setting of this switch is off.

<table>
<thead>
<tr>
<th>TERMINATOR</th>
</tr>
</thead>
<tbody>
<tr>
<td>OFF ( ○ )</td>
</tr>
<tr>
<td>ON ( )</td>
</tr>
</tbody>
</table>

OFF: Terminating resistor disable
ON: Terminating resistor enable

Figure 4.4 Terminating Resistor Setting Switch

**CAUTION**

If the built-in terminating resistor is used when this module is not placed on the terminal, the operation may not be proper.
4.7 External Dimensions

(*) When connecting the base module, the DeviceNet connector and cable leave enough space for the cable to bend properly. (See Section 5.1, "Attaching and Detaching a Module.")
5. Attaching and Wiring

The explanation about attaching/detaching a module and wiring is given in this section.

5.1 Attaching and Detaching a Module

(1) Attaching Module

Figure 5.1 shows how to attach this module to the base module. First hook the anchor slot at the bottom of the module to be attached onto the anchor pin on the bottom of the base module. Push the top of this module in the direction of the arrow shown in the figure (toward the base module) until top button clicks into place.

![Base module](F0501.EPS)

**CAUTION**

- DO NOT bend the connector pins on the rear of the module by force during the above operation. If the module is forcibly pushed with an improper connection, the pins of the connector may bend and this damage will cause a Module Installation Error during the self-diagnosis.

(2) Detaching Module

To remove this module from the base module, reverse the operation by pressing the top button to unlock it and tilting the module away from the base module. Then lift the module off of the anchor pin at the base.

(3) Attaching Module in Intense Vibration Environments

If the module is used in intense vibration environments, fasten the modules with a screw directly beneath the yellow anchor/release button as shown in the Figure 5.2.

With a Phillips screwdriver, tighten the upper side of the module. During this operation, the user must tilt the screwdriver somewhat using the guide channel at the top of the module. A clearance of approximately 80mm between the module and the duct above it is necessary to allow the screwdriver to access the screw.
CAUTION

Do not overtighten the module fixing screw.

---

Recommended screws

M4 12- to 14-mm long bind screws

Figure 5.2 Tightening Module Using Screws

(4) The Depth for Attachment

The depth between the rear surface of the base module and the front surface of this module should be 89.9 mm. When attaching a cable with the connector, leave enough space for the cable to bend. The depth of attachment of this module is shown in Figure 5.3.

<table>
<thead>
<tr>
<th>Module Name</th>
<th>D (mm)</th>
<th>d (mm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>DeviceNet Scanner module</td>
<td>89.9</td>
<td>140 *1</td>
</tr>
</tbody>
</table>

*1 The dimensions of necessary depth differ according to the thickness, hardness of the cable and length of the cable without insulation.

Figure 5.3 Attachment Depth of This Module
5.2 External Wiring

The explanation about the wiring of this module and communication cables is given in this section. For details on the network configuration, refer to Appendix1, “Network Configuration Details.”

Wiring of communication cables to the connector

Attach the cable to the connector by attaching the signal line and power supply line crimp-on terminal to the cable and processing it with vinyl tape or heat compression tube. Proper torque for tightening the screw of the cable is 0.5 N.m.

On the connector of this module, the seals corresponding to the colors of the cables are affixed. Please note that the color of the seals and the cable should match during wiring.

### CAUTION

- Switch OFF the power supply to the FA-M3 and network and then connect the communication cables.
- Use crimp-on terminals for wiring. The following products can be used as crimp-on terminals of the cables. Contact the manufacturer for details.

<table>
<thead>
<tr>
<th>Color</th>
<th>Signal Line</th>
</tr>
</thead>
<tbody>
<tr>
<td>Black</td>
<td>V-</td>
</tr>
<tr>
<td>Blue</td>
<td>CAN L</td>
</tr>
<tr>
<td>-</td>
<td>SHIELD</td>
</tr>
<tr>
<td>White</td>
<td>CAN H</td>
</tr>
<tr>
<td>Red</td>
<td>V+</td>
</tr>
</tbody>
</table>

Figure 5.4 Wiring to the Connector
• Attaching the connector to this module

Insert the connector in the module as shown below. Proper torque for tightening the screw of the cable is 0.3 N.m.

Figure 5.5 Attaching Connector to This Module
6. Module Access

The interface and access method of this module is explained in this section.

6.1 I/O Relay

The I/O relays that can be used in this module are given in Tables 6.1 and 6.2.

In this module, interrupt processing is not supported. The relays that are not defined in Tables 6.1 and 6.2 cannot be used.

In the boxes following X and Y, like X nn nn ** and Y nn nn ** in Tables 6.1 and 6.2, insert the slot number where this module is installed.

---

**CAUTION**

In a multi-CPU system, only one CPU module can be set to use this module. For details on configuration setting, refer to the Ladder Diagram Support Program M3 (IM 34M6Q13-01E).

---

**Table 6.1 Input Relays**

<table>
<thead>
<tr>
<th>Input Relay</th>
<th>Signal Name</th>
<th>Description</th>
<th>Relationship with Other Relays</th>
</tr>
</thead>
<tbody>
<tr>
<td>X nn nn 01</td>
<td>Module error</td>
<td>On if there is a hardware failure or the switch setting is incorrect.</td>
<td>Indicates the current status.</td>
</tr>
<tr>
<td>X nn nn 02</td>
<td>Communication not possible</td>
<td>On if switch number is duplicated or when a bus-off error has occurred.</td>
<td>Same as above</td>
</tr>
<tr>
<td>X nn nn 03</td>
<td>No Network power supply</td>
<td>On if the network power supply is off.</td>
<td>Same as above</td>
</tr>
<tr>
<td>X nn nn 04</td>
<td>Communication error</td>
<td>On if a communication error occurs during scanning. Stays on until scanning is stopped.</td>
<td>Same as above</td>
</tr>
<tr>
<td>X nn nn 07</td>
<td>Scan list ot set</td>
<td>On if there is no registration in the device list.</td>
<td>Same as above</td>
</tr>
<tr>
<td>X nn nn 08</td>
<td>System failure</td>
<td>On if a system failure occurs in the FA-M3.</td>
<td>Same as above</td>
</tr>
<tr>
<td>X nn nn 09</td>
<td>Scan status</td>
<td>On during scanning; Off if not scanning.</td>
<td>Off if Y nn nn 41 is off.</td>
</tr>
<tr>
<td>X nn nn 10</td>
<td>Scan operation mode</td>
<td>Operation mode on error. (On: stop scanning; Off: continue scanning on error.)</td>
<td>Y nn nn 12 request is reflected.</td>
</tr>
<tr>
<td>X nn nn 17</td>
<td>Clear error completed</td>
<td>On if error information is cleared.</td>
<td>Off if Y nn nn 49 is off.</td>
</tr>
<tr>
<td>X nn nn 26</td>
<td>Execute tool command completed</td>
<td>On if execution of tool command ends.</td>
<td>Off when Y nn nn 58 is off.</td>
</tr>
</tbody>
</table>

**Table 6.2 Output Relays**

<table>
<thead>
<tr>
<th>Output Relay</th>
<th>Signal Name</th>
<th>Description</th>
<th>Relationship with Other Relays</th>
</tr>
</thead>
<tbody>
<tr>
<td>Y nn nn 41</td>
<td>Request to start/stop scanning</td>
<td>Request to start/stop scanning (On/Off)</td>
<td>Operation status is displayed in X nn nn 09.</td>
</tr>
<tr>
<td>Y nn nn 42</td>
<td>Scan operation mode request</td>
<td>Defines operation if communication error occurs during scanning. (Off: continue; On: stop)</td>
<td>The current operation mode is displayed in X nn nn 10.</td>
</tr>
<tr>
<td>Y nn nn 49</td>
<td>Request to clear error</td>
<td>Request to clear the error information</td>
<td>Turns off after confirming that X nn nn 17 is on.</td>
</tr>
<tr>
<td>Y nn nn 58</td>
<td>Request to execute tool command</td>
<td>Request to execute a tool command</td>
<td>Turns off after confirming that X nn nn 26 is on.</td>
</tr>
</tbody>
</table>
6.2 I/O Data Register

The list of I/O data registers is shown in Table 6.3.

The I/O data registers that are not defined in Table 6.3 cannot be used.

### Table 6.3 I/O Data Register List

<table>
<thead>
<tr>
<th>Data Position Number</th>
<th>Data Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ladder</td>
<td><strong>BASIC</strong></td>
<td></td>
</tr>
<tr>
<td>1 to 99</td>
<td>Module information</td>
<td>Register which stores the operation status, error code.</td>
</tr>
<tr>
<td>100 to 299</td>
<td>Control flag</td>
<td>Flag which controls message communication and I/O communication for each node.</td>
</tr>
<tr>
<td>400 to 449</td>
<td>Tool command buffer</td>
<td>Register which stores the commands that perform settings and information reading of this module (see Section 9.1.3, “Basic Format of Commands”).</td>
</tr>
<tr>
<td>500 to 999</td>
<td>Input data buffer</td>
<td>Register in which the data input for slave is stored (see Section 7.1, “Scan List”).</td>
</tr>
<tr>
<td>1000 to 1499</td>
<td>Output data buffer</td>
<td>Register in which the data output for slave is stored (see Section 7.1, “Scan List”).</td>
</tr>
<tr>
<td>1501 to 1550</td>
<td>Message communication</td>
<td>Register in which the request command for message communication is stored (see Section 8.2, “Interface”).</td>
</tr>
<tr>
<td>1551 to 1600</td>
<td>receive buffer</td>
<td>Register in which the response command for message communication is stored (see Section 8.2, “Interface”).</td>
</tr>
</tbody>
</table>

### Module Information

Module information displays the status of the module and communication status and the settings. The list is displayed in Table 6.4.

### Table 6.4 Module Information

<table>
<thead>
<tr>
<th>Data Position Number</th>
<th>Type*1</th>
<th>Contents</th>
<th>Explanation</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>R</td>
<td>Status</td>
<td>Reflects the status of the module (X□□□□□□□□□□01 to 16).</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>R</td>
<td>Data rate</td>
<td>Data rate: 0:125; 1: 250; 2: 500 kbit/s Switch set value</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>R</td>
<td>Node address</td>
<td>Node address 0 to 63 Switch set value</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>R</td>
<td>Transmission time interval</td>
<td>Transmission time interval set value [msec] Flash memory stored value</td>
<td></td>
</tr>
<tr>
<td></td>
<td>R</td>
<td>Transmission time interval measured value</td>
<td>Transmission time interval measured value [msec]</td>
<td>0</td>
</tr>
<tr>
<td>17</td>
<td>R</td>
<td>Error code *1</td>
<td>Error code of the module (EC1) Cleared if Y□□□□□□□□□□49 is turned on. (Latched type)</td>
<td>0</td>
</tr>
<tr>
<td>18</td>
<td>R</td>
<td>Error code*2</td>
<td>Error code of the module (EC2) Cleared if Y□□□□□□□□□□49 is turned on. (Latched type)</td>
<td>0</td>
</tr>
<tr>
<td>21 to 24*2</td>
<td>R</td>
<td>Scan list information</td>
<td>The bits corresponding to the nodes registered in the scan list are on.</td>
<td>Flash memory stored value</td>
</tr>
<tr>
<td>25 to 28*2</td>
<td>R</td>
<td>Status information of other nodes</td>
<td>During scanning, the bits corresponding to the nodes that are communicating normally with this module are on.</td>
<td>0</td>
</tr>
<tr>
<td>29 to 32*2</td>
<td>R</td>
<td>Information of the node in error</td>
<td>The bits corresponding to the nodes in which a communication error has occurred, are on. Cleared if Y□□□□□□□□□□49 is turned on. (Latched type)</td>
<td>0</td>
</tr>
</tbody>
</table>

*1 Type .... Type of access available from the CPU module. (R: Read only)

*2 Bit position corresponding to the node address

Data position number

\[ n \begin{array}{cccccccccccccc} 15 & 14 & 13 & 12 & 11 & 10 & 9 & 8 & 7 & 6 & 5 & 4 & 3 & 2 & 1 & 0 \\
1 \end{array} \]

\[ n+1 \begin{array}{cccccccccccccc} 31 & 30 & 29 & 28 & 27 & 26 & 25 & 24 & 23 & 22 & 21 & 20 & 19 & 18 & 17 & 16 \\
1 \end{array} \]

\[ n+2 \begin{array}{cccccccccccccc} 47 & 46 & 45 & 44 & 43 & 42 & 41 & 40 & 39 & 38 & 37 & 36 & 35 & 34 & 33 & 32 \\
1 \end{array} \]

\[ n+3 \begin{array}{cccccccccccccc} 63 & 62 & 61 & 60 & 59 & 58 & 57 & 56 & 55 & 54 & 53 & 52 & 51 & 50 & 49 & 48 \\
1 \end{array} \]
## Control Flag

A control flag is used to control the message communications and I/O communications of each node. The list is given in Table 6.5.

<table>
<thead>
<tr>
<th>Data Position Number</th>
<th>Type*1</th>
<th>Content</th>
<th>Explanation</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>100 to 163*2</td>
<td>R/W</td>
<td>Control for each node</td>
<td>Control request for I/O communications of each node</td>
<td>0</td>
</tr>
<tr>
<td>200 to 263*3</td>
<td>R</td>
<td>Status of each node</td>
<td>Control status for communication of each node</td>
<td>0</td>
</tr>
<tr>
<td>191*4</td>
<td>R/W</td>
<td>Message communications control</td>
<td>Message communications control request</td>
<td>0</td>
</tr>
<tr>
<td>291*5</td>
<td>R</td>
<td>Message communications status</td>
<td>Message communications control status</td>
<td>0</td>
</tr>
</tbody>
</table>

*1 Type of access available from the CPU module. (R: Read only; R/W: Read/write)

*2 Data position number = 100 + node address

**Explanation of flag**

<table>
<thead>
<tr>
<th>bit</th>
<th>15</th>
<th>8</th>
<th>7</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>R</td>
<td>R</td>
<td>R</td>
<td>R</td>
<td>O</td>
</tr>
<tr>
<td>I</td>
<td>O</td>
<td>R</td>
<td>I</td>
<td>R</td>
</tr>
</tbody>
</table>

R : System reserved (Operation prohibited)
I : Request to prohibit input data update (On: Request to prohibit update; Off: Release request)
O : Request to prohibit output data access (On: Prohibit request; Off: Release request)

*3 Data position number = 200 + node address

**Explanation of flag**

<table>
<thead>
<tr>
<th>bit</th>
<th>15</th>
<th>8</th>
<th>7</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>R</td>
<td>R</td>
<td>R</td>
<td>R</td>
<td>O</td>
</tr>
<tr>
<td>R</td>
<td>R</td>
<td>R</td>
<td>I</td>
<td>F1</td>
</tr>
</tbody>
</table>

R : System reserved (Operation prohibited)
F1 : Message communications request (On: Transmission request; Off: Release request)

*4 Explanation of flag

<table>
<thead>
<tr>
<th>bit</th>
<th>15</th>
<th>8</th>
<th>7</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>R</td>
<td>R</td>
<td>R</td>
<td>R</td>
<td>F1</td>
</tr>
</tbody>
</table>

R : System reserved (Operation prohibited)
F1 : Message communications completion (On: Communication completed/response received; Off: During communication or no request (for communication))

(Supplement) When the execution flag is turned off, then the completion flag also turns off automatically.
6.3 Access from the CPU Module

The method for accessing from the CPU module is explained in this section. See the sample programs in Sections 7 to 9 for more details.

6.3.1 Access Using Ladder Sequence

When using the ladder sequence, use the following instructions to access from the CPU module. For details on each instruction, see “Sequence CPU Instruction Manual” (IM34M6P12-03E).

Using the following instructions, the reading and writing of the I/O data registers can be performed.

- **Special Module Read command (READ command)**

  \[
  \text{READ} \quad \text{SL} \quad n_1 \quad D \quad k
  \]

  SL : Slot number of the module (3 digits)
  n1 : First data position number to read
  D : First device number in which to write the data read
  k : Number of words transmitted

- **Special Module Write command (WRITE command)**

  \[
  \text{WRITE} \quad S \quad \text{SL} \quad n_2 \quad k
  \]

  S : First device number for write data
  SL : Slot number of the module (3 digits)
  n2 : First data position number to write
  k : Number of words transmitted

- **Special Module High-speed Read command (HRD command)**

  \[
  \text{HRD} \quad \text{SL} \quad n_1 \quad D \quad k
  \]

  SL : Slot number of the module (3 digits)
  n1 : First data position number to read
  D : First device number in which to write the data read
  k : Number of words transmitted

- **Special Module High-speed Write command (HWR command)**

  \[
  \text{HWR} \quad S \quad \text{SL} \quad n_2 \quad k
  \]

  S : First device number for write data
  SL : Slot number of the module (3 digits)
  n2 : First data position number to write
  k : Number of words transmitted
6.3.2 Access Using BASIC

In BASIC, the following instructions are used to access from the CPU module. For the details on each instruction, see “BASIC CPU, BASIC Programming Tool M3” (IM34M6Q22-01E).

The following BASIC statements can be used for this module. If statements other than the following are used, the operation cannot be guaranteed.

Table 6.6 Statements

<table>
<thead>
<tr>
<th>Functions</th>
<th>Statement Type</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Module use declaration</td>
<td>ASSIGN LD01=SL</td>
<td>Declares use of module and CPU module.</td>
</tr>
<tr>
<td></td>
<td>SL: Slot number</td>
<td></td>
</tr>
<tr>
<td>Reading from I/O data register</td>
<td>ENTER SL, n NOFORMAT;I</td>
<td>The data of data position number n of the module, which is installed in slot number SL, are read and stored in input variable I.</td>
</tr>
<tr>
<td></td>
<td>SL: Slot number</td>
<td></td>
</tr>
<tr>
<td></td>
<td>n: Data position number</td>
<td></td>
</tr>
<tr>
<td></td>
<td>I: Variable name to store read data</td>
<td></td>
</tr>
<tr>
<td>Writing to I/O data register</td>
<td>OUTPUT SL, n, NOFORMAT;I</td>
<td>The contents that are specified in variable I are stored in data position number n of the module which is installed in slot number SL.</td>
</tr>
<tr>
<td></td>
<td>SL: Slot number</td>
<td></td>
</tr>
<tr>
<td></td>
<td>n: Data position number</td>
<td></td>
</tr>
<tr>
<td></td>
<td>I: Integer variable name or integer</td>
<td></td>
</tr>
<tr>
<td></td>
<td>array variable name</td>
<td></td>
</tr>
<tr>
<td>Reading status of input relay</td>
<td>STATUS SL, n; P</td>
<td>The status of the input relay of the module which is installed in slot number SL is read and stored in variable P.</td>
</tr>
<tr>
<td></td>
<td>SL: Slot number</td>
<td></td>
</tr>
<tr>
<td></td>
<td>n: Data position number (101, 102)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>P: Variable to store the value read.</td>
<td></td>
</tr>
<tr>
<td>Writing to output relay</td>
<td>CONTROL SL, n; P, M</td>
<td>The value of variable P is output to the output relay status of the module which is installed in slot number SL. By using the mask pattern, you can output to only specific relays.</td>
</tr>
<tr>
<td></td>
<td>SL: Slot number</td>
<td></td>
</tr>
<tr>
<td></td>
<td>n: Data position number (101, 102)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>P: Output data</td>
<td></td>
</tr>
<tr>
<td></td>
<td>M: Mask pattern</td>
<td></td>
</tr>
</tbody>
</table>
Input relays are allocated to 2 words in bits as shown below:

![Diagram of input relays allocation](image1.png)

![Diagram of input relays allocation](image2.png)
Output relays are allocated to 2 words in bits as shown below:

---

**Diagram Description:**
- Bits are labeled from 15 to 0.
- Each bit is associated with an output relay.
- Some bits are marked as "Not in use".
- Specific relay outputs (41, 42, 49, 58) are highlighted.

---

**Notations:**
- F060302-3.EPS
- F060302-4.EPS
7. I/O Communications

The operation of this module related to scanning and I/O communications is explained here.

7.1 Scan List

The scan list of this module is explained here.

■ What is a scan list?

Node identification information, the communication type, I/O data size and I/O data storage location are stored in the scan list for each node that communicates with this module. The communication target node is managed and I/O communications and message communications are executed according to this list.

Register the following information as Node Identification Information.

- Vendor ID: Identifier for the company manufacturing the device
- Device type: Code indicating the type of device
- Product code: Code indicating device model

Register the communication type by selecting the following options.

- I/O communications (select from Polling/Bit Strobe) / None
- Message communications / None

Register the I/O data size by specifying the size of the I/O data for I/O communications in the number of bytes and register the location for storing the I/O data by specifying the relative location (Offset from the start of the input data buffer or from the start of the output data buffer) in the I/O data buffer of this module.

An example showing the registration of a device list is given below.

Example: SI Unit Compatible with DeviceNet (SMC Co., Ltd.)

<table>
<thead>
<tr>
<th>Item</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vendor ID</td>
<td>7 (SMC Co., Ltd.)</td>
</tr>
<tr>
<td>Device type</td>
<td>16</td>
</tr>
<tr>
<td>Product code</td>
<td>288</td>
</tr>
<tr>
<td>Communications type</td>
<td>polling only</td>
</tr>
<tr>
<td>Input data size (bytes)</td>
<td>2</td>
</tr>
<tr>
<td>Input data offset (bytes)</td>
<td>100</td>
</tr>
<tr>
<td>Output data size (bytes)</td>
<td>2</td>
</tr>
<tr>
<td>Output data offset (bytes)</td>
<td>100</td>
</tr>
</tbody>
</table>

SEE ALSO

For information on the node identification information, the communications type supported and the I/O data size, see also the device handling manual or data sheet (hard or soft copy). You can also inquire at the sales department of the company manufacturing the device.
Register and Save

The scan list and transmission time interval (see Section 7.5, “Transmission Time Interval”) are registered in the memory using the Register Device/Register Transmission Time Interval commands of the Tool commands (see Section 9.1.4 of this manual for details on the commands) using a Ladder/BASIC program in the CPU module. These are stored in the flash memory with the Store in Flash Memory command. The information which is stored in the flash memory is preserved even if the power supply is turned off.

CAUTION

One can write to flash memory 100,000 times. Make sure that the Ladder/BASIC program that performs flash memory storage is not created every time operation starts.

Create the program for registration/saving by referring to the programming example (Section 9.3, “Sample Program”).

Figure 7.1 Register and Save Sequence
Allocation of I/O data storage location

The storage location for I/O data is allocated in the I/O data buffer (see Section 6.2, “I/O Data Register,” in this Manual) according to the following rules.

- Select the type (polling/bit strobe/none) of the I/O communications method. (When using message communications, select separately from I/O Communications).
- Make sure to allocate data with more than 2 bytes to even addresses. Registration is not possible if an odd address is specified. One-byte data can be allocated to an odd address.
- The buffer need not be used in ascending order of the node address.
- If there is a duplicate allocation with another device outside the area, registration is not possible.
- In a multi-master environment, it is not possible to share the slave in polling/bit strobe communication.

Size and offset are specified in bytes. Offset is the offset from the start of the input data buffer or output data buffer.

<table>
<thead>
<tr>
<th>Node Address</th>
<th>Type of I/O Communications</th>
<th>Output Data</th>
<th>Input Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>P</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>B</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>P</td>
<td>2</td>
<td>10</td>
</tr>
<tr>
<td>4</td>
<td>P</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>B</td>
<td>1</td>
<td>3</td>
</tr>
</tbody>
</table>

P: polling, B: bit strobe

Figure 7.2 Example of Allocation
7.2 Module Operation

Normal Operation

Perform the operations following the steps below. The operation of this module is shown in Figure 7.3.

- If scanning is started without storing output data, system outputs '0' or the data at the time when the previous scanning is stopped.
- Determine the check time taking the following conditions into consideration:
  - It takes approximately 12 seconds to restart scanning once scanning is stopped.
  - Once this module is installed, it takes approximately 6 seconds to reach the status where scanning can start.

Start operation

Store the output data (Write to the output data buffer)*1

Execute the request to start scanning. Turns on the request to start/stop scanning flag.

Has enough time elapsed since the request to start scanning? *2

Is scanning in progress? (Is scan mode flag on?)

- Perform error processing separately for each node
- Perform error processing

Check whether node selected for I/O communications is communicating normally. (Check whether the other node status information is on)

- Write/Read I/O data (Access the I/O data buffer)
- Perform error processing separately for each node

*3 Perform this step for all the nodes which I/O communications are to be established.

*1 If scanning is started without storing output data, system outputs '0' or the data at the time when the previous scanning is stopped.

*2 Determine the check time taking the following conditions into consideration:
  - It takes approximately 12 seconds to restart scanning once scanning is stopped.
  - Once this module is installed, it takes approximately 6 seconds to reach the status where scanning can start.

*3 Perform this step for all the nodes which I/O communications are to be established.
Error processing during scanning

If an error occurs during I/O communications, two operation modes are provided as options, namely continue/stop. During I/O communications, information possessed by the slave may not be in agreement with the information of the scan list, or the stop/continue operation can be specified when a slave is absent.

If "continue" is specified, communication continues with the slaves other than the one in which the error has occurred. Communication automatically resumes when the slave in which the error has occurred becomes available again. However, the red NS LED continues to blink until scanning stops (see Section 4.5, "Displays Panels," in this document.)

Slave operation when communication stops varies with the specifications of the slave. Check the specification of the slave.

Ensuring data concurrency

I/O data are concurrent in every input/output data register (1 word). There are flags that ensure data concurrency in all nodes when handling data that exceed 2 bytes, for instance, when concurrency is required between NC data and data of a barcode reader or between multiple data, etc. However, concurrency should be ensured when the slave transmits the data.

- Input Data

Flags
  - Request to Prohibit Input Data Update flag in control flag of each node
  - Input Data Update Status flag in status flag of each node
Operation from Ladder/BASIC program of CPU module
1. When you wish to read the data, turn on the "request to prohibit input data update flag" of the target node.
2. Wait until the update status flag turns on. When it does, read the data and turn off the prohibit update flag.

Actions of the module
1. Confirms the request flag at the time of I/O communications to the target node. If it is on, updates to the latest input data, turns on the update status flag and enters the stop mode. Henceforth, the prohibit update flag is checked following the I/O communications timing of the target node and if it is found to be on, the input data are not updated.
2. When the prohibit update request turns off, turn off the update status flag and update the input data.

<table>
<thead>
<tr>
<th>CPU module</th>
<th>This module</th>
</tr>
</thead>
<tbody>
<tr>
<td>User program</td>
<td>Interface</td>
</tr>
<tr>
<td>Concurrent access to data</td>
<td>I/O of target node</td>
</tr>
<tr>
<td>Request to prohibit updating input data on</td>
<td>Has updating stopped?</td>
</tr>
<tr>
<td>Has updating stopped?</td>
<td>Request to prohibit updating input data on</td>
</tr>
<tr>
<td>NO</td>
<td>Control flag of the node</td>
</tr>
<tr>
<td></td>
<td>Updating input data status flag</td>
</tr>
<tr>
<td>YES</td>
<td>Status flag of the node</td>
</tr>
<tr>
<td>Read input data</td>
<td>Request to prohibit updating input data on</td>
</tr>
<tr>
<td>Request to prohibit updating input data off</td>
<td>Exit</td>
</tr>
</tbody>
</table>

Figure 7.4 Ensuring Concurrency of Input Data
• Output Data

Flags
- Request to Prohibit Output Data Access flag in control flag of each node.
- Output Data Access Status flag in status flag of each node.

Operation from Ladder/BASIC program of CPU module
1. When you wish to output new data, turn on the request to prohibit output data access flag of the target node.
2. Wait until the access status flag turns on, write the output data in the output data area and turn off the request to prohibit access flag.

Actions of the module
1. If the request to prohibit access flag is on during I/O communications to the target node, the previous data are transmitted to the slave. If in the output data access stop mode, the access status flag is turned on.
2. When the request to prohibit access flag is off, access prohibition is cancelled, the access status flag is turned off and the new output data are transmitted to the slave.
<table>
<thead>
<tr>
<th>CPU module</th>
<th>This module</th>
</tr>
</thead>
<tbody>
<tr>
<td>User program</td>
<td>Interface</td>
</tr>
<tr>
<td>Concurrent access to data</td>
<td></td>
</tr>
<tr>
<td>During request to prohibit access?</td>
<td>YES</td>
</tr>
<tr>
<td>NO</td>
<td></td>
</tr>
<tr>
<td>Request to prohibit access on</td>
<td></td>
</tr>
<tr>
<td>Has access stopped?</td>
<td>YES</td>
</tr>
<tr>
<td>NO</td>
<td></td>
</tr>
<tr>
<td>Write output data</td>
<td></td>
</tr>
<tr>
<td>Request to prohibit access off</td>
<td></td>
</tr>
<tr>
<td>Exit</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>I/O communications of target node</td>
<td></td>
</tr>
<tr>
<td>Has accessing stopped?</td>
<td>YES</td>
</tr>
<tr>
<td>NO</td>
<td></td>
</tr>
<tr>
<td>Request to prohibit output data access</td>
<td></td>
</tr>
<tr>
<td>Control flag of the node</td>
<td></td>
</tr>
<tr>
<td>NO</td>
<td></td>
</tr>
<tr>
<td>Output data access status flag</td>
<td></td>
</tr>
<tr>
<td>Status flag of the node</td>
<td></td>
</tr>
<tr>
<td>YES</td>
<td></td>
</tr>
<tr>
<td>Access status flag on</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>Request to prohibit update off?</td>
<td>YES</td>
</tr>
<tr>
<td>NO</td>
<td></td>
</tr>
<tr>
<td>Access status flag off</td>
<td></td>
</tr>
<tr>
<td>Transmit previous output data</td>
<td></td>
</tr>
<tr>
<td>Transmit new output data</td>
<td></td>
</tr>
<tr>
<td>Exit</td>
<td></td>
</tr>
</tbody>
</table>

Figure 7.5 Ensuring Concurrency of Output Data.
High order/low order byte swapping of multiple byte data

The encoding rules are different for data on the FA-M3 and DeviceNet. Hence in multiple-byte data it is necessary to swap the high order/low order bytes every 2 bytes.

Examples are shown in Figures 7.6 and 7.7. For programming, see Section 7.6, "I/O Data Access (Example of Programming)."

When reading data from this module, it is necessary to swap the high order/low order bytes using a ladder or BASIC program.

When writing data into this module, it is necessary to swap the high order/low order bytes using a Ladder or BASIC program.

Figure 7.6 Example of Swapping of Input Data

Figure 7.7 Example of Swapping of Output Data
7.3 Transmission Time Intervals

The transmission time interval is the period in which I/O communications are performed cyclically. The interval can be specified in the range of 0 to 5000 ms in units of 5 ms using the tool commands. The setup value and value measured at the time of actual operation (measured value) can be stored in the module information of the I/O data register and read. It can also be read using the tool command.

As the transmission time interval not only depends on the number of nodes connected, the type of communication and the capacity of data but also on the performance of the module and the response time of each connected device, it cannot be determined theoretically.

Perform the transmission under actual operating conditions and set a value which is larger than the measured value of the transmission time interval. If the measured value is unsteady, then set it to the maximum measured value.

If there is a delayed response from the slave, and if there were no response when the transmission completes, a situation occurs which is as if the slave were not present at all. In such a case set a long enough transmission time interval.

When the time needed for the transmission to all nodes is greater than the set value, then operate it with a time period that is larger than the set value.

The standard transmission time intervals for a single master configuration with data rate of 500 kbps are displayed in Table 7.1.

<table>
<thead>
<tr>
<th>Number of slave nodes</th>
<th>Transmission time interval (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Up to 9</td>
<td>10</td>
</tr>
<tr>
<td>Up to 23</td>
<td>20</td>
</tr>
<tr>
<td>Up to 37</td>
<td>30</td>
</tr>
<tr>
<td>Up to 51</td>
<td>40</td>
</tr>
<tr>
<td>Up to 63</td>
<td>50</td>
</tr>
</tbody>
</table>
7.4 I/O Response Time

The I/O response time is the time from the start of input to the slave, followed by ladder program processing, until output from the slave.

From a ladder program, read the input data from this module using the special module read command READ. Perform the required processing and write the output data to the module using the special module write command WRITE. The maximum I/O response time required for the above processing is estimated as follows.

Maximum I/O response time = transmission time interval × 2 + 1 scan time + input delay in slave + output delay in slave

It is the same whether the transmission time interval is longer or the time for one scan is longer.

![Figure 7.8 Maximum I/O Response](F0708.EPS)
7.5 Start/Stop Operation and Status Management (Example of Programming)

This sample program shows how to start/stop an operation and how to manage the status. This sample program is based on the following conditions:

- This module is installed in the 4th slot.
- The scan list is registered.
- The sample program is executed only when all the nodes registered in the scan list are communicating normally. (The scanning stops when a transmission error occurs in any node.)
- The sample program stops when normal communication for all nodes is not achieved 20 seconds after turning ON the power supply.

This sample program is described separately in:

1. Section 7.6, “I/O Data Access” (Example of Programming)
2. Section 8.1, “Module Operation,” and

Consult these sections for more details.
## Example of Ladder Programming

(0001)

********Initialization/Clear error********

(0002)

M0035

0001

SET

Y00449

(0003)

X00417

0003

RST

Y00449

(0004)

********Set scan operation mode********

(0005)

M0035

0005

SET

Y00442

(0006)

********Set Initial output data********

(0007)

********Start********

(0008)

********I/O data access and message communication********

(0009)

M0035

0007

READ

4

21

D0100

4

(0010)

X00401

X00402

X00403

X00407

I00002

0009

SET

Y00441

(0011)

TIM

T0001

20s

(0012)

X00409

0018

READ

4

25

D0104

4

(0013)

D0100

= D0104

D0102

= D0106

I00001

(0014)

SET

I00002

(0015)

******Stop*******

(0016)

I00003

0030

RST

Y00441

(0017)

T0001

0020

READ

4

17

D0110

2

(0018)

X00409

I0002

0021

READ

4

29

D0112

4

When it is necessary to set output data before starting the scanning, see Section 7.6, "I/O Data Access (Example of Programming)."

Turn on the Request to start scanning flag after confirming that there are no errors.

CONFIRM Scanning and, if scanning is in process, read the status information of other nodes.

Compare scan list information and status information of other nodes and if they are the same, turn on the data valid flag (I00001) of all the nodes.

CONFIRM 

ERROR STOP

Turn off the request to start scanning where there is an error in external request (I00003), monitor timer timeout, or scanning.

CONFIRM

When it is necessary to set output data before starting the scanning, see Section 7.6, "I/O Data Access (Example of Programming)."

CONFIRM

When it is necessary to set output data before starting the scanning, see Section 8.3, "Sample Program (Message Communication)."

CONFIRM
Example of BASIC Programming

10 ! ******************************************************************************
20 ! * F3LD01-ON SAMPLE PROGRAM
30 ! * Start/Stop Operation and Status Management
40 ! ******************************************************************************
50 ! ****Initial setup****
60 DEFINT A-Z
70 DIM DLIST (3), DSTS (3)
80 LDSLOT = 4
90 ASSIGN LD01 = LDSLOT
100 !
110 !****Initialization Clear Error****
120 CONTROL LDSLOT, 102; $0001, $FFFF : ! Request to Clear Error (Y49)
on, and other relays off
130 CONTROL LDSLOT, 101; $0000, $FFFF : ! Other relays off
140 STS=0
150 WHILE BIT (STS, 0) = 0 : ! Wait till Clear Error Confirmation(X17) is turned ON
160 STATUS LDSLOT, 102; STS
170 END WHILE
180 CONTROL LDSLOT, 102; $0000, $0001 : ! Request to clear Error (Y49)
is off

190 !
200 !****Preparations for the operation****
210 ON TIMER #1,20000 GOTO ERROR@ : ! set 20 second timer
220 !
230 FOR I=0 TO 3 : ! Read scan list
240 ENTER LDSLOT, 21+I NOFORMAT;DLIST (I)
250 NEXT I
260 !
270 !
280 GOSUB DWRITE@ : ! Write output data
290 !
300 !****Start operation****
310 CONTROL LDSLOT, 101; $0200, $0200 : ! Set scan operation mode (Y ON)
320 CONTROL LDSLOT, 101; $0100, $0100 : ! Request to start scanning (Y41 ON)

330 !
340 LOOP1@
350 STS=0
360 STATUS LDSLOT, 101; STS : ! Wait till X9 turns on
370 SCANON=BIT (STS, 8)
380 IF SCANON<>1 THEN LOOP1@
390 FOR I=0 TO 3 : ! Confirm the status information of other nodes
500 ENTER LDSLOT,25+I NOFORMAT;DSTS (I)
510 IF DLIST (I) <> DSTS (I) THEN LOOP1@ : ! Compare with scan list (Wait
till it matches)
520 NEXT  I
530 !
540 !
600 ! ****Valid data****
610 OFF TIMER #1 : ! 20 second timer interrupt off
620 !
630 GOSUB MESSEGB@ : ! Message Communication request
640 !
650 LOOP2@
660 STS=0
670 STATUS LDSLOT,101;STS : ! Confirm that X[CODE] is ON
680 SCANON=BIT (STS,8)
690 IF SCANON=0 THEN ERROR@
700 !
720 GOSUB DREAD@ : ! Read input data
730 GOSUB DWRITE@ : ! Write output data
740 GOTO LOOP2@ : ! Repeat the execution during
scanning
750 !
800 ! ****Error processing**** : ! Error processing
810 ERROR@
820 ENTER LDSLOT,17 NOFORMAT;ERCODE1
830 ENTER LDSLOT,18 NOFORMAT;ERCODE2
840 ENTER LDSLOT,29 NOFORMAT;ERCODE1 : ! Read the node address in error
850 ENTER LDSLOT,30 NOFORMAT;ERCODE2
860 ENTER LDSLOT,31 NOFORMAT;ERCODE3
870 ENTER LDSLOT,32 NOFORMAT;ERCODE4
880 DP "ERCODE1=",HEX$ (ERCODE1) : ! Error code 1(hexadecimal
display)
890 DP "ERCODE2=",HEX$ (ERCODE2) : ! Error code 2(hexadecimal
display)
900 DP "ERCODE1=",HEX$ (ERCODE1) : ! Display error node
(hexadecimal display)
910 DP "ERCODE2=",HEX$ (ERCODE2)
920 DP "ERCODE3=",HEX$ (ERCODE3)
930 DP "ERCODE4=",HEX$ (ERCODE4)
940 CONTROL LDSLOT, 101;$0000,$FFFF : ! Request to stop scanning
950 STOP
3000 END
7.6 I/O Data Access (Example of Programming)

A programming example of I/O data access is shown below. In this program, the scan list is registered in the input and output data areas and normal communication is performed.

**Input data**

<table>
<thead>
<tr>
<th>Data position no.</th>
<th>Data type</th>
<th>Data register (used in Ladder)</th>
<th>Input variable (used in BASIC)</th>
</tr>
</thead>
<tbody>
<tr>
<td>510</td>
<td>Word data</td>
<td>D0001</td>
<td>IN1</td>
</tr>
<tr>
<td>511</td>
<td>Word data</td>
<td>D0002</td>
<td>IN2</td>
</tr>
<tr>
<td>512</td>
<td>Word data</td>
<td>D0003</td>
<td>IN3</td>
</tr>
<tr>
<td>513</td>
<td>Unused</td>
<td>D0004</td>
<td>IN4</td>
</tr>
</tbody>
</table>

**Output data**

<table>
<thead>
<tr>
<th>Data position no.</th>
<th>Data type</th>
<th>Data register (used in Ladder)</th>
<th>Input variable (used in BASIC)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1021</td>
<td>Word data</td>
<td>D0011</td>
<td>OUT1</td>
</tr>
<tr>
<td>1022</td>
<td>Word data</td>
<td>D0012</td>
<td>OUT 2</td>
</tr>
<tr>
<td>1023</td>
<td>Word data</td>
<td>D0013</td>
<td>OUT 3</td>
</tr>
<tr>
<td>1024</td>
<td>Unused</td>
<td>D0014</td>
<td>OUT 4</td>
</tr>
</tbody>
</table>
Example of Ladder Programming

**Operation**

**Read data**

- **I00001**
  - READ 4 510 D0001 4

**Write Data**

- **M0033**
  - MOV 123 4 D0011
  - MOV 123 D0012
  - MOV 12 D0013
  - MOV I00117

**Data format conversion**

- I00101 to I00108 are used.

**Data format conversion**

- I00117 to I00124 are used.

**Write output data.**

- **I00003**
  - SET I00003

**Data setting completed** (First scan request)

Confirm that scan data is valid (I00001ON) and read input data.

Data format conversion

Data setting completed (First scan request)
Example of BASIC Programming

This program is only a subroutine and is called from the program described in Section 7.5, "Start/Stop Operation and Status Management."

1000 ! ****Read input data****
1010 DREAD@
1020 ENTER LDSLOT, 511 NOFORMAT ; IN1 : ! Read input data
1030 IN1=ROTATE(IN1,8) : ! Data 8-bit rotate
1040 ENTER LDSLOT, 512 NOFORMAT ; IN2
1050 IN2=ROTATE(IN2,8)
1060 ENTER LDSLOT, 513 NOFORMAT ; IN3
1070 IN3=ROTATE(IN3,8)
1080 ENTER LDSLOT, 514 NOFORMAT ; IN4
1090 IN4=BINAND ($FF00, IN4) : ! High order 8-bit truncation
1100 RETURN
1200 ! ****Write output data****
1210 DWRITE@
1220 OUT1=1234
1230 OUT2=123
1240 OUT3=12
1250 OUT4=2
1260 OUT=ROTATE(OUT1,8) : ! Data 8-bit rotation
1270 OUT=ROTATE(OUT2,8)
1280 OUT3=ROTATE(OUT3,8)
1290 OUTPUT LDSLOT, 1021 NOFORMAT ; OUT1 : ! Write output data
1300 OUTPUT LDSLOT, 1022 NOFORMAT ; OUT2
1310 OUTPUT LDSLOT, 1023 NOFORMAT ; OUT3
1320 OUTPUT LDSLOT, 1024 NOFORMAT ; OUT4
8. Message Communications

In this section, operation of message communications in this module and the interface is explained.

8.1 Module Operation

Operation during message communications in this module are explained here.

■ Conditions for request to receive communication

When the following conditions are satisfied, a request to receive communication is transmitted.

- "Message communications" are registered in the communications type of the target node in the scan list.
- Scanning is in process
- The target node and the node information match and communication is in progress.

■ Transmission procedure

Use the CPU module’s Ladder/BASIC program and follow the procedure below for communication.
Message transmission

Are both message communication request and completion flags turned off?

YES

Create a request message. (write to message communication transmission buffer.)

NO

Message communication request flag is turned on.

Is the message communication completion flag on?

YES

Read the response message (read it from message communication reception buffer).

NO

Message communication request flag is turned off.

Confirm the contents of the response message/exit code.

End
Figure 8.1 Transmission Procedure and Actions of the Module
8.2 Interface

8.2.1 Request and Response Formats

The request and response formats in the message communication transmit/receive buffer are as follows:

- **Request**

<table>
<thead>
<tr>
<th>Data position no.</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>1501</td>
<td>0 (reserved)</td>
</tr>
<tr>
<td>1502</td>
<td>Transmission destination node address</td>
</tr>
<tr>
<td>1503</td>
<td>0</td>
</tr>
<tr>
<td>1504</td>
<td>0</td>
</tr>
<tr>
<td>1505</td>
<td>Size of service data (number of bytes)</td>
</tr>
<tr>
<td>1506</td>
<td>Service code specified for DeviceNet</td>
</tr>
<tr>
<td>1507</td>
<td>Class ID of transmission destination</td>
</tr>
<tr>
<td>1508</td>
<td>Instance ID of transmission destination</td>
</tr>
<tr>
<td>1509 (up to 1550)</td>
<td>Service data</td>
</tr>
</tbody>
</table>

- **Response message**

<table>
<thead>
<tr>
<th>Data position no.</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>1551</td>
<td>0 (reserved)</td>
</tr>
<tr>
<td>1552</td>
<td>Response source node address</td>
</tr>
<tr>
<td>1553</td>
<td>Exit code</td>
</tr>
<tr>
<td>1554</td>
<td>0 (reserved)</td>
</tr>
<tr>
<td>1555</td>
<td>Size of service data (number of bytes)</td>
</tr>
<tr>
<td>1556</td>
<td>Service code specified for DeviceNet</td>
</tr>
<tr>
<td>1557 (up to 1600)</td>
<td>Service data</td>
</tr>
</tbody>
</table>

- **Limit of service data size**
  - Service data (for Request) has a maximum of 84 bytes.
  - When service data (for Response Messages) exceeds 88 bytes, code for "abnormal exit" is stored in the exit code area of the response. In such situations, the first 88 bytes of the service data is kept in the response message.

- **High order / low order byte swapping of multiple-byte data in service data**

  The encoding rules are different for data on FA-M3 and DeviceNet. Hence, in multi-byte data, it is necessary to swap the byte high order/low order byte every 2 bytes.

  See also Section 7.2, "Module Operation."

**SEE ALSO**

For information on the service code, service data, class ID and instance ID, refer to the “DeviceNet Specification” published by ODVA.
8.2.2 Exit Codes

The exit codes used for message communications are described herein.

<table>
<thead>
<tr>
<th>Error code</th>
<th>Description</th>
<th>Troubleshooting</th>
</tr>
</thead>
<tbody>
<tr>
<td>$0100</td>
<td>Connection for message communications is unavailable.</td>
<td>Target node may be absent or message communications are not registered as communication type in scan list. Confirm the network status and device list.</td>
</tr>
<tr>
<td>$0101</td>
<td>Parameter error (Class ID)</td>
<td>Class ID specification area for target node is less than or equal to 255, but a value above 256 is specified. Specify a valid value.</td>
</tr>
<tr>
<td>$0102</td>
<td>Parameter error (instance ID)</td>
<td>Instance ID specification range for the target node is less than or equal to 255, but a value above 256 is specified. Specify a valid value.</td>
</tr>
<tr>
<td>$F101</td>
<td>Receive buffer size overflow for response message</td>
<td>Normal receive is not possible for this service. First 88 bytes of the service data is stored in receive buffer.</td>
</tr>
<tr>
<td>$F102</td>
<td>Parameter error (node address)</td>
<td>Specify the correct node address in the range from 0 to 63.</td>
</tr>
<tr>
<td>$E112</td>
<td>Network power failure during communications</td>
<td>See error codes (see Section 10, “Errors and Troubleshooting”).</td>
</tr>
<tr>
<td>$E144</td>
<td>No response from target device (timeout)</td>
<td>See error codes (see Section 10, “Errors and Troubleshooting”).</td>
</tr>
</tbody>
</table>
8.3 Sample Programs

Remote setting of operation (output cut-off/maintain) during abnormal communications of a discrete output device is given as a sample program of message communications.

The target device supports the following services and attributes.

- **Service**
  - SetAttribute_Single (DeviceNet common service, $10)

- **Attribute**
  - Class ID = 9, Instance ID = 1, Attribute ID = 5
  - (Prescribed attribute value for operation during a communication error)

- **Setup value**
  - 8-bit integer (0 = Cut off, 1 = Maintain)

**Message request format**

<table>
<thead>
<tr>
<th>Data position no.</th>
<th>Data Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>1501</td>
<td>0  Reserved</td>
</tr>
<tr>
<td>1502</td>
<td>1  Transmission destination node address</td>
</tr>
<tr>
<td>1503</td>
<td>0  Reserved</td>
</tr>
<tr>
<td>1504</td>
<td>0  Reserved</td>
</tr>
<tr>
<td>1505</td>
<td>2  Service data size (byte)</td>
</tr>
<tr>
<td>1506</td>
<td>$10 Service code &quot;Set_Attribute_Single&quot;</td>
</tr>
<tr>
<td>1507</td>
<td>9  Class ID</td>
</tr>
<tr>
<td>1508</td>
<td>1  Instance ID</td>
</tr>
<tr>
<td>1509 High-order 8 bits</td>
<td>$05 Service data Attribute ID, 5</td>
</tr>
<tr>
<td>1509 Low-order 8 bits</td>
<td>$01 Service data Setup value, 1</td>
</tr>
</tbody>
</table>

**Message response format (during normal exit)**

<table>
<thead>
<tr>
<th>Data position no.</th>
<th>Data Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>1551</td>
<td>0  Reserved</td>
</tr>
<tr>
<td>1552</td>
<td>1  Transmission destination node address, 1</td>
</tr>
<tr>
<td>1553</td>
<td>0  Error code, 0</td>
</tr>
<tr>
<td>1554</td>
<td>0  Reserved</td>
</tr>
<tr>
<td>1555</td>
<td>0  Size of service data, 0 byte</td>
</tr>
<tr>
<td>1556</td>
<td>$90 Service code ($80 + service code)</td>
</tr>
</tbody>
</table>
Example of Ladder Programming

Only message communications are given in this program. Use it with the program given in Section 7.5, “Start/Stop Operation and Status Management.”

1. **Message communication**
   - Start relay I00001, confirm communication status. (Refer to 7.5 “Start/Stop Operation and Status Management” (Example of programming))
   - Reserved
   - Message receiver
   - Setting node address
   - Service data size (Byte)
   - Service code (Set_Attribute_Single)
   - Class ID
   - Instance ID
   - Service data

2. **Write service data.**
3. **Message communication request on**
4. **Confirm message status flag.**
5. **Read response after confirming completion of message communications.**
6. **Message communication request off**
Example of BASIC Programming

This program is only a subroutine and can be called from the program described in Section 7.5, “Start/Stop Operation and Status Management.”

```
2000 ! ****Message communication****
2010 MESSAGE@
2020 OUTPUT LDSLOT ,1501 NOFORMAT ;0 : ! Reserved
2030 OUTPUT LDSLOT ,1502 NOFORMAT ;1 : ! Node number
2040 OUTPUT LDSLOT ,1503 NOFORMAT ;0 : ! Reserved
2050 OUTPUT LDSLOT ,1504 NOFORMAT ;0 : ! Reserved
2060 OUTPUT LDSLOT ,1505 NOFORMAT ;2 : ! Service data size (Byte)
2070 OUTPUT LDSLOT ,1506 NOFORMAT ;$10 : ! Service code2080
    OUTPUT LDSLOT ,1507 NOFORMAT ;9 : ! Class ID
2090 OUTPUT LDSLOT ,1508 NOFORMAT ;1 : ! Instance ID
2100 SEND1=$0501 :  ! Service data
2130 OUTPUT LDSLOT ,1509 NOFORMAT ;SEND1 : ! Write service data
2140 OUTPUT LDSLOT ,1510 NOFORMAT ;SEND2 : !
2150 OUTPUT LDSLOT ,191 NOFORMAT ;1 : !Message communication request
2160 !
2170 MSTS=0
2180 WHILE MSTS=0 : ! Wait till message status flag turns ON
2190 ENTER LDSLOT 291, NOFORMAT ;MSTS
2200 END WHILE
2210 ENTER LDSLOT ,1551 NOFORMAT ;MSG1 : ! Read response
2220 ENTER LDSLOT ,1552 NOFORMAT ;MSG2
2230 ENTER LDSLOT ,1553 NOFORMAT ;MSG3
2240 ENTER LDSLOT ,1554 NOFORMAT ;MSG4
2250 ENTER LDSLOT ,1555 NOFORMAT ;MSG5
2260 ENTER LDSLOT ,1556 NOFORMAT ;MSG6
2270 !
2280 OUTPUT LDSLOT ,191 NOFORMAT ;0 : ! Release Message Comm. Request
2290 RETURN
```
9. **Configuration**

Configuration includes the following:

- Creating a scan list
- Registering the transmission time interval
- Storing in flash memory
- Remote setting of other nodes (see Section 8.3, “Sample Programs,” in this manual)

Configuration is performed by starting the Ladder/BASIC program for setting configuration on the CPU module which is defined when this module is used. Configuration is confirmed using Ladder Diagram Support Program Tool M3 or BASIC Programming Tool M3.

9.1 **Tool Commands**

Tool commands are an interface that can be used to define the actions of this module using the Ladder/BASIC program when preparing for operation.

Please do not use the tool commands during operation.

9.1.1 **Commands**

Tool commands are given in Table 9.1.

<table>
<thead>
<tr>
<th>Command no.</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>$03</td>
<td>Register Device (Scan List)</td>
</tr>
<tr>
<td>$04</td>
<td>Read Device (Scan List)</td>
</tr>
<tr>
<td>$05</td>
<td>Delete Device (Scan List)</td>
</tr>
<tr>
<td>$71</td>
<td>Store Flash Memory (Transmission Time Interval/Scan List)</td>
</tr>
<tr>
<td>$72</td>
<td>Register Transmission Time Interval</td>
</tr>
<tr>
<td>$73</td>
<td>Read Master Information</td>
</tr>
</tbody>
</table>
9.1.2 Request Procedure

The request procedure for the tool commands is given below.

```
Tool command execution

Are both tool request to execute command flag and completion flag off?

YES

Creating the requested command (writing command to tool command buffer)

Turns on tool request to execute command flag

Is the tool command execution completion flag on?

YES

Reads the response (reads the response from tool command buffer)

Turns off tool request to execute command flag

Confirms the contents of response/exit code

End
```
Figure 9.1 Tool Command Request Procedures and Actions of the Module
### 9.1.3 Basic Command Formats

The format for a request/response (normal exit/error exit) is given below.

- **Request**

<table>
<thead>
<tr>
<th>Data position no.</th>
<th>Content</th>
</tr>
</thead>
<tbody>
<tr>
<td>400</td>
<td>Command number</td>
</tr>
<tr>
<td>401 to 449</td>
<td>Parameters</td>
</tr>
</tbody>
</table>

- **Response (normal exit)**

<table>
<thead>
<tr>
<th>Data position no.</th>
<th>Content</th>
</tr>
</thead>
<tbody>
<tr>
<td>400</td>
<td>Command number</td>
</tr>
<tr>
<td>401 to 449</td>
<td>Parameters</td>
</tr>
</tbody>
</table>

- **Response (error exit)**

<table>
<thead>
<tr>
<th>Data position no.</th>
<th>Content</th>
</tr>
</thead>
<tbody>
<tr>
<td>400</td>
<td>Command number+$8000</td>
</tr>
<tr>
<td>401</td>
<td>Exit code</td>
</tr>
</tbody>
</table>

**TIP**

The tool command buffer uses the same area for requests and responses.
9.1.4 Details of Commands

1) Register device

Registers device information for each node in the scan list. Nodes already registered cannot be registered again. Perform registration after deleting the devices and emptying the information in the scan list.

- Request

<table>
<thead>
<tr>
<th>Data position no.</th>
<th>Content</th>
</tr>
</thead>
<tbody>
<tr>
<td>400</td>
<td>Command = $03</td>
</tr>
<tr>
<td>401</td>
<td>Node number</td>
</tr>
<tr>
<td>402</td>
<td>Vendor ID</td>
</tr>
<tr>
<td>403</td>
<td>Device type</td>
</tr>
<tr>
<td>404</td>
<td>Product code</td>
</tr>
<tr>
<td>405-407</td>
<td>Reserved</td>
</tr>
<tr>
<td>408</td>
<td>Communications type</td>
</tr>
<tr>
<td>409-411</td>
<td>Reserved</td>
</tr>
<tr>
<td>412</td>
<td>Size of output data (bytes)</td>
</tr>
<tr>
<td>413</td>
<td>Offset of output data (bytes)</td>
</tr>
<tr>
<td>414-415</td>
<td>Reserved</td>
</tr>
<tr>
<td>416</td>
<td>Size of input data (bytes)</td>
</tr>
<tr>
<td>417</td>
<td>Offset of input data (bytes)</td>
</tr>
<tr>
<td>418-426</td>
<td>Reserved</td>
</tr>
</tbody>
</table>

- Response (normal exit)

<table>
<thead>
<tr>
<th>Data position no.</th>
<th>Content</th>
</tr>
</thead>
<tbody>
<tr>
<td>400</td>
<td>Command = $03</td>
</tr>
</tbody>
</table>

- Response (error exit)

<table>
<thead>
<tr>
<th>Data position no.</th>
<th>Content</th>
</tr>
</thead>
<tbody>
<tr>
<td>400</td>
<td>$8003</td>
</tr>
<tr>
<td>401</td>
<td>Exit code</td>
</tr>
</tbody>
</table>

- Supplement

  - Set reserved area to 0.
  - Communications type

Use only one method of I/O communications. Both I/O communications and message communications can be selected.
R: System reserved
EX: Message communications
P: I/O communications: Polling
ST: I/O communications: Bit strobe

- The I/O data offset is the relative position from the beginning of every I/O data buffer. The unit for the I/O data offset/size is bytes.
- The input data size of a Bit Strobe should be less than or equal to 8 bytes.
- Set the size/offset to 0 when no I/O communications are being carried out.
- Registration is not possible during scanning.

2) Read device

Reads device information of a node from the scan list.

- Request

<table>
<thead>
<tr>
<th>Data position no.</th>
<th>Content</th>
</tr>
</thead>
<tbody>
<tr>
<td>400</td>
<td>Command = $04</td>
</tr>
<tr>
<td>401</td>
<td>Node number</td>
</tr>
</tbody>
</table>

- Response (normal exit)

<table>
<thead>
<tr>
<th>Data position no.</th>
<th>Content</th>
</tr>
</thead>
<tbody>
<tr>
<td>400</td>
<td>Command = $04</td>
</tr>
<tr>
<td>401</td>
<td>Node number</td>
</tr>
<tr>
<td>402</td>
<td>Vendor ID</td>
</tr>
<tr>
<td>403</td>
<td>Device type</td>
</tr>
<tr>
<td>404</td>
<td>Product code</td>
</tr>
<tr>
<td>405-407</td>
<td>Reserved</td>
</tr>
<tr>
<td>408</td>
<td>Communications type</td>
</tr>
<tr>
<td>409-411</td>
<td>Reserved</td>
</tr>
<tr>
<td>412</td>
<td>Size of output data (bytes)</td>
</tr>
<tr>
<td>413</td>
<td>Offset of output data (bytes)</td>
</tr>
<tr>
<td>414-415</td>
<td>Reserved</td>
</tr>
<tr>
<td>416</td>
<td>Size of input data (bytes)</td>
</tr>
<tr>
<td>417</td>
<td>Offset of input data (bytes)</td>
</tr>
<tr>
<td>418-426</td>
<td>Reserved</td>
</tr>
</tbody>
</table>

- Response (error exit)

<table>
<thead>
<tr>
<th>Data position no.</th>
<th>Content</th>
</tr>
</thead>
<tbody>
<tr>
<td>400</td>
<td>$8004</td>
</tr>
<tr>
<td>401</td>
<td>Exit code</td>
</tr>
</tbody>
</table>

T090104-4.EPS
T090104-5.EPS
T090104-6.EPS
• Supplement
  • For information on communications type/offset, refer to the description of device registration.

3) Delete device

Deletes the device information of a node from the scan list.

• Request

<table>
<thead>
<tr>
<th>Data position no.</th>
<th>Content</th>
</tr>
</thead>
<tbody>
<tr>
<td>400</td>
<td>Command = $05</td>
</tr>
<tr>
<td>401</td>
<td>Node number</td>
</tr>
</tbody>
</table>

• Response (normal exit)

<table>
<thead>
<tr>
<th>Data position no.</th>
<th>Content</th>
</tr>
</thead>
<tbody>
<tr>
<td>400</td>
<td>Command = $05</td>
</tr>
</tbody>
</table>

• Response (error exit)

<table>
<thead>
<tr>
<th>Data position no.</th>
<th>Content</th>
</tr>
</thead>
<tbody>
<tr>
<td>400</td>
<td>$8005</td>
</tr>
<tr>
<td>401</td>
<td>Exit code</td>
</tr>
</tbody>
</table>

• Supplement
  • Deletion is not possible during scanning.

4) Store in flash memory

Writes the transmission time interval/device list to flash memory.

• Request

<table>
<thead>
<tr>
<th>Data position no.</th>
<th>Content</th>
</tr>
</thead>
<tbody>
<tr>
<td>400</td>
<td>Command = $71</td>
</tr>
</tbody>
</table>

• Response (normal exit)

<table>
<thead>
<tr>
<th>Data position no.</th>
<th>Content</th>
</tr>
</thead>
<tbody>
<tr>
<td>400</td>
<td>Command = $71</td>
</tr>
</tbody>
</table>
• Response (error exit)

<table>
<thead>
<tr>
<th>Data position no.</th>
<th>Content</th>
</tr>
</thead>
<tbody>
<tr>
<td>400</td>
<td>$8071</td>
</tr>
<tr>
<td>401</td>
<td>Exit code</td>
</tr>
</tbody>
</table>

• Supplement
  - Writing is not possible during scanning.

5) Register transmission time interval

Registers transmission time intervals.

• Request

<table>
<thead>
<tr>
<th>Data position no.</th>
<th>Content</th>
</tr>
</thead>
<tbody>
<tr>
<td>400</td>
<td>Command = $72</td>
</tr>
<tr>
<td>401-402</td>
<td>Reserved</td>
</tr>
<tr>
<td>403</td>
<td>Transmission time interval [msec]</td>
</tr>
</tbody>
</table>

• Response (normal exit)

<table>
<thead>
<tr>
<th>Data position no.</th>
<th>Content</th>
</tr>
</thead>
<tbody>
<tr>
<td>400</td>
<td>Command = $72</td>
</tr>
</tbody>
</table>

• Response (error exit)

<table>
<thead>
<tr>
<th>Data position no.</th>
<th>Content</th>
</tr>
</thead>
<tbody>
<tr>
<td>400</td>
<td>$8072</td>
</tr>
<tr>
<td>401</td>
<td>Exit code</td>
</tr>
</tbody>
</table>

• Supplement
  - Set reserved areas to 0.
  - Valid range for the transmission time interval is 0 to 5000 [ms].
  - Registration cannot be carried out during scanning.

6) Read master information

Reads information related to the master.

• Request

<table>
<thead>
<tr>
<th>Data position no.</th>
<th>Content</th>
</tr>
</thead>
<tbody>
<tr>
<td>400</td>
<td>Command = $73</td>
</tr>
</tbody>
</table>
• Response (normal exit)

<table>
<thead>
<tr>
<th>Data position no.</th>
<th>Content</th>
</tr>
</thead>
<tbody>
<tr>
<td>400</td>
<td>Command = $73</td>
</tr>
<tr>
<td>401-402</td>
<td>Reserved</td>
</tr>
<tr>
<td>403</td>
<td>Transmission time interval [msec]</td>
</tr>
<tr>
<td>404</td>
<td>Reserved</td>
</tr>
<tr>
<td>405</td>
<td>Vendor ID</td>
</tr>
<tr>
<td>406</td>
<td>Device type</td>
</tr>
<tr>
<td>407</td>
<td>Product code</td>
</tr>
<tr>
<td>508</td>
<td>Revision</td>
</tr>
<tr>
<td>409</td>
<td>Status</td>
</tr>
<tr>
<td>410-411</td>
<td>Serial number</td>
</tr>
<tr>
<td>412-429</td>
<td>Product name</td>
</tr>
<tr>
<td>430</td>
<td>Reserved</td>
</tr>
</tbody>
</table>

• Response (error exit)

<table>
<thead>
<tr>
<th>Data position no.</th>
<th>Content</th>
</tr>
</thead>
<tbody>
<tr>
<td>400</td>
<td>$8073</td>
</tr>
<tr>
<td>401</td>
<td>Exit code</td>
</tr>
</tbody>
</table>

• Supplement

This module keeps the module information shown in Table 9.2, in accordance with the DeviceNet specification. This information can be read using the read master information command.

Table 9.2 Module Information Summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Value of this module</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vendor ID</td>
<td>Identifier of device manufacturer</td>
<td>$FA</td>
</tr>
<tr>
<td>Device type</td>
<td>Code indicating device type</td>
<td>$C</td>
</tr>
<tr>
<td>Product code</td>
<td>Code indicating device model</td>
<td>1</td>
</tr>
<tr>
<td>Revision</td>
<td>Product revision of device</td>
<td>*1</td>
</tr>
<tr>
<td>Status</td>
<td>Module status of device</td>
<td>*2</td>
</tr>
<tr>
<td>Serial number</td>
<td>Serial number of device</td>
<td>*3</td>
</tr>
<tr>
<td>Product name</td>
<td>Product name of device</td>
<td>*4</td>
</tr>
</tbody>
</table>

*1 The revision starts from 1.1 ($0101) in the DeviceNet manual.
*2 The status codes given in the DeviceNet manual are used.
*3 Each device is given a unique number.
*4 FA-M3 LD01 Scanner Module
9.1.5 Exit Codes

The unique exit codes used in response to tool commands are given in Table 9.3. In the case of an incorrect response to a request, either confirm the command number/parameter or delete device/stop scanning before the request, according to the exit code.

Table 9.3 Summary of Exit Codes

<table>
<thead>
<tr>
<th>Item</th>
<th>Exit code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unsupported command</td>
<td>$0001</td>
<td>The specified command is not supported.</td>
</tr>
<tr>
<td>Node number error</td>
<td>$0002</td>
<td>The specified node number is beyond the range.</td>
</tr>
<tr>
<td>Device already</td>
<td>$0005</td>
<td>The device is already registered.</td>
</tr>
<tr>
<td>Device is not set</td>
<td>$0006</td>
<td>The scan list is empty.</td>
</tr>
<tr>
<td>Scanning in progress</td>
<td>$000A</td>
<td>Execution is not possible during scanning.</td>
</tr>
<tr>
<td>Outside the data</td>
<td>$000D</td>
<td>Value specified is outside the range of the I/O data area.</td>
</tr>
<tr>
<td>Communications type</td>
<td>$000F</td>
<td>The specified communications type is not supported.</td>
</tr>
<tr>
<td>Flash memory error</td>
<td>$1283</td>
<td>Unable to write to flash memory due to exceeding the limit allowed for writing to flash memory.</td>
</tr>
<tr>
<td>Connection not used</td>
<td>$8000</td>
<td>Offset/size are specified even though I/O communications is not selected.</td>
</tr>
<tr>
<td>Data area undefined</td>
<td>$8001</td>
<td>Offset/size are not specified even though I/O communications is selected.</td>
</tr>
<tr>
<td>Duplication of data area</td>
<td>$E131</td>
<td>The data area being used is duplicated in another node.</td>
</tr>
<tr>
<td>Master information</td>
<td>$E133</td>
<td>Beyond the range of master information parameter (transmission time interval).</td>
</tr>
</tbody>
</table>
9.2 Configuration Procedure

Configuration involves the following steps:

1. Create a scan list and set the transmission time interval.

Register the nodes with which communication is to be established (create a scan list) and set the transmission time interval using the tool command. Set the transmission time interval to either 0 msec. or to a specific value (it is necessary to set the values again using values measured during the test run).

In the sample program in Section 9.3 of this manual, the steps for registering the devices and setting the transmission time interval are given. After the setting is done, execute the program.

2. Test Run

Start scanning and verify whether communication is taking place properly with the devices registered in the scan list. If there is an error, check the settings, wiring, scan list, etc. of this module and other devices, then re-adjust and set again.

Once normal communication with all the nodes is confirmed, compute the transmission time interval measured value (Data position no. 8 of I/O data register) (If the measured values are unsteady, record the maximum value).

For operation, see the sample program in Section 7.5, “Start/Stop Operation and Status Management (Example of Programming),” in this manual.

3. Evaluate Configuration Results

If communication does not take place properly and if the measured value of the transmission time interval (actual value during scanning) exceeds the set value, set an appropriate value again, and then test run and confirm the operation.

CAUTION

(1) In the sample program given in Section 9.3 of this manual, the processes necessary for configuration (Setting transmission time interval / Initialising scan list / device registration / storing in flash memory) are carried out in one program. When changing the configuration, re-run the program after making the necessary changes.

(2) The transmission time interval/scan list is stored in the flash memory all at once. Please note that this deletes all the previous data in the flash memory.
9.3 Sample Program

A sample program for configuration is given here. Configuration is carried out as follows:

1. Configuration
2. Clear error
3. Setup transmission time interval
4. Initialize scan list
5. Register device
6. Store in flash memory
7. End

- **Clear error**: Clears the error flags of this module.
- **Setup transmission time interval**: Sets up transmission time interval.
- **Initialize scan list**: Deletes all the registered devices from the scan list.
- **Register device**: Registers device and creates a scan list.
- **Store in flash memory**: Stores the register scan list and transmission time interval in flash memory. If it is not stored in flash memory, the scan list is lost when power is switched to off.
• Clear Error

The clear error command is carried out as follows:

1. **Clear error**
2. **Request to clear error relay (Ynnn49) on**
3. **Clear error completed confirmation relay(Xnnn17) on**
4. **Release request to clear error relay (Xnnn49) off**
5. **Release request to clear error confirmation relay (Xnnn17) off**
6. **End**

• Setting Transmission Time Interval

Setting the transmission time interval is carried out as shown below.
In the sample program, the transmission time interval is set to 20 ms.
9-14

Setting transmission time

Write a request to command buffer

Request to execute command relay (Y×××58) on

Execute command completed confirmation relay (X×××26) on

Read response

Any errors? (command codes of request/response do not match)

NO

Release request to execute command relay (Y×××58) off

Release request to execute command confirmation relay (X×××26) off

END

400 Command $72
403 Set Transmission Time Interval

Error processing

Write a request to command buffer

Request to execute command relay (Y×××58) on

Execute command completed confirmation relay (X×××26) on

Read response

Any errors? (command codes of request/response do not match)

NO

Release request to execute command relay (Y×××58) off

Release request to execute command confirmation relay (X×××26) off

END

Error processing

F0903-3.EPS
• Initializing Scan list

If there are registered devices, they are deleted from the scan list. Deletion from the scan list is carried out in the following manner.

![Flowchart diagram]

- Initialising scan list
- If there are registered devices, they are deleted from the scan list.
- Deletion from the scan list is carried out in the following manner.
- **Scan information not set relay (X00007) ON?**
  - NO
  - Node address N=0
    - Write a request to command buffer
    - Request to execute command relay (Y00058) on
    - Execute command completed confirmation relay (X00026) on
    - Read response
    - Any errors?
      - NO
        - Release request to execute command relay (Y00058) off
        - Release request to execute command confirmation relay (X00026) off
        - N=N+1
      - YES
        - Exit Code=6?
          - NO
            - Error processing
          - YES
            - Device Not Set (the request for deletion was for a node for which device was not set, hence Exit Code=6 is normal)
    - YES
      - All nodes completed? N>=64
          - NO
          - END
          - YES

400 Command $05 401 Node address, N
Registering Devices

The procedure for device registration is given below.

In the sample program, the following two devices are being registered.

<table>
<thead>
<tr>
<th>Item</th>
<th>Data</th>
<th>Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>Node no</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>Vendor ID</td>
<td>47 (Omron Corporation)</td>
<td>47 (Omron Corporation)</td>
</tr>
<tr>
<td>Device type</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Product code</td>
<td>100 (Type DRT1-ID08)</td>
<td>101 (Model DRT1-OD08)</td>
</tr>
<tr>
<td>Communication type</td>
<td>4 (Bit Strobe)</td>
<td>3 (Polling and message communications)</td>
</tr>
<tr>
<td>Input data size (bytes)</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Input data offset (bytes)</td>
<td>27 (Input data position 513 low order)</td>
<td>0</td>
</tr>
<tr>
<td>Output data size (bytes)</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Output data offset (bytes)</td>
<td>0</td>
<td>49 (Output data position 1024 low order)</td>
</tr>
</tbody>
</table>
Registering devices

Read device registration data

Is data available?

NO

YES

Write a request to command buffer

Request to execute command relay (Y\textsuperscript{nnn}58) on

Execute command completed confirmation relay (X\textsuperscript{nnn}26) on

Read response

Any errors?

YES

Error processing

NO

Request to execute command relay (Y\textsuperscript{nnn}58) off

Execute command completed confirmation relay (X\textsuperscript{nnn}26) off

END
- **Storing in flash memory**

The steps for storing in flash memory are given below.

1. **Storing in flash memory**
2. Write a request to command buffer
3. Request to execute command relay (Y58) on
4. Execute command completed confirmation relay (X26) on
5. Read response
6. **Any errors?**
   - **NO**
   - **YES**
     - Error processing
   - Release request to execute command relay (Y58) off
   - Release request to execute command confirmation relay (X26) off
7. **END**
Sample Program Using Ladder Program

A sample program using ladder programs is shown here. The sample program is created based on the following conditions.

- This module is installed in slot 4.
- The contents for device registration should be set in advance as given below.
  
  D0051 ~ information for first item
  D0101 ~ information for second item
  D0051+50×N information for Nth item
  
  Set -1 at the end of the data.

- Handling internal register
  
  D0001 = 6 Normal exit
  D0001 = 7 Error has occurred
  D0011: Process number where an error has occurred
  D0012: Node address where an error has occurred (during device deletion)
  Information of (N-1)th device where an error has occurred (during device registration)
  D0013: Command where an error has occurred ($8000+command number)
  D0014: Exit code when an error has occurred
********Initialisation********

M0035 | BSET 0 D0001 2048

Clear data register.

BSET 0 D2049 1200

Clear Y relay of this module.

MOV 0 Y00433

Clear relay.

MOV 0 I00001

Specify the slot number of this module.

MOV 4 D0007

********Clear error********

M0035 | SET Y00449

Clear error request on.

M00417 | RST Y00449

Confirm that clear error has completed, and then clear error request off.

MOV 1 D0001

Go to next processing.

********Execute command********

I00001 X00426 | SET Y00458

Request to execute command.

I00001 | RST I00001
********Setting transmission time interval********

D0001 = 1

WRITE 0 D0007 400 50

100002

********Check that device is not registered********

D0001 = 2

X00407

MOV 4 D0001

X00407

MOV 3 D0001

********Delete device********

D0001 = 3

WRITE 0 D0007 400 50

100002

1st Edition : Feb.01,1999-00
**Register Device**

- Clear D2 register (N-th item).
- Specify data position of N-th item.
- Proceed to data registration if data is valid (if it does not begin with -1).
- Proceed to the next process after the last data is registered (if it does not begin with -1).
- Clear tool command buffer.
- Write registration data.
- Tool command $3
- Execute tool command.
- Proceed to next data on normal exit.
- Proceed to error processing on error.
- Clear tool command buffer.
- Tool command $71
- Execute tool command.
- Proceed to next process on normal exit.
- Proceed to error processing on error.

**Write ROM**

- Proceed to the next process if node number is 64.
********Address 2********

Node address

Vendor ID

Device type

Produce code

Communication type

Size of output data

Offset of output data

Size of input data

Offset of input data

Store -1 at the end of data

********End of data********
Sample BASIC Program

An example of a BASIC program is given here.

- This module is installed in slot 4.
- The contents to be registered in the scan list are set using a DATA statement.
- -1 is set at the end of the data.
- "END OK" is displayed on normal exit.
- If an error occurs, "ERROR" and error code are displayed.

```
10 ! **********************************************
20 ! *              F3LD01-ON SAMPLE PROGRAM         *
30 ! *              Scan list registration         *
40 ! **********************************************
50 !
60 ! **** Initialisation ****
70 DEFINT A-Z
80 DIM SDATA (17)
90 LDSLOT=4
100 ASSIGN LD01=LDSLOT
110 !
120 CONTROL LDSLOT, 102; $0000, $FFFF           : ! All output relays OFF
130 CONTROL LDSLOT, 101; $0000, $FFFF
140 GOSUB TBCLEAR@                                : ! Clear Tool Command Buffer
150 !
160 !
200 ! **** Clear error****
210 CONTROL LDSLOT, 102; $0001, $FFFF           : ! Request to Clear Error (Y49) ON
220 STS = 0
230 WHILE BIT (STS, 0) = 0                       : ! Clear Error Confirmation (Wait for
                                                   X49 ON)
240 STATUS LDSLOT, 102; STS
250 END WHILE
260 CONTROL LDSLOT, 102; $0000, $0001           : ! Request to Clear Error (Y49) OFF
270 !
300 ! **** Set transmission time interval ****
310 MODE$="Set CycleTime"
320 NA = -1
330 GOSUB TBCLEAR@                                : ! Clear Tool Command Buffer
340 OUTPUT LDSLOT, 400 NOFORMAT; $72            : ! Command $72
350 OUTPUT LDSLOT, 403 NOFORMAT; 20             : ! Set transmission time interval to 20ms
360 GOSUB COMMAND@                                : ! Execute command
370 IF RES1<>$72 THEN GOTO ERROR@                : ! Proceed to error processing on error
380 !
400 ! ******** Delete Device *********
410 MODE$ = "Del Device"
420 STATUS LDSLOT, 101; STS                      : ! Read output relay
```
9-26  Configuration  

430 IF BIT (STI, 6) = 1 THEN GOTO SSET@ : ! Skip if scan list is not registered  
440 FOR NA = 0 TO 63  
450 GOSUB TBCLEAR@  
460 OUTPUT LDSLOT, 400 NOFORMAT; $05 : ! Command $05  
470 OUTPUT LDSLOT, 401 NOFORMAT; NA : ! Node address of device to be deleted  
480 GOSUB COMMAND@  
490 IF (RES1<>$5) AND (RES2<>$6) THEN GOTO ERROR@  
500 NEXT NA : ! Proceed to error processing if error occurs and next node if normal  
510 !  
600 ! **** Register Device ****  
610 SSET@  
620 MODE$="Add Device"  
630 LOOP@  
640 READ SDATA (0) : ! Confirm first data item (Node number)  
650 NA  =  SDATA (0)  
660 IF NA = -1 THEN GOTO SAVEROM@ : ! Proceed to Store to flash memory if (?! is stored at the end of data  
670 GOSUB TBCLEAR@ : ! Clear Tool Command Buffer  
680 OUTPUT LDSLOT, 400 NOFORMAT; $03 : ! Command $03  
690 OUTPUT LDSLOT, 401 NOFORMAT; NA : ! Write node address  
700 FOR N = 1 TO 16  
710 READ SDATA (N) : ! Read and write data one by one  
720 GOSUB COMMAND@ : ! Execute command  
730 IF RES1 <> $03 THEN GOTO ERROR@ : ! Proceed to an error processing if an error occurs  
740 GOTO LOOP@ : ! Proceed to registration of next data  
770 !  
800 ! **** Store in Flash memory ****  
810 SAVEROM@  
820 MODE$="Rom Write"  
830 NA = -1  
840 OUTPUT LDSLOT, 400 NOFORMAT; $71 : ! Command $71  
850 GOSUB COMMAND@ : ! Execute command  
860 IF RES1 <> $71 THEN GOTO ERROR@ : ! Proceed to error processing if an error occurs  
870 PRINT "END OK" : ! Normal exit  
880 STOP  
890 !  
900 ! **** Subroutine - Clear tool command buffer****  
910 TBCLEAR@  
920 FOR I   =  0  TO  49  
930 OUTPUT LDSLOT, 400+I NOFORMAT; 0 : ! Clear Tool Command Buffer  
940 NEXT I  
950 I = 0
960 RETURN
970 !
1000 ! **** Subroutine - Execute command****
1010 COMMAND@
1020 CONTROL LDSLOT, 102; $0200, $FFFF : ! Request to Execute Command
         relay(Y58) ON
1030 STS = 0
1040 WHILE BIT (STS, 9) = 0 : ! Clear Error Confirmation (Wait
         for Clear Error Confirmation
         (X26) ON)
1050 STATUS LDSLOT, 102; STS
1060  END WHILE
1070 ENTER LDSLOT, 400 NOFORMAT; RES1 : ! Read response
1080 ENTER LDSLOT, 401 NOFORMAT; RES2
1090 CONTROL LDSLOT, 102; $0000, $FFFF : ! Request to Execute Command
         Relay (Y58) OFF
1100 RETURN
1110 !
1200 ! **** Error processing****
1210 ERROR@
1220 PRINT "ERROR" : ! Display error
1230 PRINT MODE$ : ! Display Error operation
1240 PRINT "RES1=", HEX$ (RES1) : ! Display response1 (Hexadecimal)
1250 PRINT "RES2=", HEX$ (RES2) : ! Display response2 (Hexadecimal)
1260 PRINT "NA=", NA : ! Display node address where an
         error has occurred (decimal)
1270 STOP
1280 !
1300 ! **** Registration data****
1310 ! * Position 401, 402, 403, 404, 405, 406, 407, 408, 409, 410, 411, 412, 413,
         414, 415, 416, 417,
1320 DATA 1, 47, 0, 100, 0, 0, 0, 4, 0, 0, 0, 0, 0, 1, 27,
1330 DATA 2, 47, 0, 101, 0, 0, 3, 0, 0, 1, 49, 0, 0,
1340 DATA -1, : ! End of data (write -1 at the
         end)
1350 END
10. Errors and Troubleshooting

Troubleshooting of this module is shown in the following flowcharts.

Error

Is RDY LED on?

YES

Sequence CPU error?

YES

Remedy the cause of the sequence CPU error

NO

Status of NS LED?

Off

NO Go to 9.3 flowchart when NS LED is not on

NO

Red light on

Go to 9.4 flowchart when red NS LED is on

Red light flashing

Is it after Request to start scanning?

YES Green light flashing

NO Green light on

Normal

Go to 9.1 Flowchart when RDY LED is not on

Go to 9.2 flowchart when MS LED is blinking

NO
10.1 Flowchart When RDY LED Is Not ON

RDY LED is not on

- Is the required voltage applied to the power module? [NO → Apply the correct voltage, YES → ]

- Are the other modules RDY? [NO → Replace the power module or the base module, YES → ]

- Is module properly installed in the slot? [NO → Fitting is correct when you hear a click, YES → ]

- Is RDY LED still not on when installed in other slots [NO → Replace the base module, YES → Replace the module]
10.2 Flowchart When MS LED Is Flashing

1. **MS LED is flashing**
   - **Is the setting of node address setting switch correct?**
     - **YES**
     - **NO**
       - Correct the setting of the switch (See 5.2.1, “Node Number Set Switch”)
   - **YES**
   - **NO**
     - Correct the setting of the switch (See 5.2.2, “Action Mode Setting Switch”)
  - Replace the module
10.3 Flowchart When NS LED Is Not ON

NS LED is not on

Is the power supply for communication correct?

YES

Is the network power supply correctly set?

YES

Is the network power supply working properly?

YES

Correct the wiring

NO

Reset the wiring and power supply of network

NO

Restart power to network or replace the network power supply

NO

Replace the module

YES

Is the network power supply correctly set?

YES

Correct the wiring

NO

Reset the wiring and power supply of network

NO

Restart power to network or replace the network power supply

YES

Replace the module

Error Code = $E112
10.4 Flowchart When Red NS LED Is ON

Red NS LED is on

Yes

Is the node address duplicated?

Yes

Correct the node address

No

Are the data rates of all nodes the same?

Yes

Make the transmission speeds of all nodes the same

No

Is the trunk/brunchline cable length correct?

Yes

Correct the cable length

No

Is there a cable break?

Yes

Change the cable

No

Is the wiring proper?

Yes

Correct the wiring

No

Are the terminating resistors set only at the two terminals of the trunk line?

Yes

Set the terminating resistors correctly

No

Is it noisy?

Yes

Improve the network environment

No

Replace the module
10.5 Flowchart When Red NS LED Is Flashing

1. Red NS LED is flashing and 'NS' LED is not on after request to start scanning.
2. Is the scan list registered? 
   - NO: Register scan list
   - YES: 
     3. Is the information registered in the scan list correct? 
        - NO: Correct the scan list
        - YES: 
          4. Are the data rates of all nodes the same? 
             - NO: Make the data rates of all nodes the same
             - YES: 
               5. Is the trunk/brunch line cable length correct? 
                  - NO: Correct the cable length
                  - YES: 
                    6. Is there a cable break? 
                       - NO: Change the cable
                       - YES: 
                         7. Is the cabling correct? 
                            - NO: Correct the cabling
                            - YES: 
                              8. Is the terminal resistance set only at the two terminals of the trunk line? 
                                 - NO: Set the terminating resistors correctly
                                 - YES: 
                                   9. Is there noise? 
                                      - NO: Improve the network environment
                                      - YES: Change the module

10.6 Error Codes

This module conveys error status not only by LED but also by input relay (Xnnn01 to 16), error codes and other node error information.

The error codes within the module information of the input/output data register and the node error information is of latched type. The latest error information is maintained until a Request to Clear Error is executed.

The error codes are given in Table 10.1.

Table 10.1 Error Codes

<table>
<thead>
<tr>
<th>Item</th>
<th>EC1</th>
<th>EC2</th>
<th>Relay</th>
<th>OT</th>
<th>MS-LED</th>
<th>NS-LED</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Internal error</td>
<td>$1005</td>
<td>–</td>
<td>1</td>
<td>No</td>
<td>Red</td>
<td>Off</td>
<td>Internal discrepancy in the basic program</td>
</tr>
<tr>
<td>Memory error</td>
<td>$1203</td>
<td>–</td>
<td>1</td>
<td>No</td>
<td>Red</td>
<td>Off</td>
<td>RAM error</td>
</tr>
<tr>
<td>Flash memory error 1</td>
<td>$1281</td>
<td>–</td>
<td>1</td>
<td>No</td>
<td>Red</td>
<td>Off</td>
<td>Error in the part of flash memory which stores the program</td>
</tr>
<tr>
<td>Flash memory error 2</td>
<td>$1282</td>
<td>–</td>
<td>1</td>
<td>No</td>
<td>Red</td>
<td>Off</td>
<td>Error in the part of the flash memory which stores DeviceNet information</td>
</tr>
<tr>
<td>Flash memory error 3</td>
<td>$1283</td>
<td>–</td>
<td>1</td>
<td>No</td>
<td>Red</td>
<td>Off</td>
<td>Failure in storing scan list and transmission time interval in flash memory</td>
</tr>
<tr>
<td>Watchdog timer error</td>
<td>$1285</td>
<td>–</td>
<td>1</td>
<td>No</td>
<td>Red</td>
<td>Off</td>
<td>Watchdog timer error</td>
</tr>
<tr>
<td>CAN error 1</td>
<td>$1286</td>
<td>–</td>
<td>1</td>
<td>No</td>
<td>Red</td>
<td>Off</td>
<td>CAN controller error (cannot cancel reset)</td>
</tr>
<tr>
<td>CAN error 2</td>
<td>$1287</td>
<td>–</td>
<td>1</td>
<td>No</td>
<td>Red</td>
<td>Off</td>
<td>CAN controller error (data bus error)</td>
</tr>
<tr>
<td>CAN error 3</td>
<td>$1288</td>
<td>–</td>
<td>1</td>
<td>No</td>
<td>Red</td>
<td>Off</td>
<td>CAN controller error (address bus error)</td>
</tr>
<tr>
<td>System failure</td>
<td>$1289</td>
<td>–</td>
<td>8</td>
<td>No</td>
<td>–</td>
<td>–</td>
<td>System failure occurred</td>
</tr>
<tr>
<td>Node Busy</td>
<td>$1305</td>
<td>Node no.</td>
<td>4</td>
<td>Yes</td>
<td>Green</td>
<td>Red flashing</td>
<td>Communications not possible because of communicating with another master</td>
</tr>
<tr>
<td>Switch setting invalid</td>
<td>$E111</td>
<td>Data position no.</td>
<td>1</td>
<td>No</td>
<td>Red flashing</td>
<td>Off</td>
<td>Switch setting error (node address, data rate, etc)</td>
</tr>
<tr>
<td>Network power error</td>
<td>$E112</td>
<td>–</td>
<td>3</td>
<td>No</td>
<td>Green</td>
<td>Off</td>
<td>No power to network</td>
</tr>
<tr>
<td>Duplicate node address</td>
<td>$E121</td>
<td>Node no.</td>
<td>2</td>
<td>No</td>
<td>Green</td>
<td>Red</td>
<td>The current node has the same address as another node and as a result cannot join the network</td>
</tr>
<tr>
<td>Bus-off detected</td>
<td>$E122</td>
<td>–</td>
<td>2</td>
<td>No</td>
<td>Green</td>
<td>Red</td>
<td>Numerous data errors on the network</td>
</tr>
<tr>
<td>Node absent</td>
<td>$E141</td>
<td>Node no.</td>
<td>4</td>
<td>Yes</td>
<td>Green</td>
<td>Red flashing</td>
<td>No response from the registered node in the scan list</td>
</tr>
<tr>
<td>Node information mismatch</td>
<td>$E142</td>
<td>Node address</td>
<td>4</td>
<td>Yes</td>
<td>Green</td>
<td>Red flashing</td>
<td>The information read from the scan list and node differs.</td>
</tr>
<tr>
<td>Scan list not registered</td>
<td>$E143</td>
<td>–</td>
<td>7</td>
<td>No</td>
<td>–</td>
<td>–</td>
<td>There are no registered devices in the scan list. There is nothing stored in the flash memory.</td>
</tr>
</tbody>
</table>

Relay: The input relay no. (** in Xnnn**) that is on in the status.

OT: Reflected in the error code storage information of other nodes (input/output data registers 29 to 32)
SEE ALSO

1) See also Section 8.2.2, “Exit Codes,” in this manual for the exit codes of message communications.
2) See also Section 9.1.5, “Exit Codes,” in this manual for the exit codes of tool commands.
11. DeviceNet Glossary

■ Bus-off:

Indicates that the error rate in the network is very high.

■ CAN:

CAN is short for Controller Area Network. It is a communications protocol developed as a LAN for use in an automobile. DeviceNet employs CAN technology.

■ I/O communications:

This is a communications function which allows the exchange of control information (I/O data) between masters and slaves in real time. This module supports the following two types of I/O communications protocols of DeviceNet.

Polling:
Sends a polling instruction containing output data from the master to a specified slave; the master then receives a response containing the input data from the slave (1 : 1 communication).

Bit Strobe:
Multicasts (broadcasts) a bit strobe request instruction from the master to multiple slave systems and receives a response containing input data from each slave. Improves network throughput as a request can be communicated to multiple slave systems simultaneously (1: N communication). However, this is possible only in the case of slave systems with input data of less than 8 bytes.

■ ODVA:

ODVA is short for Open DeviceNet Vendor Association, Inc. It is a non-profit organization formed by machine vendors with the aim to administer and popularize the DeviceNet specification.

■ Connection:

This is a logical communication channel for facilitating communications between nodes. Maintains and manages communications between nodes.

■ Device Profile:

Standardizes the configuration and behavior (the smallest data configuration and operation) of devices of the same type (equipment, etc.). Provides mutual exchangeability between devices of the same type. Also known as a device model.

■ Node:

Devices (equipment) linked in DeviceNet are called nodes.
Master/Slave:
A node can be either a master which collects and distributes data or a slave which outputs and inputs data according to the instructions received from the master.

Message communications:
These are functions for parameter setting/reading, operation control/management, information exchange, etc. between nodes (1:1) as and when necessary. It is also called an explicit peer-to-peer message.

In this module, it is possible to request service from other nodes (master and slaves) using the explicit message defined in DeviceNet.
Appendix 1. Details on Network Configuration

The network configuration of a DeviceNet is given below. For configuration components, refer to Section 2.1, “Network Configuration.”

![Network Media Topology](FAP0101.EPS)

Appendix 1.1 Cable Length

- **Cable to be used exclusively for DeviceNet**

  Two types of cables are exclusively used for DeviceNet; thick cable and thin cables. Thin cables are softer as compared to thick cables. When using these cables, take note that the network distance differs for the thick and thin cables.

  **CAUTION**

  Use the cables that are exclusively recommended for DeviceNet; if other cables are used, they may cause a communications error.
Maximum cable length

The maximum allowable node distance is called the maximum cable length. Maximum cable lengths according to the type of cable used are given in Table 1.1.

Table 1.1 Maximum Cable Length

<table>
<thead>
<tr>
<th>Data rate</th>
<th>Thick cable</th>
<th>Thin cable</th>
</tr>
</thead>
<tbody>
<tr>
<td>125 kbps</td>
<td>500 m</td>
<td></td>
</tr>
<tr>
<td>250 kbps</td>
<td>250 m</td>
<td>100 m</td>
</tr>
<tr>
<td>500 kbps</td>
<td>125 m</td>
<td></td>
</tr>
</tbody>
</table>

For instances where both thick and thin cables are used, see Figure 1.2 for maximum cable length.

![Figure 1.2 Maximum Cable Length](TA0101.EPS)

Length of thick cable used (m)

\[
L (\text{thick}) + 5 \times L (\text{thin}) = 500 \\
L (\text{thick}) + 2.5 \times L (\text{thin}) = 250 \\
L (\text{thick}) + L (\text{thin}) = 100 \\
\]

at 125 kbps

at 250 kbps

at 500 kbps

L (thick) and L (thin) are the length of thick and thin cables respectively.

Maximum length of drop line

The maximum length of individual drop line is 6 m. However the total length of the drop line generally changes with the data rate.

![Figure 1.3 Maximum Drop Line Length](FAP0102.EPS)

Trunk line

Node

Node
Appendix 1.2 Connection Methods

There are two types of connection methods for DeviceNet; T-branching (using T-branching tap) and daisy chain method. The two types of methods can be mixed for trunk line/ drop line.

■ T-branching method

T-branching is achieved by using a branching tap that is exclusively used for DeviceNet. T-branching is possible from both trunk lines and drop lines.

![T-branching Diagram](FAP0104.EPS)

**Figure 1.4 T-branching**

**CAUTION**

Use the T-branching tap that is exclusively recommended for DeviceNet. If other T-branching tap is used, it may cause a communications error.


Daisy Chain method

Both trunk lines and drop lines can be connected using the daisy chain method.

![Figure 1.5 Daisy Chain](image)

---

**CAUTION**

The DeviceNet connector provided cannot be used for Daisy Chain type connection using thick cables as the cables are too thick. In this case, the connection can be made using the COMBICON plug mentioned below as the connector for multi-drop wiring.

However, as the connectors protrude from both sides of the module, pay special attention to the two adjacent modules. For details on the equipment, contact the manufacturer.

Model No. : TMSTBP 2.5/5-STF-5.08AU

Manufactured by : Phoenix Contact Co., Ltd.
Appendix 1.3 Terminating Resistor

Installing terminating resistors (121Ω) at both ends of a trunk line reduces reflection and stabilizes communications.

The terminating resistor should satisfy the following specifications.

- 121Ω
- 1% of metallic film resistance
- 1/4 W

Terminating resistors are in-built in this module. When connecting to the terminals of a trunk line using the daisy chain method, turn on the terminating resistor setting switch and use the built-in terminating resistors. In situations other than this, make sure that the switch is off and the terminating resistors are not in use.
Appendix 1.4 Configuration of Network Power Supply

Two lines out of the five special cables are 24 V DC power supply lines used for communication. It is necessary to provide power supply to the connectors of all nodes through these cables.

When using only one power supply to a single network, you can connect it directly to the trunk line. However, when using multiple network power supply, use the special power supply tap.

## Power Supply Specifications

Power supply specifications of DeviceNet is shown in the following table. Use the power supply that meets these specifications.

<table>
<thead>
<tr>
<th>Specifications</th>
<th>Parameter</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initial tolerance</td>
<td>24 V ± 1% or adjustable to 0.2%</td>
</tr>
<tr>
<td>Line regulation</td>
<td>0.3% max.</td>
</tr>
<tr>
<td>Load regulation</td>
<td>0.3% max.</td>
</tr>
<tr>
<td>Temperature coefficient</td>
<td>0.03% per deg. C max.</td>
</tr>
<tr>
<td>Input voltage range</td>
<td>120 V ± 10% 230 V ± 10% (if necessary) or automatic switching in range 95 to 250V</td>
</tr>
<tr>
<td>Input frequency range</td>
<td>48 to 62 Hz minimum.</td>
</tr>
<tr>
<td>Output ripple</td>
<td>250 mVp-p</td>
</tr>
<tr>
<td>Load capacitance</td>
<td>7000 µF max.</td>
</tr>
<tr>
<td>Temperature limits</td>
<td>When operating : 0 to 60°C*  When not operating: -40 to 85°C  * Operating at 60°C will lower the rated output.</td>
</tr>
<tr>
<td>Inrush current limit</td>
<td>Less than 65 amps peak</td>
</tr>
<tr>
<td>Overvoltage protection</td>
<td>Available (No set value)</td>
</tr>
<tr>
<td>Overcurrent protection</td>
<td>Available (Current Limit : 125% max.)</td>
</tr>
<tr>
<td>Turn-on time (with full load)</td>
<td>250 m sec max./ 5% of final value</td>
</tr>
<tr>
<td>Turn-on overshoot</td>
<td>0.2% max.</td>
</tr>
<tr>
<td>Stability</td>
<td>0 to 100% of load (for all conditions)</td>
</tr>
<tr>
<td>Insulation</td>
<td>Insulated from AC and chassis ground</td>
</tr>
<tr>
<td>Agency approvals</td>
<td>Required : UL  Optional : FCC Class B, CSA, TUV, VDE</td>
</tr>
<tr>
<td>Humidity</td>
<td>5 to 95% (non-condensing)</td>
</tr>
<tr>
<td>Output voltage</td>
<td>24V ± 1%</td>
</tr>
<tr>
<td>Output current</td>
<td>Up to 16A continuous</td>
</tr>
<tr>
<td>Surge current capacity</td>
<td>10% reverse capability</td>
</tr>
</tbody>
</table>
Special Purpose Power Supply Tap

Special purpose power supply taps are configured as shown in Figure 1.6. Always use it when there are more than two-power supply sources on a single network.

Network Power Supply Configuration

Network power supply configuration depends on electricity consumption of all nodes and by the maximum current capacity of the network cable. Please use the calculation in configuration 1 and 2 as reference when installing the power supply.

TIP

Maximum current capacity of cables in DeviceNet is as follows.

<table>
<thead>
<tr>
<th>Cable length (m)</th>
<th>0</th>
<th>25</th>
<th>50</th>
<th>100</th>
<th>150</th>
<th>200</th>
<th>250</th>
<th>300</th>
<th>350</th>
<th>400</th>
<th>450</th>
<th>500</th>
</tr>
</thead>
<tbody>
<tr>
<td>Max. current capacity (amp)</td>
<td>8.00</td>
<td>8.00</td>
<td>5.42</td>
<td>2.93</td>
<td>2.01</td>
<td>1.53</td>
<td>1.23</td>
<td>1.03</td>
<td>0.89</td>
<td>0.78</td>
<td>0.69</td>
<td>0.63</td>
</tr>
</tbody>
</table>

Maximum current capacity (amps)

![Graph showing maximum current available on thick cable DeviceNet power supply bus](FAP0107.EPS)
### Table 1.5 Maximum Current (amps) Available Based on Network Thin Cable Length

<table>
<thead>
<tr>
<th>Cable length (m)</th>
<th>0</th>
<th>10</th>
<th>20</th>
<th>30</th>
<th>40</th>
<th>50</th>
<th>60</th>
<th>70</th>
<th>80</th>
<th>90</th>
<th>100</th>
</tr>
</thead>
<tbody>
<tr>
<td>Max. current (amps)</td>
<td>3.00</td>
<td>3.00</td>
<td>3.00</td>
<td>2.06</td>
<td>1.57</td>
<td>1.26</td>
<td>1.06</td>
<td>0.91</td>
<td>0.80</td>
<td>0.71</td>
<td>0.64</td>
</tr>
</tbody>
</table>

Figure 1.8 Maximum Current Available on Thin Cable DeviceNet power Supply Bus

- **Configuration example 1**

![Network Power Supply Configuration (Example 1)](FAP0109.EPS)

Total length of network = 300 m  
Total current consumption = 0.1 A + 0.1 A + 0.1 A + 0.05 A + 0.1 A = 0.45 A  
Maximum current capability over the 300 m of a thick cable = 1.03 A  
The total current consumption is less than the maximum current capacity of cables, so this network configuration is possible.
Configuration example 2

Figure 1.10  Network Power Supply Configuration (Example 2)

Total length of network on right side = 150 m
Total current consumption on right side = 0.1 A + 0.05 A + 0.1 A = 0.25 A
Maximum current capability of 150 m of a thick cable = 2.01 A

Total length of network on left side = 100 m
Total current consumption on left side = 0.1 A + 0.1 A = 0.2 A
Maximum current capability of 100 m of a thick cable = 2.93 A

In both the right and left networks, the total electricity consumption is less than the max. current capability of cables, so this network configuration is possible.
Appendix 1.5  Network Grounding

DeviceNet uses one-point grounding. As far as possible, ground it in the vicinity of the center of the network through the SHIELD terminal of the power supply tap.

If the tap is not used, ground the GND cable directly.

Retain the grounding resistance up to 100 Ω (JIS Class 3 grounding).
Appendix 2. Information on Third-party Products

- Obtaining Information on Third-party Products

DeviceNet is a multi-vendor network, and devices conforming to DeviceNet specifications are supplied by many vendors.

A list of devices that conform to the DeviceNet specifications and the relevant contacts are given in the “Product Catalog (Catalog of devices conforming to DeviceNet)” published by ODVA and it can be obtained from ODVA or ODVA Japan. It is also described in the homepages of ODVA and ODVA Japan.

URL of ODVA: http://www.odva.org/
URL of ODVA Japan: http://www.odva.astem.or.jp

TIP

The homepage address may be changed without prior notice. Please verify the address.

- Cable/T-branching Tap/Power Supply Tap/Terminating Resistor

The items minimally required in a network configuration are cables, T-branching tap, power supply tap and terminating resistor. One third-party supplier is given here for each item. For more details, make inquiries to the respective manufacturers provided by ODVA.

<table>
<thead>
<tr>
<th>Type</th>
<th>Manufacturer</th>
<th>Model name</th>
<th>Specification/Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Thick cable</td>
<td>Showa Electric wire &amp;</td>
<td>TDN18-100G</td>
<td>Length: 100 m, Color: Light gray</td>
</tr>
<tr>
<td></td>
<td>Cable Co., Ltd.</td>
<td>TDN18-300G</td>
<td>Length: 300 m, Color: Light gray</td>
</tr>
<tr>
<td></td>
<td></td>
<td>TDN18-500G</td>
<td>Length: 500 m, Color: Light gray</td>
</tr>
<tr>
<td>Thin cable</td>
<td></td>
<td>TDN24-100G</td>
<td>Length: 100 m, Color: Light gray</td>
</tr>
<tr>
<td></td>
<td></td>
<td>TDN24-300G</td>
<td>Length: 300 m, Color: Light gray</td>
</tr>
<tr>
<td></td>
<td></td>
<td>TDN24-500G</td>
<td>Length: 500 m, Color: Light gray</td>
</tr>
<tr>
<td>T-Branching tap</td>
<td>Omron Corporation</td>
<td>DCN1-1C</td>
<td>With three connectors for connection (You can mount one terminating resistor with one drop line.)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DCN1-3C</td>
<td>With five connectors for connection (You can mount one terminating resistor with three drop lines.)</td>
</tr>
<tr>
<td>Terminating resistor</td>
<td>Omron Corporation</td>
<td>DRS1-T</td>
<td>Terminal block type terminating resistor</td>
</tr>
</tbody>
</table>

TIP

The model name may be changed without prior notice. Please verify the model names.
## Revision History

<table>
<thead>
<tr>
<th>Edition</th>
<th>Date</th>
<th>Revised Item</th>
</tr>
</thead>
<tbody>
<tr>
<td>1st</td>
<td>Feb. 1999</td>
<td>New publication</td>
</tr>
</tbody>
</table>

Written by    Product Marketing Section, PLC Center  Industrial Automation Business Head Quarter.  Yokogawa Electric Corporation  
Published by  Yokogawa Electric Corporation  2-9-32 Nakacho, Musashino-shi, Tokyo 180-8750, JAPAN  
Printed by    Yokogawa Graphic Arts Co., Ltd.