

HITACHI

User's Manual

Option

FL.NET

(LQE702-E)

S10VE

SEE-1-104 (A)

User's Manual

Option

FL.NET

(LQE702-E)

S10VE

If you export this product, please check all restrictions (for example, Japan's Foreign Exchange and Foreign Trade Law, and USA export control laws and regulations), and carry out all required procedures. If you require more information or clarification, please contact your Hitachi sales representative.

Edition 1 (October 2019): SEE-1-104 (A)

- No part of this material may be reproduced in any form or by any means without permission in writing from the publisher.
- Information in this document is subject to change without notice.

For inquiries about this product, please visit the following URL:

<https://www.hitachi.com/s10/>

All Rights Reserved, Copyright © 2019, Hitachi, Ltd.

Safety Precautions

- Before installation, operation, maintenance, and inspection of this product, you must carefully read through this manual and other related manuals. When using the product, make sure that you are familiar with all the information concerning this product, safety information, and precautions provided in those manuals.
- Keep this manual in a readily accessible place for future reference when using the product.
- Safety precautions in this manual are classified into four levels according to the severity of potential hazards: DANGER, WARNING, CAUTION, and Notice.

Definitions of the safety labels



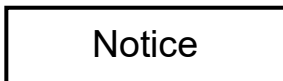
: Indicates a hazardous situation which, if not avoided, will result in death or serious injury.




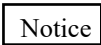
: Indicates a hazardous situation which, if not avoided, could result in death or serious injury.



: Indicates a hazardous situation which, if not avoided, could result in minor or moderate injury.



: Indicates a danger (resulting from incorrect use of the product) that can cause property damage or loss not related to personal injury if the safety precautions are not observed.

Failure to observe precautions marked with  or  could also lead to a serious consequence depending on the situation in which the product is used. Therefore, you must observe all of those precautions without fail.

The following are definitions of *serious injury*, *minor or moderate injury*, and *property damage or loss not related to personal injury* used in the safety labels.

Serious injury:

Is an injury that has aftereffects and requires hospitalization for medical treatment or long-term follow-up care.

Examples of serious injuries are as follows: vision loss, burns (caused by dry heat), low-temperature burns, electric-shock injuries, broken bones, and poisoning.

Minor or moderate injury:

Is an injury that does not require either hospitalization for medical treatment or long-term follow-up care.

Examples of minor or moderate injuries include burns and electric-shock injuries.

Property damage or loss not related to personal injury:

Is damage other than personal injury. Examples of property damage or loss not related to personal injury are as follows: damage or loss of personal property, failure or damage of the main unit of the product, and loss of data.


The safety precautions stated in this manual are based on the general rules of safety applicable to this product. These safety precautions are a necessary complement to the various safety measures included in this product. Although they have been considered carefully, the safety precautions posted on this product and in the manual do not cover every possible hazard. Common sense and caution must be used when operating this product. For safe operation and maintenance of this product, establish your own safety rules and regulations according to your specific needs. A variety of industry standards are available to aid in establishing such safety rules and regulations.

1. General Safety Guidelines

Before installing, operating, inspecting, or conducting maintenance on this unit, read the following instructions carefully:

Before starting work

- Follow all the instructions and operating procedures provided in this manual and related manuals.
- Pay special attention to all the precautions on the machine and in this manual.
Failure to do so could cause personal injury or damage to the machine.
- Do not perform any operation or action that is not described in this manual. When in doubt, contact the relevant store or your sales staff.
- The hazard warnings on the machine and in the manual cannot cover every possible case, as it is impossible to predict and evaluate all circumstances.
You must be alert and use your common sense.
- Do not perform any installation, wiring, handling, or internal customization that is not described in this manual. Hitachi will not be responsible for any damage to Hitachi equipment or peripherals and personal injury resulting from such a practice.
- In case this product fails, configure an emergency stop circuit or interlock circuit outside the product. Failure of this product might damage the machine or cause an accident.
- Maintenance work must be done only by appropriately trained persons who have practical experience and understand the potential hazards (and how to avoid them) during operation.

 During work

- Follow the sequence of steps specific to each procedure.
- Use the relevant tools and instruments for each task as specified in the manual. If no particular tools are specified, use commercially available tools and instruments which fit the purpose.
- Make sure that all measurement instruments and powered tools have been properly calibrated or periodically inspected prior to use.
- Keep the maintenance area neat and tidy.
- Make sure that maintenance parts, materials, and removed parts are not placed in a passageway so they are not tripped on by personnel.
- Wear eye protection if there is a risk of flying debris.
- When using sharp objects or cutting tools, keep fingers and other parts of your body away from the path of the blade bit or point.
- After maintenance work is completed, before turning on power, make sure that all parts removed during maintenance have been installed back in their original positions in the machine. Also make sure that no tool or foreign material remains in the machine.



Prevention of electric shock

- Before starting work, make sure that there is no potential electrical hazard in the maintenance area.
Example: Insufficient grounding line or a wet floor
- Before starting work, check the location of the emergency power-off switches and how to operate them.
- Unless otherwise specifically instructed, isolate the machine from all power sources before starting maintenance. Turn off not only the power switches on the machine but also the switch on the distribution panel.
After turning off the switch on the power distribution panel, attach a notice saying *Do not turn on this switch* on the power distribution panel. If the power distribution panel is equipped with a lockout device, turn off the switch on the power distribution panel, lock the device, and then bring the key with you.
If you are resuming maintenance work performed by someone else, do not assume that the power is off. Make sure that the above-mentioned conditions, such as switches turned off, are satisfied. If necessary, use a measurement tool to ensure that the power is off.
- Some parts in the machine remain charged for a certain time even after the power supply to the machine is disconnected. (Follow the displayed instructions).
- When working on a machine which has a grounding terminal, make sure that the terminal is properly connected to the facility's ground.
- When working near an exposed live electric circuit, do not work alone. Work with another person who can immediately turn off the power.
- To prevent electric shock during work, do not wear any metallic items or an accessory such as a wristwatch with a metallic surface. If you wear eyeglasses with a metallic frame, take care not to let the frame touch an uninsulated surface.
- Make sure that your hands and arms are dry.
- Use only one hand when it is necessary to work near an exposed live electric circuit. This mitigates the risk of current passing through your heart if you accidentally touch the circuit.
- Do not use a dental mirror near an exposed live electric circuit. The mirror surface is conductive and can become hazardous even if the mirror is made of plastic.
- Unless otherwise specifically instructed, do not supply power to any subassembly such as a power supply unit or a motor while it is removed from the machine.



Procedure in an emergency

In the case of electric shock

- Do not panic. Do not become another victim through contact with the injured person.
- First, shut off the electric current passing through the victim by using the emergency power-off switch. If there is no emergency power-off switch, use the normal power-off switch.
If this cannot be done, push the victim away from the source of the electric current by using a nonconductive object such as a dry wooden stick.
- Call an ambulance.
- If the victim is unconscious, artificial respiration may be necessary.
A proper method for performing artificial respiration or resuscitation should be learned beforehand.
If the victim's heart is not beating, cardio-pulmonary resuscitation must be performed by a trained and qualified person.

In the case of fire

- Call the fire department, and then take action to extinguish the fire.
- Shut off power to the machine by using the emergency power-off switch. If there is no emergency power-off switch, use the normal power-off switch.

2. Notes About Warning Indications

The following summarizes the warnings contained in this manual and their locations.

2.1 WARNING

WARNING

- Turn off the power switch of the power supply module before replacing any module. This is to avoid the risk of electrical shock, and to prevent damage and malfunction.

(See page 6-1.)

2.2 Notice

Notice

- To avoid damaging the product, protect it with cushioning, and then store it in the environment shown in Table 2-1.
- To avoid damaging the product during transit, protect it with cushioning and handle it as a precision machine.

(See page 2-1.)

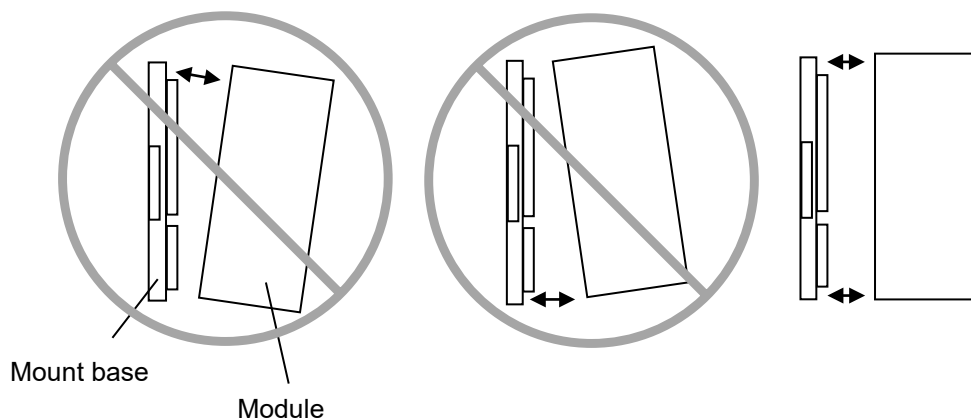
- Switch off the power to the power supply module before setting the MAIN/SUB setting switch. Setting the MAIN/SUB setting switch while the power is switched on might result in a malfunction.

(See pages 2-7 and 3-3.)

- Poor contact might cause malfunction. Mount the module and connect cabling to the module immediately after the module is unpacked so that dust or other foreign substances do not accumulate on connectors.
- To avoid damaging a module when removing or mounting it, turn off the power switch of the power supply module beforehand.
- To avoid damaging a module, observe the following precautions when mounting or removing the module:
 - Before mounting the module on the mount base connector, check that the connector pins are properly aligned and not bent, broken, or soiled.
 - Connecting or disconnecting a module that is tilted might damage connector pins. When moving the module, hold it vertically parallel to the mount base as shown below.

Bad examples

Good example



(See page 3-2.)

Notice

- A loose connection might cause a malfunction. Do not use communication cables that have any broken cable lock pins. After connecting the connector, confirm that the connector is locked.

- Do not touch the connectors when the system is powered on. This might cause a malfunction.

(See page 4-1.)

- As circuit load increases, so does the risk of communication failure. Because the FL.NET module (LQE702-E) only supports auto-negotiation settings, do not specify the 100-Mbps full-duplex setting in the port settings of the switching hub.

(See page 4-3.)

- If the CPU memory's area word count differs from the common memory size setting (link parameter area word count), the FL.NET module transfers data from the common memory. The amount of this data is determined by the number of words specified by the CPU memory area word count setting.
- If there is a node not specified in the **Other nodes reception setting FL.NET (Main)** tab, the FL.NET module cannot transfer the data received in its common memory to the CPU memory allocated in the CPU module.

(See page 5-38.)

- When you specify the area for the self-node status, the self-node status flag ((1) *Contents of the self-node status flag* in A.4 *Contents of each status flag*) is transferred to the CPU memory allocated as one word of data.
- When you specify the area for the FA link status, the FA link status flag of the self-node ((2) *Contents of the FA link status flag* in A.4 *Contents of each status flag*) is transferred to the lower byte of the allocated CPU memory.
- If you do not specify the area for the self-node status or the FA link status, the FL.NET module cannot transfer the self-node status flag or the FA link status flag to the allocated CPU memory.

(See page 5-40.)

Notice

- The range of register/extension memory addresses that can be specified to the higher layer status area and FA link status area is the same as for the CPU memory allocation area.
- When you specify the area for the higher layer status, the description of the higher layer status flag ((3) *Contents of the higher layer status flag* in A.4 *Contents of each status flag*) is transferred to the allocated CPU memory of each node as one word of data.
- When you specify the area for the FA link status, the description of the FA link status flag of each node ((2) *Contents of the FA link status flag* in A.4 *Contents of each status flag*) is transferred to the lower byte of the allocated CPU memory location.
- The specified higher layer status and FA link status area must not be a duplicate of the allocated CPU memory area for another node's reception setting, the allocated CPU memory area for the self-node, or the setting for any other area.
- For a node that is not specified in the **Other nodes reception setting FL.NET (Main)** tab, the FL.NET module cannot transfer that node's higher layer status or FA link status to the allocated CPU memory of other nodes.

(See page 5-42.)

- To install or uninstall FL.NET SYSTEM/S10VE, use the administrator account. Using the standard account might result in incorrect installation or uninstallation.
- Before installing FL.NET SYSTEM/S10VE, terminate all Windows® programs, including resident programs in the memory (such as virus monitoring software). Installing FL.NET SYSTEM/S10VE while any of these programs are running might cause an error. If an error occurs, uninstall the tool being installed, terminate all Windows® programs, and then reinstall the tool. For details about the uninstallation procedure, see 5.4.3 *Uninstallation*.
- Do not specify, as the FL.NET SYSTEM/S10VE installation folder, any of the following folders protected by User Account Control:
 - Program file folder (for example, C:¥Program Files)
 - System root folder (for example, C:¥Windows)
 - System drive root folder (for example, C:¥)
 - Program data folder (for example, C:¥ProgramData)

(See page 5-44.)

Notice

- FL.NET SYSTEM/S10VE cannot be installed on a per-user basis. To install FL.NET SYSTEM/S10VE successfully, you must first log on to the system with an administrator account.

FL.NET SYSTEM/S10VE might not be installed properly in any of the following cases: 1)

Administrator permissions are acquired by using User Account Control# from a standard user account,

2) The administrator account was created from a standard user account by using User Account Control.

In this case, log on with the administrator account that was first created on your computer, and then reinstall FL.NET SYSTEM/S10VE.

If you log on with a user account other than that used for installing FL.NET SYSTEM/S10VE, the installed program might not appear in the program menu. In this case, log off and log on again with the administrator account that was first created on your computer, uninstall the installed program, and then install the program again.

When you want to create a new account, log on with an administrator account without using User Account Control.

#: User Account Control is a Microsoft Windows feature that temporarily grants administrative rights to standard user accounts.

(See page 5-46.)

- Static electricity could cause damage to the module. Before handling the module, discharge any static electricity from your body.

(See page 6-1.)

Application Conditions

- When using the product explained in this manual (called *this product* hereafter), we ask you to provide a backup or fail-safe system outside the product so that any failure or problem in this product will not cause serious consequences.
- This product is a general-purpose product designed for general industrial applications. This product shall not be used in applications that require a high level of safety or responsibility or special quality assurance. Hitachi assumes no responsibility for any loss or damage arising from the use of this product in such applications. Examples of such applications are as follows:

Applications where a high level of safety is required

Example: Power station control system (nuclear power, thermal power, or hydro power), combustion facility, aviation or space facility, railway facility, lift facility, facility for recreation and amusement, medical facility, safety equipment, on-vehicle equipment, ship facility, traffic light system, and other facilities where death or bodily harm might result in the event of an emergency

Applications where a high level of responsibility is required

Example: Systems that supply gas, water, or electricity, systems requiring round-the-clock operation, system responsible for legal settlement or other purpose of handling rights and properties

Applications under severe conditions or environment

Example: Outdoor facility environment that meets any of the following conditions:
Environment that is chemically contaminated, subject to electromagnetic interference, or subject to constant vibration or shock

However, use of this produce in any of the applications described above can be approved by the decision of Hitachi if the purpose is specifically limited, the customer has responsibility for providing redundancy, or no special quality is required. For details, contact a Hitachi sales representative.

Warranty and Servicing

1. Warranty period and scope

Warranty period

The warranty period of this product shall be one year after the product has been delivered to the specified site. The warranty period of repaired products shall be six months from the date of repair. The warranty period for repaired products takes precedence over the warranty period prior to repair.

Scope

If the product malfunctions during the warranty period described above while using this product as instructed by this manual, the product shall be repaired free of charge.

Repair service

Return repair is supported, and requires the customer to send the malfunctioning product to a designated repair service.

- Fill in the required items in the *Hitachi Programmable Controller S10VE Repair Request Sheet* in *Appendix A* in the *S10VE User's Manual General Description* (manual number SEE-1-001), and then enclose it in the package with the product to be returned for repair.
- The customer must cover the costs for sending the product for repair to Hitachi.
- Hitachi will pay the transport cost for returning the repaired product to the customer.
- Repair is limited to replacement of malfunctioning parts.
- Work other than replacement of malfunctioning parts, such as investigation into the cause of failure, shall be charged even during the warranty period.

2. Exception of warranty obligation

Regardless of the warranty period, Hitachi bears no responsibility in any of the following cases.

The warranty mentioned here means the warranty for the individual product that is delivered. Therefore, we cannot be held responsible for any losses or lost profits that result from operation of this product or from malfunctions of this product. This warranty is valid only in Japan and is not transferable.

- The malfunction was caused by handling or use of the product in a manner not specified in the product specifications.
- The malfunction was caused by a unit other than that which was delivered.
- The malfunction was caused by modifications or repairs made by a vendor other than the vendor that delivered the unit.
- The malfunction was caused by a relay or other consumable which has passed the end of its service life.
- The malfunction was caused by a disaster, natural or otherwise, for which the vendor is not responsible.
- The malfunction was caused by reasons that could not be anticipated, based on conventional technological understanding, at the time of shipment from Hitachi.

3. Range of services

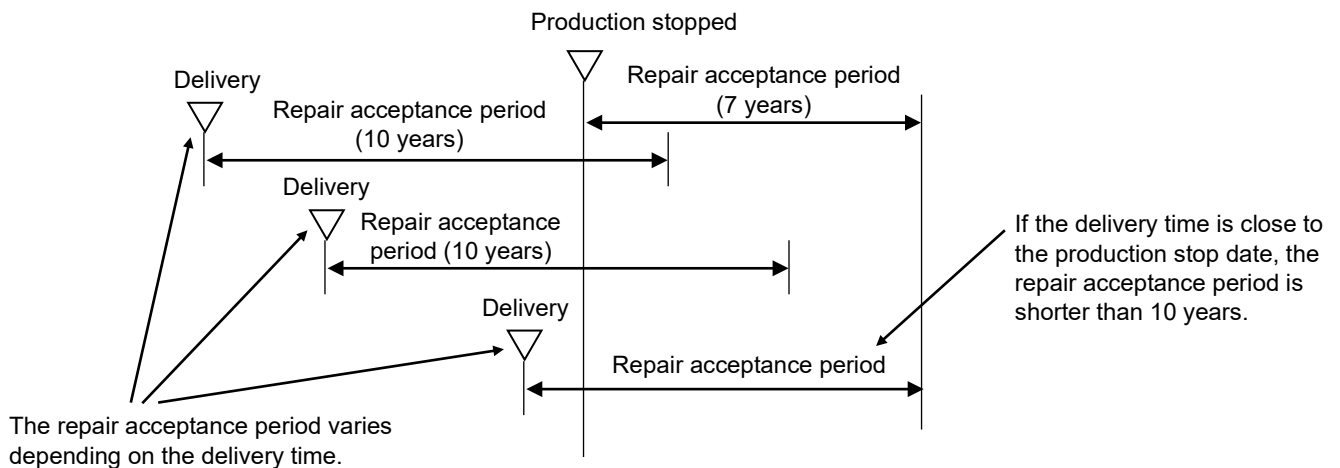
The price of the delivered product does not include on-site servicing fees by engineers. Extra fees will be charged for the following:

- Instruction for installation and adjustments, and witnessing trial operations
- Inspections, maintenance, and adjustments
- Technical instruction, technical training, and training facilities
- Examinations and repairs after the warranty period expires
- Examination of malfunctions caused by reasons outside the scope of the preceding warranty

4. Repair acceptance period

The repair acceptance period of S10VE products is 10 years after the product is delivered to the specified site or 7 years after production of the product is stopped, whichever comes earlier. The standard service life of S10VE is 10 years. We cannot accept repair of products whose designated repair acceptance periods (as described here) have been exceeded.

Payment for any repair after expiration of the warranty is the responsibility of the customer.



5. Service after production of the product is stopped

Products (including spare parts) cannot be supplied after production of those products is stopped.

6. Specification changes

Specifications described in this manual are subject to change without prior notice.

Revision History

Revision No.	History (revision details)	Issue date	Remarks
A	New edition	Oct. 2019	

Preface

Thank you for purchasing the FL.NET module, which is an option for use with the S10VE.

This manual *S10ve User's Manual FL.NET* describes how to use the FL.NET module. For proper use of the FL.NET module, thoroughly read this manual.

Notes on FL-net protocol versions

The S10VE FL.NET module (model: LQE702-E) uses FL-net protocol version 3.01.

This module cannot be connected to devices that use FL-net protocol version 1.00.

The following table shows which devices can be combined in a connection, according to their version of the FL-net protocol. Take care to reference this table when setting up a connection.

	Version 1.00 devices ^{#1}	Version 2.00 devices ^{#2}	Version 3.01 devices ^{#3}
Version 1.00 devices ^{#1}	Y	N	N
Version 2.00 devices ^{#2}	N	Y	Y
Version 3.01 devices ^{#3}	N	Y	Y

Y: Can be connected; N: Cannot be connected

#1: FL-net protocol version 1.00 devices (hereafter *version 1.00 devices*)

#2: FL-net protocol version 2.00 devices (hereafter *version 2.00 devices*)

#3: FL-net protocol version 3.01 devices (hereafter *version 3.01 devices*)

Version 1.00: FL-net protocol version of devices manufactured based on the standards in the *FA Control Network [FL-net(OPCN-2)] - Protocol Specification JEM 1479 (formulated in 2000)* issued by JEMA (Japan Electrical Manufacturers' Association)

Version 2.00: FL-net protocol version of devices manufactured based on the standards in the *FA Control Network [FL-net(OPCN-2)] - Protocol Specification JEM 1479 (formulated in 2002)* issued by JEMA (Japan Electrical Manufacturers' Association)

Version 3.01: FL-net protocol version of devices manufactured based on the standards in the *FA Control Network [FL-net(OPCN-2)] - Protocol Specification JEM 1479 (formulated in 2012)* issued by JEMA (Japan Electrical Manufacturers' Association)

Trademarks

- Microsoft® and Windows® are either a registered trademark or trademarks of Microsoft Corporation in the United States and/or other countries.
- Ethernet® is a product name of Xerox Corp.

Note about storage capacity calculations

- Memory capacities and requirements, file sizes and storage requirements must be calculated according to the formula 2^n . The following examples show the results of such calculations by 2^n (to the right of the equal signs).
 - 1 KB (kilobyte) = 1,024 bytes
 - 1 MB (megabyte) = 1,048,576 bytes
 - 1 GB (gigabyte) = 1,073,741,824 bytes
 - 1 TB (terabyte) = 1,099,511,627,776 bytes
- As for disk capacities, they must be calculated using the formula 10^n . Listed below are the results of calculating the above example capacities using 10^n in place of 2^n .
 - 1 KB (kilobyte) = 1,000 bytes
 - 1 MB (megabyte) = $1,000^2$ bytes
 - 1 GB (gigabyte) = $1,000^3$ bytes
 - 1 TB (terabyte) = $1,000^4$ bytes

Contents

1. Overview	1-1
1.1 What is FL-net?	1-1
1.2 Features of FL-net	1-2
1.3 System software specifications	1-4
1.3.1 System overview	1-4
1.3.2 Hardware and software requirements	1-4
2. FL.NET Module	2-1
2.1 System configuration	2-1
2.1.1 FL.NET module specifications	2-2
2.1.2 Link data specifications	2-3
2.1.3 Area for setting link parameters	2-6
2.1.4 Area for setting profile system parameters	2-6
2.2 FL.NET module component names and their functions	2-7
3. Mounting the FL.NET Module	3-1
3.1 Mounting base	3-1
3.2 Mounting the module	3-1
3.3 Setting the MAIN/SUB setting switch	3-3
3.4 External dimensions of the FL.NET module	3-4
4. Wiring the FL.NET Module	4-1
4.1 Connecting the communication cable	4-1
4.2 Network configuration components	4-2
4.3 Connection configuration using a hub	4-2
5. User's Guide	5-1
5.1 About FL-net	5-1
5.1.1 Overview of FL-net	5-1
5.1.2 Connection capacity and node numbers	5-3
5.1.3 Supported types of data communications	5-4
5.1.4 Amount of data that can be transmitted	5-5
5.1.5 The transfer cycle and its monitoring	5-6
5.1.6 Data area and memory	5-7
5.1.7 Communication management tables	5-8
5.1.8 Cyclic transmission and area	5-10

5.1.9 Message transmission	5-15
5.1.10 Using bit data	5-27
5.1.11 Communication time of the FL.NET module	5-28
5.2 Setting up the FL.NET module	5-29
5.2.1 Startup procedure	5-29
5.3 Using the FL.NET module	5-31
5.3.1 Setting link parameters	5-31
5.3.2 Allocating CPU memory	5-33
5.4 Installing and starting up the system	5-43
5.4.1 Notes on installation	5-43
5.4.2 Installation	5-45
5.4.3 Uninstallation	5-47
5.4.4 Reinstallation	5-47
5.4.5 Starting up the system	5-48
5.4.6 Closing the system	5-48
5.5 Commands	5-49
5.5.1 Structure of the FL.NET SYSTEM/S10VE functions	5-49
5.5.2 Specifying the self-node information settings	5-50
5.5.3 Specifying other nodes' reception settings	5-52
5.5.4 Registering parameters	5-54
5.5.5 Reading parameters	5-57
5.5.6 Deleting parameters	5-59
5.5.7 Saving parameters to a file	5-61
5.5.8 Reading parameters from a file	5-64
5.5.9 Printing	5-67
5.5.10 CSV output	5-70
5.5.11 Self-node information	5-73
5.5.12 Information on other participating nodes	5-75
5.5.13 Displaying data on other nodes	5-76
5.5.14 Network status	5-78
5.5.15 RAS information	5-80
6. Maintenance	6-1
6.1 Maintenance and inspection	6-1
6.2 Replacing or adding the module	6-2
6.3 Troubleshooting	6-2

7. Test and Maintenance Diagnosis	7-1
7.1 Details of Test and Maintenance diagnosis	7-1
7.2 Wiring method	7-2
7.3 Starting Test and Maintenance	7-4
7.4 Checking errors	7-6

Appendix

A.1 Network system definitions	A-1
A.1.1 Compliance with communication protocol standards	A-1
A.1.2 Layer structure of the communication protocol	A-1
A.1.3 FL-net physical layer	A-2
A.1.4 IP addresses	A-2
A.1.5 Subnet mask	A-3
A.1.6 TCP/IP and UDP/IP communication protocols	A-3
A.1.7 Port numbers	A-3
A.1.8 FL-net data format	A-4
A.1.9 FL-net transaction codes	A-6
A.1.10 Transaction code receive operation at a UDP port	A-8
A.2 FL-net network management	A-9
A.2.1 FL-net token management	A-9
A.2.2 Joining and leaving an FL-net network	A-12
A.2.3 Node status management	A-14
A.2.4 FL-net self-node management table	A-14
A.2.5 Participating node management table for an FL-net network	A-15
A.2.6 FL-net status management	A-16
A.2.7 FL-net message sequence number management	A-16
A.3 Profile of FL-net	A-17
A.3.1 Types of device communication information	A-17
A.3.2 Details of common parameters	A-18
A.3.3 Details of device-specific parameters (when used)	A-18
A.3.4 Example system parameters (for PLC)	A-19
A.3.5 Example system parameters (for CNC)	A-24
A.4 Contents of each status flag	A-27

List of Figures

Figure 1-1 Example configuration of an FA control network	1-1
Figure 1-2 Basic structure of the FL-net protocol	1-2
Figure 2-1 Example configuration for an FA control network	2-1
Figure 3-1 Mounting the optional module	3-1
Figure 3-2 Setting the MAIN/SUB setting switch	3-3
Figure 3-3 External dimensions of the FL.NET module	3-4
Figure 4-1 Wiring by using a 100BASE-TX communication cable	4-1
Figure 4-2 Connection configuration using a hub	4-2
Figure 4-3 Example configuration using a 100-Mbps hub (class 1)	4-2
Figure 4-4 Example configuration using a 100-Mbps hub (class 2)	4-2
Figure 4-5 Example configuration using a 100-Mbps switching hub	4-3
Figure 5-1 Concepts behind FL-net	5-1
Figure 5-2 FA link protocol	5-2
Figure 5-3 FL-net IP addresses	5-3
Figure 5-4 FL-net connection capacity and node numbers	5-3
Figure 5-5 Types of data communications supported by FL-net	5-4
Figure 5-6 Example of common memory and cyclic transmission	5-4
Figure 5-7 Example of message transmission	5-5
Figure 5-8 Amount of data that can be transmitted with cyclic transmission	5-5
Figure 5-9 Amount of data that can be transmitted with message transmission	5-6
Figure 5-10 Example of a refresh cycle time limit	5-6
Figure 5-11 Data area and memory	5-7
Figure 5-12 Token rotation and cyclic transmission 1	5-11
Figure 5-13 Token rotation and cyclic transmission 2	5-11
Figure 5-14 Common memory area for cyclic transmission: Example 1	5-12
Figure 5-15 Common memory area for cyclic transmission: Example 2	5-13
Figure 5-16 Common memory areas 1 and 2	5-13
Figure 5-17 Data synchronicity assurance	5-14
Figure 5-18 Overview of message transmission	5-15
Figure 5-19 Byte block read	5-17
Figure 5-20 Byte block write	5-18
Figure 5-21 Word block read	5-19
Figure 5-22 Word block write	5-20
Figure 5-23 Network parameter read	5-21
Figure 5-24 Network parameter write	5-22
Figure 5-25 Run/stop directive	5-23

Figure 5-26 Profile read	5-24
Figure 5-27 Communication log data read	5-25
Figure 5-28 Communication log data clear	5-25
Figure 5-29 Message return function	5-26
Figure 5-30 Transparent message transmission	5-26
Figure 5-31 Bit conversion of area 1 data	5-27
Figure 5-32 Startup procedure	5-29
Figure 5-33 [Node setting] tab	5-30
Figure 5-34 Example of FL-net module operation	5-32
Figure 5-35 Overview of common memory data transfer with the FL.NET module	5-33
Figure 5-36 [Node setting FL.NET (Main)] tab	5-35
Figure 5-37 [Other nodes reception setting FL.NET (Main)] tab	5-37
Figure 5-38 [Node setting FL.NET (Main)] tab	5-39
Figure 5-39 [Other nodes reception setting FL.NET (Main)] tab	5-41
Figure 5-40 User Account Control message	5-45
Figure 5-41 SETUP window	5-45
Figure 5-42 Window for choosing whether to install FL.NET SYSTEM/S10VE	5-46
Figure 5-43 Window for choosing whether to uninstall FL.NET SYSTEM/S10VE	5-47
Figure 5-44 FL.NET module selection window	5-48
Figure 5-45 Structure of the FL.NET SYSTEM/S10VE functions	5-49
Figure 5-46 FL.NET module selection window	5-50
Figure 5-47 [Node setting FL.NET (Main)] tab (default display)	5-50
Figure 5-48 Example of editing in the [Node setting FL.NET (Main)] tab	5-51
Figure 5-49 [Other nodes reception setting FL.NET (Main)] tab (default display)	5-52
Figure 5-50 [Other nodes reception setting FL.NET (Main)] tab (when editing Node 2)	5-53
Figure 5-51 Message confirming whether to discard changes to node settings	5-53
Figure 5-52 Clicking [PCs write] in the [Node setting FL.NET (Main)] tab	5-54
Figure 5-53 Option module parameter setup list window (registering settings)	5-55
Figure 5-54 Confirmation message for overwriting parameters (when overwriting the parameters of the OD.RING (main) module)	5-55
Figure 5-55 Reset confirmation message	5-56
Figure 5-56 Clicking the [PCs read] button in the [Node setting FL.NET (Main)] tab	5-57
Figure 5-57 Message confirming whether to discard changes	5-58
Figure 5-58 Receiving agency selection window	5-58
Figure 5-59 Clicking the [Delete] button in the [Node setting FL.NET (Main)] tab	5-59
Figure 5-60 Option module parameter setup list window (deleting parameters)	5-59
Figure 5-61 Message confirming whether to delete parameters	5-60
Figure 5-62 Reset confirmation message	5-60

Figure 5-63 Clicking the [File save] button in the [Node setting FL.NET (Main)] tab	5-61
Figure 5-64 Save As window	5-62
Figure 5-65 File save window	5-62
Figure 5-66 Clicking the [File open] button in the [Node setting FL.NET (Main)] tab	5-64
Figure 5-67 Message confirming whether to discard changes	5-64
Figure 5-68 Open window	5-65
Figure 5-69 Message indicating that the selected file is incorrect	5-65
Figure 5-70 File open window	5-66
Figure 5-71 Clicking the [Print] button in the [Node setting FL.NET (Main)] tab	5-67
Figure 5-72 Print dialog box	5-68
Figure 5-73 Sample printout	5-69
Figure 5-74 Clicking the [CSV Output] button in the [Node setting FL.NET (Main)] tab	5-70
Figure 5-75 Save As window	5-71
Figure 5-76 Sample CSV file output	5-72
Figure 5-77 [Self-node information FL.NET (Main)] tab	5-73
Figure 5-78 [Other nodes information FL.NET (Main)] tab	5-75
Figure 5-79 [Node data FL.NET (Main)] tab (selecting the self-node)	5-76
Figure 5-80 [Node data FL.NET (Main)] tab (selecting other node)	5-77
Figure 5-81 [Network status FL.NET (Main)] tab	5-78
Figure 5-82 [RAS information FL.NET (Main)] tab	5-80
Figure 7-1 Wiring required for stand-alone checks	7-2
Figure 7-2 Wiring required for the loopback connector	7-2
Figure 7-3 Wiring required for inter-module connection check	7-3
Figure 7-4 Switch settings on the FL.NET module	7-5
Figure 7-5 LER LED and ERR LED of the FL.NET module	7-6
Figure A-1 Positioning of the FA link protocol	A-1
Figure A-2 FL-net IP address	A-2
Figure A-3 Overview of the FL-net data format	A-4
Figure A-4 Example of FL-net data (one frame)	A-5
Figure A-5 FL-net header	A-6
Figure A-6 Token circulation	A-10
Figure A-7 Startup time chart 1	A-12
Figure A-8 Startup time chart 2	A-13
Figure A-9 Types of device communication information	A-17

List of Tables

Table 1-1 Devices that can be combined in a connection, by FL-net protocol version	1-3
Table 1-2 System software (tool)	1-4
Table 2-1 Environment specifications	2-1
Table 2-2 FL.NET module specifications	2-2
Table 2-3 Registers in CPU memory available for common memory area allocation	2-3
Table 2-4 Virtual address spaces and physical memory	2-4
Table 2-5 Symbol names and their corresponding virtual address spaces	2-5
Table 2-6 Area for setting profile system parameters (example)	2-6
Table 2-7 Names of components and their functions	2-7
Table 2-8 Settings of the MAIN/SUB setting switch	2-7
Table 3-1 Mounting base for the FL.NET module	3-1
Table 3-2 Meanings of the MAIN/SUB setting switch settings	3-3
Table 5-1 Self-node management table	5-8
Table 5-2 Participating node management table	5-9
Table 5-3 Network management table	5-9
Table 5-4 List of supported message transmissions	5-16
Table 5-5 Network parameter information	5-21
Table 5-6 Ranges of values that can be entered for link parameters	5-31
Table 5-7 Address ranges where a common memory area can be allocated in CPU memory	5-34
Table 5-8 Allocating CPU memory for the common memory of the self-node	5-35
Table 5-9 Allocating CPU memory for the common memory of other nodes	5-38
Table 5-10 PCs allocation for referencing the self-node status and the FA link status	5-40
Table 5-11 PCs allocation for referencing the FA link status and the higher layer status of each node	5-41
Table 6-1 Maintenance and inspection items	6-1
Table 7-1 Details of Test and Maintenance diagnosis for FL.NET modules	7-1
Table 7-2 Test items selected by the MAIN/SUB setting switch	7-5
Table 7-3 Information displayed in the tool windows when T/M1 detects an error	7-6
Table 7-4 Information displayed in the tool windows when T/M2 detects an error	7-8
Table A-1 FL-net communication protocols	A-1
Table A-2 Classes of IP addresses	A-2
Table A-3 FL-net port numbers	A-3
Table A-4 List of transaction codes	A-7
Table A-5 Processing of transaction code reception at a UDP port	A-8
Table A-6 Tokens and data	A-11
Table A-7 Overview of the node status management table	A-14
Table A-8 Self-node management table	A-14

Table A-9 Participating node management table	A-15
Table A-10 Network management table	A-16
Table A-11 Transmission management data for message sequence number management	A-16
Table A-12 Reception management data for message sequence number management	A-16
Table A-13 Details of common parameters	A-18
Table A-14 Details of device-specific parameters	A-18
Table A-15 System parameters in tabular document notation (for PLC)	A-19
Table A-16 System parameters in tabular document notation (for CNC)	A-24

1. Overview

1.1 What is FL-net?

FL-net is an open FA network standardized by the FA Open Systems Promotion Forum (FAOP) in the Manufacturing Science and Technology Center (MSTC), an auxiliary organization of the Ministry of Economy, Trade and Industry. The specifications for FL-net were prepared by the Japan Electrical Manufacturers' Association (JEMA).

As shown in Figure 1-1, FL-net enables the interconnection of various FA controllers such as programmable controllers (PLCs), computer numerical controls (CNCs), and personal computers from a variety of different manufacturers. FL-net also enables the control and monitoring of these computers, controllers, and devices.

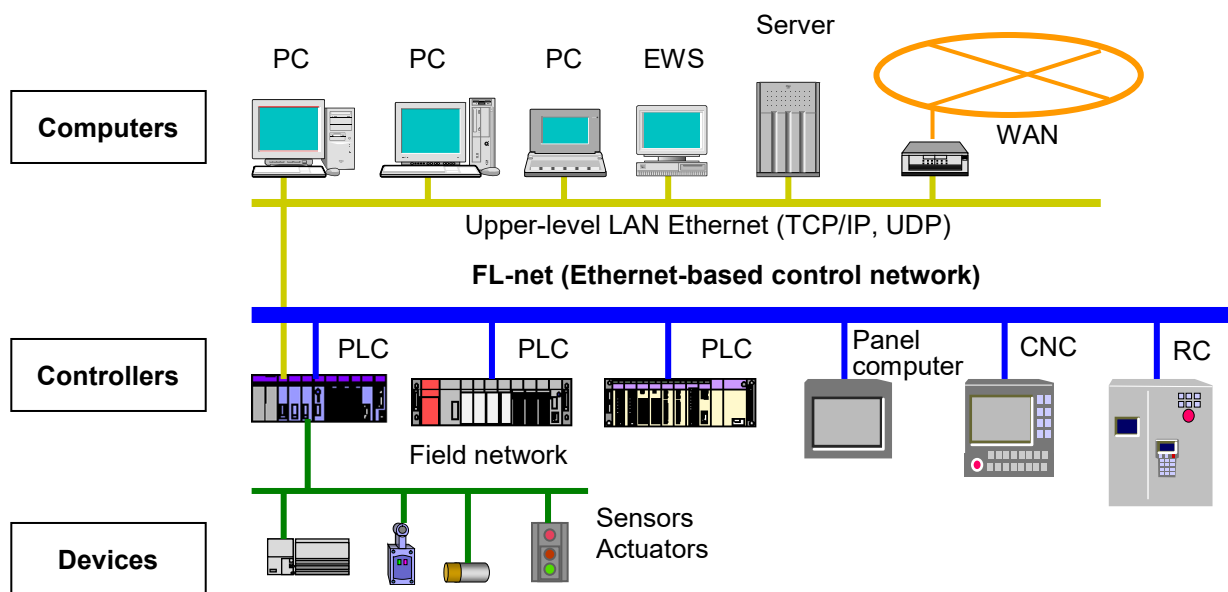


Figure 1-1 Example configuration of an FA control network

1.2 Features of FL-net

FL-net offers the following features:

- Support for open-source solutions
- Multi vendor solutions
- Interconnection of controllers such as programmable controllers (PLC), and computer numerical controls (CNC), and personal computers from a variety of different manufacturers, as well as their control and monitoring

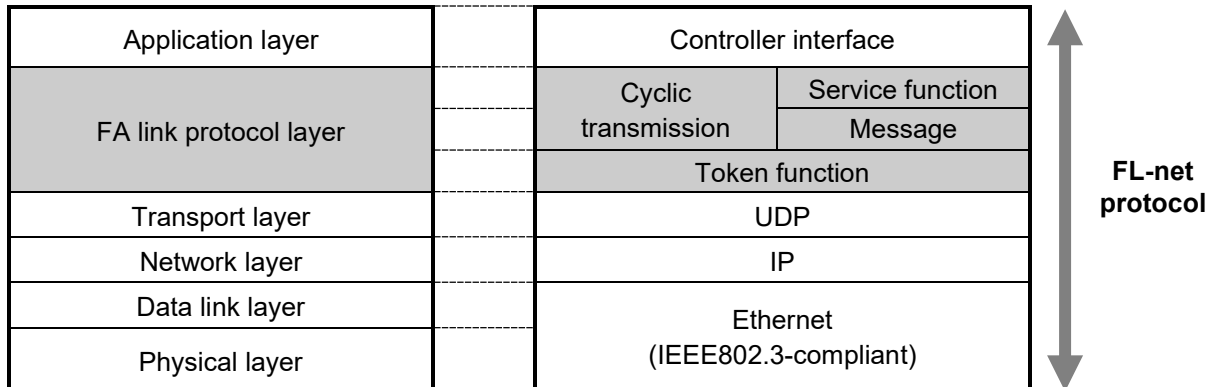


Figure 1-2 Basic structure of the FL-net protocol

<Compliance with widely accepted standards>

FL-net is based on Ethernet, a standard technology used in OA (office automation) equipment, and uses standard UDP/IP to offer efficient communications.

- Low price

FL-net is available at a low price because it allows configurations composed of common communications devices.

- Use of widespread network devices

Widely available Ethernet devices such as transceivers, hubs, cables, and LAN cards for personal computers can be used.

- Future speed increases

Transfer speeds are expected to increase from 100 Mbps to 1 Gbps in the future.

- Introduction of optical fiber communications

Optical repeaters and other widespread Ethernet devices can be used to replace relevant parts of the network with fiber-optic technologies. This enables long-distance transmission over distances of 500 m or more, improvements to noise immunity, and protection against lightning surges for outdoor wiring.

<Support for functions required for transmission among FA controllers>

Because the initial objective of FL-net is to meet user requirements, it offers various features necessary for FA.

- Large-scale networks

Up to 254 devices (nodes) can be connected.

- A choice of two types of communication functions for different purposes

FL-net supports both a common memory function and a message communication function. The common memory function allows nodes to continuously share the same data through cyclic communications, and the message communication function allows only the necessary information to be exchanged as needed.

- Large-capacity common memory

The common memory has a large capacity of 8 kbits + 8,000 words.

- High-speed response

When FL-net-compliant modules are interconnected, performance varies depending on the connected devices. When S10VE FL.NET modules (version 3.01 class 1: model LQE702-E) are interconnected, response speed is as fast as 42 ms/32 nodes (in the case of 2 kbits + 2,000 words).

- High reliability due to masterless systems

Because there is no master, individual nodes can join or leave the network without affecting other nodes' communications. As a result, nodes can be freely turned on and off or serviced.

<Notes on FL-net protocol versions>

S10VE FL.NET modules (model: LQE702-E) use FL-net protocol version 3.01.

These cannot be connected to devices that use FL-net protocol version 1.00.

Table 1-1 shows which devices can be combined in a connection, according to their version of the FL-net protocol. Take care to reference this table when setting up a connection.

Table 1-1 Devices that can be combined in a connection, by FL-net protocol version

	Version 1.00 devices	Version 2.00 devices	Version 3.01 devices
Version 1.00 devices	Y	N	N
Version 2.00 devices	N	Y	Y
Version 3.01 devices	N	Y	Y

Y: Can be connected; N: Cannot be connected

Version 1.00: FL-net protocol version of devices manufactured based on the standards in the *FA Control Network [FL-net(OPCN-2)] - Protocol Specification JEM 1479 (formulated in 2000)* issued by JEMA (Japan Electrical Manufacturers' Association)

Version 2.00: FL-net protocol version of devices manufactured based on the standards in the *FA Control Network [FL-net(OPCN-2)] - Protocol Specification JEM 1479 (formulated in 2002)* issued by JEMA (Japan Electrical Manufacturers' Association)

Version 3.01: FL-net protocol version of devices manufactured based on the standards in the *FA Control Network [FL-net(OPCN-2)] - Protocol Specification JEM 1479 (formulated in 2012)* issued by JEMA (Japan Electrical Manufacturers' Association)

1.3 System software specifications

1.3.1 System overview

To use the FL.NET module, you must register hardware switch settings and various software information to the module. You can register various module information by using the system software (tool) shown in Table 1-2. The operating procedures for this tool are equivalent to those for general Windows® applications.

Table 1-2 System software (tool)

Package name	Model
FL.NET SYSTEM/S10VE	S-7898-30

1.3.2 Hardware and software requirements

The following are the hardware and software requirements for using the system software of the FL.NET module:

- Personal computer (hereafter *computer*) with a 1-GHz or faster CPU
- Display with at least 1366 x 768 dots (FWXGA) of resolution
- At least 2 GB of RAM
- At least 200 MB of free hard disk space
- Cable for connecting the computer to the CPU module (UTP crossed twisted-pair cable of category 5e or greater)
- Microsoft® Windows® 7 (64-bit) operating system, Microsoft® Windows® 10 (64-bit) operating system
- Microsoft .NET Framework 4
- Microsoft Visual C++ 2010 redistributable package (x64)

Users of this product must have adequate knowledge of Windows® environments and user interfaces. This system complies with Windows® standards. This manual is written for users familiar with basic Windows® operating procedures.

2. FL.NET Module

2.1 System configuration

The S10VE FL.NET module (model: LQE702-E) is a module developed in compliance with the FL-net protocol. It is hereafter referred to as the *FL.NET module*.

Figure 2-1 shows an example system configuration for using FL-net communications.

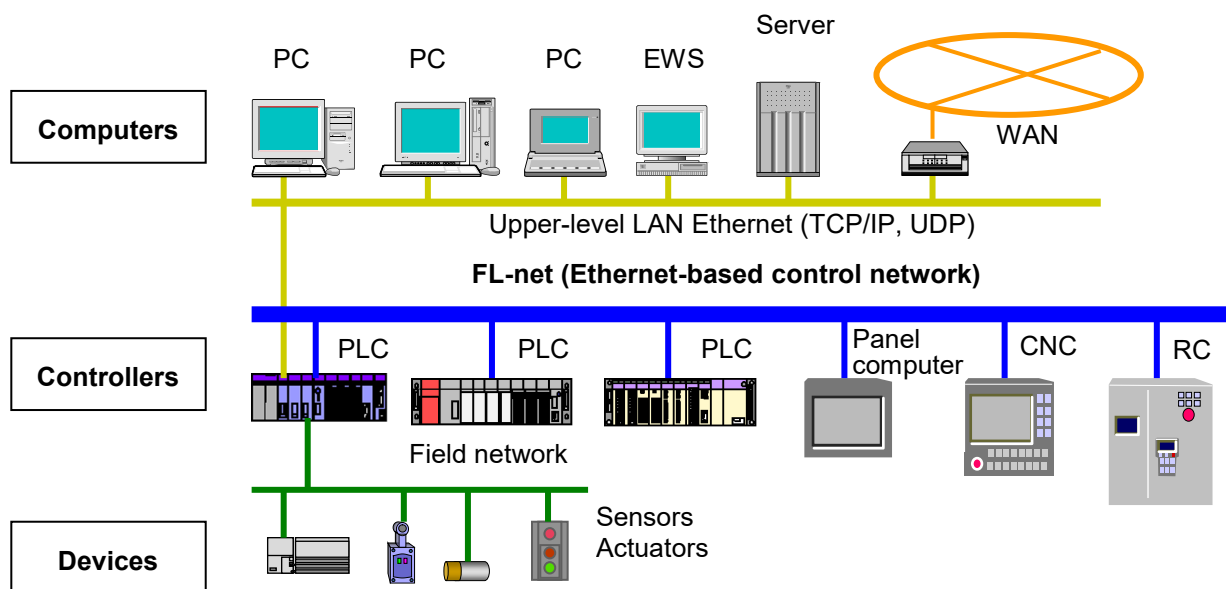


Figure 2-1 Example configuration for an FA control network

Table 2-1 Environment specifications

Item	Specification
Operating ambient temperature	0 to 55°C
Storage ambient temperature	-20 to 75°C
Relative humidity	10 to 90% RH (when operating and not operating) (non-condensing)
Vibration resistance	Frequency: 10 to 150 Hz; Acceleration: 10 m/s ² Axes: X/Y/Z; Sweep time: 8 minutes; Number of sweep cycles: 20
Shock resistance	Peak acceleration: 147 m/s ² ; Half-sine shock time: 11 ms; three times in X/Y/Z directions
Ambient air	Dust: 0.1mg/m ³ or less
Corrosive gas	JEITA IT-1004A Class B (no corrosive gas)

Notice

- To avoid damaging the product, protect it with cushioning, and then store it in the environment shown in Table 2-1.
- To avoid damaging the product during transit, protect it with cushioning and handle it as a precision machine.

2. FL.NET Module

2.1.1 FL.NET module specifications

Table 2-2 FL.NET module specifications

Item	Specification
Model	LQE702-E
Maximum number of mountable modules	2 modules per CPU
Current consumption	1,300 mA or less
Module dimensions	34 (width) x 130 (height) x 140 (depth) mm
Weight	275 g or less
Transfer speed	100 Mbps
Electrical interface	IEEE802.3-compliant (CSMA/CD-compliant)
Transmission protocol	UDP/IP, FA link protocol version 2.00
Number of connectable units	Up to 254 units per network
Communication connector	RJ45 connector
Transfer word count	Cyclic: Up to 8,500 words (node) Message: Up to 1,024 bytes (node)
Cable length	Twisted-pair cable: Up to 100 m per segment (category 5e or greater)
FL-net protocol version	Version 3.01 class 1
Hot swapping	Not supported

Note: To use the FL.NET module, you must make various settings such as node numbers and common memory settings by using the FL.NET SYSTEM/S10VE setup tool.
For detailed specifications of the setup tool, see *5.4 Installing and starting up the system* and *5.5 Commands*.

2.1.2 Link data specifications

(1) Common memory area

With the FL.NET module, you can set up a common memory area for each node.

For how to set up a common memory area, see *5.3.1 Setting link parameters*. For how to allocate part of the CPU's internal memory as a common memory area, see *5.3.2 Allocating CPU memory*.

Table 2-3 Registers in CPU memory available for common memory area allocation

No.	Available register/address	Remarks
1	XW0000 to XWFFF0	External input
2	YW0000 to YWFFF0	External output
3	JW000 to JWFF0	Transfer register
4	QW0000 to QWFFF0	Receive register
5	GW000 to GWFF0	Global link register
6	RW000 to RWFF0	Internal register
7	EW0400 to EWFFF0	Event register
8	MW0000 to MWFFF0	Internal register
9	DW000 to DWFFF	Function data register
10	FW000 to FWBFF	Function work register
11	LBW0000 to LBWFFF0	Word work register
12	LWW0000 to LWWFFFF	Word work register
13	LXW0000 to LXW3FFF	Word work register

(2) Virtual address spaces and physical memory

Table 2-4 Virtual address spaces and physical memory

Item	Description	
Area name	Memory access (0x000000 to 0xFFFFFFFF)	
Unit of access	Words	
Area size	16,777,216 bytes	
Access attributes	Read/write Conditions: Write operations cannot be performed on some ROM areas and OS areas.	
Correspondence with virtual addresses (word blocks) [#]	Vendor-specific notation	Virtual address
	0x000000, 0x000001	0x00000000
	0x000002, 0x000003	0x00000001
	⋮	⋮
	0xFFFFFE, 0xFFFFF	0x007FFFFFF
Data sequence	A 2-byte area corresponds to one word of a word block. <div style="display: flex; justify-content: space-around; align-items: center;"> <div style="border: 1px solid black; padding: 2px; text-align: center;"> <small>MSB</small> Address <i>n</i> </div> <div style="border: 1px solid black; padding: 2px; text-align: center;"> <small>LSB</small> Address <i>n</i> + 1 </div> </div>	
Other access methods	None in particular	

[#]: Correspondences with virtual addresses do not support byte blocks.

Table 2-5 shows the relationships among virtual address spaces, memory addresses, and registers.

Table 2-5 Symbol names and their corresponding virtual address spaces

Register name	Symbol name	Memory address	Virtual address space
External input	X0000 to XFFFF	0x240000 to 0x241FFE	0x120000 to 0x120FFF
External output	Y0000 to YFFFF	0x242000 to 0x243FFE	0x121000 to 0x121FFF
Internal register	R000 to RFFF	0x0AC000 to 0x0ADFFE	0x056000 to 0x056FFF
Global register	G000 to GFFF	0x0A8000 to 0x0A9FFE	0x054000 to 0x054FFF
On-delay timer	T000 to T1FF	0x0B3000 to 0x0B33FE	0x059800 to 0x0599FF
One-shot timer	U000 to U0FF	0x0B5000 to 0x0B51FE	0x05A800 to 0x05A8FF
Counter	C000 to C0FF	0x0B7000 to 0x0B71FE	0x05B800 to 0x05B8FF
Keep	K000 to KFFF	0x0B0000 to 0x0B1FFE	0x058000 to 0x058FFF
System register	S0000 to SBFFF	0x0BE800 to 0x0BFFFE	0x05F400 to 0x05FFFF
Zee register	Z000 to Z3FF	0x0BE000 to 0x0BE7FE	0x05F000 to 0x05F3FF
Internal register	M0000 to MFFFF	0x0AE000 to 0x0AFFFE	0x057000 to 0x057FFF
Transfer register	J000 to JFFF	0x0A2000 to 0x0A3FFE	0x051000 to 0x051FFF
Receive register	Q0000 to QFFFF	0x0A6000 to 0x0A7FFE	0x053000 to 0x053FFF
External input (word)	XW0000 to XWFFF0	0x414000 to 0x4141FE	0x20A000 to 0x20A0FF
External output (word)	YW0000 to YWFFF0	0x414200 to 0x4143FE	0x20A100 to 0x20A1FF
Internal register (word)	RW000 to RWFF0	0x0E0C00 to 0x0E0DFE	0x070600 to 0x0706FF
Global register (word)	GW000 to GWFF0	0x0E0800 to 0x0E09FE	0x070400 to 0x0704FF
On-delay timer (word)	TW000 to TW1F0	0x0E1300 to 0x0E133E	0x070980 to 0x07099F
One-shot timer (word)	UW000 to UW0F0	0x0E1500 to 0x0E151E	0x070A80 to 0x070A8F
Counter (word)	CW000 to CW0F0	0x0E1700 to 0x0E171E	0x070B80 to 0x070B8F
Keep (word)	KW000 to KWFF0	0x0E1000 to 0x0E11FE	0x070800 to 0x0708FF
System register (word)	SW0000 to SWBFF0	0x0E1E80 to 0x0E1FFE	0x070F40 to 0x070FFF
Zee register (word)	ZW000 to ZW3F0	0x0E1E00 to 0x0E1E7E	0x070F00 to 0x070F3F
Internal register (word)	MW0000 to MWFFF0	0x0E0E00 to 0x0E0FFE	0x070700 to 0x0707FF
Transfer register (word)	JW000 to JWFF0	0x0E0200 to 0x0E03FE	0x070100 to 0x0701FF
Receive register (word)	QW0000 to QWFFF0	0x0E0600 to 0x0E07FE	0x070300 to 0x0703FF
Function data register	DW000 to DWFFF	0x061000 to 0x062FFE	0x030800 to 0x0317FF
Function work register	FW000 to FWBFF	0x0E2000 to 0x0E37FE	0x071000 to 0x071BFF
Work register	LB0000 to LBFFFF	0x220000 to 0x23FFFE	0x110000 to 0x11FFFF
Word work register	LBW0000 to LBWFFF0	0x412000 to 0x413FFE	0x209000 to 0x209FFF
Word work register	LWW0000 to LWWFFFF	0x450000 to 0x46FFFE	0x228000 to 0x237FFF
Word work register	LXW0000 to LXW3FFF	0x4A0000 to 0x4A7FFE	0x250000 to 0x253FFF

2. FL.NET Module

(3) Status memory area

You can view information on the self-node status, FA link status, and higher layer status of the FL.NET module by using the FL.NET SYSTEM/S10VE setup tool (for details, see *5.3 Using the FL.NET module*).

2.1.3 Area for setting link parameters

You can view the link parameters for the FL.NET module by using the FL.NET SYSTEM/S10VE setup tool (for details, see *5.3 Using the FL.NET module*).

2.1.4 Area for setting profile system parameters

You can view the profile system parameters for the FL.NET module by using the FL.NET SYSTEM/S10VE setup tool (for details, see *5.3 Using the FL.NET module*).

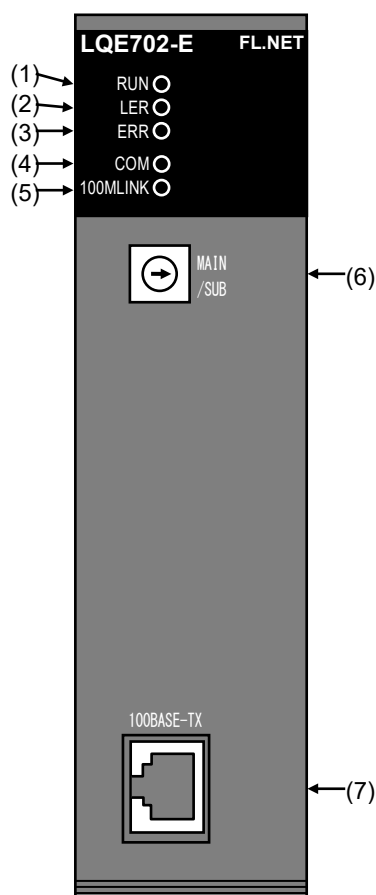
For the FL.NET module, the vendor name, manufacturer model, and protocol version in the profile system parameters are fixed data. The FL.NET module automatically changes the FA link status and self-node status according to the communication status and other conditions. Users can use the setup tool to set the node name only.

Table 2-6 Area for setting profile system parameters (example)

Item	Byte length	Data	Description
Vendor name	10 bytes	"HITACHI"	Vendor name
Manufacturer model	10 bytes	"LQE702"	Manufacturer model, device name
Node name (equipment name)	10 bytes	--	User-defined node name
Protocol version	1 byte	--	Fixed at 0x80
FA link status	1 byte	--	Joined or left the network, etc.
Self-node status	1 byte	--	Duplicate node number detected, etc.

2.2 FL.NET module component names and their functions

Table 2-7 Names of components and their functions



No.	Name	Function
(1)	RUN LED	This LED is lit when the module is up and running normally.
(2)	LER LED	This LED is lit when an FA link is not connected, or in the event of an error in FA link communications.
(3)	ERR LED	This LED is lit in the event of a hardware error.
(4)	COM LED	This LED is lit when data is flowing through the transmission lines (a carrier is detected).
(5)	100MLINK LED	This LED is lit when a circuit is connected for communication at 100 Mbps.
(6)	MAIN/SUB setting switch	This switch is used to define the module as the main or sub module. For the setting values, see Table 2-8.
(7)	Communication connector	This connector is for 100BASE-TX communications.

Table 2-8 Settings of the MAIN/SUB setting switch

Setting value	Description
0	Main setting
1	Sub setting
2 to B, E	Setting prohibited
C, D, F	Setting prohibited (for maintenance only)

Notice

- Switch off the power to the power supply module before setting the MAIN/SUB setting switch. Setting the MAIN/SUB setting switch while the power is switched on might result in a malfunction.

This page is intentionally left blank.

3. Mounting the FL.NET Module

3.1 Mounting base

The FL.NET module can be mounted on the mounting base shown in Table 3-1.

Table 3-1 Mounting base for the FL.NET module

Name	Model	Specifications
7-slot mounting base	HSC-1770	Power supply + RI/O-IF + CPU + 7 I/O slots

3.2 Mounting the module

Mount the optional module in one of the I/O slots (slots 1 to 7) of the mounting base shown in Figure 3-1.

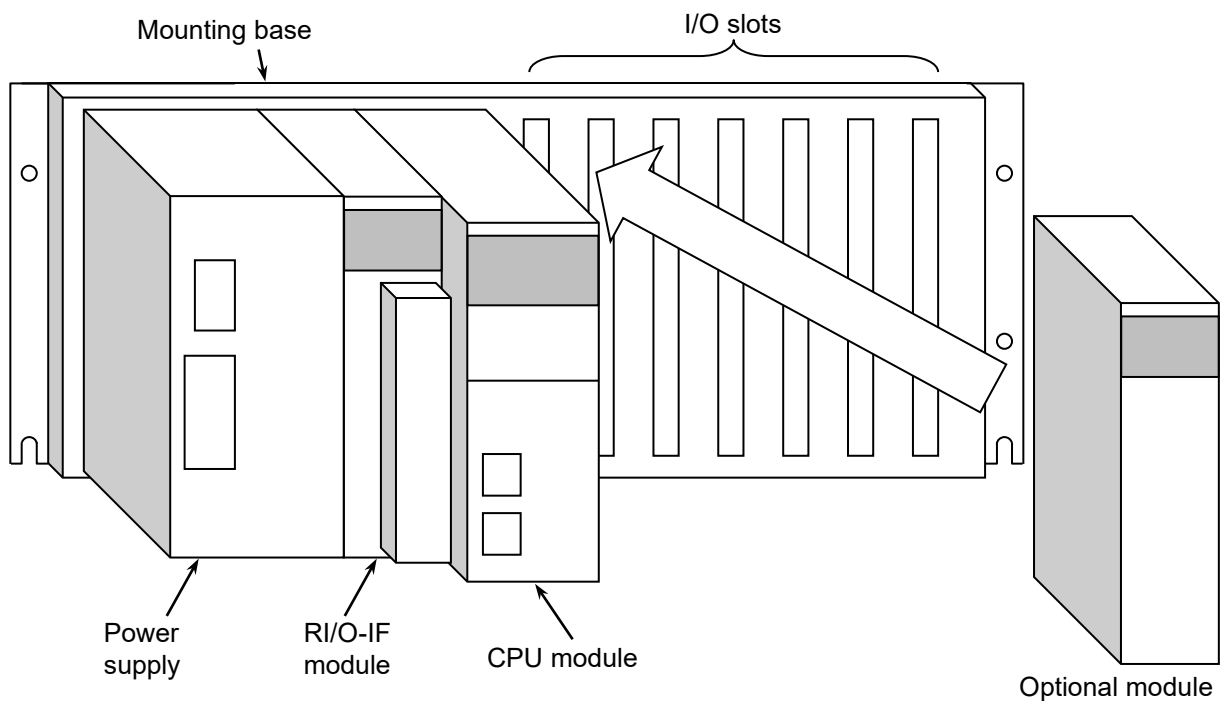


Figure 3-1 Mounting the optional module

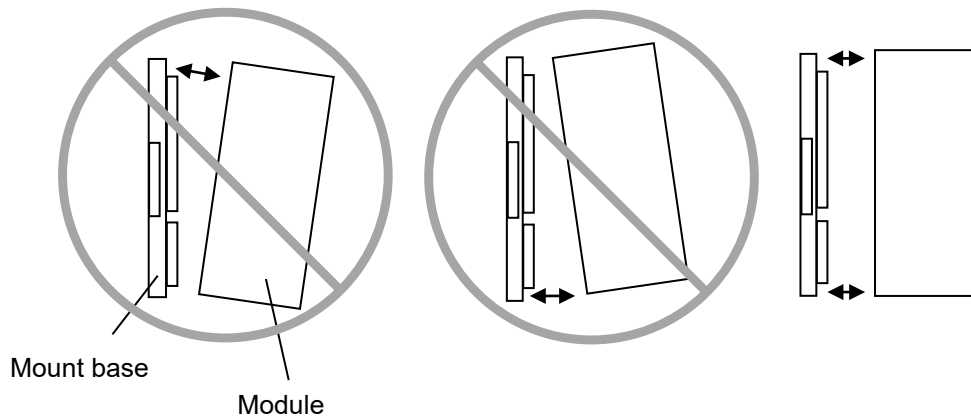
3. Mounting the FL.NET Module

Notice

- Poor contact might cause malfunction. Mount the module and connect cabling to the module immediately after the module is unpacked so that dust or other foreign substances do not accumulate on connectors.
- To avoid damaging a module when removing or mounting it, turn off the power switch of the power supply module beforehand.
- To avoid damaging a module, observe the following precautions when mounting or removing the module:
 - Before mounting the module on the mount base connector, check that the connector pins are properly aligned and not bent, broken, or soiled.
 - Connecting or disconnecting a module that is tilted might damage connector pins. When moving the module, hold it vertically parallel to the mount base as shown below.

Bad examples

Good example



3.3 Setting the MAIN/SUB setting switch

When using the FL.NET module, set it as either the main module or the sub module.

You can do so by aligning the arrow on the MAIN/SUB setting switch to either 0 (main module) or 1 (sub module).

When only one FL.NET module is mounted, set the MAIN/SUB setting switch to 0.

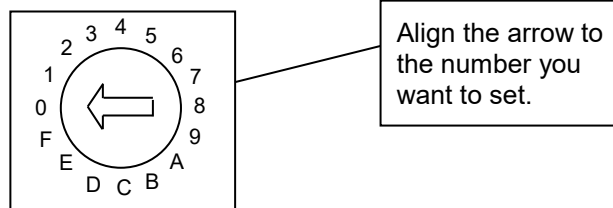


Figure 3-2 Setting the MAIN/SUB setting switch

Table 3-2 shows the meanings of the MAIN/SUB setting switch settings.

Table 3-2 Meanings of the MAIN/SUB setting switch settings

MAIN/SUB		Setting
Main module	Sub module	
0	1	100BASE-TX communication
2 to B, E		These values are invalid settings for the MAIN/SUB setting switch. Do not set these values. If you set these values, an incorrect setting error occurs in the MAIN/SUB setting switch, making communication impossible.
C	D	Do not set these values. They are for maintenance purposes.
F	F	

Notice

- Switch off the power to the power supply module before setting the MAIN/SUB setting switch. Setting the MAIN/SUB setting switch while the power is switched on might result in a malfunction.

3. Mounting the FL.NET Module

3.4 External dimensions of the FL.NET module

Figure 3-3 shows the external dimensions of the FL.NET module.

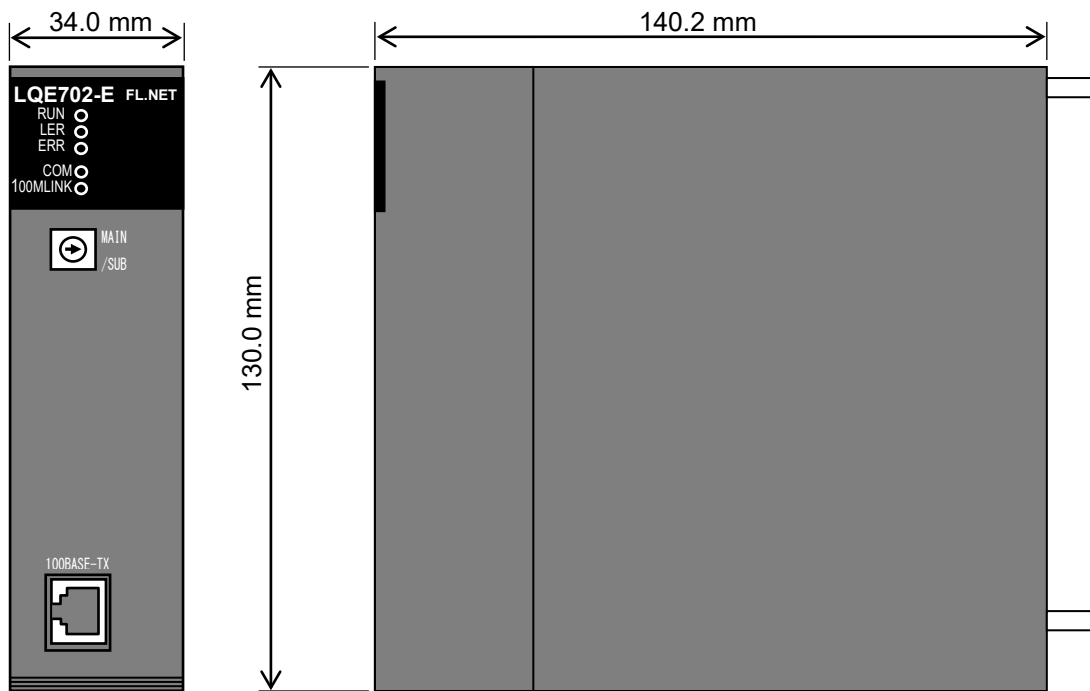


Figure 3-3 External dimensions of the FL.NET module

4. Wiring the FL.NET Module

4.1 Connecting the communication cable

The FL.NET module communicates with other modules by using 100BASE-TX interface connectors.

Insert the 100BASE-TX communication cable at the position shown in Figure 4-1.

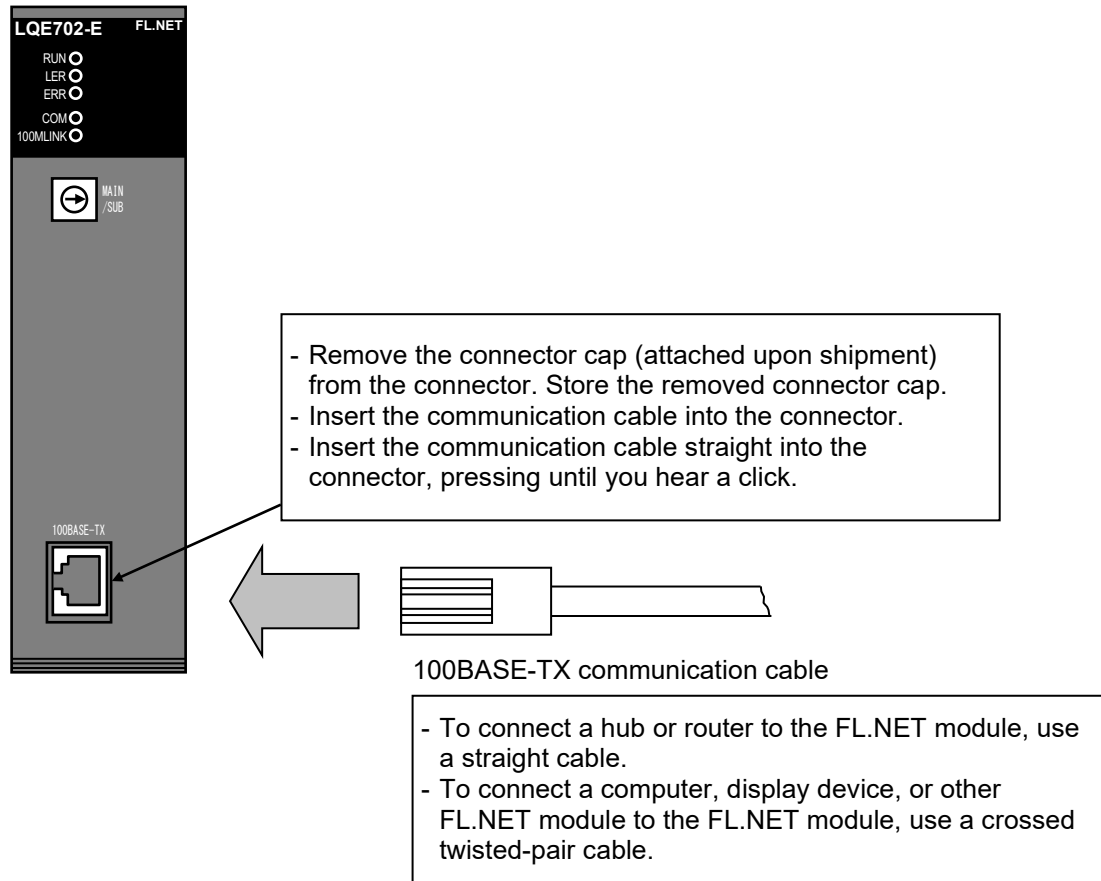


Figure 4-1 Wiring by using a 100BASE-TX communication cable

Notice

- A loose connection might cause a malfunction. Do not use communication cables that have any broken cable lock pins. After connecting the connector, confirm that the connector is locked.
- Do not touch the connectors when the system is powered on. This might cause a malfunction.

4.2 Network configuration components

The FL.NET module is a standard product in compliance with the IEEE802.3 international standards.

Any network configuration components such as hubs and twisted-pair cables used in combination with the FL.NET module must comply with the IEEE802.3 standards.

Make sure that you sufficiently verify the operation of network components before use.

4.3 Connection configuration using a hub

As shown in Figure 4-2, you can enable communication among Ethernet devices by connecting multiple Ethernet devices to a hub. To connect an Ethernet device to a hub, use a straight cable.

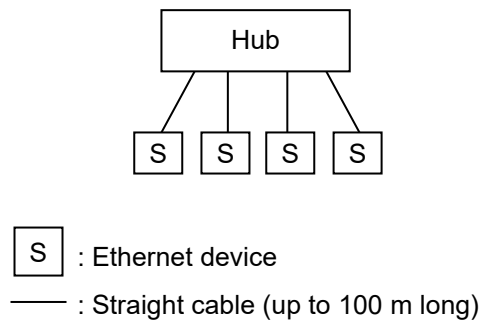


Figure 4-2 Connection configuration using a hub

■ Configuration when using a 100-Mbps hub

- When using a 100-Mbps hub (class 1), you cannot connect multiple tiers of hubs. Connect devices to the hub as shown in Figure 4-3.

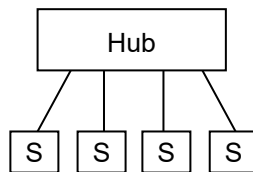


Figure 4-3 Example configuration using a 100-Mbps hub (class 1)

- When using a 100-Mbps hub (class 2) hub, you can connect up to two tiers of hubs. Connect devices to and use the hubs as shown in Figure 4-4. (When connecting hubs together, use crossed twisted-pair cables of up to 5 m in length.)

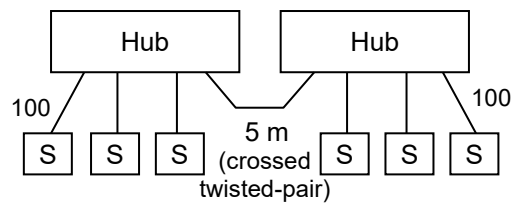


Figure 4-4 Example configuration using a 100-Mbps hub (class 2)

- When using a 100-Mbps switching hub, as a rule, you can use an unlimited number of cascade connection tiers. However, refer to the manual for the switching hub before connecting devices.

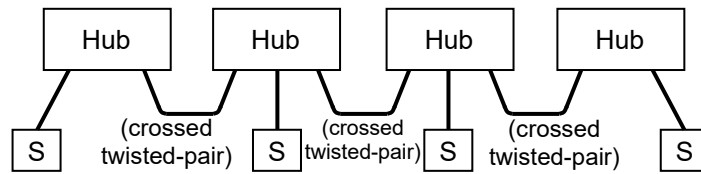


Figure 4-5 Example configuration using a 100-Mbps switching hub

Notice

- As circuit load increases, so does the risk of communication failure. Because the FL.NET module (LQE702-E) only supports auto-negotiation settings, do not specify the 100-Mbps full-duplex setting in the port settings of the switching hub.

This page is intentionally left blank.

5. User's Guide

5.1 About FL-net

5.1.1 Overview of FL-net

(1) Concepts behind FL-net

FL-net is an Ethernet-based FA (factory automation) control network protocol.

It has a cyclic transmission function and a message transmission function.

The following are the basic ideas behind FL-net:

- Ethernet is used as the medium (physical and data link layers) of communication among FA controllers.
- UDP/IP, which is commonly used with Ethernet, is used to provide a basic means of data transmission.
- Transmission is assured by managing and controlling access to the communication medium by each node in the network (collision avoidance) while making use of the aforementioned basic means of data transmission.

FL-net is designed for FA control networks that facilitate data exchange among nodes in production systems. These nodes include control devices such as programmable controllers (PLCs), robot controllers (RCs), and computer numerical controls (CNCs), and personal computers for control purposes.

Figure 5-1 shows where the FL-net network belongs in a production system.

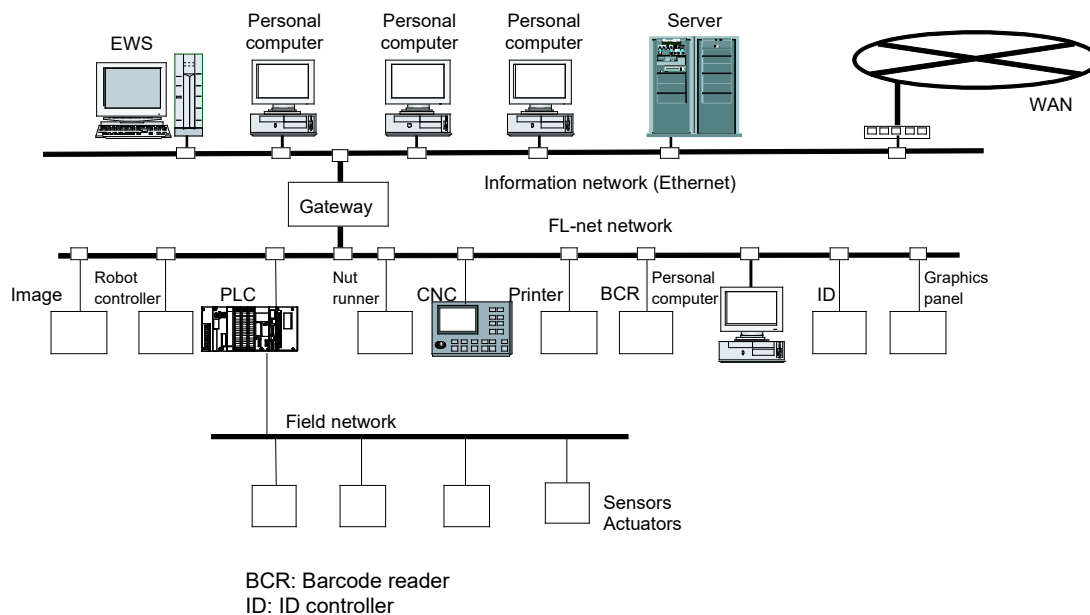


Figure 5-1 Concepts behind FL-net

(2) FL-net protocol

FL-net consists of six protocol layers as shown in Figure 5-2.

The transport and network layers use UDP/IP, whereas the data link and physical layers use Ethernet.

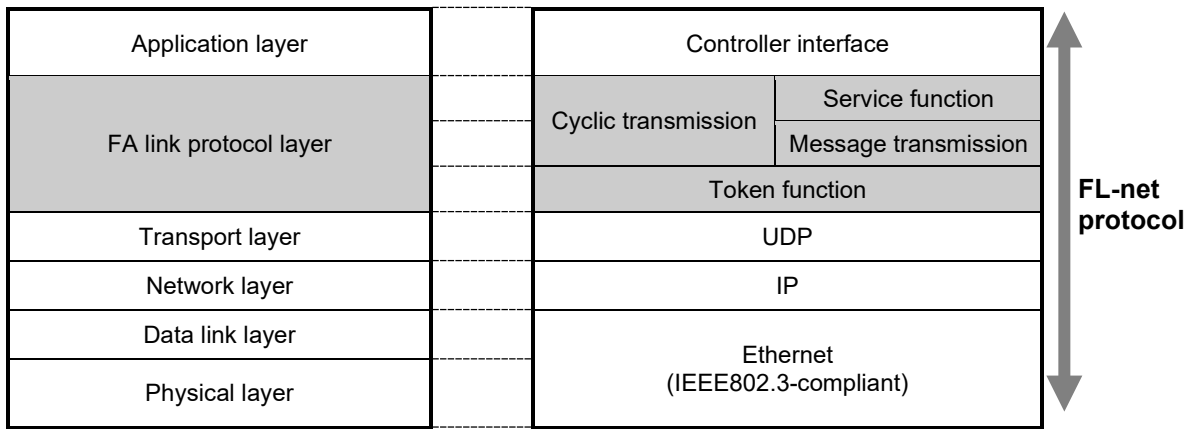


Figure 5-2 FA link protocol

(3) Features of the FL-net transmission system

FL-net's FA link protocol layer offers the following features:

- Collision avoidance is achieved through masterless, token-based transmission management.
- Refresh cycle time can be regulated by circulating tokens.
- A predetermined token is transmitted together with cyclic data.
- At startup, a token is transmitted from the node with the smallest node number.
- If a token timeout is detected, the next node transmits a token.
- Thanks to the use of a masterless, token-based system, the failure of some nodes will not interrupt network operations.
- Tables are provided for managing information such as the operating mode (RUN/STOP) and hardware errors (ALARM). These tables can be used to check the operational status of other nodes.

(4) FL-net IP addresses

IP addresses are addresses used in IP (Internet Protocol) transmission to designate particular nodes (stations). You must therefore set and manage IP addresses to make sure there are no duplicate addresses.

An IP address needs to be set individually for each FL-net node. It is recommended that you use Class C IP addresses in accordance with the FL-net protocol conventions.

The default values of FL-net IP addresses follow the pattern 192.168.250.***, where the asterisks (***) represent the node number. (The recommended value according to the FL-net protocol conventions is set by default.)

Network address	Node number (host number)
192.168.250.	***: 1 to 254

Figure 5-3 FL-net IP addresses

5.1.2 Connection capacity and node numbers

Up to 254 units can be connected. The available node numbers range from 1 to 254.

Node numbers 1 to 249: For normal FL-net devices

Node numbers 250 to 254: For maintenance of FL-net

Node number 255: For internal use by FL-net; unavailable to users (This number is used for global address broadcasting.)

Node number 0: For internal use by FL-net; unavailable to users

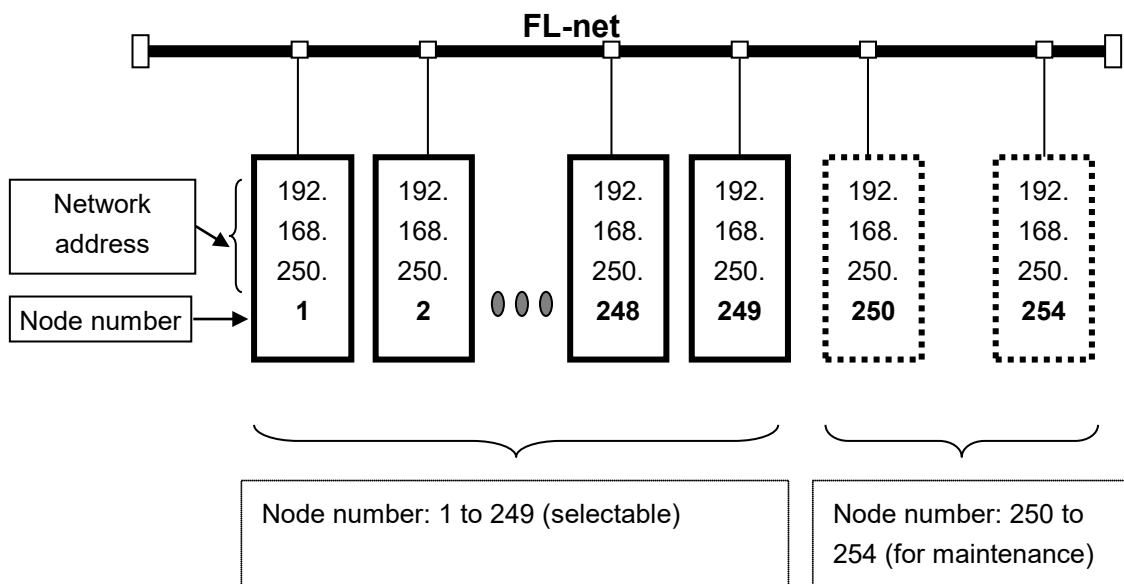


Figure 5-4 FL-net connection capacity and node numbers

5.1.3 Supported types of data communications

The types of data communications supported by FL-net are cyclic transmission and message transmission.

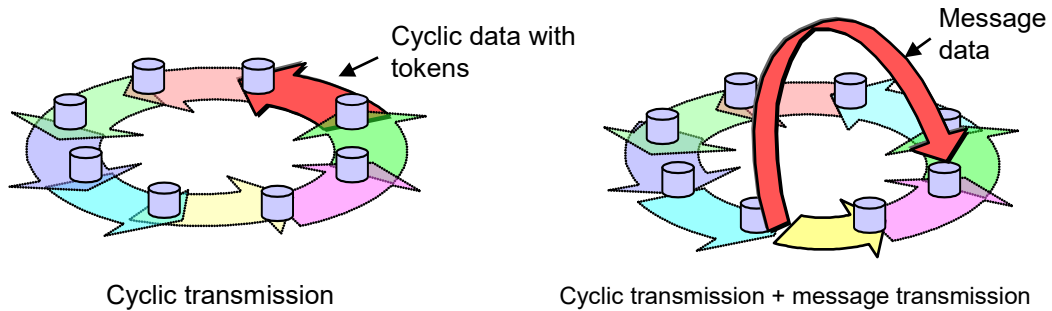


Figure 5-5 Types of data communications supported by FL-net

(1) Cyclic transmission

Cyclic transmission provides cyclic (periodic) transmission of data. All nodes can share data via the common memory (shared memory).

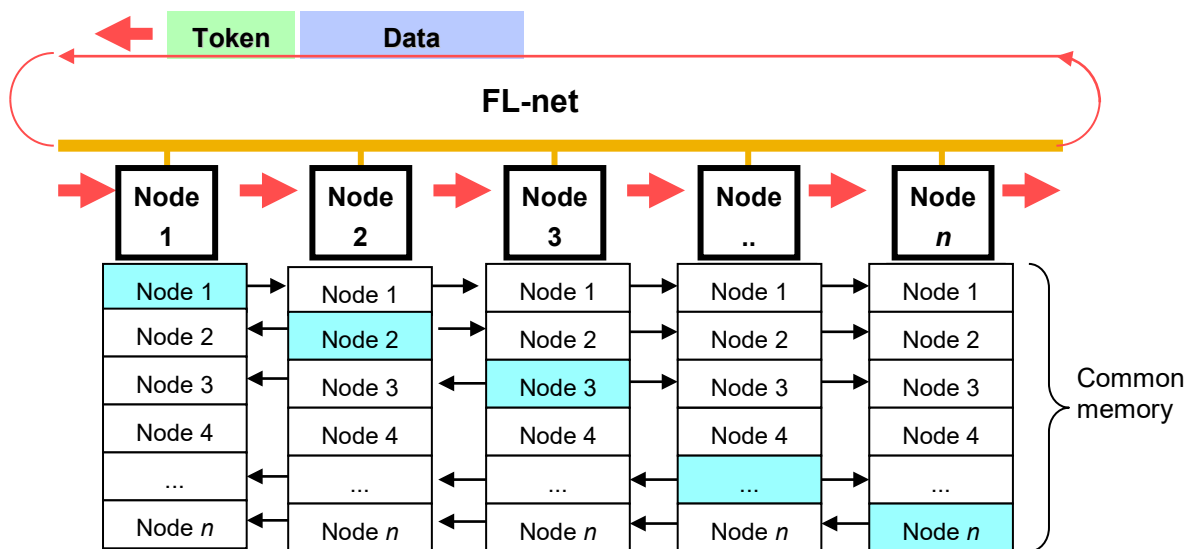


Figure 5-6 Example of common memory and cyclic transmission

(2) Message transmission

Message transmission provides non-cyclic (non-periodic) transmission of data.

Under normal conditions, communications are transmitted to a particular node upon request.

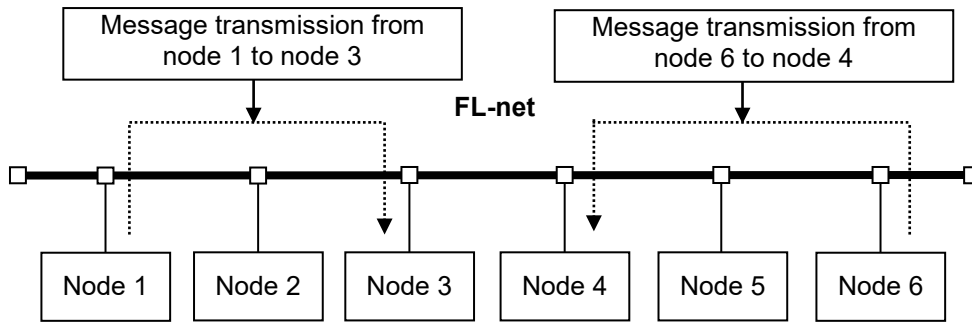


Figure 5-7 Example of message transmission

5.1.4 Amount of data that can be transmitted

(1) Cyclic transmission

The entire network contains 8,500 words (= 8 kbits + 8,000 words).

Up to 8,500 words of data can be transmitted per node.

Note that each word consists of two bytes.

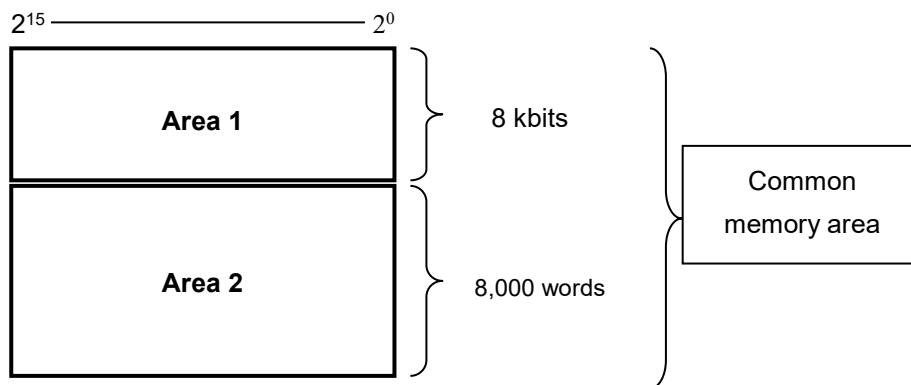


Figure 5-8 Amount of data that can be transmitted with cyclic transmission

(2) Message transmission

One message can contain up to 1,024 bytes of frame data (excluding the header).

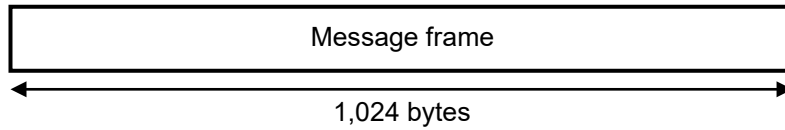


Figure 5-9 Amount of data that can be transmitted with message transmission

5.1.5 The transfer cycle and its monitoring

In cyclic data communications, the common memory is refreshed at mostly fixed intervals. Message transmission is controlled to prevent a single message communication from exceeding the time limit on the common memory refresh cycle.

Each node continuously monitors message communication frames flowing through the network, from when a token addressed to the self-node is received, to when the token addressed to the next self-node is received.

When no message communication frame flows through the network in one such cycle, a value of 120% of the duration of this cycle is set as the refresh cycle time limit.

Thanks to the aforementioned monitoring process, the refresh cycle time limit is determined dynamically according to the number of nodes that have joined the network.

Example: When only cyclic data communications are maintained by five nodes
(None of the nodes perform message communication.)

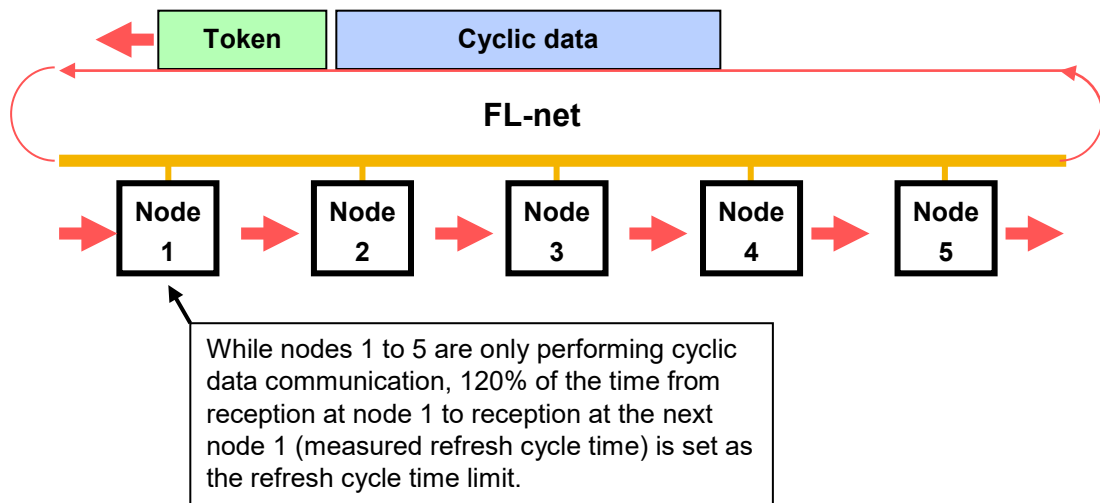


Figure 5-10 Example of a refresh cycle time limit

For the calculation method, see 5.1.11 *Communication time of the FL.NET module.*

5.1.6 Data area and memory

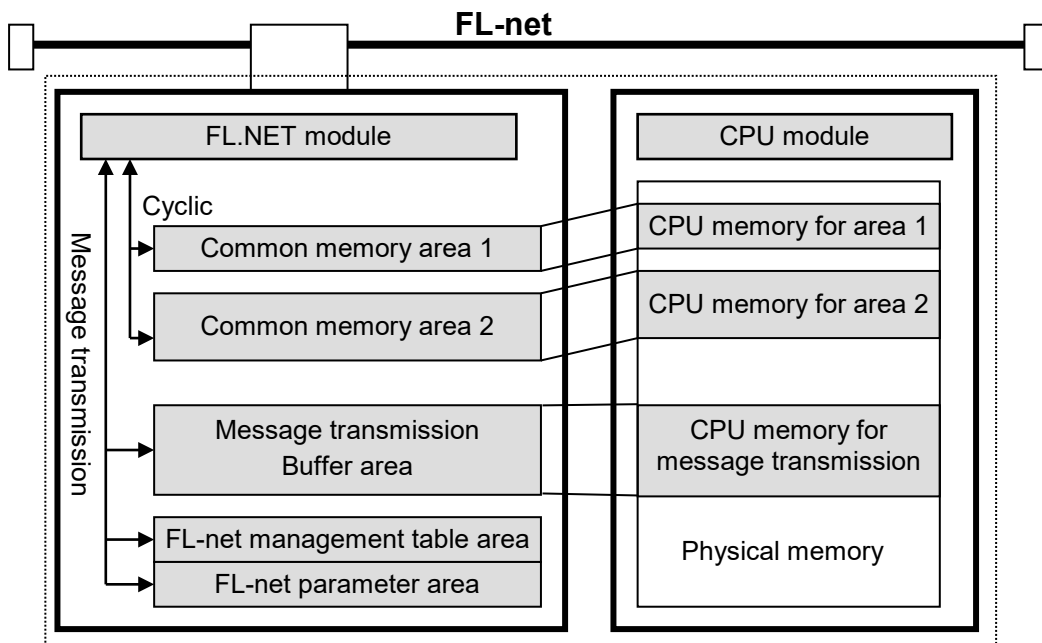


Figure 5-11 Data area and memory

5.1.7 Communication management tables

Node statuses are managed by using the self-node management table, participating node management table, and network management table.

(1) Self-node management table

The Self-node management table (Table 5-1) is used to manage the self-node settings.

Table 5-1 Self-node management table

Name	Length	Description/data range
Node number	1 byte	1 to 254
Starting address of area 1 data for the common memory	2 bytes	Word address (0 to 0x1FF)
Size of area 1 data for the common memory	2 bytes	Size (0 to 0x1FF)
Starting address of area 2 data for the common memory	2 bytes	Word address (0 to 0x1FFF)
Size of area 2 data for the common memory	2 bytes	Size (0 to 0x1FFF)
Higher layer status	2 bytes	RUN/STOP/ALARM/WARNING/NORMAL
Token monitoring timeout	1 byte	In units of 1 ms
Minimum frame interval	1 byte	In units of 100 us
Vendor name	10 bytes	Vendor name
Manufacturer model	10 bytes	Manufacturer model, device name
Node name (equipment name)	10 bytes	User-defined node name
Protocol version	1 byte	Fixed at 0x80
FA link status	1 byte	Joined or left the network, etc.
Self-node status	1 byte	Duplicate node number detected, etc.

(2) Participating node management table

The participating node management table (Table 5-2) is used to manage information about nodes that have joined the network.

Table 5-2 Participating node management table

Name	Length	Description/data range
Node number	1 byte	1 to 254
Higher layer status	2 bytes	RUN/STOP/ALARM/WARNING/NORMAL
Starting address of area 1 data for the common memory	2 bytes	Word address (0 to 0x1FF)
Size of area 1 data for the common memory	2 bytes	Size (0 to 0x1FF)
Starting address of area 2 data for the common memory	2 bytes	Word address (0 to 0x1FFF)
Size of area 2 data for the common memory	2 bytes	Size (0 to 0x1FFF)
Refresh cycle time limit	2 bytes	In units of 1 ms
Token monitoring timeout	1 byte	In units of 1 ms
Minimum frame interval	1 byte	In units of 100 us
Link status	1 byte	Joined or left the network, etc.

(3) Network management table

The network management table (Table 5-3) is used to manage information shared with the network.

Table 5-3 Network management table

Name	Length	Description/data range
Number of node holding the token	1 byte	Node currently holding the token
Minimum frame interval	1 byte	In units of 100 us
Refresh cycle time limit	2 bytes	In units of 1 ms
Measured refresh cycle time (current value)	2 bytes	In units of 1 ms
Measured refresh cycle time (maximum value)	2 bytes	In units of 1 ms
Measured refresh cycle time (minimum value)	2 bytes	In units of 1 ms

5.1.8 Cyclic transmission and area

(1) Overview of cyclic transmission

The cyclic transmission function uses the common memory to cyclically (periodically) exchange data.

- A node transmits data when it has the token.
- Nodes not performing cyclic transmission are still allowed to participate in the network.
- When a node has the token, it transmits all pending cyclic data.

Token: As a rule, there is only one token per network. If a network has two or more tokens, the token with the smallest destination node number takes precedence, and all other tokens are discarded.

Token frame: A token frame (frame containing a token) contains the token's destination node number, and a node number to which the token is transmitted. A node becomes the holder of a token when its node number matches the destination node number of the token frame it receives.

Token sequence: The token rotation sequence is determined based on node numbers. The token is passed around in a rotation in ascending order, among the nodes registered in the participating node management table. The node with the largest node number passes the token to the node with the smallest node number.

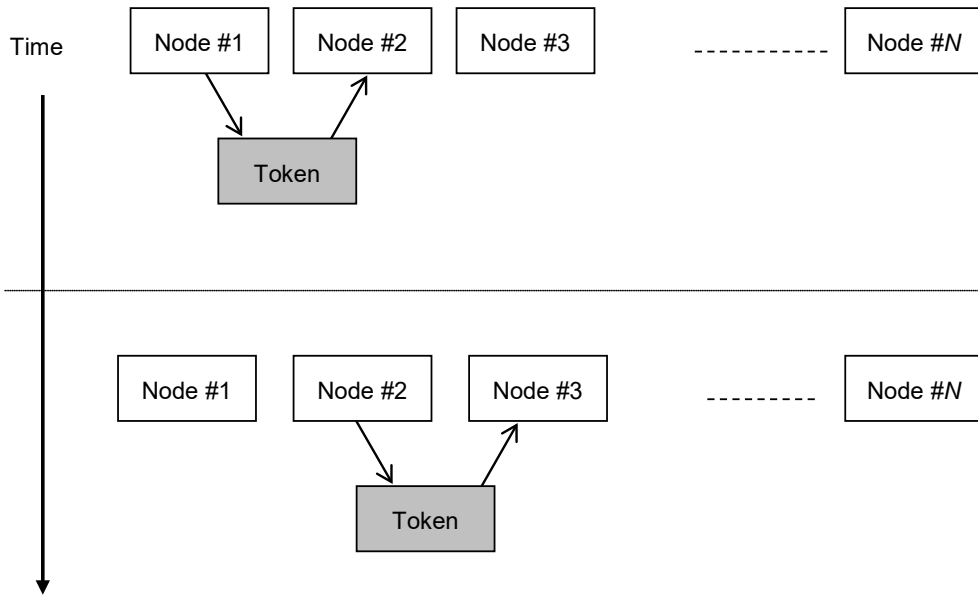


Figure 5-12 Token rotation and cyclic transmission 1

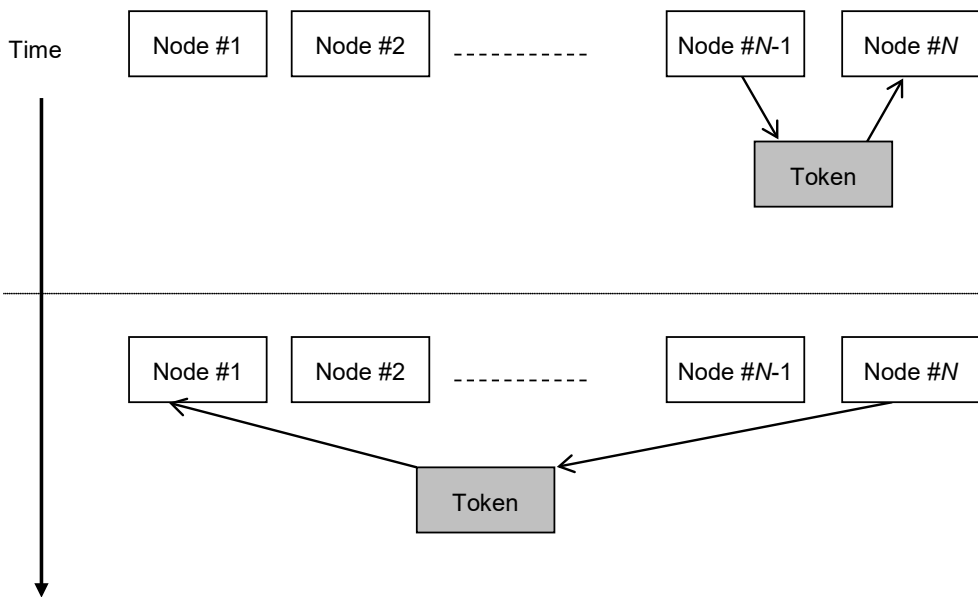


Figure 5-13 Token rotation and cyclic transmission 2

(2) Common memory

The following are the ideas behind common memory:

- Common memory provides nodes performing cyclic transmission with functions that can be treated as a shared memory.
- Two types of areas (areas 1 and 2) can be allocated for one node.
- If the 1,024-byte transmission size limit of one frame is exceeded by an area transmitted by one node, two or more frames are used for data transmission.
- In the reception of data divided into two or more frames, the common memory is not refreshed until all the frames coming from a node are successfully received. In other words, synchronicity is assured on a per-node basis.
- The common memory size provided by the communication section of one node is fixed at 8,500 words (= 8 kbits + 8,000 words).
- Areas 1 and 2 can both be freely specified as the common memory's transmission area for one node as long as the maximum size is not exceeded.
- The same data can be shared across the entire system by having each node broadcast data at fixed intervals. Nodes on the FL-net network exchange data by using their own transmission areas that do not overlap with those of the other nodes. In a common memory operation, the transmission area allocated for a given node is the reception area for another node.

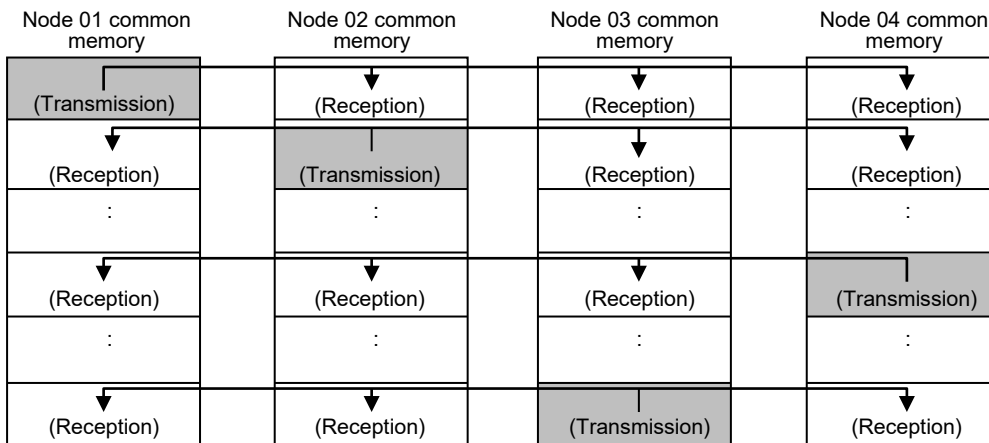


Figure 5-14 Common memory area for cyclic transmission: Example 1

It is also possible to use only the reception area of the common memory.

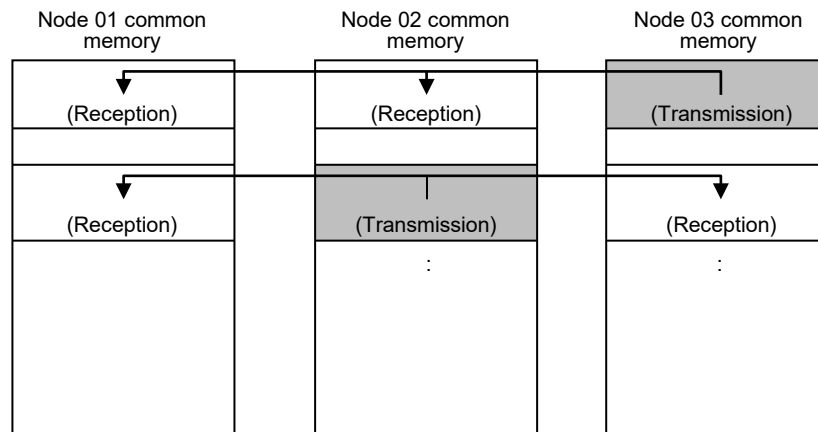


Figure 5-15 Common memory area for cyclic transmission: Example 2

(3) Areas 1 and 2

One node can allocate two data areas (areas 1 and 2) to the common memory. The transmission area is set up in accordance with the area starting address and size.

For area access, a word address is used. Area 1 consists of 500 words, and area 2 consists of 8,000 words.

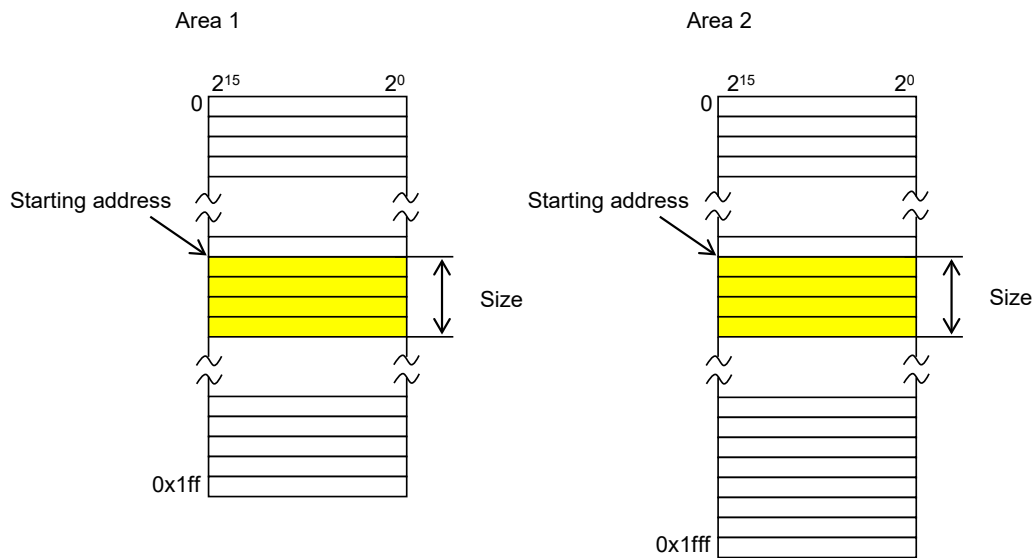


Figure 5-16 Common memory areas 1 and 2

(4) Data synchronicity assurance

In cyclic transmission, the data to be transmitted is divided into frames depending on its size.

The following procedures are used to assure common memory (internal memory such as the FL.NET module) synchronicity on a per-node basis:

- Transmission timing

When a data transmission is requested by a higher layer, the cyclic data of the self-node is copied to a buffer to prepare for transmission, and then transmitted sequentially. If the data retained by a transmitting node exceeds the maximum size for transmitting one frame, the buffered data is divided into two or more frames for transmission.

- Refresh timing for reception sequences

When a receiving node successfully receives all cyclic data from another node, it refreshes the associated area while maintaining synchronicity with a higher layer.

Even when cyclic data is transmitted in two or more frames, the associated area is refreshed when all the frames transmitted from a node are successfully received. If not all of these transmitted frames are received, all data received from the transmitting node is discarded.

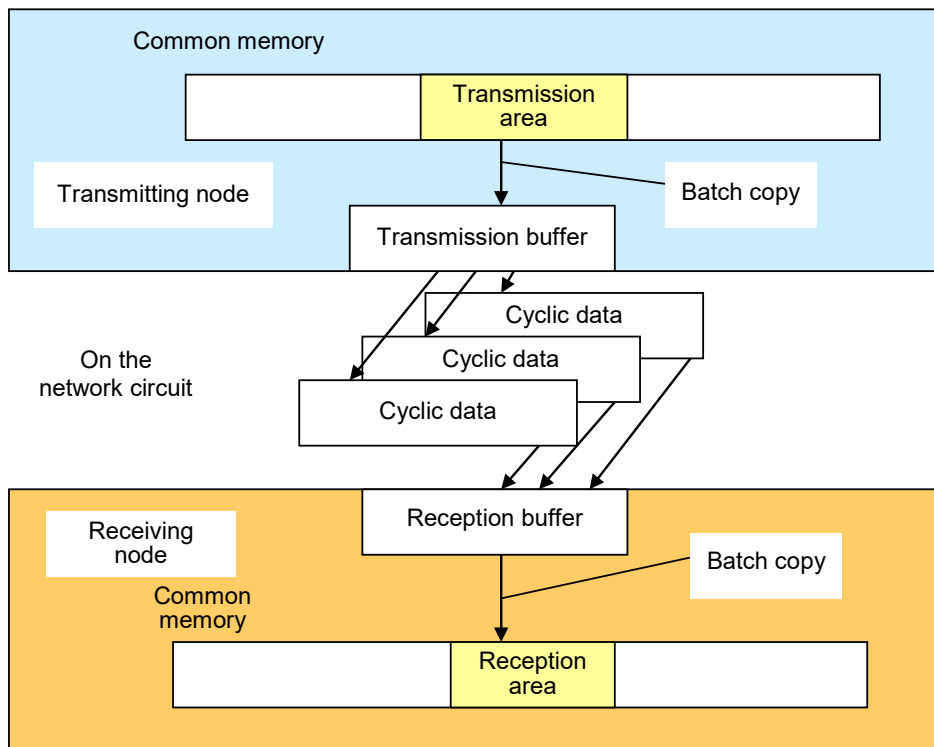


Figure 5-17 Data synchronicity assurance

5.1.9 Message transmission

(1) Overview of message transmission

The message transmission function supports asynchronous data exchange among nodes.

The following describes the basic functions of message transmission:

- Upon receiving a token, a node can transmit up to one frame of data before cyclic frame transmission.
- Up to 1,024 bytes of data can be transmitted in one session.
- A special algorithm is employed to ensure that the refresh cycle time limit for cyclic transmission is not exceeded.
- Functions are provided for both one-to-one transmission for sending data to a specified destination node, and one-to- n transmission for sending data to all nodes.
- In one-to-one message transmission, the delivery verification (response) function is available for checking whether transmitted data is successfully received at its destination.

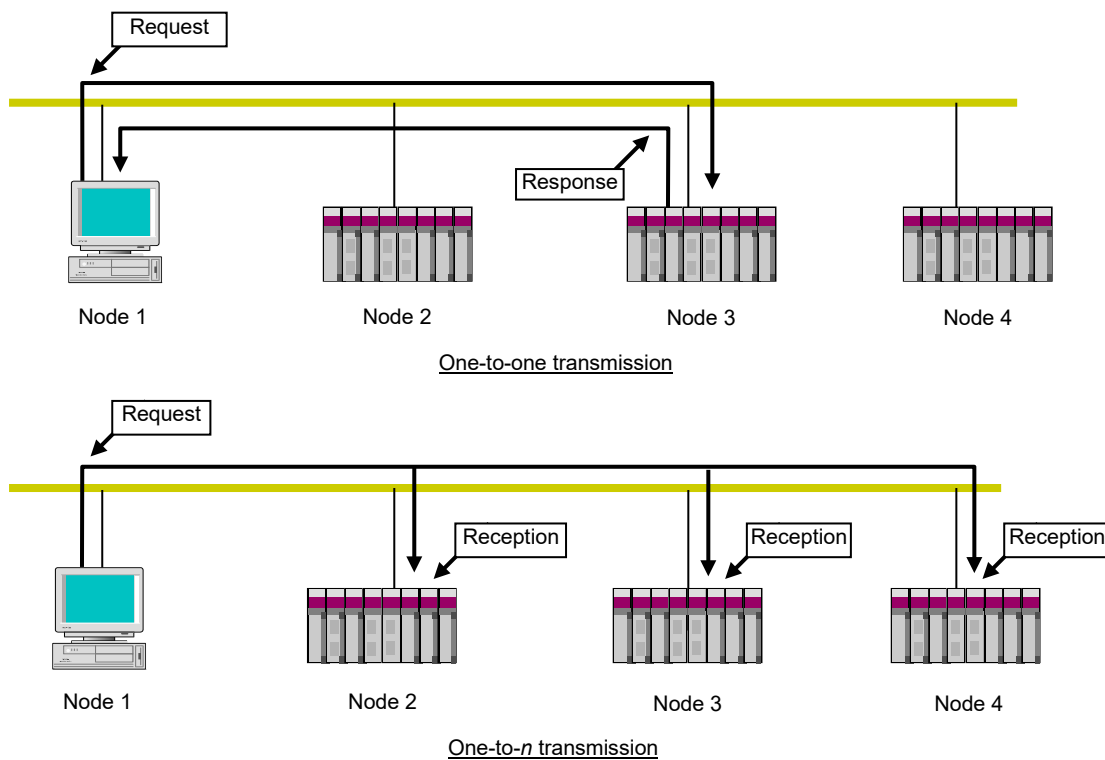


Figure 5-18 Overview of message transmission

5. User's Guide

(2) List of supported messages

Table 5-4 lists the types of messages that are supported by the FL.NET module.

Table 5-4 List of supported message transmissions

Message	Request	Response
Byte block read	N	N
Byte block write	N	N
Word block read	N	Y
Word block write	N	Y
Network parameter read	N	Y
Network parameter write	N	Y
Run/stop directive	N	N
Profile read	N	Y
Communication log data read	N	Y
Communication log data clear	N	Y
Message return	N	Y
Transparent message	N	Y

Y: Supported; N: Not supported

(3) Explanation of message functions

- Byte block read

This message function reads a destination node's virtual address space (32-bit address space) from the network byte by byte in 1-address 8-bit units). This function is not supported by the FL.NET module.

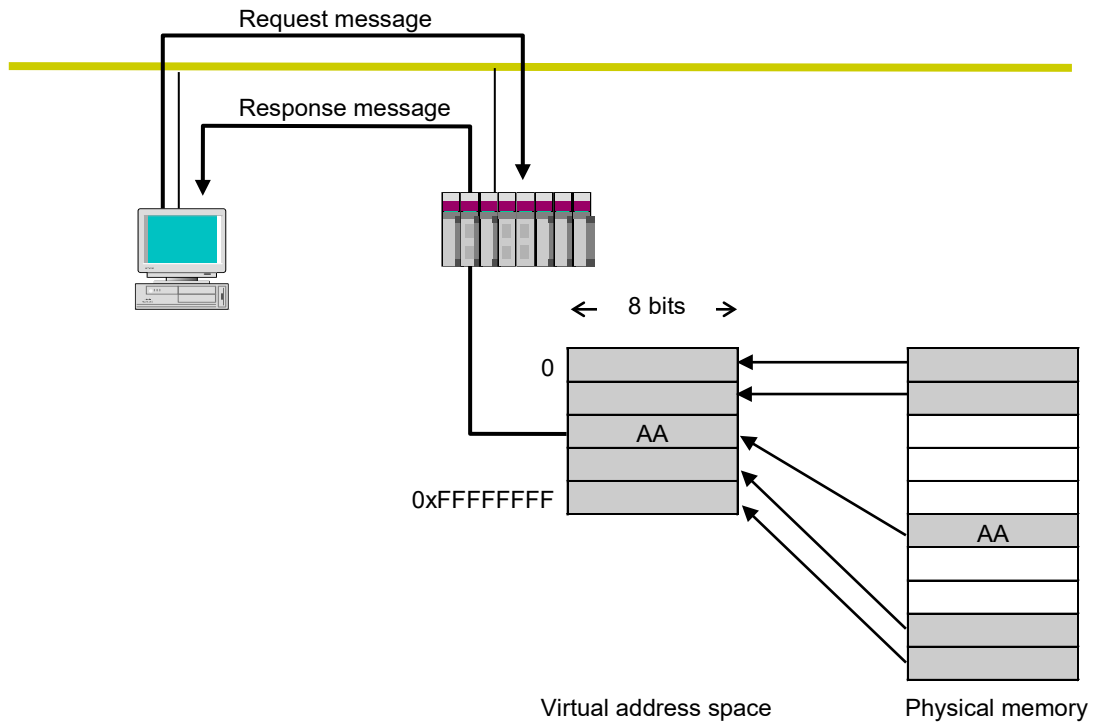


Figure 5-19 Byte block read

- Byte block write

This message function writes to a destination node's virtual address space (32-bit address space) from the network byte by byte (in 1-address 8-bit units). This function is not supported by the FL.NET module.

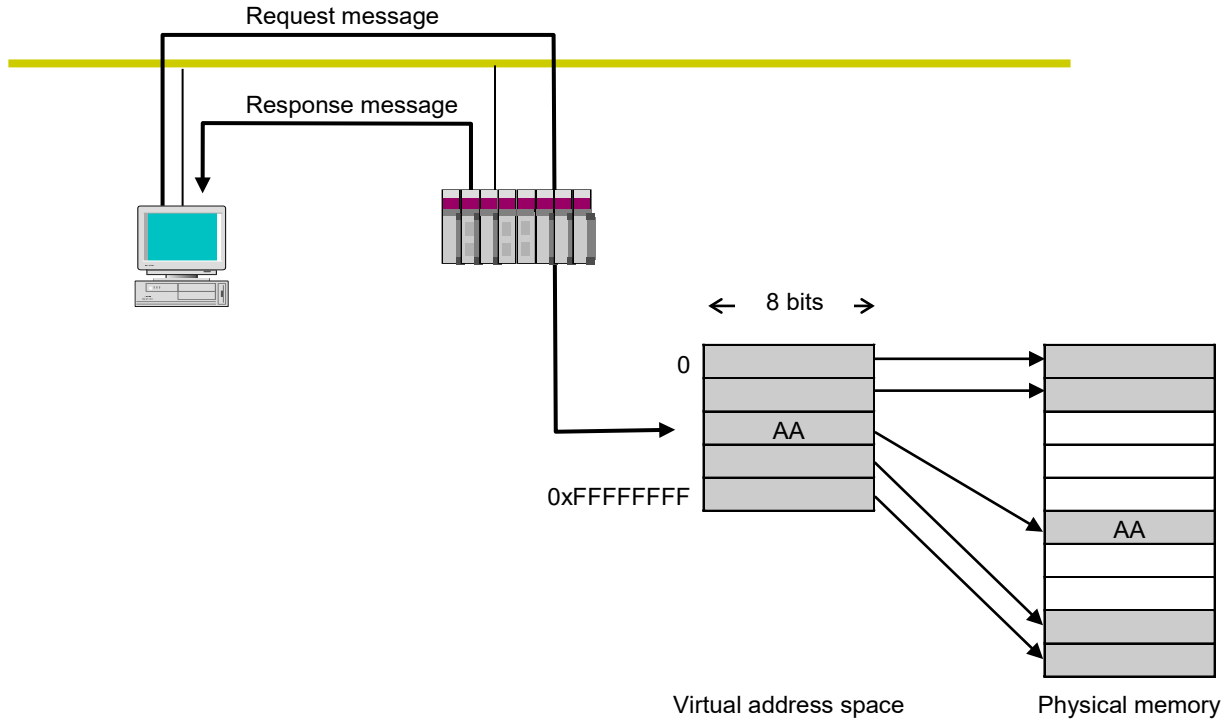


Figure 5-20 Byte block write

- Word block read

This message function reads a destination node's virtual address space (32-bit address space) from the network word by word (in 1-address 16-bit units). For use instructions, see 5.3 *Using the FL.NET module*.

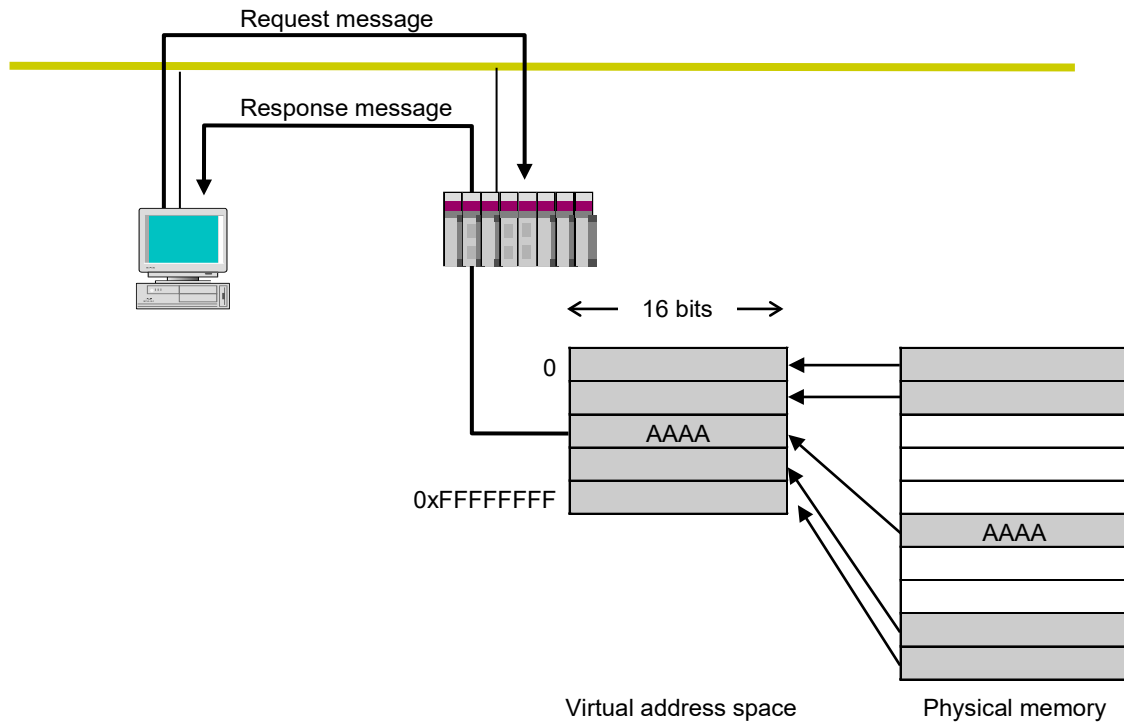


Figure 5-21 Word block read

- Word block write

This message function writes into a destination node's virtual address space (32-bit address space) from the network word by word (in 1-address 16-bit units). For use instructions, see 5.3 *Using the FL.NET module*.

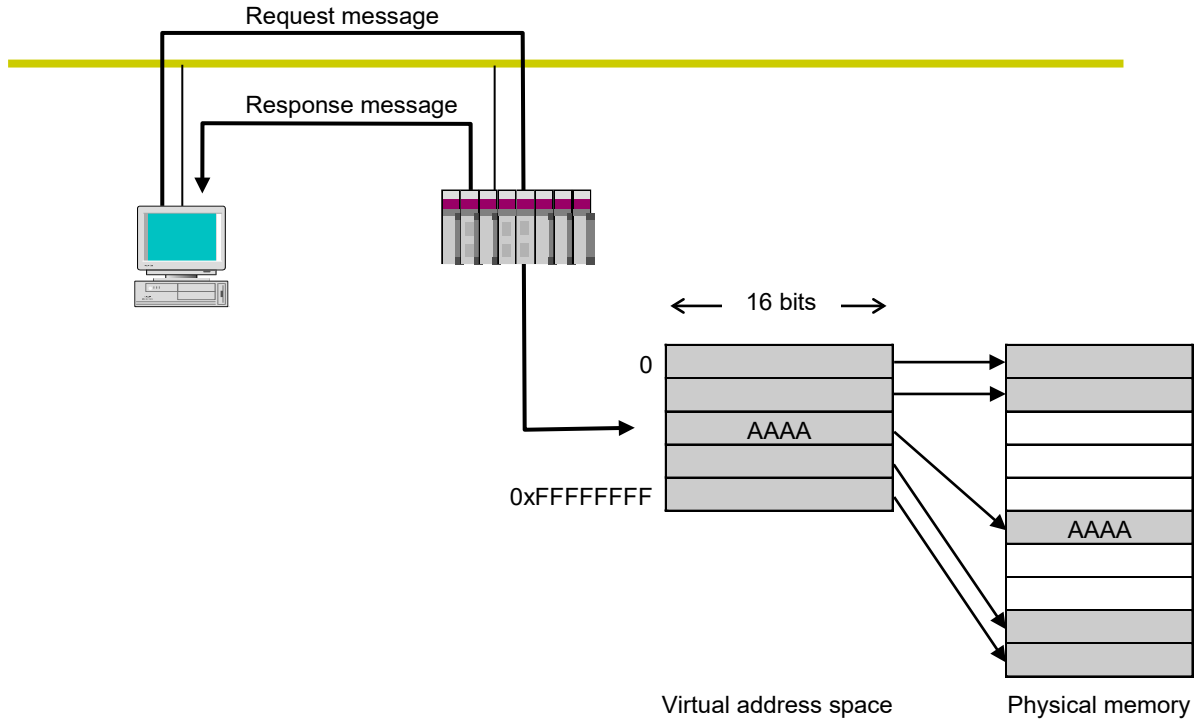


Figure 5-22 Word block write

- Network parameter read
 This message function reads a destination node's network parameter information from the network. Table 5-5 lists the information this function reads. For use instructions, see 5.3 *Using the FL.NET module*.

Table 5-5 Network parameter information

Node number
Vendor name
Manufacturer model
Node name (equipment name)
Common memory address and size
Token monitoring timeout
Refresh cycle time limit
Measured refresh cycle time
Minimum frame interval
Higher layer status
FL-net status
Protocol version

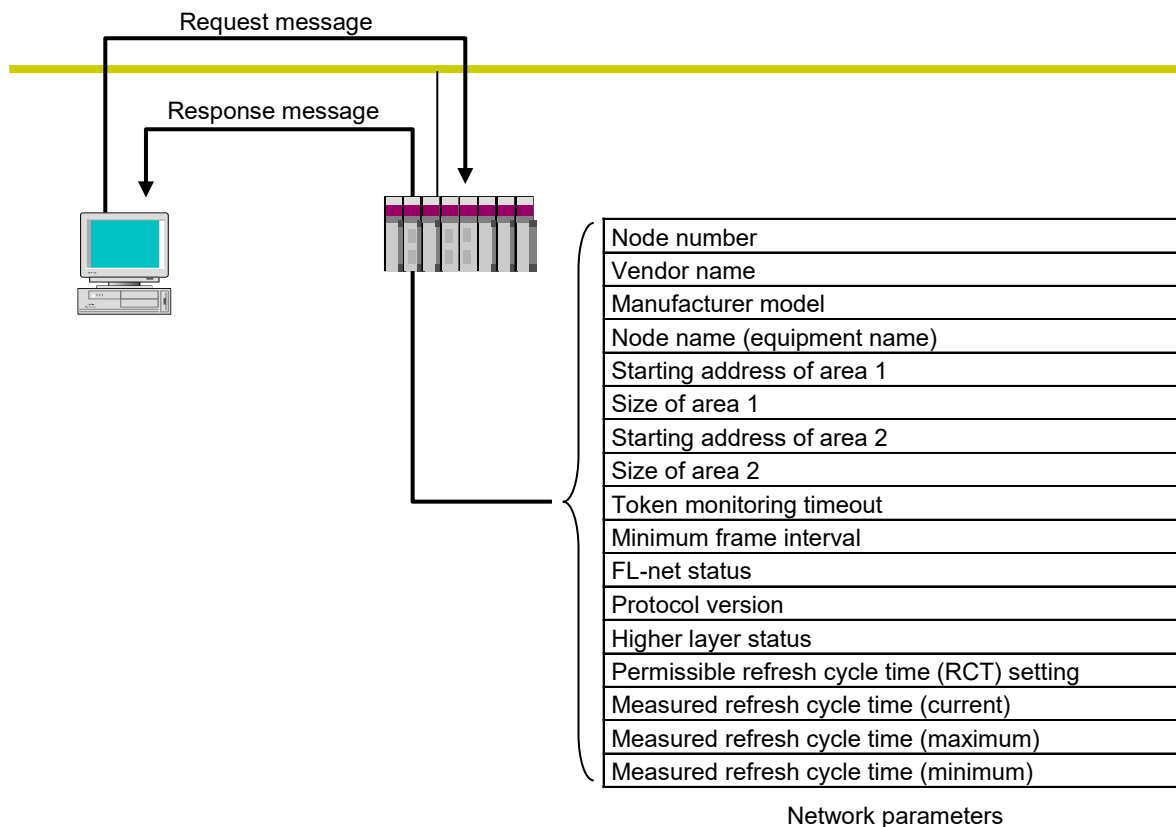


Figure 5-23 Network parameter read

- Network parameter write

This message function changes a destination node's network parameter information from the network. The following information can be changed. For use instructions, see 5.3 *Using the FL.NET module*.

- Node name (equipment name)
- Address and size of the common memory area

If the starting address of area 1 or 2 is outside the common memory, an error occurs.

However, the node name is not checked.

If the address or size of the common memory is changed, the destination node leaves the network and then joins it again. If only the node name is changed, the destination node does not leave the network.

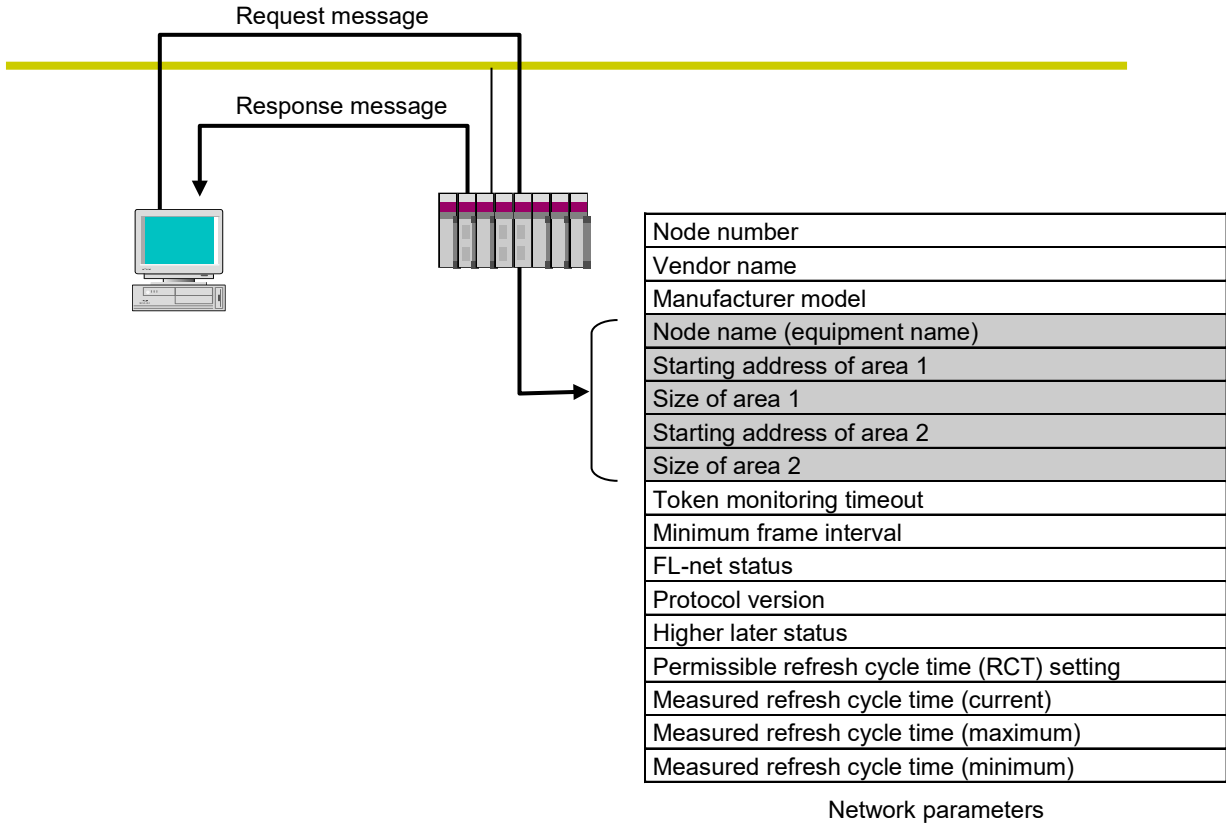


Figure 5-24 Network parameter write

- Run/stop directive

This message function allows the network to remotely run or stop the operation of devices connected to the FL-net network. This function is not supported by the FL.NET module.

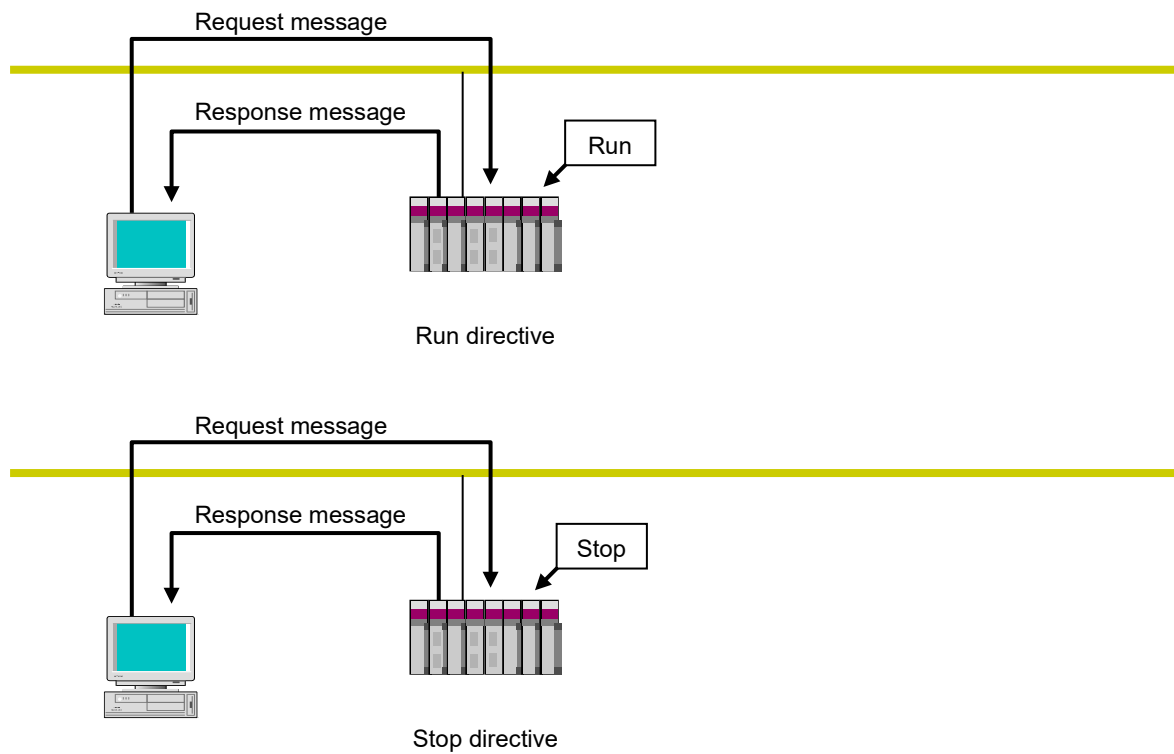
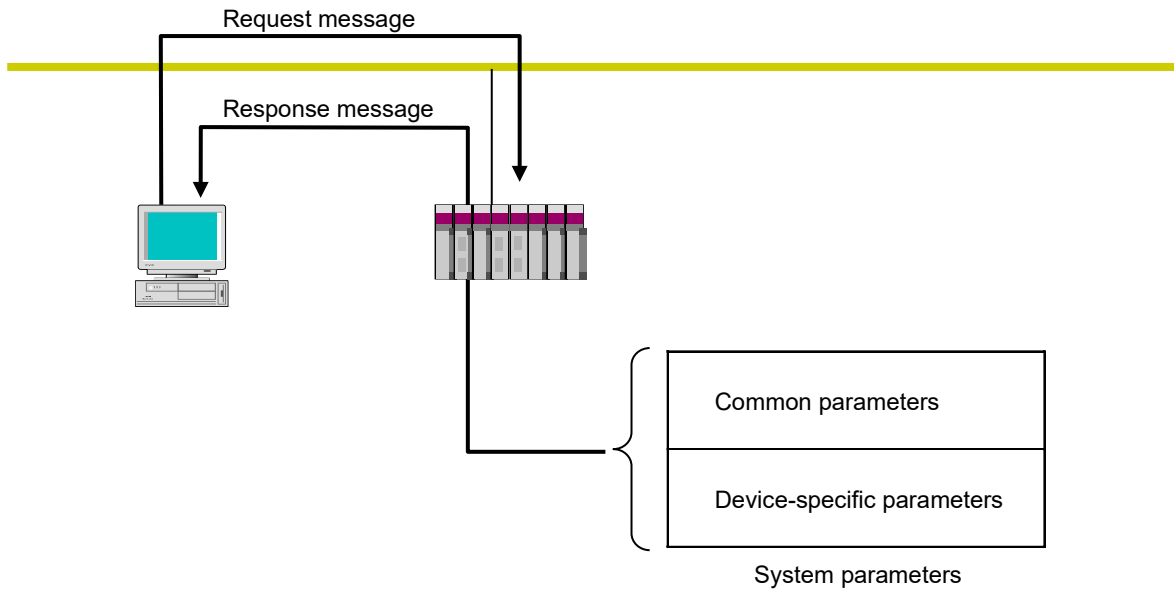


Figure 5-25 Run/stop directive

- Profile read

This message function reads from the network the device profile system parameters representing information about a destination node. Only the associated response is supported by the FL.NET module. The system parameters represent the following information:

- Common parameters (essential)
- Device-specific parameters (optional)



For details about system parameter device profiles, see *Appendix*.

Figure 5-26 Profile read

- Communication log data read
This message function reads a destination node's log information from the network.
For use instructions, see 5.3 *Using the FL.NET module*.

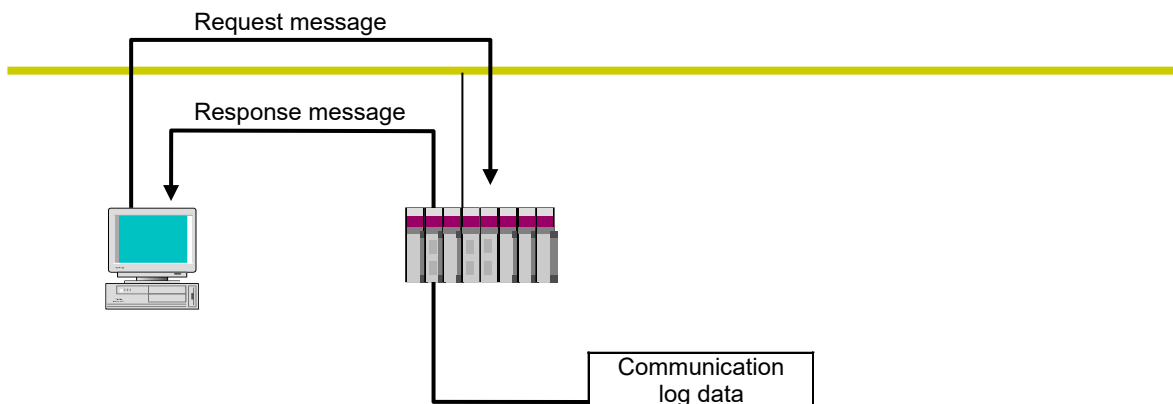


Figure 5-27 Communication log data read

- Communication log data clear
This message function clears a destination node's log information from the network.
For use instructions, see 5.3 *Using the FL.NET module*.

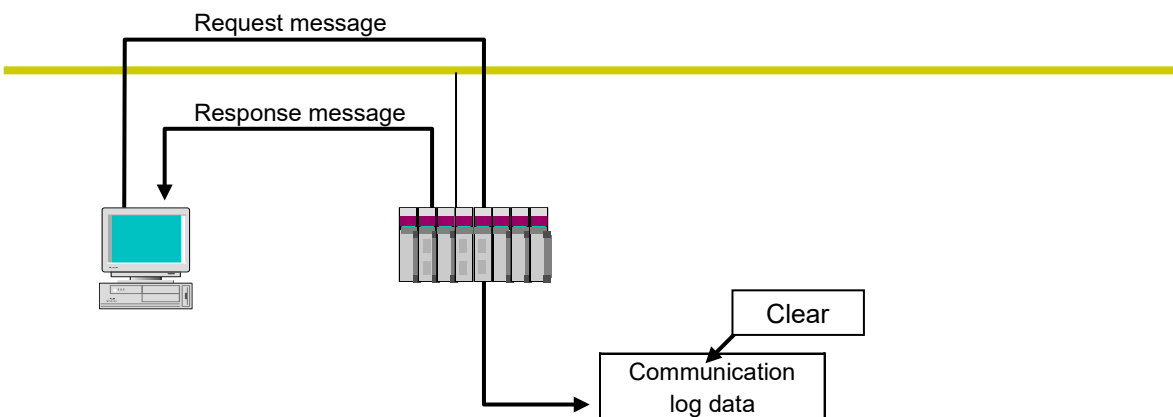


Figure 5-28 Communication log data clear

- Message return
 This message function returns a received message.
 Messages are returned automatically within the FL.NET module.
 For use instructions, see *5.3 Using the FL.NET module*.

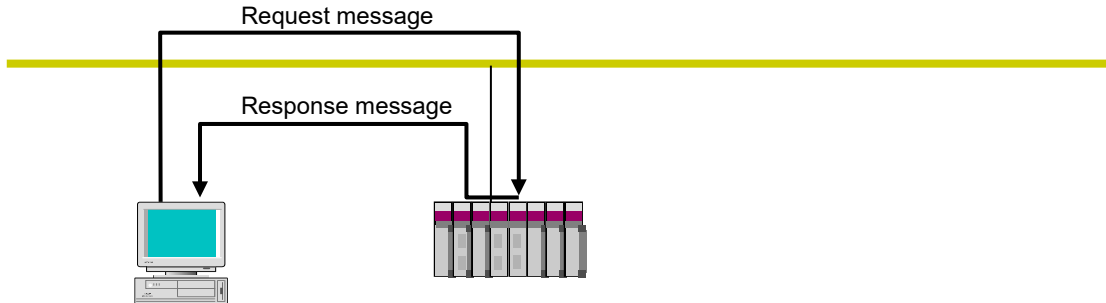


Figure 5-29 Message return function

- Transparent message transmission
 This message function offers a transparent message transmission service to higher layers of FL-net.
 This service notifies a higher layer of FL-net of a received message. Upon receiving the notification, the higher layer of FL-net sends the notification to the user interface level. When the notification arrives at the user interface level, it is necessary to create and return an associated response.
 For use instructions, see *5.3 Using the FL.NET module*.

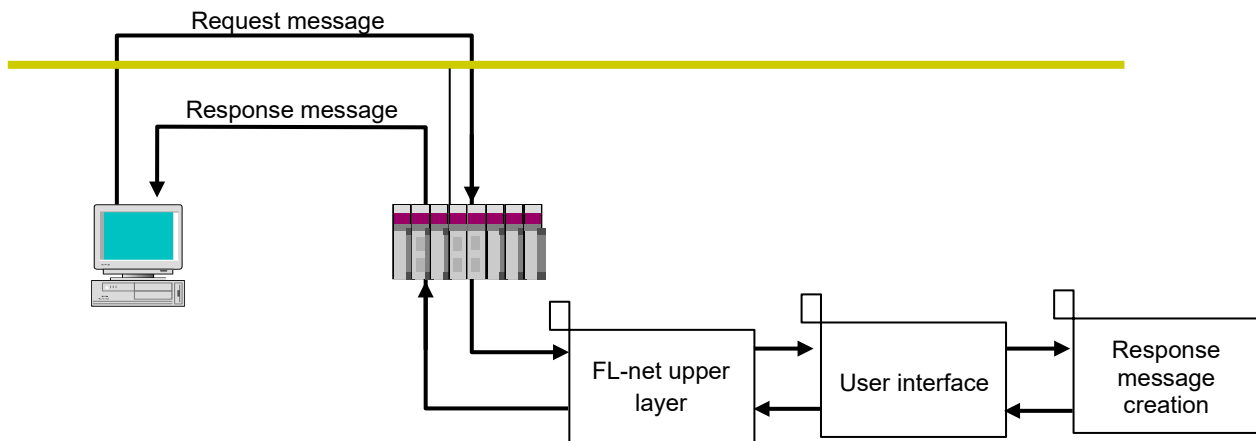


Figure 5-30 Transparent message transmission

5.1.10 Using bit data

The FL.NET module performs bit conversion on the data in common memory area 1 at the time of transmission and reception.

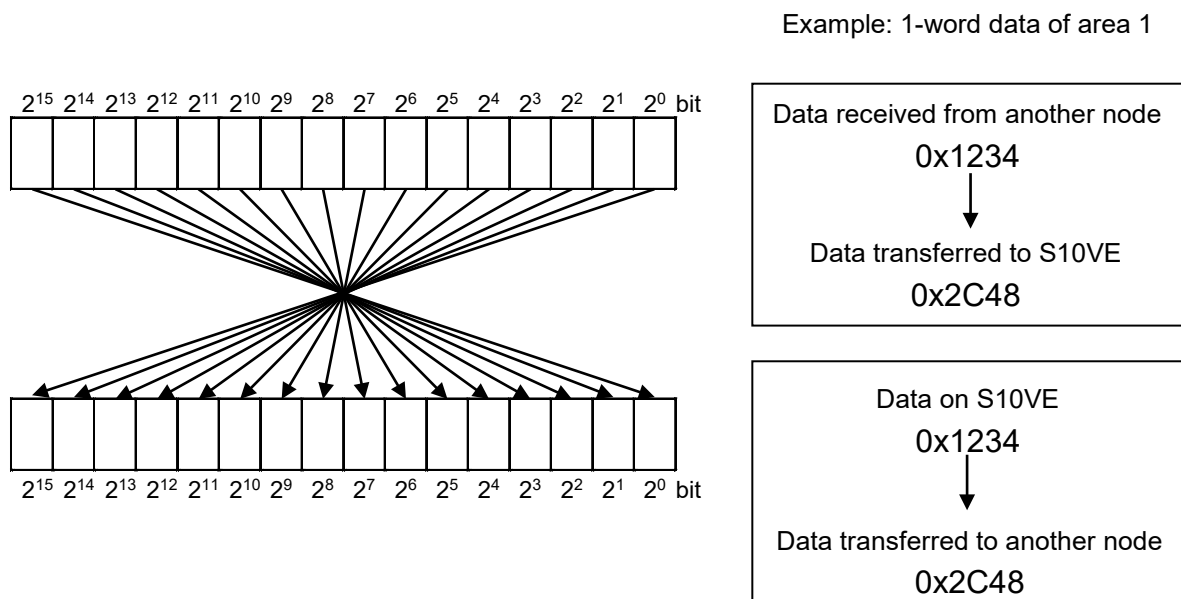


Figure 5-31 Bit conversion of area 1 data

5.1.11 Communication time of the FL.NET module

Each of the optional modules, including the FL.NET module, updates I/O data in its specific communication time in regular cycles. The CPU module uses ladder programs with the I/O data updated by the optional modules to perform control. Because ladder programs are repeatedly executed in cycles (sequence cycles), the communication time of each optional module must be, at most, half the length of the sequence cycle to prevent I/O data from going missing.

The communication time of an optional module is determined based on its specified communication word count and the number of optional modules mounted in the CPU unit. Calculate the communication time (refresh cycle time) of the FL.NET module as shown in the following section. For details about how to calculate the communication time of optional modules other than the FL.NET module, see the manual for each option module.

The following is a legend for the symbols used in the formulas:

J: Total number of optional modules (except the relevant FL.NET module) mounted in the CPU unit

S: Number of words transmitted from the self-node

R: Number of words received by the self-node from the previous node

F: Result of dividing (S-1) by 512

(e.g., F = 0 when S = 512; F = 1 when S = 513)

MFT: Minimum frame interval (ms)

For the settings, see *5.3.1 Setting link parameters*.

- **Communication time (refresh cycle time) (ms):** Sum of the token hold times of all nodes

$$\text{Communication time (refresh cycle time)} = \text{Token hold time of node 1} + \text{token hold time of node 2} + \dots + \text{token hold time of node n}$$

- **Token hold time (ms):** Time for which the token was held by the self-node

$$\text{Token hold time} = 1.1 + (0.001 + 0.0013 \times J) \times S + (0.001 + 0.0013 \times J) \times R + (MFT + 0.2) \times F + MFT$$

5.2 Setting up the FL.NET module

5.2.1 Startup procedure

The following describes the startup procedure for the FL.NET module:

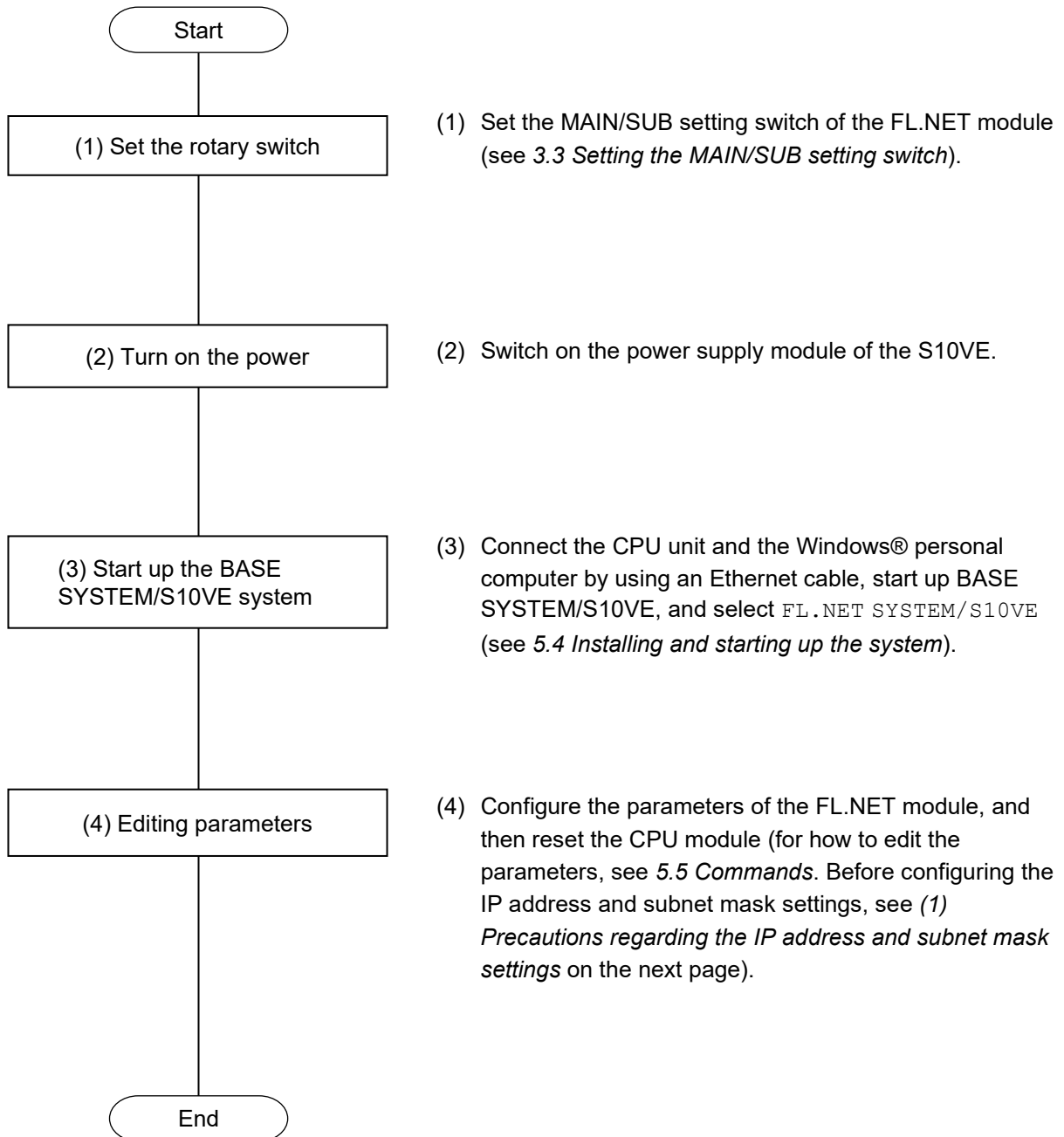


Figure 5-32 Startup procedure

(1) Precautions regarding the IP address and subnet mask settings

For the FL.NET module, you can freely set the IP address and subnet mask. However, you must comply with the FL-net protocol. It is therefore recommended that you use a Class C IP address, and set the network address to 192 . 168 . 250 . Also, make sure that the subnet mask is set to 255 . 255 . 255 . 0 . To set the IP address, start the FL.NET SYSTEM/S10VE setup tool and click the **Node setting** tab (see Figure 5-33).

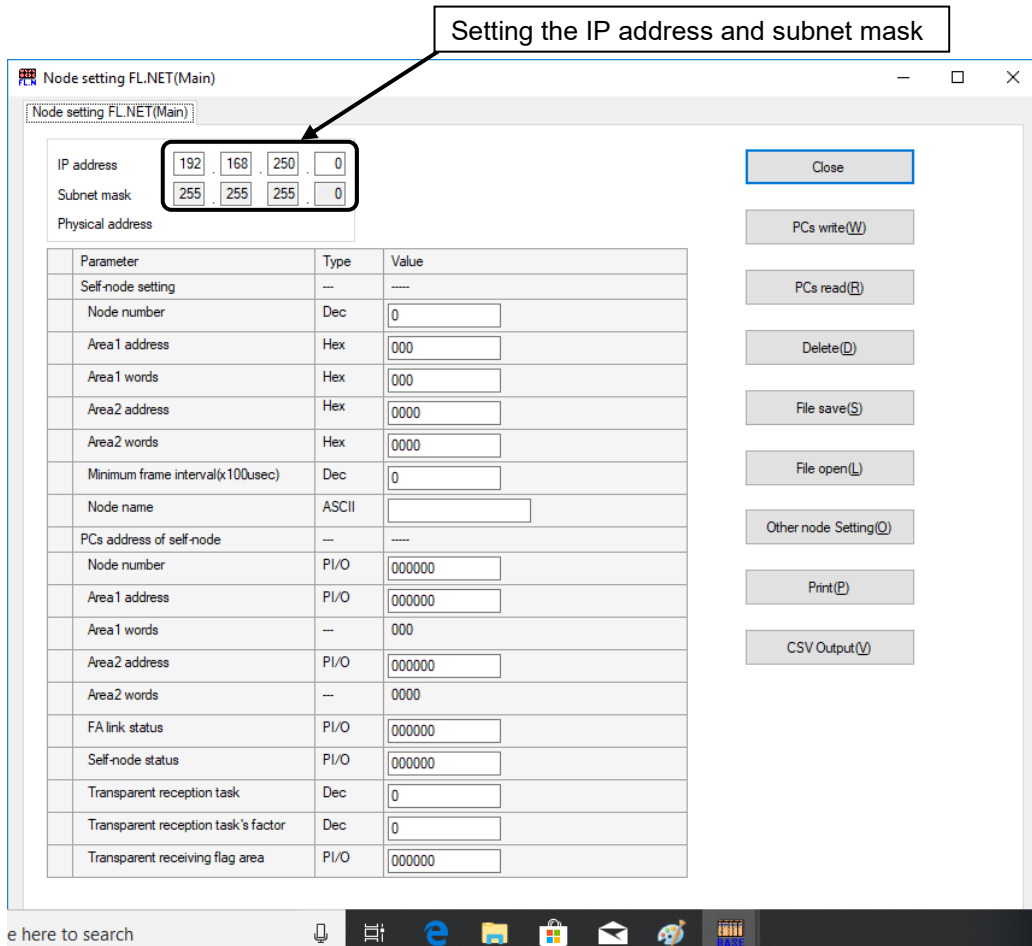


Figure 5-33 [Node setting] tab

- Note 1: If you set anything other than a Class C IP address, an error occurs indicating that the specified IP address is outside the valid range.
- Note 2: If you specify 0 or 255 for the host number portion, an error occurs indicating that the specified IP address is outside the valid range.
- Note 3: If you start the FL.NET SYSTEM/S10VE setup tool while the FL.NET module has not been mounted, click the **Note setting** tab of the setup tool, and click the **PCs read** button to receive the parameter values, the physical address is shown as - : - : - : - : - : - .
- Note 4: When referencing the physical address, mount the FL.NET module and set the IP address. The IP address and subnet mask can be referenced even if the FL.NET module is not mounted.
- Note 5: If the IP address is not set, the ERR LED on the FL.NET module blinks.

5.3 Using the FL.NET module

5.3.1 Setting link parameters

Link parameters are values used to set up the FL.NET module's common memory area for each node. Use the FL.NET SYSTEM/S10VE setup tool to set the link parameters.

To specify the values of the link parameters, click the **Node setting** tab and enter the fields shown in the self-node settings (see 5.5.2 *Specifying the self-node information settings*).

Table 5-6 shows the ranges of values you can enter for the parameters.

Table 5-6 Ranges of values that can be entered for link parameters

Input item	Description	Valid setting range
Node number	This number is used to identify nodes for FL-net communication purposes.	1 to 254
Minimum frame interval	This value (specified in units of 100 us) represents the minimum time intervals at which frames can be consecutively transmitted, or the minimum time interval between the instant at which the token addressed to the self-node is received and the instant at which frame transmission starts.	0 to 50
Area 1 address	This value specifies the starting address for the area 1 transmission area. Enter a hexadecimal number.	0 to 0x1FF (511)
Area 1 word count	This value specifies the size of the area 1 transmission area. Enter a hexadecimal number.	0 to 0x200 (512)
Area 2 address	This value specifies the starting address for the area 2 transmission area. Enter a hexadecimal number.	0 to 0x1FFF (8191)
Area 2 word count	This value specifies the size of the area 2 transmission area. Enter a hexadecimal number.	0 to 0x2000 (8192)
Node name	This value specifies the self-node name.	String of no more than 10 single-byte alphanumeric characters

For detailed procedure descriptions, see 5.5 *Commands*.

Note 1: If the minimum frame interval is set to 0, the frame transmission interval will automatically be changed to its minimum value at each device.

Note 2: When no common memory transmission is to be initiated from the self-node, set both the starting address and size to 0.

5. User's Guide

Example: When the following link parameter settings are specified, the FL.NET module operates as shown in the subsequent figure:

Node number	: 1	Node number	: 2
Area 1 address	: 0x000	Area 1 address	: 0x000
Area 1 word count	: 0x000	Area 1 word count	: 0x000
Area 2 address	: 0x0000	Area 2 address	: 0x0040
Area 2 word count	: 0x0040	Area 2 word count	: 0x0040
Minimum frame interval	: 30	Minimum frame interval	: 30
Node name	: CPU01	Node name	: CPU02

Node number	: 3
Area 1 address	: 0x000
Area 1 word count	: 0x000
Area 2 address	: 0x0080
Area 2 word count	: 0x0040
Minimum frame interval	: 30
Node name	: CPU03

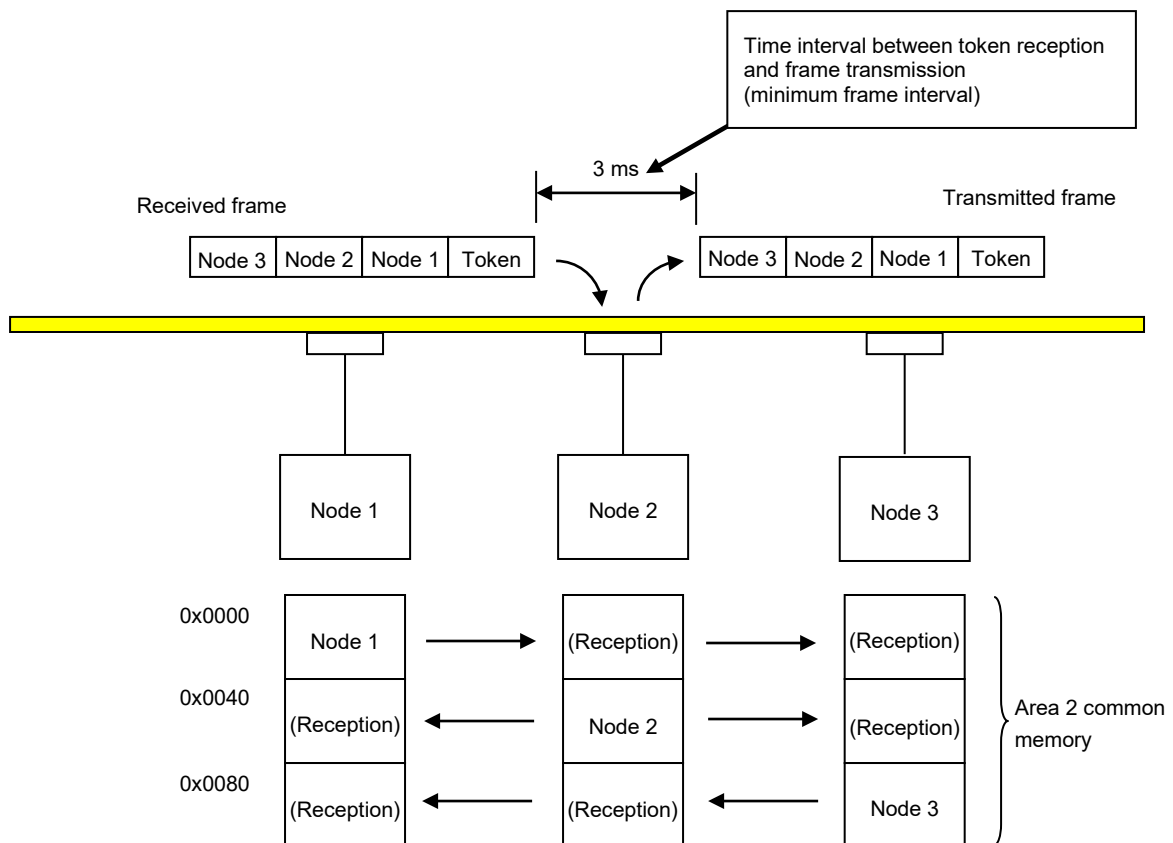


Figure 5-34 Example of FL-net module operation

Data is exchanged using the area 2 common memory area as shown in the preceding figure.

5.3.2 Allocating CPU memory

The FL.NET module allocates a common memory area in the CPU memory.

Data received by the FL.NET module is transferred to the allocated area in the CPU memory.

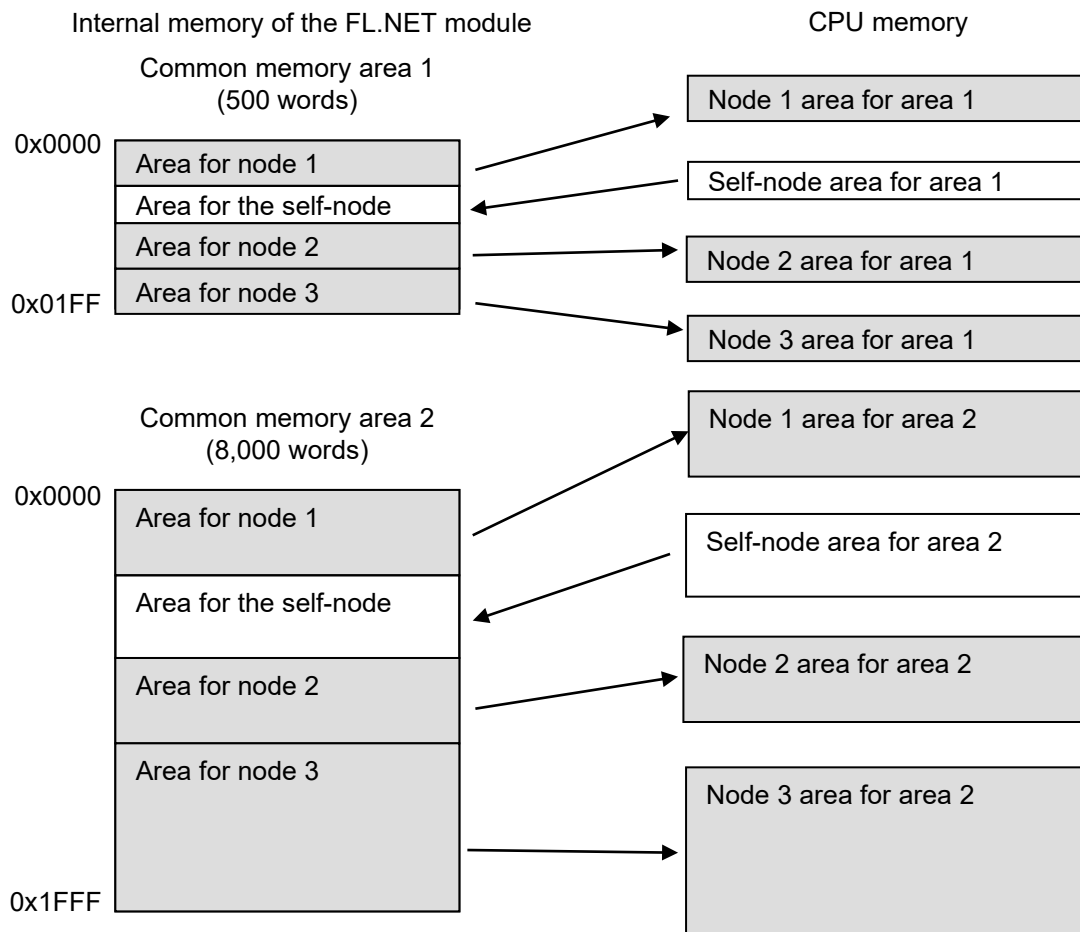


Figure 5-35 Overview of common memory data transfer with the FL.NET module

Configure the settings for allocating a common memory area in the CPU memory in the FL.NET SYSTEM/S10VE setup tool.

For details about the procedures to be performed in various configuration windows, see *5.5 Commands*.

Table 5-7 shows the addresses that are available for allocating a common memory area in the CPU memory.

Table 5-7 Address ranges where a common memory area can be allocated in CPU memory

Specifiable register	Allocatable range of addresses	Remarks
XW0000 to XWFFF0	0x0044 1000 to 0x0044 2FFE	External input
YW0000 to YWFFF0	0x0044 3000 to 0x0044 4FFE	External output
JW000 to JWFF0	0x000E 0200 to 0x000E 03FE	Transfer register
QW0000 to QWFFF0	0x0044 5000 to 0x0044 6FFE	Receive register
GW000 to GWFF0	0x000E 0800 to 0x000E 09FE	Global link register
RW000 to RWFF0	0x000E 0C00 to 0x000E 0DFE	Internal register
EW0400 to EWFFF0	0x0044 9080 to 0x0044 AFFE	Event register
MW0000 to MWFFF0	0x0044 7000 to 0x0044 8FFE	Internal register
DW000 to DWFFF	0x0006 1000 to 0x0006 2FFE	Function data register
FW000 to FWBFF	0x000E 2000 to 0x000E 37FE	Function work register
LBW0000 to LBWFFF0	0x0041 2000 to 0x0041 3FFE	Work register
LWW0000 to LWWFFFF	0x0041 2000 to 0x0041 3FFE	Word work register
LXW0000 to LXW3FFF	0x004A 0000 to 0x004A 7FFE	Word work register

(1) Allocating CPU memory for the self-node

Using the FL.NET SYSTEM/S10VE setup tool, display the **Node setting FL.NET (Main)** tab (see Figure 5-36). In the section for allocation to the self-node's PCs, enter the addresses for the PI/O to be allocated as the self-node's CPU memory area. Then, click the **PCs write** button. The entered settings are saved in the FL.NET module.

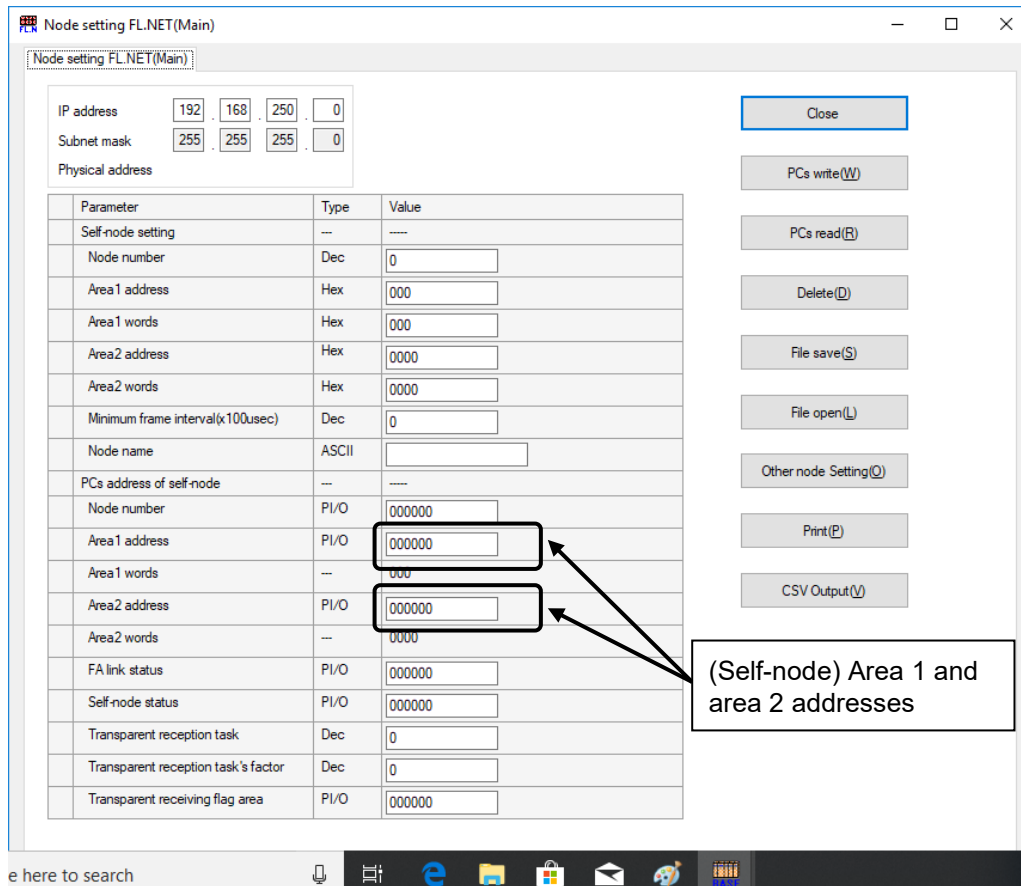


Figure 5-36 [Node setting FL.NET (Main)] tab

Table 5-8 Allocating CPU memory for the common memory of the self-node

Input item	Description	Specifiable range
Area 1 address (Self-node's PCs allocation)	This value specifies the starting address for the allocation of area 1's common memory and CPU memory.	See Table 5-7 Address ranges where a common memory area can be allocated in CPU memory.
Area 2 address (Self-node's PCs allocation)	This value specifies the starting address for the allocation of area 2's common memory and CPU memory.	See Table 5-7 Address ranges where a common memory area can be allocated in CPU memory.

5. User's Guide

Note 1: The area 1 word count (area 2 word count) for allocating the self-node PCs is the same as the link parameter area 1 word count (area 2 word count).

Note 2: The allocation address specified for the self-node CPU memory must not be a duplicate of any allocation address specified for the CPU memory of other nodes.

Note 3: CPU memory cannot be allocated over more than one PI/O range.

Example: The following is an example of an invalid setting:

When the word count (size) setting for area 1 is 3 words, `RWFF0` cannot be specified for CPU memory allocation (because `RWFF0` exceeds the RW area limit).

Note 4: If the link parameter area 1 word count (area 2 word count) is set to 0, the area 1 address (area 2 address) for allocating PCs cannot be set.

(2) Setting the CPU memory area for other nodes

Start the FL.NET SYSTEM/S10VE setup tool, click the **Other nodes reception setting FL.NET (Main)** tab (Figure 5-37), and specify the necessary settings so that the FL.NET module can transfer data received from other nodes to the CPU memory.

Data received from various nodes is transferred to the area allocated in the CPU memory. If the CPU memory's area word count differs from the common memory size setting (link parameter area word count), the FL.NET module transfers data from the common memory. The amount of this data transfer is determined by the number of words specified by the CPU memory area word count setting.

To acquire all data received by the common memory, ensure that the specified size is equal to the common memory setting (link parameter setting) for the other nodes.

For the range of values you can enter for each parameter, see the following figure.

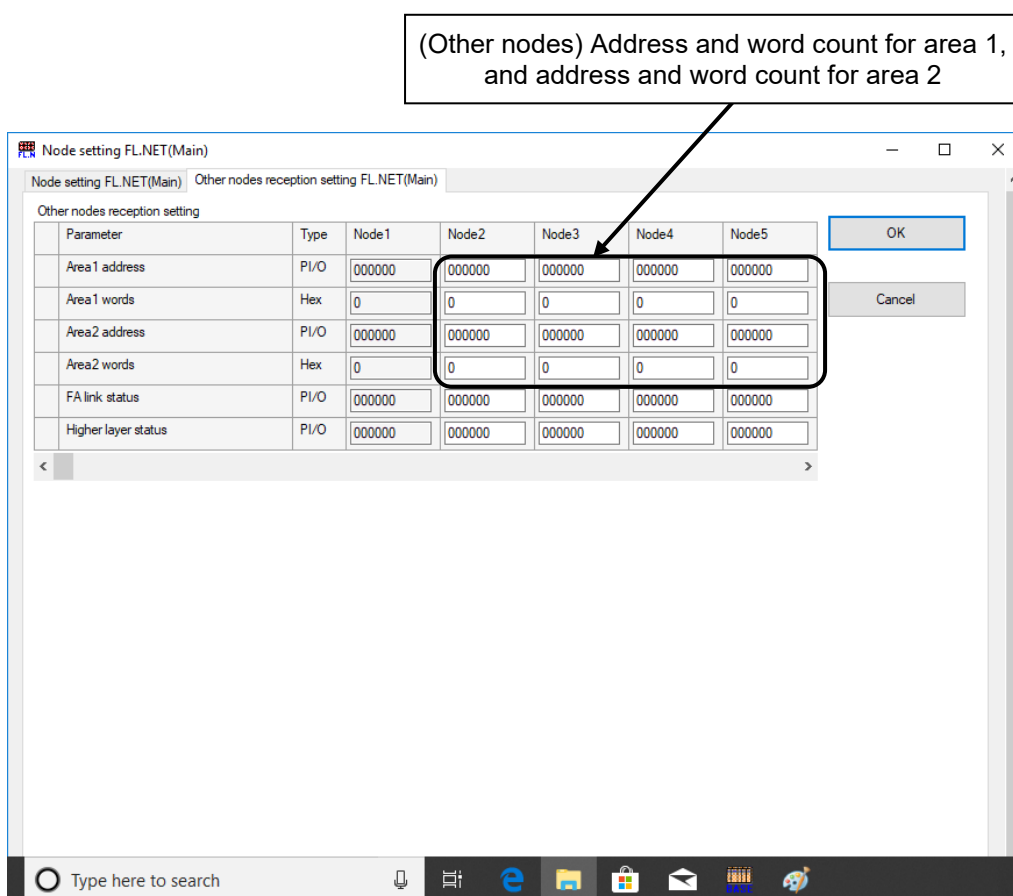


Figure 5-37 [Other nodes reception setting FL.NET (Main)] tab

Table 5-9 Allocating CPU memory for the common memory of other nodes

Input item	Description	Specifiable range
Area 1 address	This value specifies the starting address of the area 1 reception area for the other node.	See Table 5-7 Address ranges where a common memory area can be allocated in CPU memory.
Area 1 word count	This value specifies the size (in number of words) of the area 1 reception area for the other node. Enter a hexadecimal number.	0 to 0x200 (512)
Area 2 address	This value specifies the starting address of the area 2 reception area for the other node.	See Table 5-7 Address ranges where a common memory area can be allocated in CPU memory.
Area 2 word count	This value specifies the size (in number of words) of the area 2 reception area for the other node. Enter a hexadecimal number.	0 to 0x2000 (8192)

Note 1: The specified area in the reception settings of other nodes must not be a duplicate of the area allocated in the self-node's CPU memory area.

Note 2: CPU memory cannot be allocated over more than one PI/O range.

Example: The following is an example of an invalid setting:

When the word count (size) for area 1 is 3, RWFF0 cannot be specified as the area 1 address (because RWFF0 exceeds the RW area range).

Notice

- If the CPU memory's area word count differs from the common memory size setting (link parameter area word count), the FL.NET module transfers data from the common memory. The amount of this data is determined by the number of words specified by the CPU memory area word count setting.
- If there is a node not specified in the **Other nodes reception setting FL.NET (Main)** tab, the FL.NET module cannot transfer the data received in its common memory to the CPU memory allocated in the CPU module.

(3) Referencing the self-node status and FA link status

The self-node status and FA link status of the FL.NET module, the FA link status of each node, and the higher layer status flag can be referenced by using the FL.NET SYSTEM/S10VE setup tool.

When you wish to reference various items of status information on S10VE, you can learn the status of participating nodes and the like by setting the transfer area for each flag.

To set the self-node status and FA link status transfer area for the self-node, use the **Node setting FL.NET (Main)** tab (Figure 5-38).

For details about the on-screen information about the self-node status and the FA link status (bit allocation), see *A.4 Contents of each status flag*. For the range of values you can enter for each parameter, see the following figure.

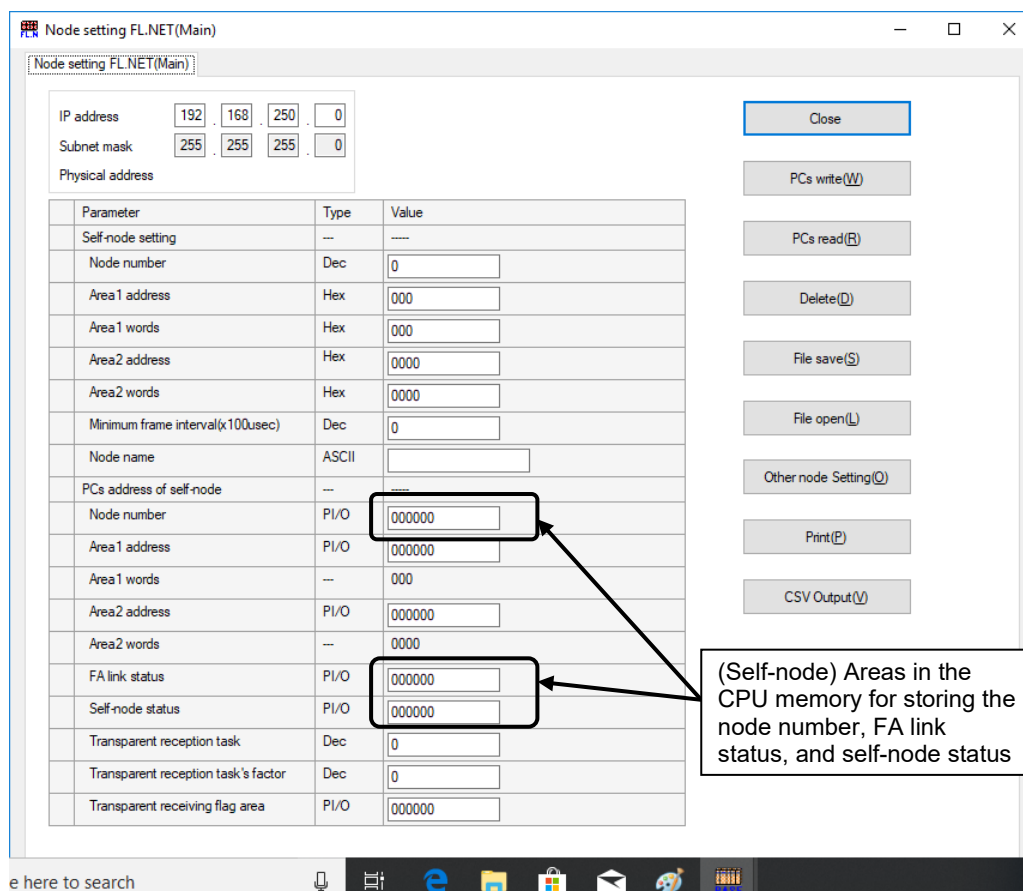


Figure 5-38 [Node setting FL.NET (Main)] tab

Table 5-10 PCs allocation for referencing the self-node status and the FA link status

Input item	Description	Specifiable range
Node number (Self-node PCs allocation)	This value specifies the area to which the self-node number is transferred (the area size is 1 word).	See Table 5-7 Address ranges where a common memory area can be allocated in CPU memory.
FA link status (Self-node PCs allocation)	This value specifies the area to which the FA link status of the self-node is transferred (the area size is 1 word).	See Table 5-7 Address ranges where a common memory area can be allocated in CPU memory.
Self-node status (Self-node PCs allocation)	This value specifies the area to which the self-node status is transferred (the area size is 1 word)	See Table 5-7 Address ranges where a common memory area can be allocated in CPU memory.

Note 1: The register range that can be specified for the self-node area and FA link status area is the same as for the CPU memory allocation area.

Note 2: The specified self-node status area and FA link status area must not be a duplicate of the specified CPU memory allocation area for any other participating nodes or the self-node.

Notice

- When you specify the area for the self-node status, the self-node status flag ((1) Contents of the self-node status flag in A.4 Contents of each status flag) is transferred to the CPU memory allocated as one word of data.
- When you specify the area for the FA link status, the FA link status flag of the self-node ((2) Contents of the FA link status flag in A.4 Contents of each status flag) is transferred to the lower byte of the allocated CPU memory.
- If you do not specify the area for the self-node status or the FA link status, the FL.NET module cannot transfer the self-node status flag or the FA link status flag to the allocated CPU memory.

To specify the areas in the CPU memory for storing the FA link status and the higher layer status flags for each node, use the **Other nodes reception setting FL.NET (Main)** tab (Figure 5-39).

Table 5-11 shows the range of values you can enter for each parameter.

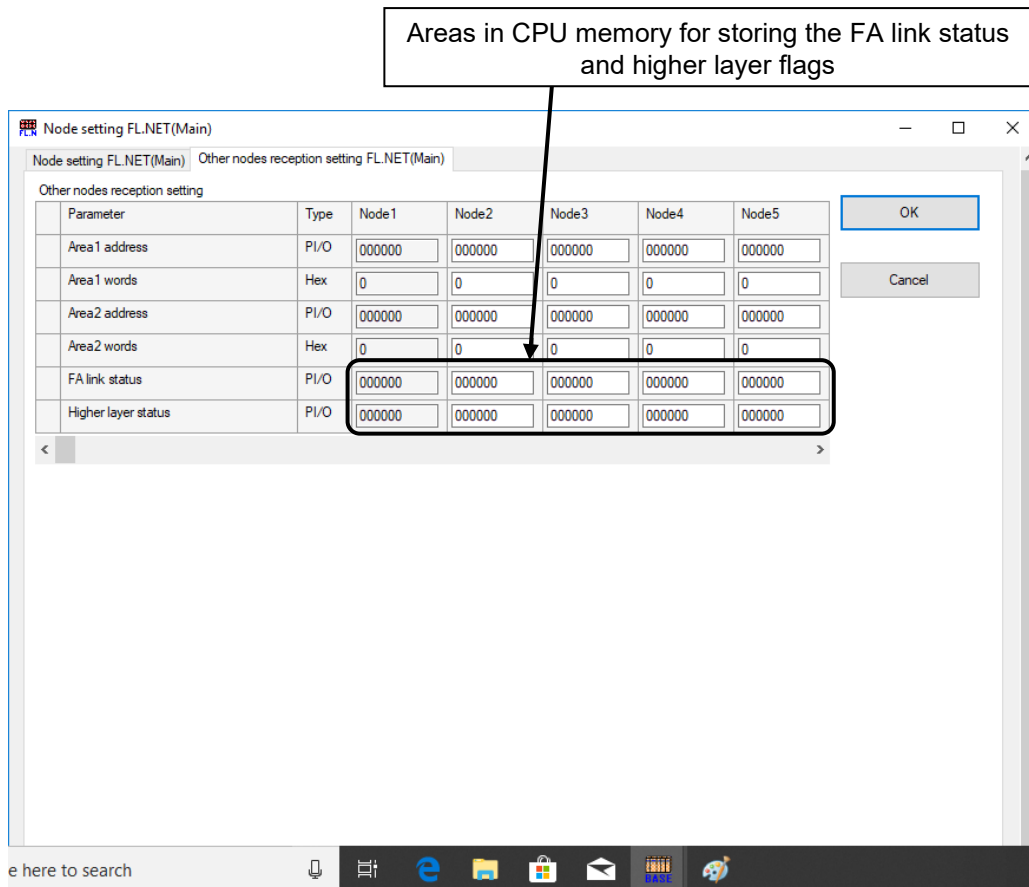


Figure 5-39 [Other nodes reception setting FL.NET (Main)] tab

Table 5-11 PCs allocation for referencing the FA link status and the higher layer status of each node

Input item	Description	Specifiable range
FA link status	This value specifies the area to which the FA link status is transferred (the area size is 1 word).	See Table 5-7 Address ranges where a common memory area can be allocated in CPU memory.
Higher layer status	This value specifies the area to which the higher layer status is transferred (the area size is 1 word).	See Table 5-7 Address ranges where a common memory area can be allocated in CPU memory.

For details about the on-screen information about the FA link status and higher layer status (bit allocation), see *A.4 Contents of each status flag*.

Notice

- The range of register/extension memory addresses that can be specified to the higher layer status area and FA link status area is the same as for the CPU memory allocation area.
- When you specify the area for the higher layer status, the description of the higher layer status flag ((3) *Contents of the higher layer status flag in A.4 Contents of each status flag*) is transferred to the allocated CPU memory of each node as one word of data.
- When you specify the area for the FA link status, the description of the FA link status flag of each node ((2) *Contents of the FA link status flag in A.4 Contents of each status flag*) is transferred to the lower byte of the allocated CPU memory location.
- The specified higher layer status and FA link status area must not be a duplicate of the allocated CPU memory area for another node's reception setting, the allocated CPU memory area for the self-node, or the setting for any other area.
- For a node that is not specified in the **Other nodes reception setting FL.NET (Main)** tab, the FL.NET module cannot transfer that node's higher layer status or FA link status to the allocated CPU memory of other nodes.

5.4 Installing and starting up the system

5.4.1 Notes on installation

FL.NET SYSTEM/S10VE supports the Microsoft® Windows® 7 (64-bit) operating system and Microsoft® Windows® 10 (64-bit) operating system.

FL.NET SYSTEM/S10VE must be installed from the CD of BASE SET/S10VE. The Microsoft Visual C++ 2010 redistributable package (x64) is a prerequisite program for operating BASE SET/S10VE. If you double-click `SETUP.exe` (in BASE SET/S10VE) with this program not installed, the following message appears and installation fails: *The program can't start because MSVCR100.dll is missing from your computer. Try reinstalling the program to fix this problem.* Install the Microsoft Visual C++ 2010 redistributable package (x64) in advance from the Microsoft download center. Before you attempt to install FL.NET SYSTEM/S10VE, make sure that BASE SYSTEM/S10VE (program product model: S-7898-38) has been installed. If BASE SYSTEM/S10VE has not been installed, install it. FL.NET SYSTEM/S10VE cannot be installed if BASE SYSTEM/S10VE has not been installed.

The following runtime package must be installed before you can operate FL.NET SYSTEM/S10VE. If this runtime package has not been installed, install it in advance from the Microsoft download center.

- Microsoft .NET Framework 4

If you attempt to start BASE SYSTEM/S10VE with the Microsoft .NET Framework 4 not installed, an error message indicating a .NET Framework initialization error appears and startup fails

- If you attempt to start BASE SET/S10VE with the Microsoft Visual C++ 2010 redistributable package (x64) not installed, the following error message appears during startup: *The program can't start because MSVCR100.dll is missing from your computer. Try reinstalling the program to fix this problem.* Then, BASE SET/S10VE terminates abnormally.
- If you attempt to start BASE SET/S10VE with the Microsoft .NET Framework 4 not installed, an error message indicating a .NET Framework initialization error appears and startup fails.

Notice

- To install or uninstall FL.NET SYSTEM/S10VE, use the administrator account. Using the standard account might result in incorrect installation or uninstallation.
- Before installing FL.NET SYSTEM/S10VE, terminate all Windows® programs, including resident programs in the memory (such as virus monitoring software). Installing FL.NET SYSTEM/S10VE while any of these programs are running might cause an error. If an error occurs, uninstall the tool being installed, terminate all Windows® programs, and then reinstall the tool. For details about the uninstallation procedure, see *5.4.3 Uninstallation*.
- Do not specify, as the FL.NET SYSTEM/S10VE installation folder, any of the following folders protected by User Account Control:
 - Program file folder (for example, C:\Program Files)
 - System root folder (for example, C:\Windows)
 - System drive root folder (for example, C:\)
 - Program data folder (for example, C:\ProgramData)

5.4.2 Installation

This subsection describes how to install FL.NET SYSTEM/S10VE.

Before installation, make sure that you are logged on as an administrator account.

- (1) To install FL.NET SYSTEM/S10VE, in the CD (BASE SET/S10VE), double-click `SETUP.exe`.
- (2) When you double-click `SETUP.exe`, the following message (Figure 5-40) might appear. If this happens, click the **Yes** button to execute the setup program.

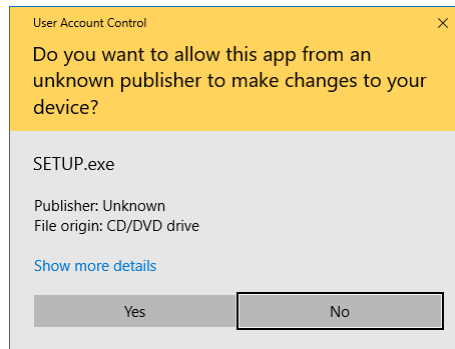


Figure 5-40 User Account Control message

- (3) The Setup window appears (Figure 5-41). Select **FL.NET SYSTEM/S10VE** and click the **Install** button.

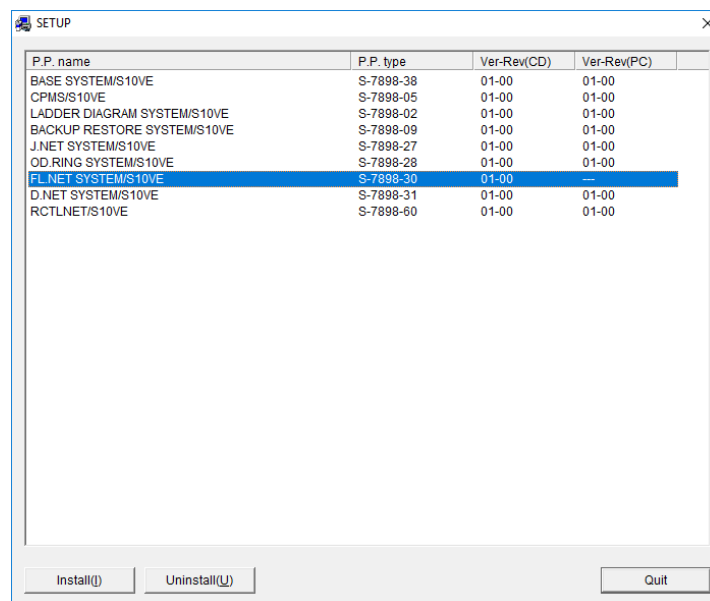


Figure 5-41 SETUP window

- (4) The window for choosing whether to install FL.NET SYSTEM/S10VE appears (Figure 5-42). Click the **Next** button.

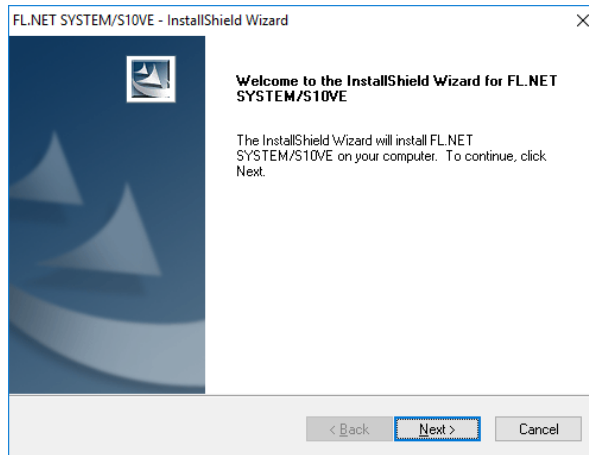


Figure 5-42 Window for choosing whether to install FL.NET SYSTEM/S10VE

If you click the **Cancel** button, a message appears, asking whether you want to cancel setup. Click either the **Yes** or **No** button.

Yes button: Installation of FL.NET SYSTEM/S10VE is canceled.

In the InstallShield Wizard Completion window, click the **Complete** button.

No button: Installation of FL.NET SYSTEM/S10VE is resumed.

- (5) The Setup Status window appears.
- (6) When you complete the installation, the InstallShield Wizard Completion window appears. Click the **Complete** button.

Notice

- FL.NET SYSTEM/S10VE cannot be installed on a per-user basis. To install FL.NET SYSTEM/S10VE successfully, you must first log on to the system with an administrator account.

FL.NET SYSTEM/S10VE might not be installed properly in any of the following cases: 1)

Administrator permissions are acquired by using User Account Control# from a standard user account,

2) The administrator account was created from a standard user account by using User Account Control.

In this case, log on with the administrator account that was first created on your computer, and then reinstall FL.NET SYSTEM/S10VE.

If you log on with a user account other than that used for installing FL.NET SYSTEM/S10VE, the installed program might not appear in the program menu. In this case, log off and log on again with the administrator account that was first created on your computer, uninstall the installed program, and then install the program again.

When you want to create a new account, log on with an administrator account without using User Account Control.

#: User Account Control is a Microsoft Windows feature that temporarily grants administrative rights to standard user accounts.

5.4.3 Uninstallation

This subsection describes how to uninstall FL.NET SYSTEM/S10VE.

To uninstall FL.NET SYSTEM/S10VE, log on as an administrator account and perform the following:

- (1) Click **Start** and then **Control Panel**. Click **Uninstall a program** and double-click FL.NET SYSTEM/S10VE.

The message Do you want to completely remove the selected application and all of its features? appears (Figure 5-43). Click either the **Yes** or **No** button.

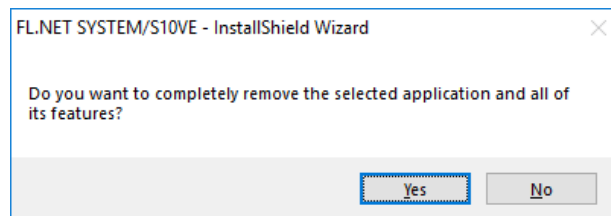


Figure 5-43 Window for choosing whether to uninstall FL.NET SYSTEM/S10VE

Yes button: FL.NET SYSTEM/S10VE is uninstalled.

No button: Uninstallation of FL.NET SYSTEM/S10VE is canceled.

- (2) When the uninstallation is completed, a message appears indicating that uninstallation is complete. Click the **Complete** button.

5.4.4 Reinstallation

To reinstall FL.NET SYSTEM/S10VE, first make sure that FL.NET SYSTEM/S10VE has been uninstalled.

For the uninstallation procedure, see *5.4.3 Uninstallation*. For the installation procedure, see *5.4.2 Installation*.

5.4.5 Starting up the system

To start up FL.NET SYSTEM/S10VE, perform the following procedure:

- (1) Start BASE SYSTEM/S10VE.

Start BASE SYSTEM/S10VE. Create or select the target project and open it. For details about how to start BASE SYSTEM/S10VE or how to create or select a project, see the *S10VE User's Manual General Description* (manual number SEE-1-001).

- (2) Select commands.

- (a) To display the FL.NET module parameter settings or RAS information:

From the **Program** menu of BASE SYSTEM/S10VE, click **Setup Tool**.

The Setup Tool window appears. Select **FL.NET SYSTEM/S10VE** and click the **Execute** button.

The FL.NET module selection window appears (Figure 5-44). For what to do next, see 5.5 *Commands*.

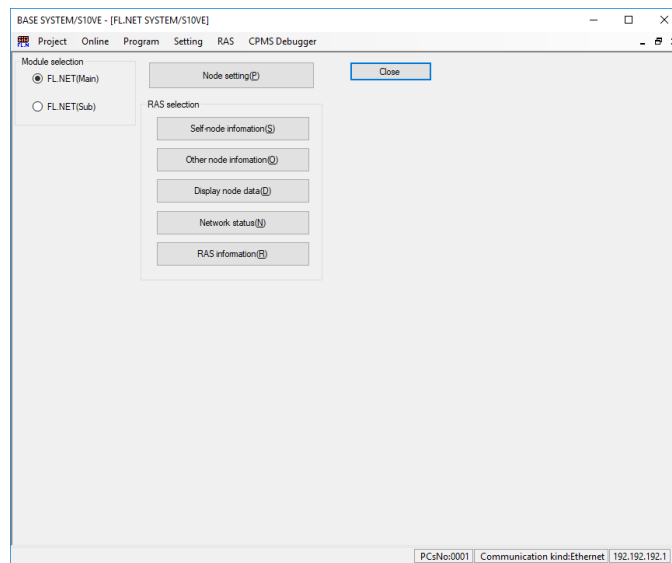


Figure 5-44 FL.NET module selection window

5.4.6 Closing the system

From the FL.NET module selection window (Figure 5-44), click the **Close** button.

5.5 Commands

5.5.1 Structure of the FL.NET SYSTEM/S10VE functions

Figure 5-45 shows the structure of the FL.NET SYSTEM/S10VE functions.

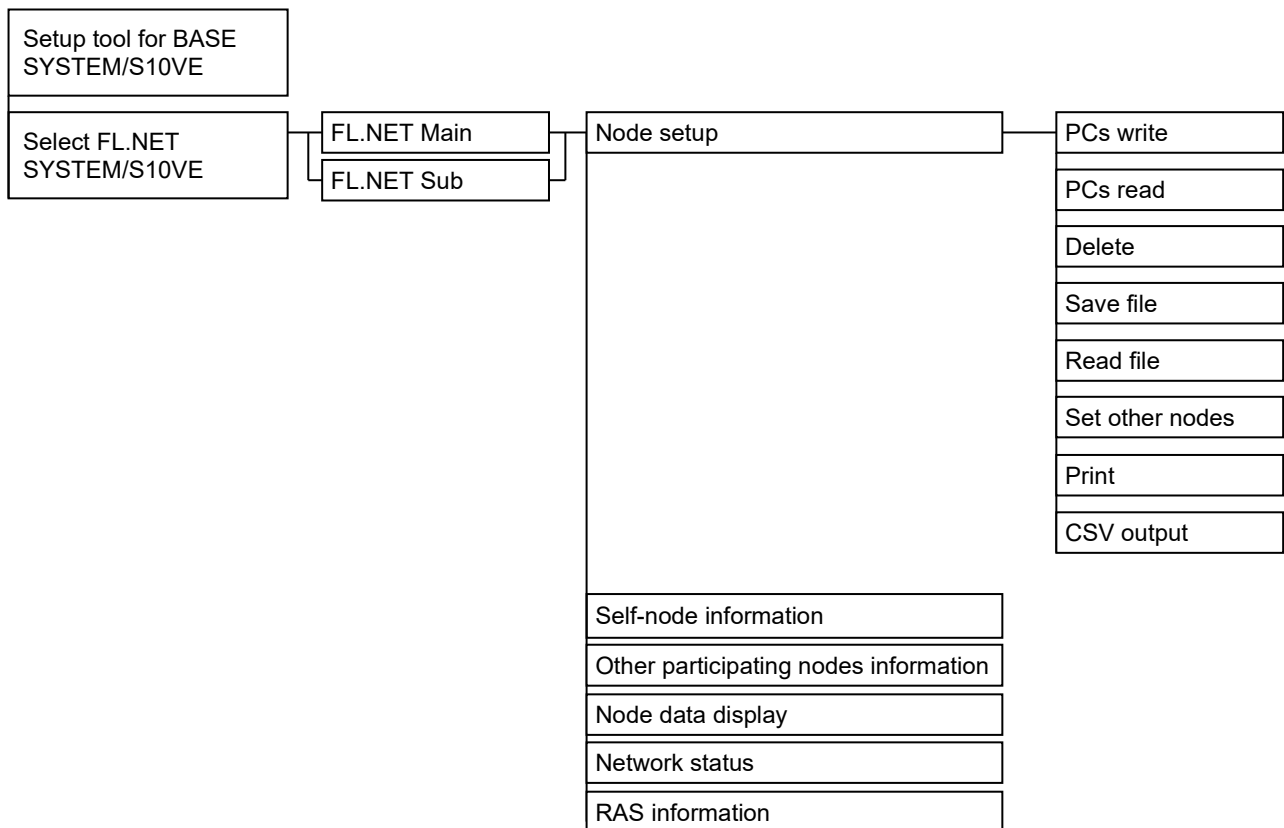


Figure 5-45 Structure of the FL.NET SYSTEM/S10VE functions

5.5.2 Specifying the self-node information settings

Function: This procedure sets the parameters required for the FL.NET module and sets the allocation of common memory.

Operation: The procedure is shown in the following subsection.

- (1) In the FL.NET module selection window (Figure 5-46), click **FL.NET (Main)** or **FL.NET (Sub)**, and then click the **Node setting** button.

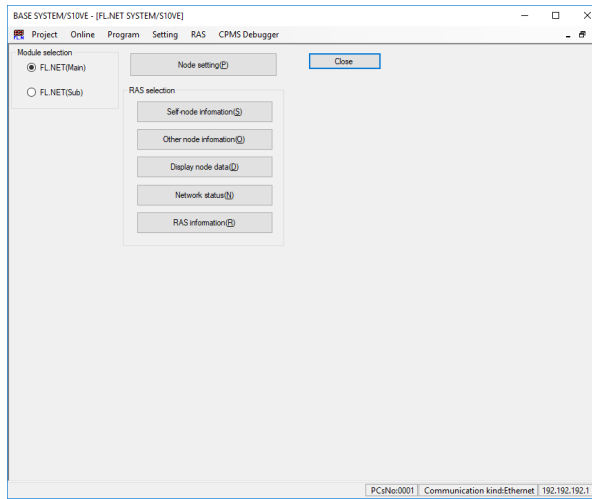


Figure 5-46 FL.NET module selection window

- (2) The **Node setting FL.NET (Main)** tab appears (Figure 5-47).

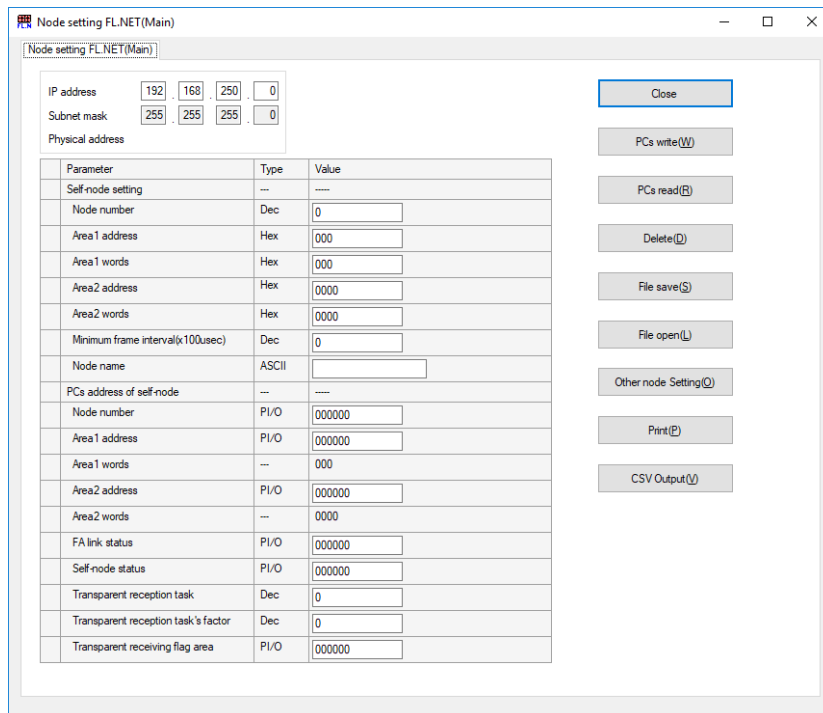


Figure 5-47 [Node setting FL.NET (Main)] tab (default display)

In the **Node setting FL.NET (Main)** tab (Figure 5-48), set each parameter value.

Node setting FL.NET(Main)

IP address: 192 . 168 . 250 . 0
Subnet mask: 255 . 255 . 255 . 0
Physical address:

Parameter	Type	Value
Self-node setting	--	----
Node number	Dec	1
Area1 address	Hex	000
Area1 words	Hex	10
Area2 address	Hex	0000
Area2 words	Hex	30
Minimum frame interval(x100usec)	Dec	10
Node name	ASCII	
PCs address of self-node	--	----
Node number	PI/O	XW0000
Area1 address	PI/O	XW0100
Area1 words	--	10
Area2 address	PI/O	XW1100
Area2 words	--	30
FA link status	PI/O	XW2000
Self-node status	PI/O	XW2100
Transparent reception task	Dec	10
Transparent reception task's factor	Dec	15
Transparent receiving flag area	PI/O	XW2200

Buttons: Close, PCs write(W), PCs read(R), Delete(D), File save(S), File open(L), Other node Setting(O), Print(P), CSV Output(V)

Figure 5-48 Example of editing in the [Node setting FL.NET (Main)] tab

For further operations, see the next page onwards.

5.5.3 Specifying other nodes' reception settings

Function: This procedure sets the common memory allocation for each node.

Operation: The procedure is shown in the following subsection.

- (1) In the **Node setting FL.NET (Main)** tab (Figure 5-47), click the **Other nodes Setting** button. The **Other nodes reception setting FL.NET (Main)** tab appears (Figure 5-49).

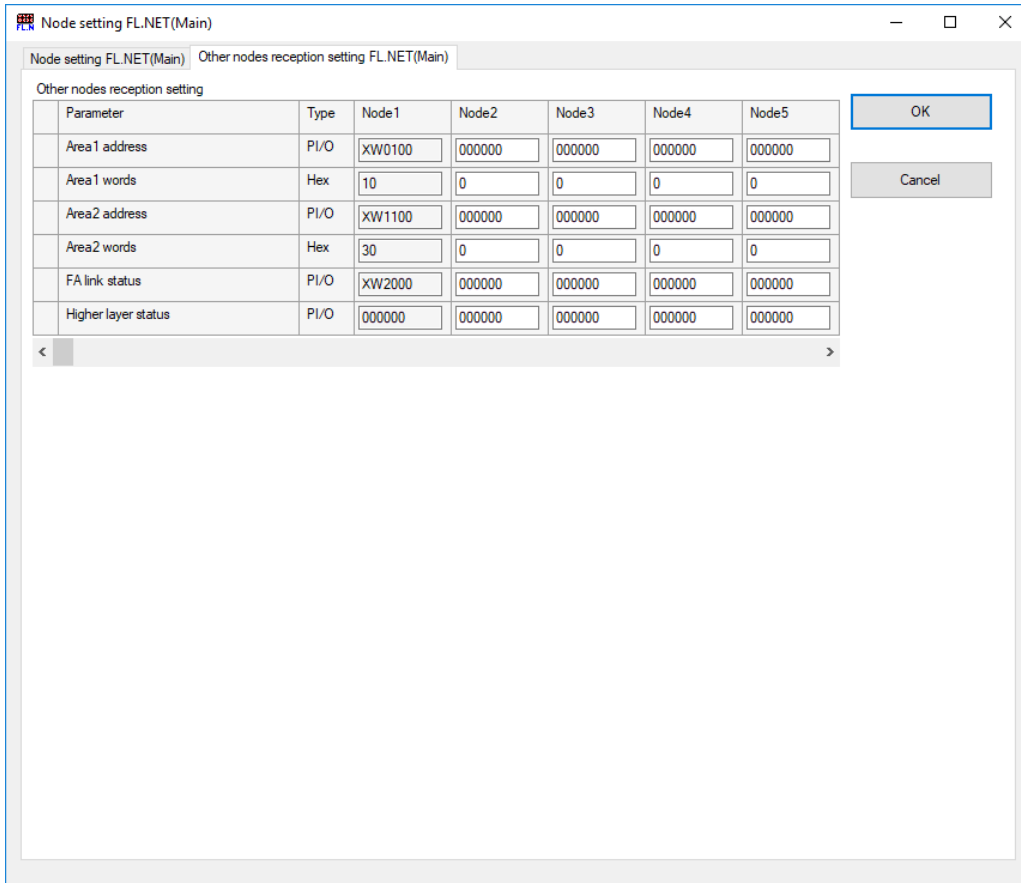


Figure 5-49 [Other nodes reception setting FL.NET (Main)] tab (default display)

- (2) Items whose settings have been changed are displayed in a different color.
- The data of the self-node (Node 1 in Figure 5-50) is shaded and cannot be changed.
- To change the settings of the self-node, use the **Node setting FL.NET (Main)** tab (Figure 5-47).

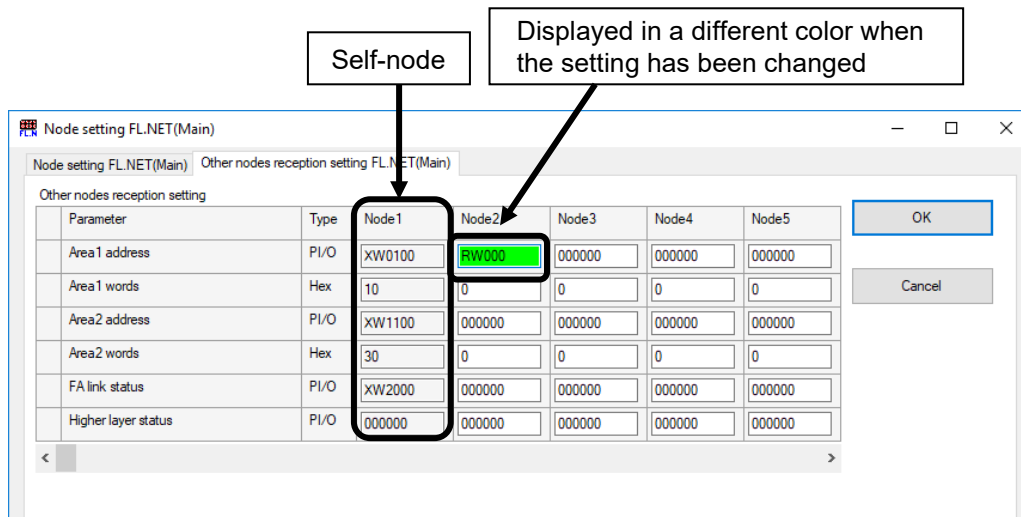


Figure 5-50 [Other nodes reception setting FL.NET (Main)] tab (when editing Node 2)

- (3) If the desired node number is not displayed, use the scroll bar.
- (4) When you have finished specifying the settings, click the **OK** button. To apply your changes to the FL.NET module, from the **Node setting FL.NET (Main)** tab, click the **PC's write** button.
- To discard your changes, click the **Cancel** button.
- A confirmation message appears (Figure 5-51).

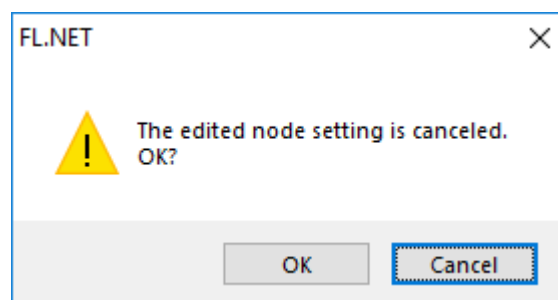


Figure 5-51 Message confirming whether to discard changes to node settings

If you click the **OK** button, the edited settings are discarded, and the **Other nodes reception setting FL.NET (Main)** tab closes.

If you click the **Cancel** button, the **Other nodes reception settings FL.NET (Main)** tab does not close.

5.5.4 Registering parameters

Function: This procedure registers the values of the parameters currently displayed in the **Node setting FL.NET (Main)** tab and the parameters set in the **Other nodes reception setting FL.NET (Main)** tab to the S10VE CPU module.

Operation: The procedure for registering parameters to the S10VE CPU module is shown in the following subsection.

- (1) From the **Node setting FL.NET (Main)** tab (Figure 5-52), click the **PCs write** button.

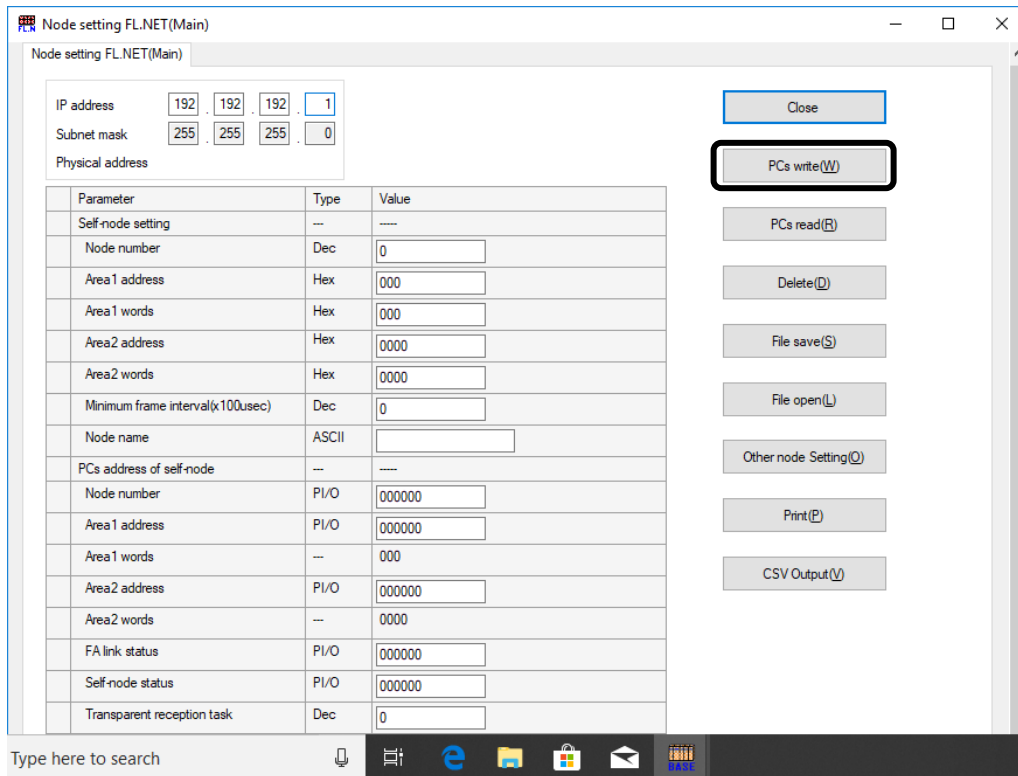


Figure 5-52 Clicking [PCs write] in the [Node setting FL.NET (Main)] tab

- (2) If any of the edited parameter values are incorrect, an error message appears. Correct the error and click the **PCs write** button again.

- (3) The Option module parameter setup list window appears (Figure 5-53).

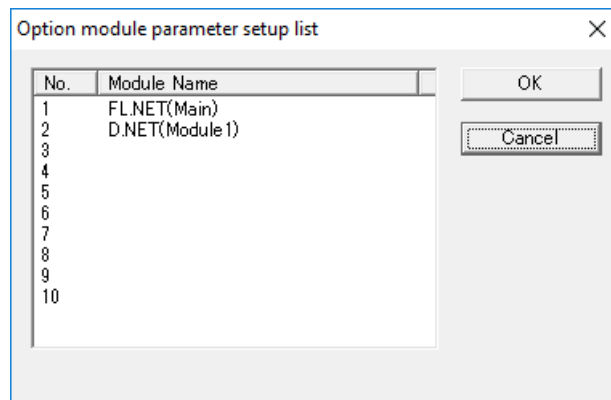


Figure 5-53 Option module parameter setup list window (registering settings)

- (4) Select the desired number and click the **OK** button or the **Cancel** button.
- OK** button: The edited parameter values are written to the S10VE CPU module[#].
When the writing is complete, a message appears confirming whether to reset the CPU module (Figure 5-55).

Cancel button: The edited parameter values are not written to the S10VE CPU module.

[#]: If you attempt to overwrite existing registered parameters, a confirmation message appears (Figure 5-54). Click either the **OK** or **Cancel** button.

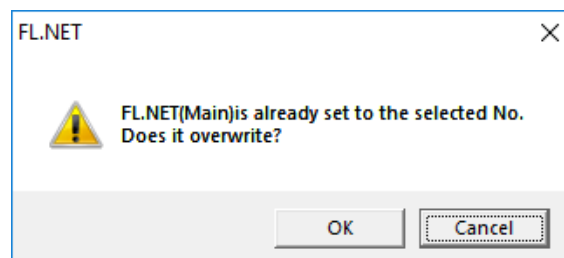


Figure 5-54 Confirmation message for overwriting parameters (when overwriting the parameters of the OD.RING (main) module)

OK button: The existing parameters in the S10VE CPU module are overwritten with the edited values.

Cancel button: You are returned to the Optional module parameter setup list window.

- (5) From the reset confirmation message (Figure 5-55), click the **OK** or **Cancel** button.

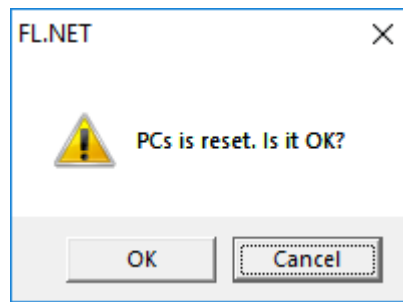


Figure 5-55 Reset confirmation message

OK button: The CPU module is reset.

The edited parameter values are applied to the FL.NET module, and the FL.NET module automatically joins the network.

Cancel button: The CPU module is not reset.

The edited parameter values are not applied to the FL.NET module.

The CPU module needs to be reset to apply the edited parameter values to the FL.NET module.

- (6) This concludes the registration of FL.NET module parameter values to the S10VE CPU module.

5.5.5 Reading parameters

Function: This procedure reads the FL.NET module parameters from the S10VE CPU module, and displays them in the Node setting window.

Operation: The procedure for reading FL.NET module parameters from the S10VE CPU module is shown in the following subsection.

- (1) From the **Node setting FL.NET (Main)** tab (Figure 5-56), click the **PCs read** button.

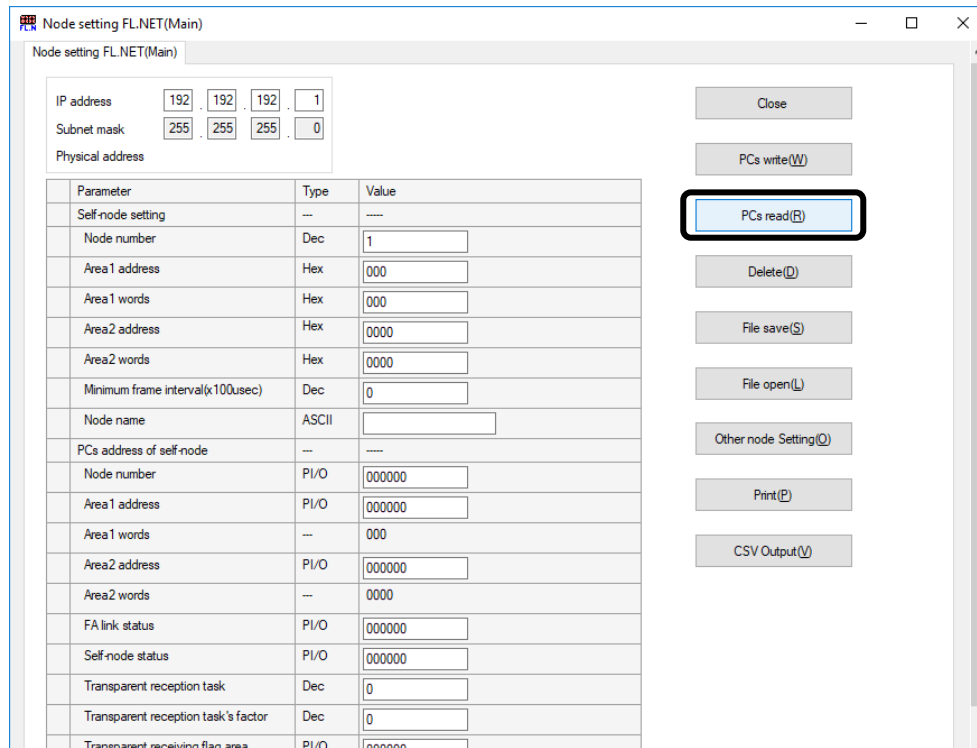


Figure 5-56 Clicking the [PCs read] button in the [Node setting FL.NET (Main)] tab

- (2) If you have edited any parameters without applying your changes, the following message appears confirming whether to discard the changes. Click either the **Yes** or **No** button.

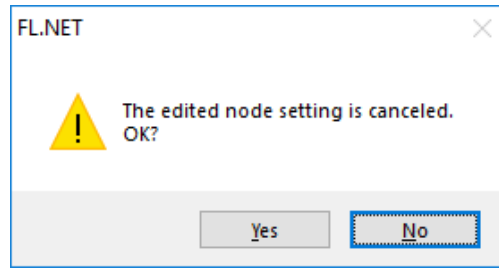


Figure 5-57 Message confirming whether to discard changes

Yes button: Changes to the parameters (that is, any edited parameters) are discarded. However, changes are not discarded if you click the **Cancel** button when the Receiving agency selection window appears.

No button: Reading of parameters from the S10VE CPU module or FL.NET module is canceled.

- (3) If the FL.NET module selected in the FL.NET module selection window (Figure 5-46) is already installed and registered in the S10VE CPU module, the Receiving agency selection window appears (Figure 5-58). If the FL.NET module selected in the FL.NET module selection window is either not installed or not registered in the S10VE CPU module, the Receiving agency selection window does not appear. In this case, the parameter values are read from the FL.NET module or the S10VE CPU module, and displayed in the **Node setting FL.NET (Main)** tab and the **Other nodes reception setting FL.NET (Main)** tab.

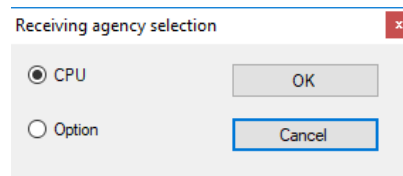


Figure 5-58 Receiving agency selection window

- (4) From the Receiving agency selection window, select either the **CPU** or **Option** radio button
- CPU**: Registered parameter values are read from the S10VE CPU module of the FL.NET module selected in the FL.NET module selection window.
CPU is selected by default.
- Option**: Parameter values are read from the FL.NET module selected in the FL.NET module selection window.
- (5) From the Receiving agency selection window, click either the **OK** or **Cancel** button.
- OK** button: FL.NET module parameters are received from your selection in (4), and displayed in the **Note setting FL.NET (Main)** tab and the **Other nodes reception setting FL.NET (Main)** tab.
- Cancel** button: Reception of FL.NET module parameters is canceled.
- (6) This concludes the reading of parameter values from the S10VE CPU module or FL.NET module.

5.5.6 Deleting parameters

Function: This procedure deletes FL.NET module parameters from the registration table of the S10VE CPU module.

Operation: The procedure for deleting the FL.NET module parameters from the registration table of the S10VE CPU module is shown in the following subsection.

- (1) From the **Node setting FL.NET (Main)** tab (Figure 5-59), click the **Delete** button.

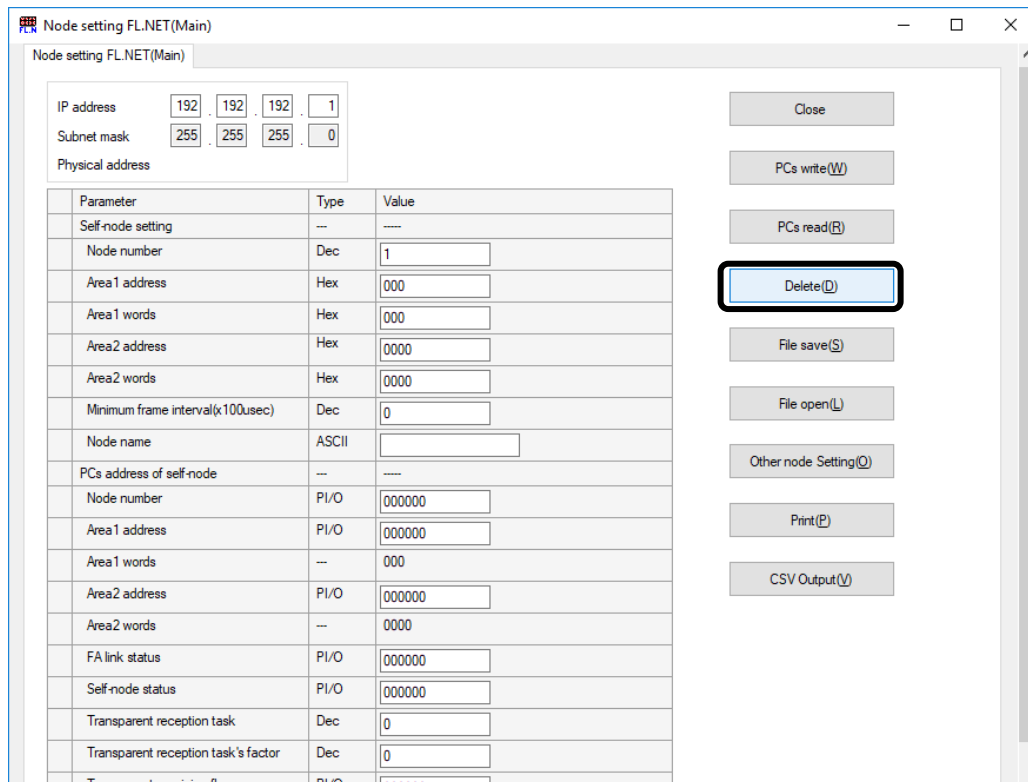


Figure 5-59 Clicking the [Delete] button in the [Node setting FL.NET (Main)] tab

- (2) The Option module parameter setup list window appears (Figure 5-60).

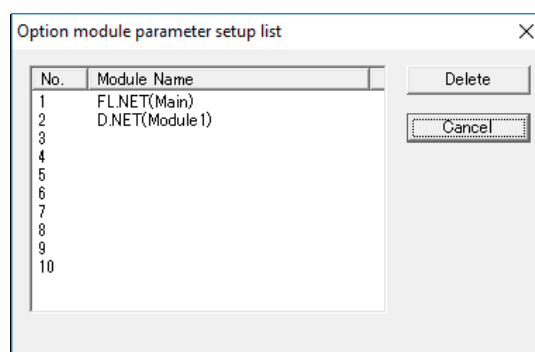


Figure 5-60 Option module parameter setup list window (deleting parameters)

- (3) Select the option parameters you want to delete. To cancel deletion of option parameters, click the **Cancel** button.

If you click the **Delete** button, a message confirming whether to delete the parameters appears (Figure 5-61).

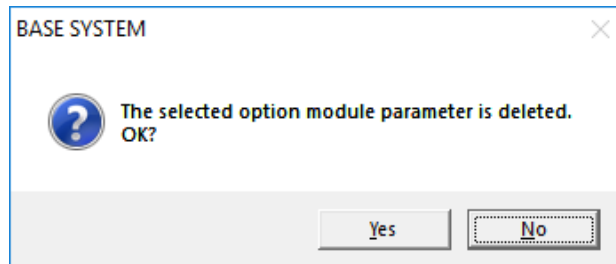


Figure 5-61 Message confirming whether to delete parameters

In the confirmation message dialog box, click either the **Yes** or **No** button.

Yes button: The selected option parameters are deleted, and the message appears confirming whether to reset the CPU module.

No button: The parameters are not deleted, and you are returned to the Optional module parameter setup list window.

- (4) In the reset confirmation message dialog box (Figure 5-62), click either the **OK** or **Cancel** button.

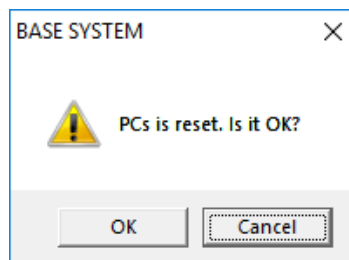


Figure 5-62 Reset confirmation message

OK button: The CPU module is reset.

Note: Parameters stored in the FL.NET module are not deleted.

Cancel button: The CPU module is not reset.

- (5) This concludes the deletion of option parameters from the S10VE CPU module.

5.5.7 Saving parameters to a file

Function: This procedure saves the parameters currently displayed in the **Node setting FL.NET (Main)** tab and the parameters in the **Other nodes reception setting FL.NET (Main)** tab to a file under a user-specified file name. The file is saved in binary format.

The saved file can be used by clicking the **File open** button or by using the BASE SYSTEM data exchange function.

Operation: The procedure for saving parameters in a file is shown in the following subsection.

- (1) From the **Node setting FL.NET (Main)** tab (Figure 5-63), click the **File save** button.

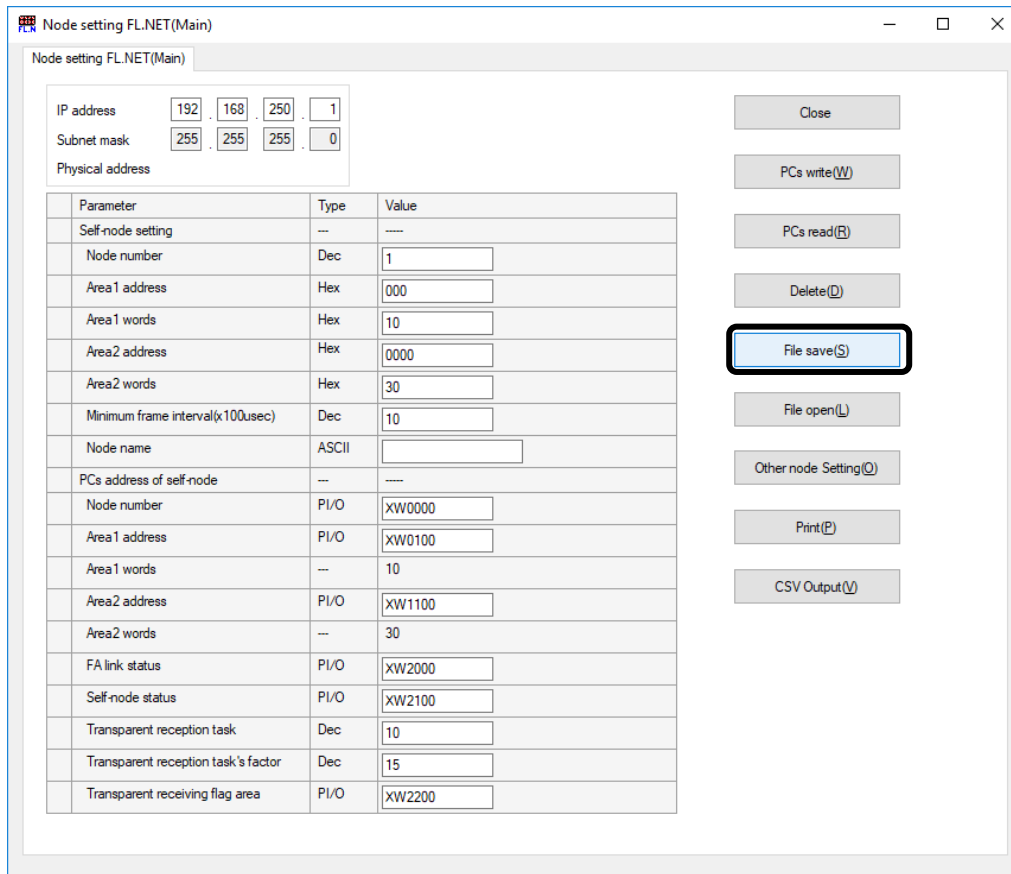


Figure 5-63 Clicking the [File save] button in the [Node setting FL.NET (Main)] tab

- (2) If any of the edited parameter values are incorrect, an error message appears. Correct the error and click the **File save** button again.

- (3) The Save As window appears (Figure 5-64).

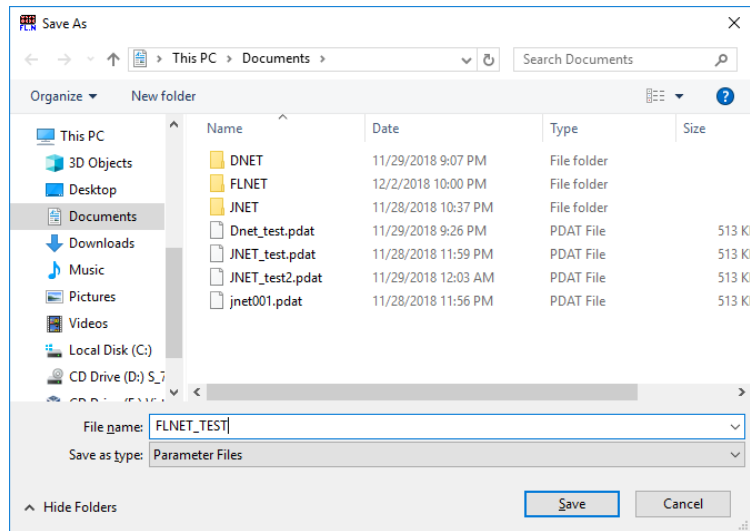


Figure 5-64 Save As window

- (4) Enter a name for the file to be saved, and click the **Save** button.
 The File save window appears (Figure 5-65).

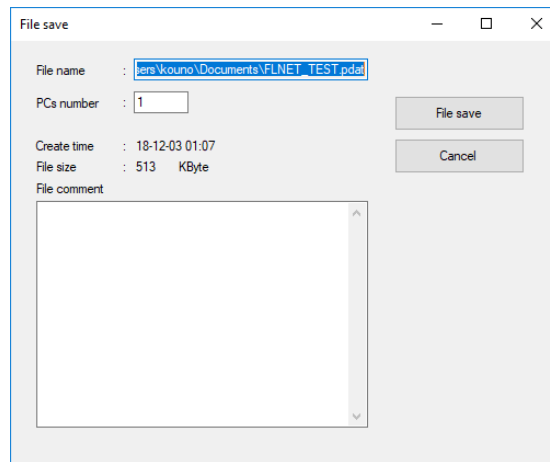


Figure 5-65 File save window

- PCs number:** The PCs number for the selected project is displayed by default.
 You can change the PCs number to any value in the following range.
- Range of PCs numbers:** 0 to 9999
- Create time:** This displays the date and time the File save window appeared.
- File comment:** You can enter a comment for the file (optional).
 You can enter up to 512 characters.

- (5) From the File save window, click either the **File save** or **Cancel** button.

File save button: The parameters currently displayed in the **Node setting FL.NET (Main)** tab and the parameters in the **Other nodes reception setting FL.NET (Main)** tab are saved to a file. If the specified file already exists, the Save As window appear. Determine whether to overwrite the existing file.

Cancel button: The parameters currently displayed in the **Node setting FL.NET (Main)** tab and the parameters in the **Other nodes reception setting FL.NET (Main)** tab are not saved to a file.

- (6) This concludes the saving of FL.NET parameters to a file.

5.5.8 Reading parameters from a file

Function: This procedure reads the FL.NET module parameters that were saved in a file from the File save window and displays them in the **Node setting FL.NET (Main)** tab and the **Other nodes reception setting FL.NET (Main)** tab.

Operation: The procedure for reading parameters from a file are shown in the following subsection.

- (1) From the **Node setting FL.NET (Main)** tab (Figure 5-66), click the **File open** button.

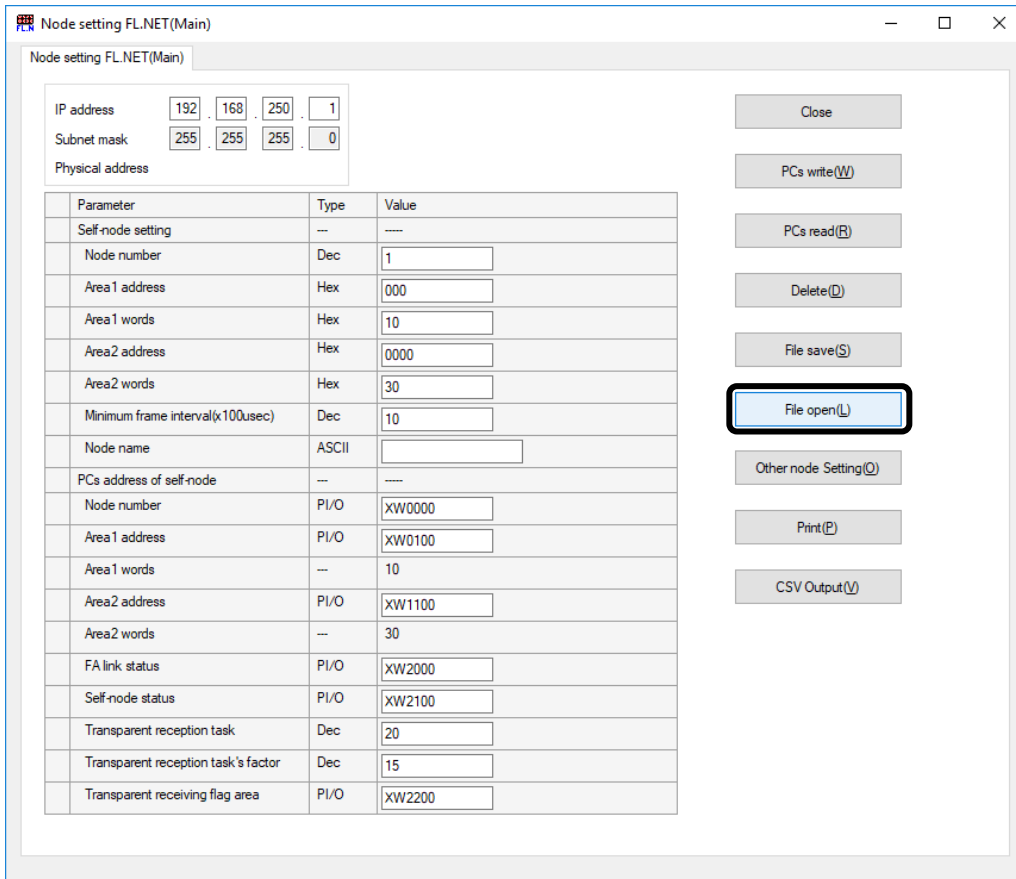


Figure 5-66 Clicking the [File open] button in the [Node setting FL.NET (Main)] tab

- (2) If you are currently editing the parameters, a message appears confirming whether to discard your changes (Figure 5-67). Click either the **OK** or **Cancel** button.

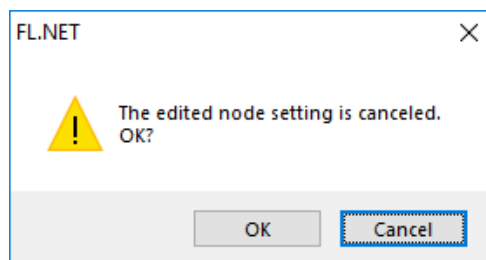


Figure 5-67 Message confirming whether to discard changes

OK button: Changes to the parameters are discarded.

However, if you click the **Cancel** button before clicking the **File open** button in the File open window (Figure 5-70), the changes are not discarded.

Cancel button: Opening of the parameter file is canceled.

- (3) The Open window appears (Figure 5-68).

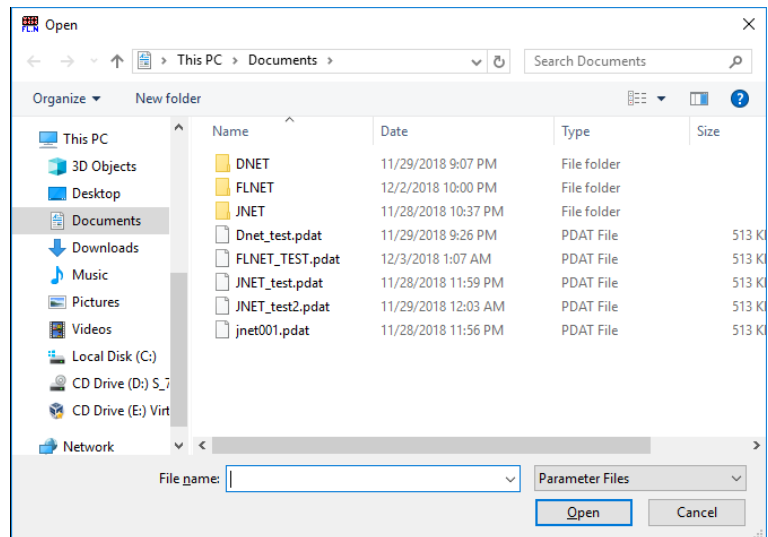


Figure 5-68 Open window

Select the parameter file you want to open and click the **Open** button.

To cancel opening of the file, from the Open window, click the **Cancel** button.

- (4) If the selected parameter file does not contain the parameters of the FL.NET module selected in the FL.NET module selection window (Figure 5-46), a message appears indicating that the selected file is incorrect (Figure 5-69).

Click the **OK** button to reload the parameters.

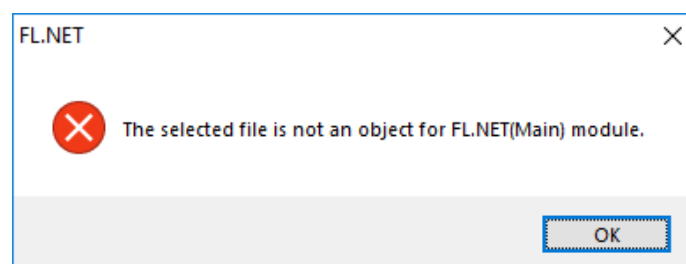


Figure 5-69 Message indicating that the selected file is incorrect

5. User's Guide

- (5) If the selected parameter file contains the parameter values of the FL.NET module selected in the FL.NET module selection window (Figure 5-46), the File open window appears (Figure 5-70). This window displays the header information of the selected parameter file.

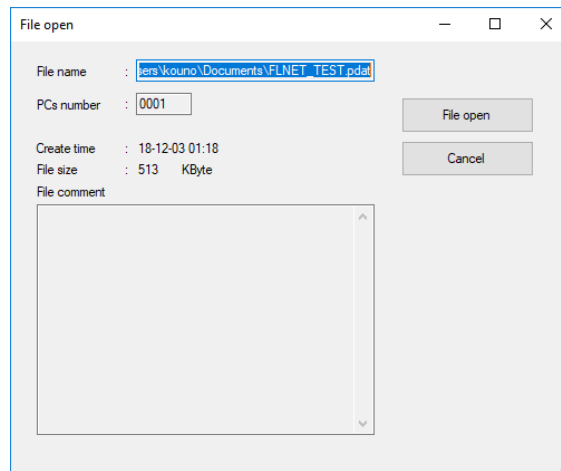


Figure 5-70 File open window

- File name:** The name of the selected parameter file is displayed.
- PCs number:** The PCs number of the selected parameter file is displayed.
- Create time:** The date and time the selected parameter file was saved is displayed.
- File comment:** The file comment of the selected parameter file is displayed.
- (6) From the File open window, click either the **File open** or **Cancel** button.
- File open** button: The contents of the selected parameter file are read and displayed in the Node setting window.
- Cancel** button: The contents of the selected parameter file are not read, and the File open window is closed.
- (7) This concludes the reading of FL.NET parameters from a file.

5.5.9 Printing

Function: This procedure prints the displayed parameters and their values in the **Node setting FL.NET (Main)** tab and the **Other nodes reception setting FL.NET (Main)** tab.

Operation: The procedure for printing the parameters and their values is shown in the following subsection.

- (1) From the **Node setting FL.NET (Main)** tab (Figure 5-71), click the **Print** button.

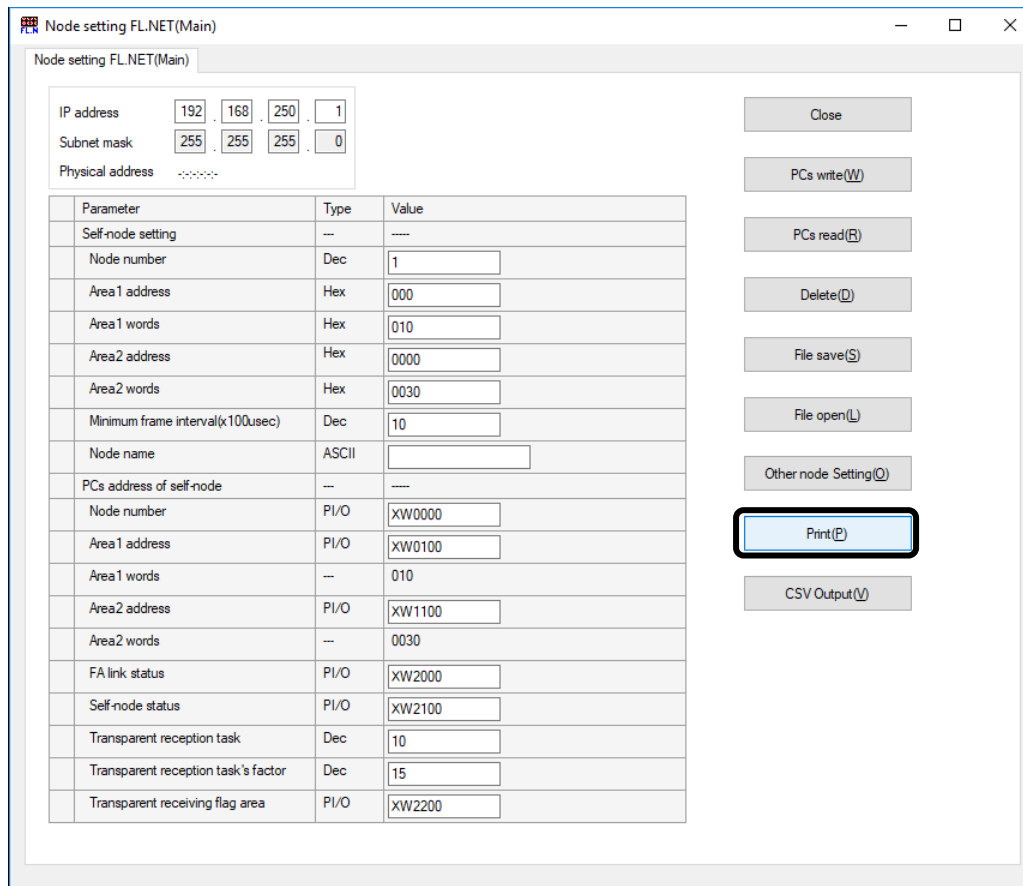


Figure 5-71 Clicking the [Print] button in the [Node setting FL.NET (Main)] tab

- (2) If any of the edited parameter values are incorrect, an error message appears. Correct the error and click the **Print** button again.

5. User's Guide

- (3) The Print dialog box appears (Figure 5-72). In this dialog box, specify the target printer, any relevant properties, and then click the **Print** button.

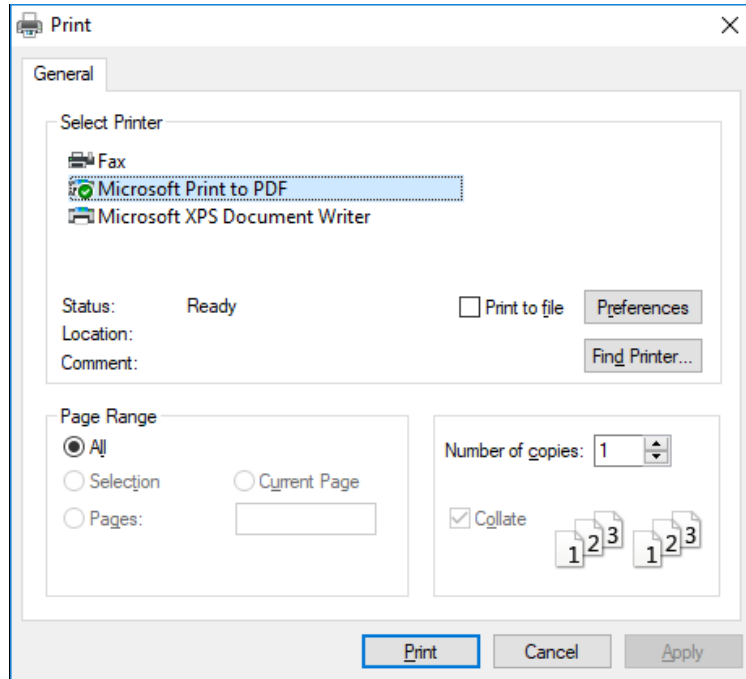


Figure 5-72 Print dialog box

- (4) The parameters and their values currently displayed in the **Node setting FL.NET (Main)** tab (Figure 5-71) are printed.

<Sample printout>

```
FL.NET 12/6/2018 9:54:14 PM
PCsNo=1
```

Main module

IP address setting

```
IP address      192.192.192.1
Subnet mask    255.255.255.0
Physical address
```

Self-node setting

```
Node number      1
Area1 address    0x000
Area1 words      0x000
Area2 address    0x0000
Area2 words      0x0000
Minimum frame interval(x100usec) 0
Node name
```

PCs address of self-node

```
Node number      000000
Area1 address    000000
Area1 words      0x000
Area2 address    000000
Area2 words      0x0000
FA link status   000000
Self-node status 000000
Transparent reception task 0
Transparent reception task's factor 0
Transparent receiving flag area 000000
```

Other nodes reception setting

```
Node number Area1 address Area1 words Area2 address Area2 words FA link status Higher layer status
```

Figure 5-73 Sample printout

5.5.10 CSV output

Function: This procedure saves the parameters currently displayed in the **Node setting FL.NET (Main)** tab (Figure 5-74) and the parameters in the **Other nodes reception settings FL.NET (Main)** tab to a file in CSV format.

Operation: The procedure for outputting parameters in CSV format is shown in the following subsection.

- (1) From the **Node setting FL.NET (Main)** tab, click the **CSV Output** button.

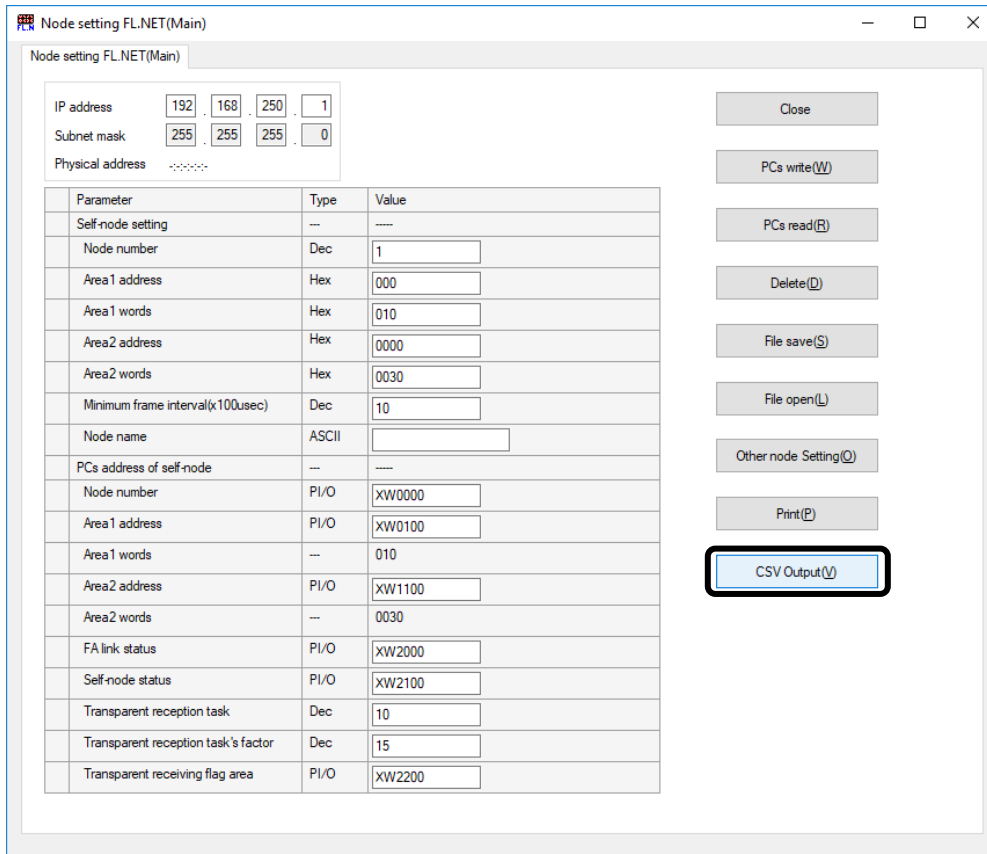


Figure 5-74 Clicking the [CSV Output] button in the [Node setting FL.NET (Main)] tab

- (2) If any of the edited parameter values are incorrect, an error message appears. Correct the error and click the **CSV Output** button again.

- (3) The Save As window appears (Figure 5-75).

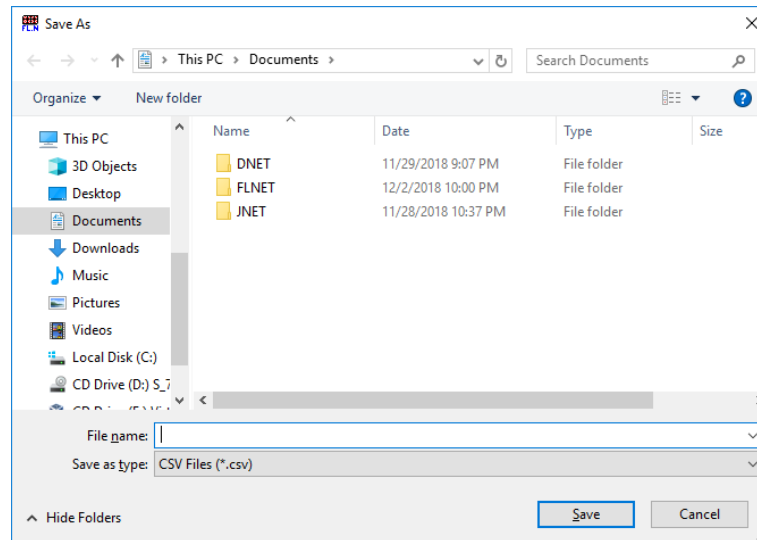


Figure 5-75 Save As window

- (4) Enter a name for the file to be saved, and click the **Save** button.
When you click the **Cancel** button, output in CSV format is canceled.
- (5) The parameter values currently displayed in the **Node setting FL.NET (Main)** tab (Figure 5-74) are saved to a file in CSV format.

5. User's Guide

<Sample CSV file output>

FL.NET 12/3/2018 1:24:37 AM
PCsNo=1

Main module

IP address setting

IP address,192.168.250.1
Subnet mask,255.255.255.0
Physical address,-:-:-:-:-

Self-node setting

Node number,1
Area1 address,0x000
Area1 words,0x010
Area2 address,0x0000
Area2 words,0x0030
Minimum frame interval(x100usec),10
Node name,

PCs address of self-node

Node number,XW0000
Area1 address,XW0100
Area1 words,0x010
Area2 address,XW1100
Area2 words,0x0030
FA link status,XW2000
Self-node status,XW2100
Transparent reception task,10
Transparent reception task's factor,15
Transparent receiving flag area,XW2200

Other nodes reception setting

Node number,Area1 address,Area1 words,Area2 address,Area2 words,FA link status,Higher layer status
1,XW0100,0x010,XW1100,0x0030,XW2000,000000

Figure 5-76 Sample CSV file output

5.5.11 Self-node information

Function: This monitors information about the FL.NET module. This also allows you to add or remove the FL.NET module to or from the network.

As a prerequisite for using this function, the PCs and computer must be connected by a cable.

Operation: The procedure is shown in the following subsection.

- (1) From the FL.NET module selection window (Figure 5-44), select **FL.NET (Main)** or **FL.NET (Sub)**, and click the **Self-node information** button.
- (2) The **Self- node information FL.NET (Main)** tab (Figure 5-77) appears.

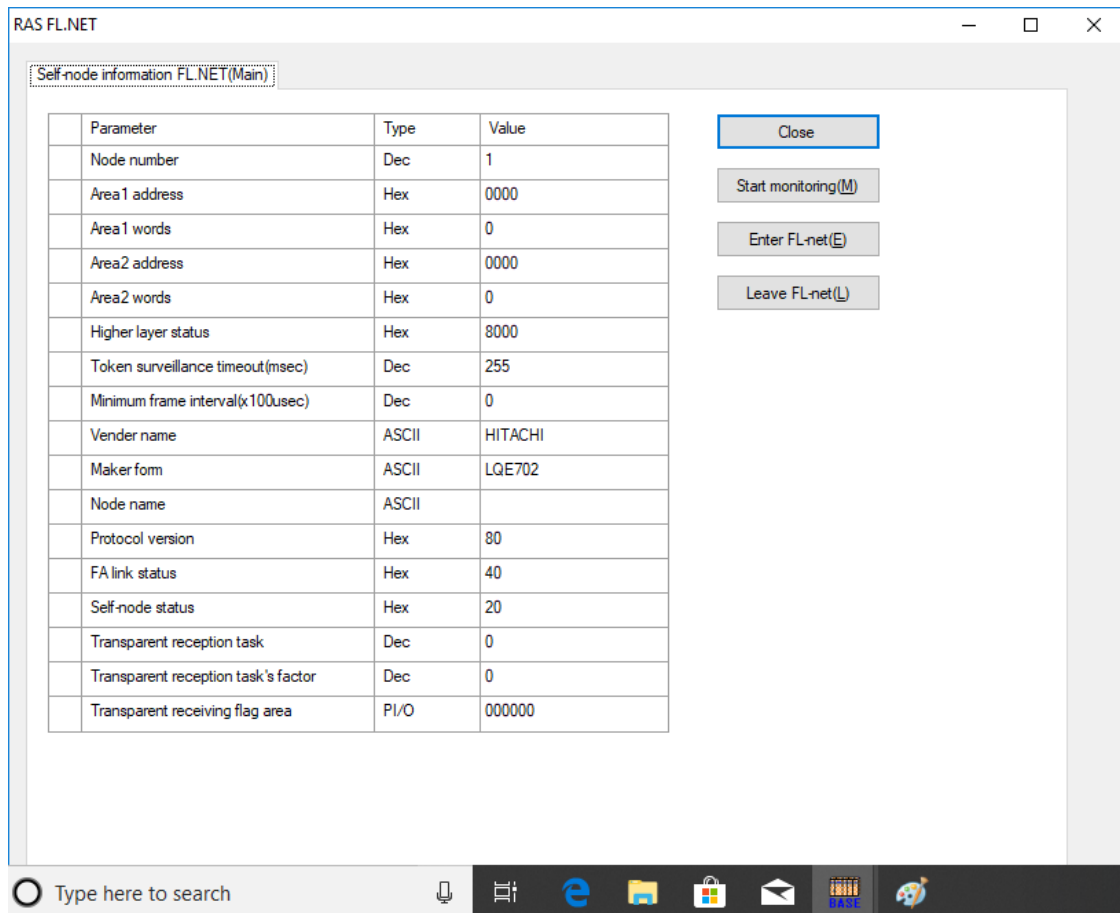


Figure 5-77 [Self-node information FL.NET (Main)] tab

5. User's Guide

- (3) To monitor the self-node information, click the **Start monitoring** button.

The **Start monitoring** button changes to the **Stop monitoring** button. The tab monitors and displays information on the self-node.

To stop monitoring, click the **Stop monitoring** button.

The **Stop monitoring** button changes to the **Start monitoring** button. The tab stops monitoring information on the self-node.

- (4) To join the network, click the **Enter FL-net** button.

- (5) To leave the network, click the **Leave FL-net** button.

5.5.12 Information on other participating nodes

Function: This procedure monitors information on other nodes in the network.

A prerequisite for using this function is that the PCs and computer are connected by a cable.

Operation: The procedure is shown in the following subsection.

- (1) From the FL.NET module selection window (Figure 5-44), select either **FL.NET (Main)** or **FL.NET (Sub)**, and click the **Other node information** button.
- (2) The **Other nodes information FL.NET (Main)** tab (Figure 5-78) appears.

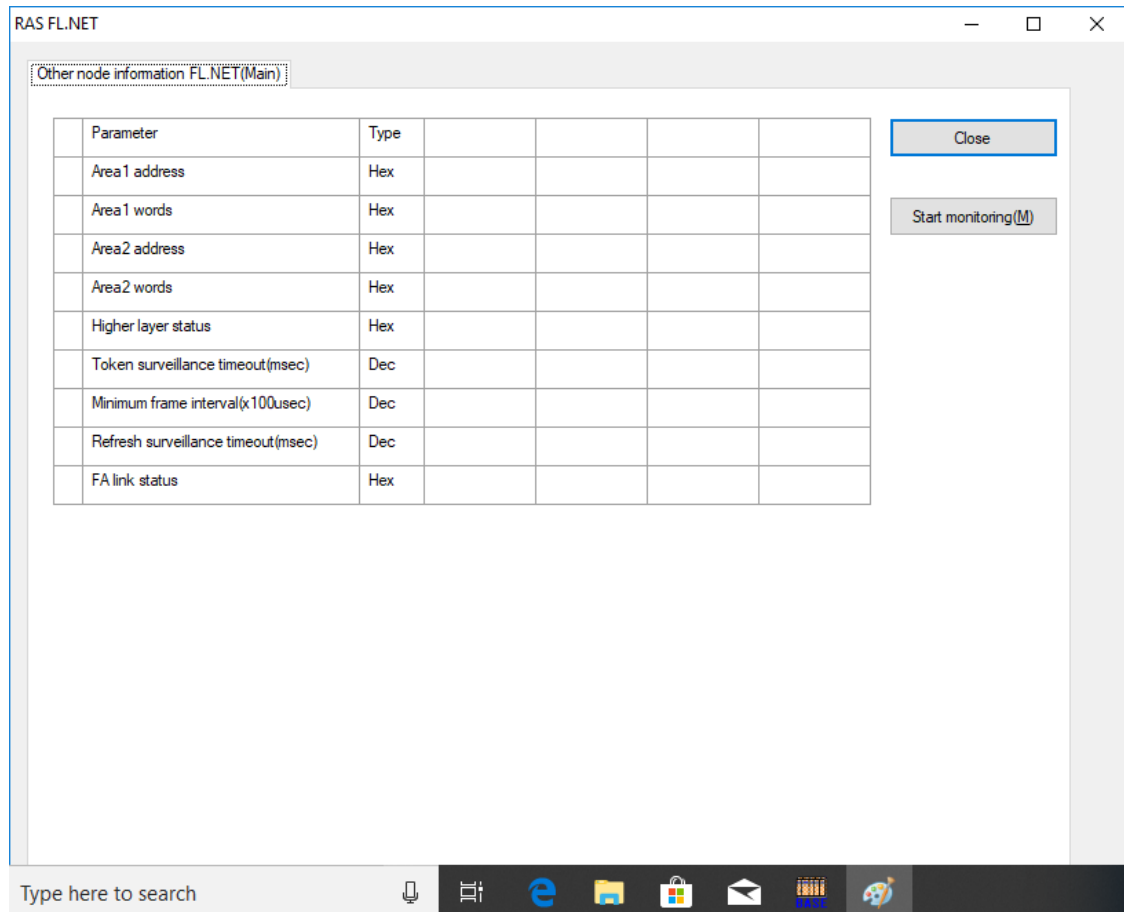


Figure 5-78 [Other nodes information FL.NET (Main)] tab

- (3) To monitor the joining other nodes information, click the **Start monitoring** button. The **Start monitoring** button changes to the **Stop monitoring** button. The tab monitors and displays information on other nodes in the network. To stop monitoring, click the **Stop monitoring** button. The **Stop monitoring** button changes to the **Start monitoring** button. The tab stops monitoring information on other nodes in the network.

5.5.13 Displaying data on other nodes

Function: This procedure allows the display and editing of data on areas 1 and 2 of either the self-node or another node.

A prerequisite for using this function, the PCs and computer must be connected by a cable.

Operation: The procedure is shown in the following subsection.

- (1) From the FL.NET module selection window (Figure 5-44), select either **FL.NET (Main)** or **FL.NET (Sub)**, and click the **Display node data** button.
- (2) The **Node data FL.NET (Main)** tab appears (Figure 5-79, Figure 5-80).

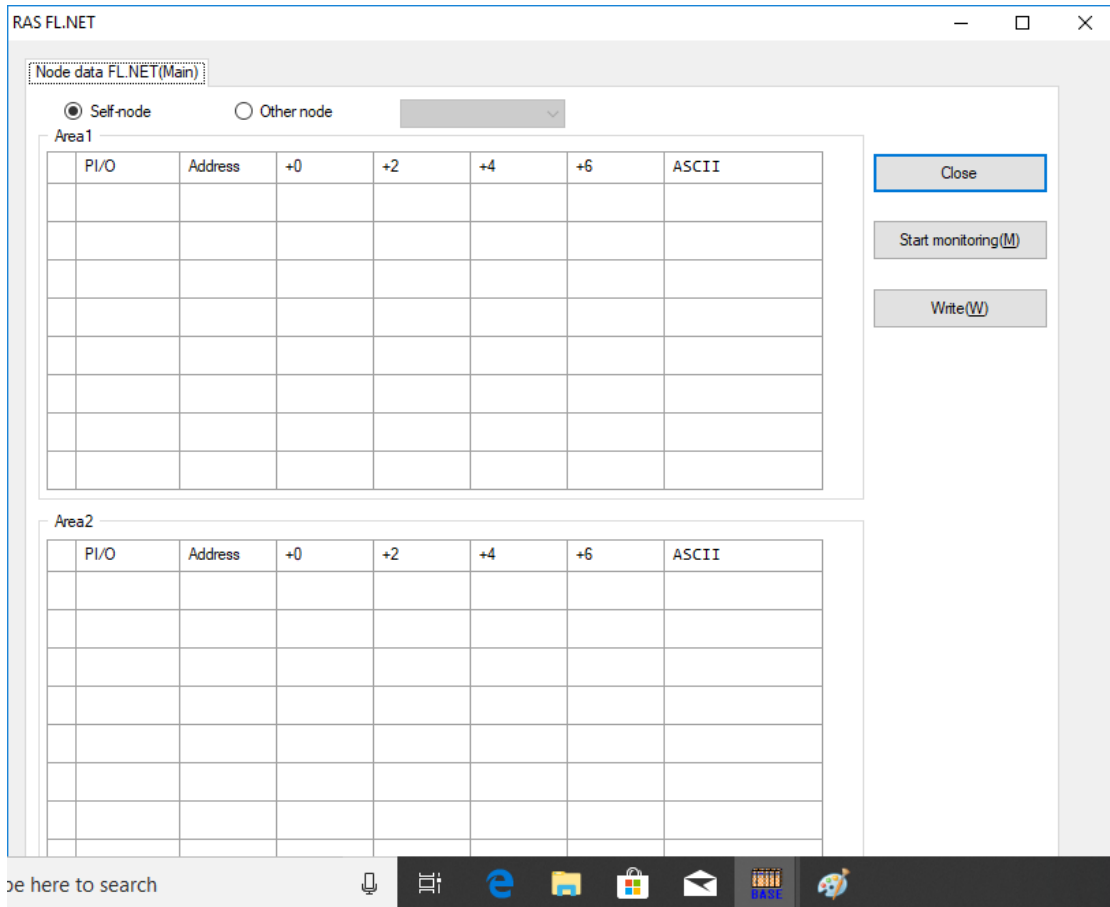


Figure 5-79 [Node data FL.NET (Main)] tab (selecting the self-node)

By default the area 1 and area 2 data of **Self-node** is displayed. To display data for other nodes in the network, click the **Other node** radio button. Select the node whose data to display from the combo box. The **Other node** combo box is not available while **Self-node** is selected.

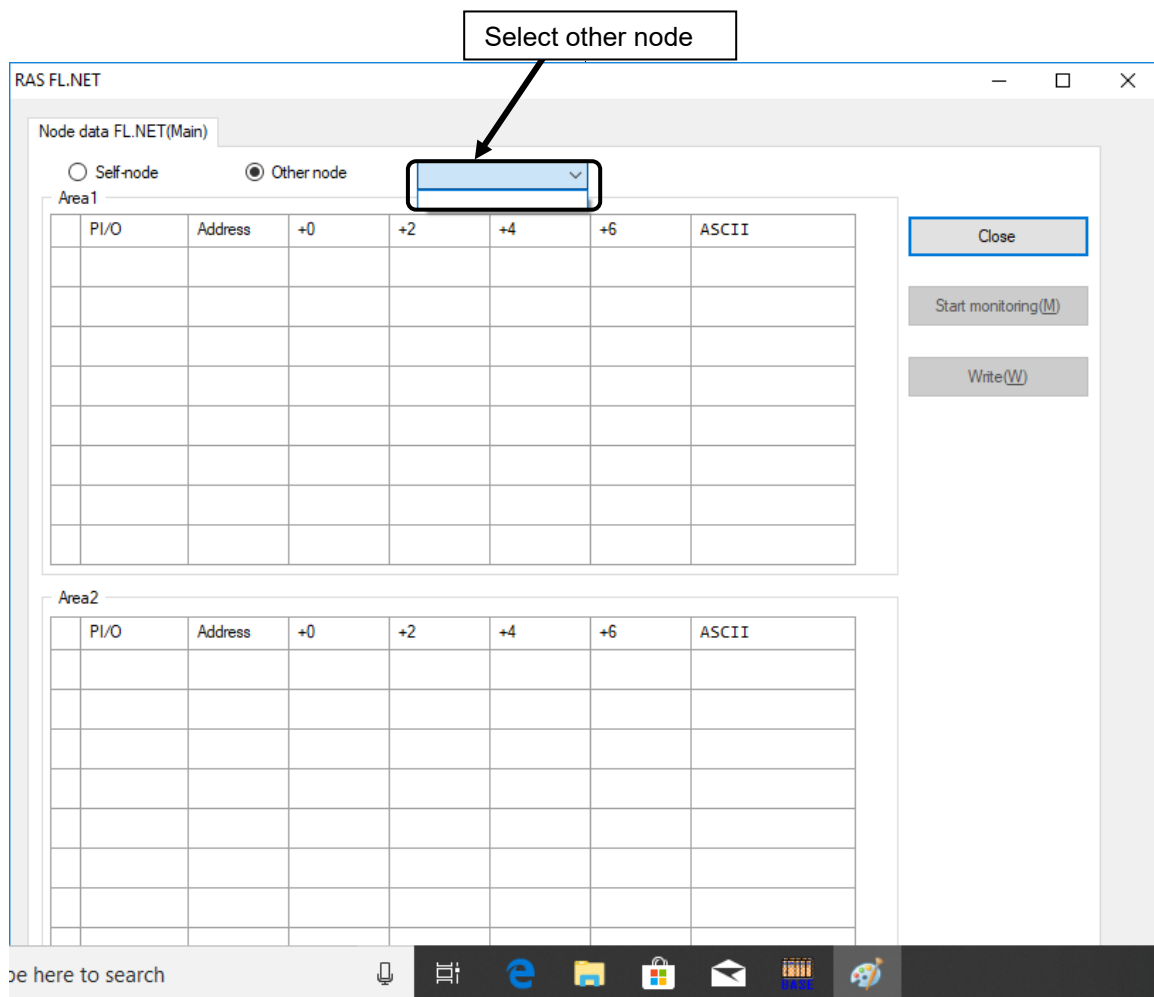


Figure 5-80 [Node data FL.NET (Main)] tab (selecting other node)

- (3) If the data in areas 1 and 2 is too large to be displayed at one time, use the scroll bar.
- (4) To monitor the data in areas 1 and 2, click the **Start monitoring** button.

The **Start monitoring** button changes to the **Stop monitoring** button. The tab monitors and displays the node data.

To stop monitoring, click the **Stop monitoring** button.

The **Stop monitoring** button changes to the **Start monitoring** button. The tab stops monitoring the node data.

You cannot change the data in areas 1 and 2 of the self-node during monitoring.

You can never change the data of other nodes. You can only reference the data.

If you want to change the data of the self-node, click the **Write** button. Areas 1 and 2 are overwritten by the data currently displayed for areas 1 and 2. If you press the **Tab** key after changing node data, only that new data overwrites the existing data.

- (5) To close the **Node data FL.NET (Main)** tab (Figure 5-79, Figure 5-80), click the **Close** button.

5.5.14 Network status

Function: This procedure displays various information about the network status during participation in the network, such as the refresh cycle time limit.

As a prerequisite for using this function, the PCs and computer must be connected by a cable.

Operation: The procedure is as shown in the following subsection.

- (1) From the FL.NET module selection window (Figure 5-44), select either **FL.NET (Main)** or **FL.NET (Sub)**, and click the **Network status** button.
- (2) The **Network status FL.NET (Main)** tab (Figure 5-81) appears.

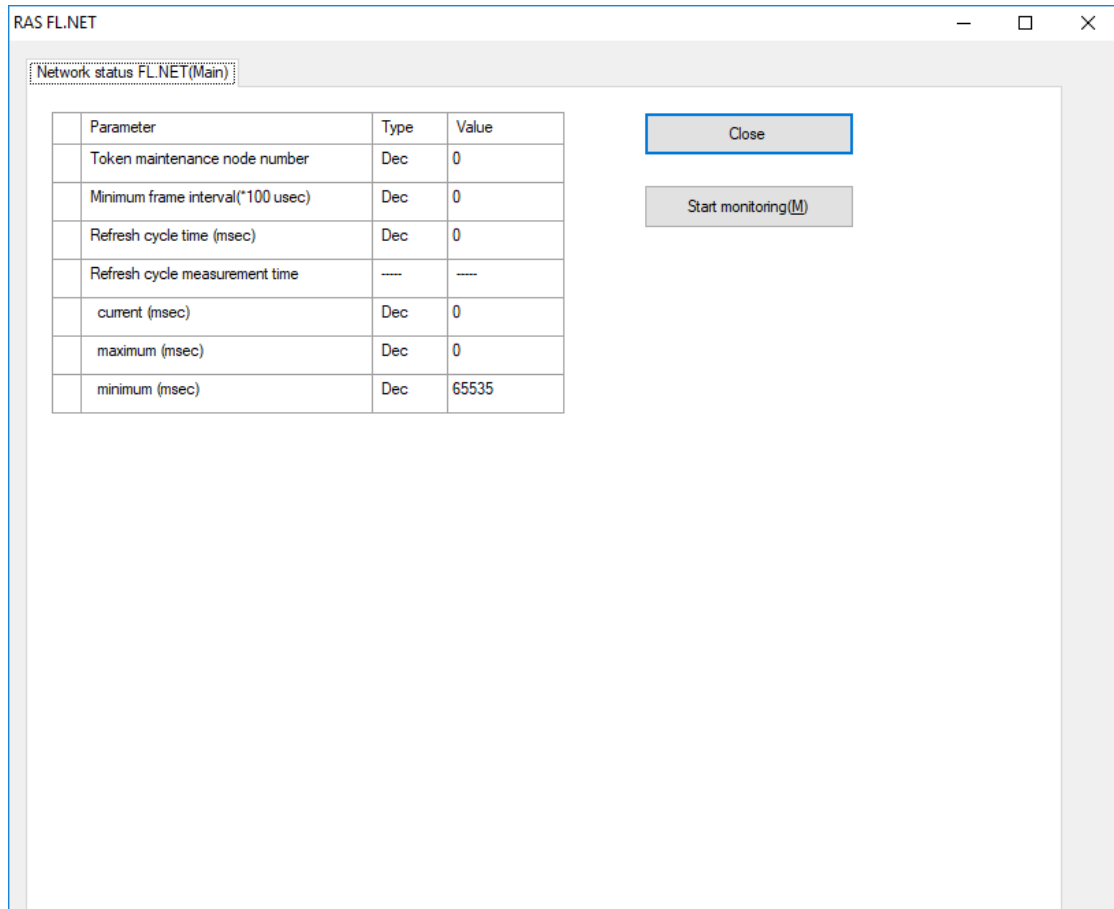


Figure 5-81 [Network status FL.NET (Main)] tab

- (3) To monitor the network status, click the **Start monitoring** button.

The **Start monitoring** button changes to the **Stop monitoring** button. The tab monitors and displays the network status.

To stop monitoring, click the **Stop monitoring** button.

The **Stop monitoring** button changes to the **Start monitoring** button. The tab stops monitoring the network status.

- (4) To close the **Network status FL.NET (Main)** tab (Figure 5-81), click the **Close** button.

5.5.15 RAS information

Function: This procedure displays the RAS information.

As a prerequisite for using this function, the PCs and computer must be connected by a cable.

Operation: The procedure is shown in the following subsection.

- (1) From the FL.NET module selection window (Figure 5-44), select either **FL.NET (Main)** or **FL.NET (Sub)**, and click the **RAS information** button.
- (2) The **RAS information FL.NET (Main)** tab (Figure 5-82) appears.

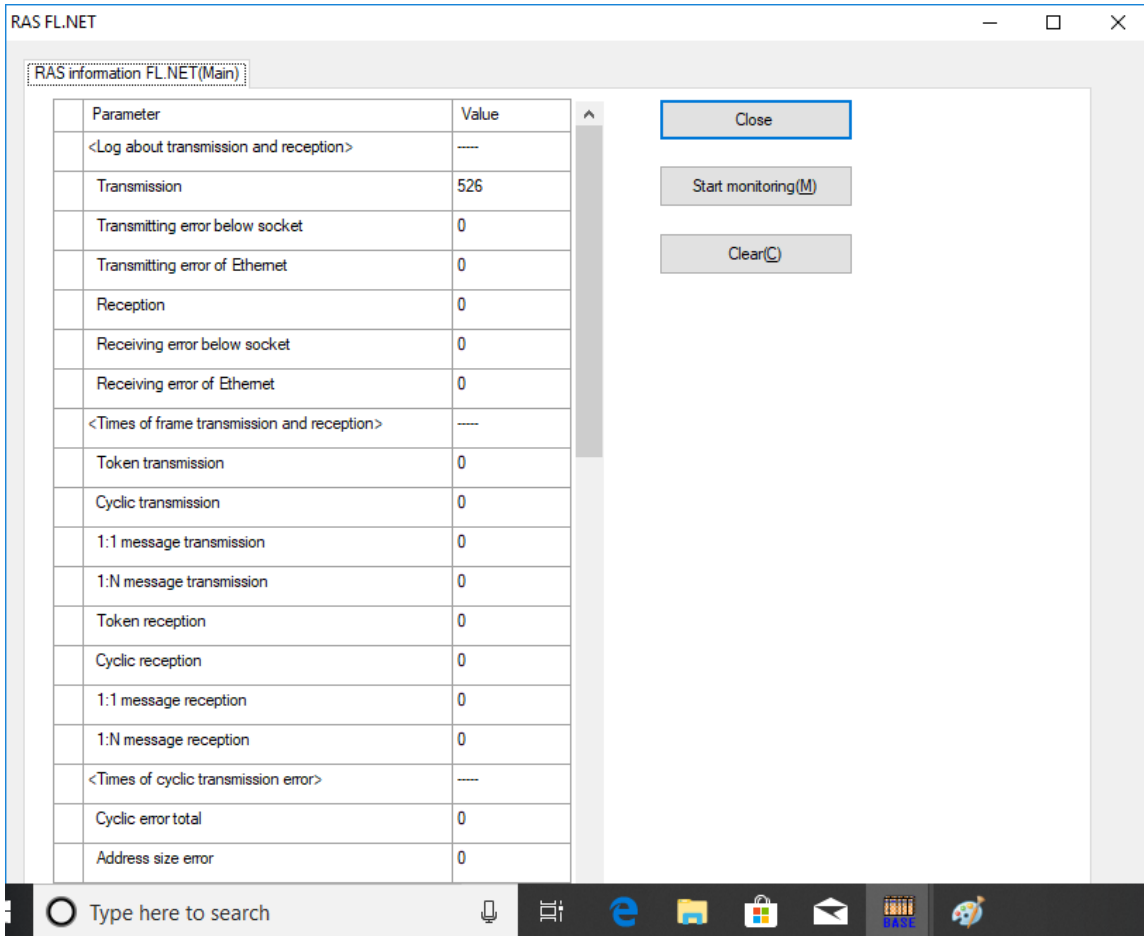


Figure 5-82 [RAS information FL.NET (Main)] tab

- (3) To monitor the RAS information, click the **Start monitoring** button.
 The **Start monitoring** button changes to the **Stop monitoring** button. The tab monitors and displays the RAS information.
 To stop monitoring, click the **Stop monitoring** button.
 The **Stop monitoring** button changes to the **Start monitoring** button. The tab stops monitoring the RAS information.

- (4) To clear the RAS information, click the **Clear** button.

Note that all the RAS information returns to 0 when you reset the PCs or turn it off and then on.

- (5) To close the **RAS information FL.NET (Main)** tab (Figure 5-82), click the **Close** button.

This page is intentionally left blank.

6. Maintenance

6.1 Maintenance and inspection

To use FL.NET modules under optimum conditions, perform the inspections shown in Table 6-1 periodically (at least twice a year).

Table 6-1 Maintenance and inspection items

Item	Description	Frequency
Unit cleaning	Turn off all the power. Use a vacuum cleaner to suck dust from the openings in the case of the FL.NET module. Be careful not to stir up the dust.	At least twice a year
Mechanical check	Check whether the screws and communication cables of the FL.NET module are loose or damaged. If a screw or cable is loose, tighten it. Replace any damaged screws or cables.	At least twice a year



WARNING

- Turn off the power switch of the power supply module before replacing any module. This is to avoid the risk of electrical shock, and to prevent damage and malfunction.

Notice

- Static electricity could cause damage to the module. Before handling the module, discharge any static electricity from your body.

6. Maintenance

6.2 Replacing or adding the module

For details about how to replace or add the FL.NET module, see *Chapter 14. Adding and Replacing Modules* in the *SIOVE User's Manual General Description* (manual number SEE-1-001)).

6.3 Troubleshooting

As a guide for troubleshooting a FL.NET module, use the indicator of the CPU module and LED indicators (ERR LED and LER LED) of the FL.NET module.

For details about how to perform troubleshooting, see *Chapter 13. Troubleshooting* in the *SIOVE User's Manual General Description* (manual number SEE-1-001).

7. Test and Maintenance Diagnosis

7.1 Details of Test and Maintenance diagnosis

Table 7-1 describes details of Test and Maintenance diagnosis for a FL.NET module. You can select between two types of Test and Maintenance by using the MAIN/SUB setting switch of the FL.NET module.

For details about how to start Test and Maintenance, see *7.3 Starting Test and Maintenance*.

Table 7-1 Details of Test and Maintenance diagnosis for FL.NET modules

No.	Test item	Description	Wiring required?
1	T/M1 Stand-alone check	Use this function to check the read write operations for the internal memory and Ethernet LSI of the FL.NET module. You can use this function on a single mounted FL.NET module.	Yes
2	T/M2 Inter-module connection check	Use this function to check Ethernet communications between two mounted FL.NET modules (from the main module to the sub module). This function requires two FL.NET modules to be mounted.	Yes

Note: Make sure that you start Test and Maintenance only after performing wiring according to *7.2 Wiring methods*.

7.2 Wiring method

- Wiring required for T/M1 stand-alone check (No. 1)

Connect the loopback connector to the communication connector on the FL.NET module as shown in Figure 7-1.

Figure 7-2 shows the wiring required for the loopback connector.

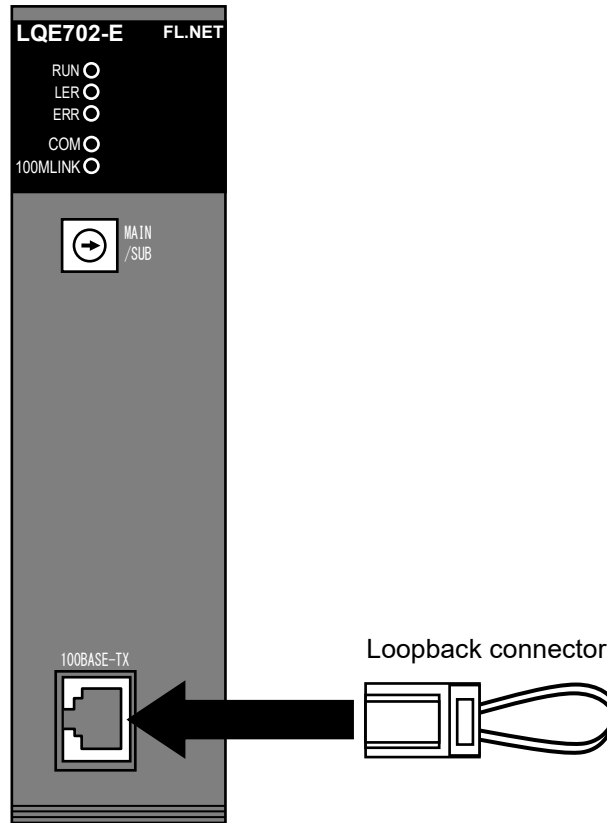


Figure 7-1 Wiring required for stand-alone checks

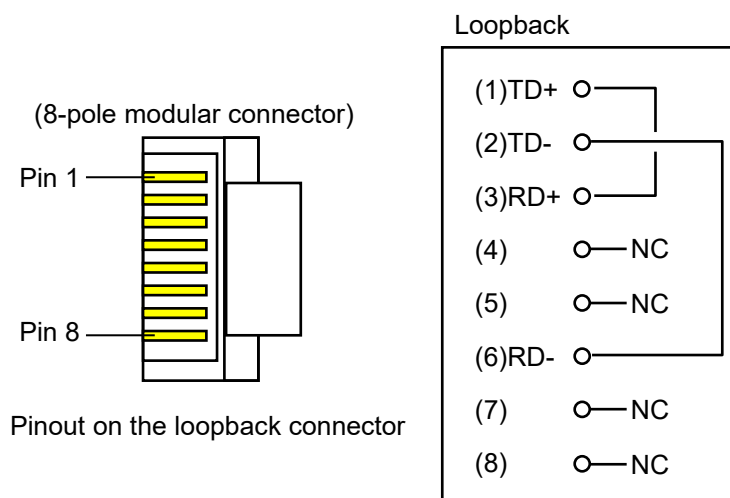


Figure 7-2 Wiring required for the loopback connector

- Wiring required for T/M2 inter-module connection check (No.2)
Connect the FL.NET modules as shown in Figure 7-3. Two FL.NET modules are necessary. For the cable used to connect the modules, make sure that you use a category 5e or greater cross twisted-pair cable.

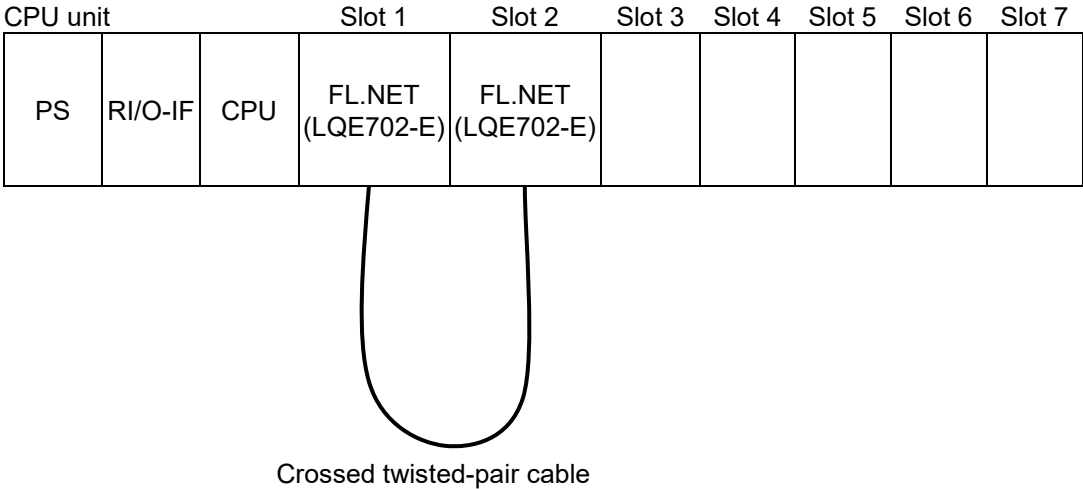


Figure 7-3 Wiring required for inter-module connection check

7.3 Starting Test and Maintenance

Observe the following and perform Test and Maintenance by using steps (1) to (6):

- Back up data.
- Do not perform Test and Maintenance for other modules at the same time.
- Before starting Test and Maintenance, record the settings of all switches and module mount positions of the FL.NET module. This ensures that the original settings can be restored after Test and Maintenance is completed.
- Turn off the power switch of the power supply module.

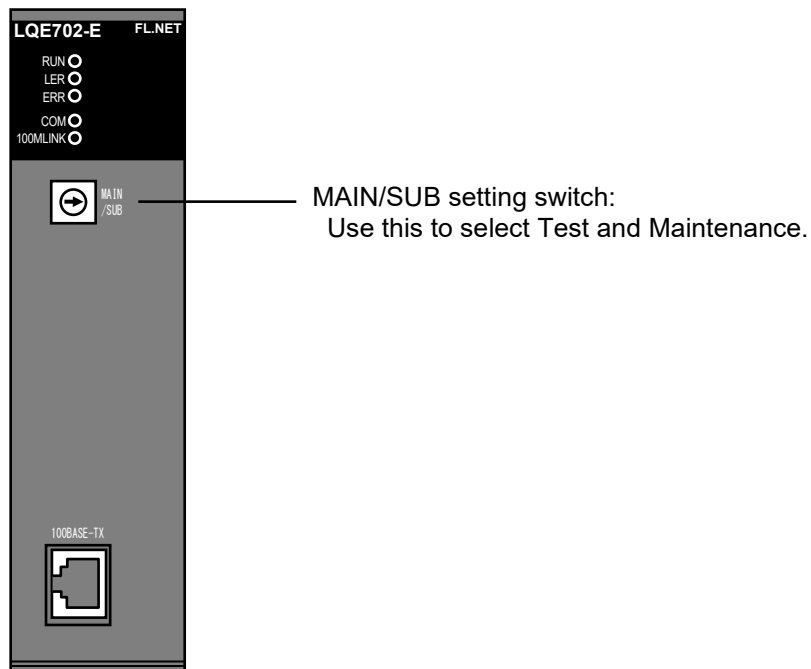
- (1) Connect wires appropriate to the Test and Maintenance operation you will start, as shown in *7.2 Wiring method*.
- (2) Use the MAIN/SUB setting switch of the FL.NET module shown in Figure 7-4 to select the Test and Maintenance operation you want to start. Table 7-2 shows the Test and Maintenance operation (test items) selected by using the MAIN/SUB setting switch, and the CPU module switch settings for starting Test and Maintenance operations.
- (3) Set the CPU RUN/STOP switch and LADDER RUN/STOP switch of the CPU module to RUN.
- (4) Switch on the power to the power supply module. This starts the Test and Maintenance operation. When T/M1 is running, the RUN, LER, and ERR LEDs (three LEDs) blink, and the COM LED is lit. When T/M2 is running, the RUN and COM LEDs (two LEDs) are lit.
- (5) To stop the Test and Maintenance operation, switch off the power to the power supply module.
- (6) To return the system to normal operation, return any modules you removed to their original locations, return each setting switch to its setting before performing Test and Maintenance, and restore the backup data.

Table 7-2 Test items selected by the MAIN/SUB setting switch

MAIN/SUB setting switch	Test item	Module type	Switch settings of the CPU module	
			CPU RUN/STOP	LADDER RUN/STOP
C	T/M2	Main module	RUN	RUN
D		Sub module		
F	T/M1	Main module		
		Sub module		

Up to two FL.NET modules can be mounted in a CPU unit.

For the main module, select C or F on the MAIN/SUB setting switch. For the sub module, select D or F on the MAIN/SUB setting switch.

**Figure 7-4 Switch settings on the FL.NET module**

7.4 Checking errors

If an error is detected during the Test and Maintenance operation, either the LER LED or the ERR LED of the FL.NET module is lit.

If the LER LED or ERR LED is lit as a result of running the Test and Maintenance operation, this means that a failure occurred in the FL.NET module. You need to replace the FL.NET module.

For more detailed information, connect the tool to the CPU unit and check the error log information. For details about the tool, see the *SIOVE User's Manual General Description* (manual number SEE-1-001).

Table 7-3 lists the types of errors that can be detected by T/M1.

Table 7-4 lists the types of errors that can be detected by T/M2.

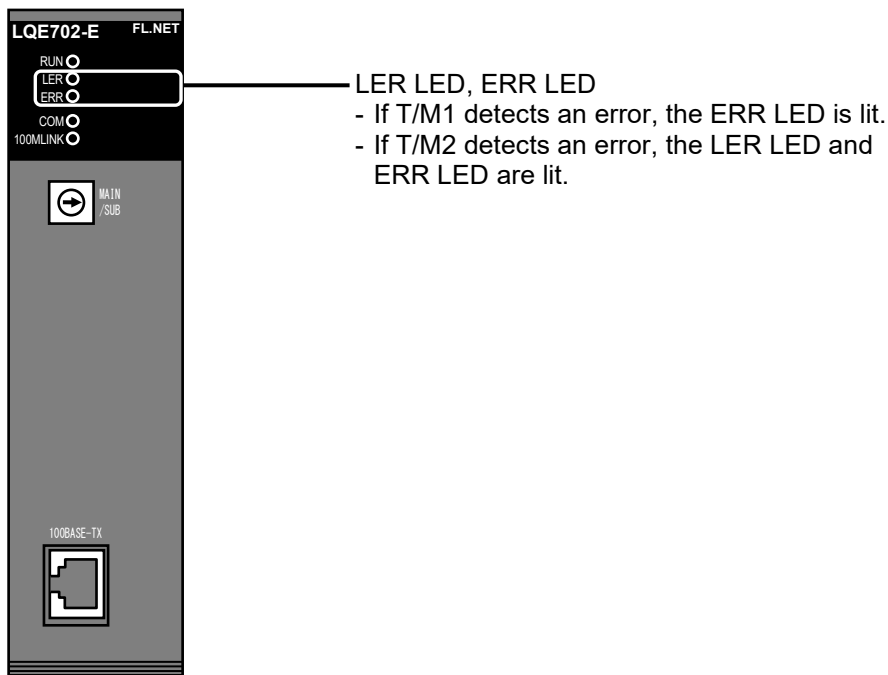


Figure 7-5 LER LED and ERR LED of the FL.NET module

Table 7-3 Information displayed in the tool windows when T/M1 detects an error

Reference address	Displayed information
/00416XX0	Error code of the error that occurred
/00416XX4	Extended information 1
/00416XX8	Extended information 2
/00416XXC	Extended information 3

XX: The number shown here depends on the slot number where the FL.NET module is mounted. The table on the right lists the values that might replace XX for each slot number.

Slot No. where module is mounted	No. at XX
Slot 1	/E4
Slot 2	/E8
Slot 3	/EC
Slot 4	/F0
Slot 5	/F4
Slot 6	/F8
Slot 7	/FC

Table 7-3 Information displayed in the tool windows when T/M1 detects an error (continued)

Error code	Description	Extended information 1	Extended information 2	Extended information 3
/01220001	ECC 2-bit error detected in internal memory (when address data is set)	Starting address of area where error occurred	Not used (fixed at /00000000)	Not used (fixed at /00000000)
/01220002	ECC 1-bit error detected in internal memory (when address data is set)	Starting address of area where error occurred	Not used (fixed at /00000000)	Not used (fixed at /00000000)
/01220003	ECC 2-bit error detected in internal memory (when inverse address data is set)	Starting address of area where error occurred	Not used (fixed at /00000000)	Not used (fixed at /00000000)
/01220004	ECC 1-bit error detected in internal memory (when inverse address data is set)	Starting address of area where error occurred	Not used (fixed at /00000000)	Not used (fixed at /00000000)
/01220011	ECC 1-bit error detected in internal memory (when an address data copy pair is set)	Starting address of area where error occurred	Not used (fixed at /00000000)	Not used (fixed at /00000000)
/01220012	ECC 2-bit error detected in internal memory (when an address data copy pair is set)	Starting address of area where error occurred	Not used (fixed at /00000000)	Not used (fixed at /00000000)
/01220013	ECC 1-bit error detected in internal memory (when an inverse address data copy pair is set)	Starting address of area where error occurred	Not used (fixed at /00000000)	Not used (fixed at /00000000)
/01220014	ECC 2-bit error detected in internal memory (when an inverse address data copy pair is set)	Starting address of area where error occurred	Not used (fixed at /00000000)	Not used (fixed at /00000000)
/01220015	Copy pair error detected in internal memory (address data)	Address where error occurred	Write data	Read data
/01220016	Copy pair error detected in internal memory (inverse address data)	Address where error occurred	Write data	Read data
/01230001	Ethernet LSI reset failure	Not used (fixed at /00000000)	Not used (fixed at /00000000)	Not used (fixed at /00000000)
/01230002	Ethernet link not established	Not used (fixed at /00000000)	Not used (fixed at /00000000)	Not used (fixed at /00000000)
/01230003	Ethernet transmission/reception timeout error (10 ms)	Not used (fixed at /00000000)	Not used (fixed at /00000000)	Not used (fixed at /00000000)
/01230004	Ethernet transmission/reception error	Not used (fixed at /00000000)	Not used (fixed at /00000000)	Not used (fixed at /00000000)
/01230005	Invalid length of Ethernet reception data	Not used (fixed at /00000000)	Not used (fixed at /00000000)	Not used (fixed at /00000000)
/01230006	Ethernet transmission/reception data compare error	Not used (fixed at /00000000)	Not used (fixed at /00000000)	Not used (fixed at /00000000)

Table 7-4 Information displayed in the tool windows when T/M2 detects an error

Reference address		Displayed information
Main module	Sub module	
/00417000	/00417100	Test count
/00417004	/00417104	Join error count
/00417008	/00417108	Compare error count for common memory area 1
/0041700C	/0041710C	Compare error count for common memory area 2
/00417010	/00417110	Word block read error count
/00417014	/00417114	Word block write error count
/00417018	/00417118	Network parameter read error count
/0041701C	/0041711C	Network parameter write error count
/00417020	/00417120	Message return request error count
/00417024	/00417124	Error count for transparent message transmission requests
/00417028	/00417128	Error count for transparent message reception requests
/0041702C	/0041712C	Not used
/00417030	/00417130	FA link status flag
/00417034	/00417134	Self-node status flag
/00417038	/00417138	Message error code
/0041703C	/0041713C	Not used

Appendix

A.1 Network system definitions

A.1.1 Compliance with communication protocol standards

A communication protocol is a set of rules (communication conventions) that enable a system to exchange information with another system through a communication line. The communication protocols adopted by FL-net comply with the standards shown in Table A-1.

Table A-1 FL-net communication protocols

FL-net communication protocol	Standard complied with
FL-net	FA Control Network [FL-net(OPCN-2)] - Protocol Specifications (issued by JEMA (Japan Electrical Manufacturers' Association))
UDP	RFC768
IP, ICMP, etc.	RFC791, RFC792, RFC919, RFC922, RFC950
ARP, etc.	RFC826, RFC894
--	IEEE802.3

A.1.2 Layer structure of the communication protocol

A communication protocol is modeled in a layer structure, standardizing the communication process by systematically dividing it into various levels. FL-net consists of the following six protocol layers:

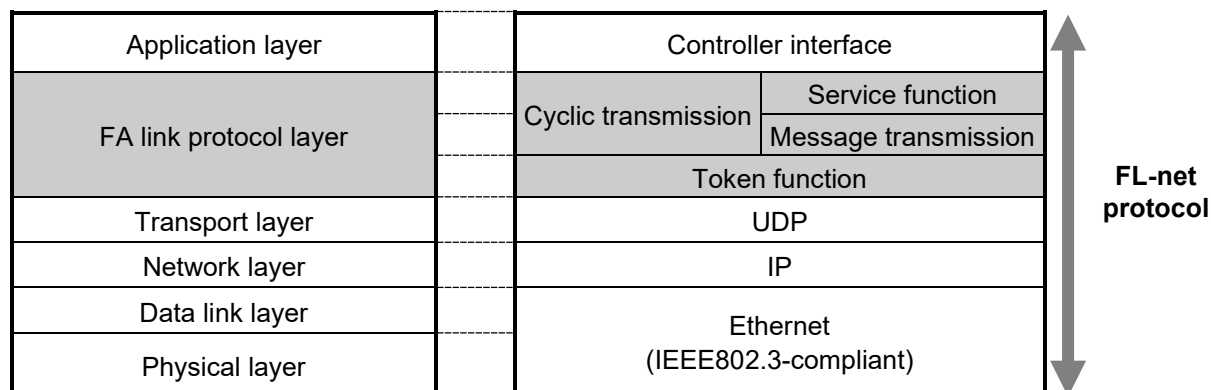


Figure A-1 Positioning of the FA link protocol

A.1.3 FL-net physical layer

For transmission speeds of 100 Mbps, the Ethernet physical layer offers the following five transmission methods:

100BASE-TX, 100BASE-FX, 100BASE-T, 100BASE-T2, 100BASE-T4

Of these, FL-net adopts 100BASE-TX.

A.1.4 IP addresses

Addresses called "IP addresses" ("INET addresses") are used to differentiate and designate the many communication devices connected to each other in an Ethernet network. For this reason, every communication device connected to an Ethernet network must be configured with its own unique IP address. An IP address is divided into two sections: the section containing the network address to which the communication device is connected, and the section containing the host address of the communication device. IP addresses can be categorized into three different network classes (A, B, and C), which are chosen according to the network size. There are additional classes, D and E, to be used for special purposes.

Table A-2 Classes of IP addresses

	First 1-octet value	Network address section	Host address section
Class A	0 to 127	xxx . xxx . xxx . xxx	xxx . xxx . xxx . xxx
Class B	128 to 191	xxx . xxx . xxx . xxx	xxx . xxx . xxx . xxx
Class C	192 to 223	xxx . xxx . xxx . xxx	xxx . xxx . xxx . xxx

Shaded xxx . portions represent the associated address sections.

The IP addresses of communication devices connected to a network all have the same network address section. Each IP address has a unique host address section that is not duplicated within the network. FL-net uses a Class C IP address. It is recommended that you set the network address to 192 . 168 . 250 . *N* (where *N* is a node number between 1 and 254). It is also recommended that the low-order host address matches the FL-net protocol node number.

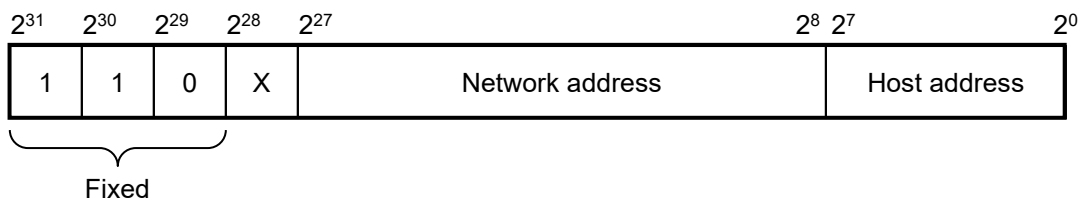


Figure A-2 FL-net IP address

A.1.5 Subnet mask

The subnet mask is fixed at 255 . 255 . 255 . 0 in compliance with the FL-net protocol. FL-net users must always use a subnet mask setting of 255 . 255 . 255 . 0.

This value has the same network address section and host address section as for Class C.

A.1.6 TCP/IP and UDP/IP communication protocols

TCP, UDP, and IP are important protocols used in Ethernet.

IP corresponds to the network layer of the communication protocol, and controls the flow of communications data.

TCP and UDP correspond to the transport layer. They both use IP for network layer communications, but greatly differ in the services they provide.

TCP offers highly reliable services while keeping higher layers unaware of its data segments.

UDP, on the other hand, functions by transmitting a datagram (one pack or unit of information) from IP to a higher layer, without guaranteeing delivery of the data to its destination. UDP lets a higher layer perform data reception acknowledgment, retransmission, and other processes.

UDP is not as reliable as TCP. However, it provides communication services with a minimum of overhead.

FL-net uses UDP. The reason for this is that TCP's complicated data verification and retransmission procedures are redundant for FL.NET. FL-net provides faster data exchange speeds by skipping such complicated procedures and having the FL-net protocol layer on a higher level perform token-based transmission permissions management, multiple-frame division and synthesis, and other relevant processes.

A.1.7 Port numbers

FL-net predetermines port numbers as shown in Table A-3 to ensure that services are implemented by the FL-net protocol layer, which is positioned higher than the transport layer. Note, however, that the FL-net user does not have to set these port numbers by using parameters and the like.

Table A-3 FL-net port numbers

Name	Port number
Cyclic transmission port number	55000 (fixed)
Message communication port number	55001 (fixed)
Join request frame port number	55002 (fixed)
Transmission port number	55003 (fixed)

A.1.8 FL-net data format

(1) Overview of the FL-net data format

Data transmitted and received by FL-net is encapsulated in each communication protocol layer as shown in Figure A-3.

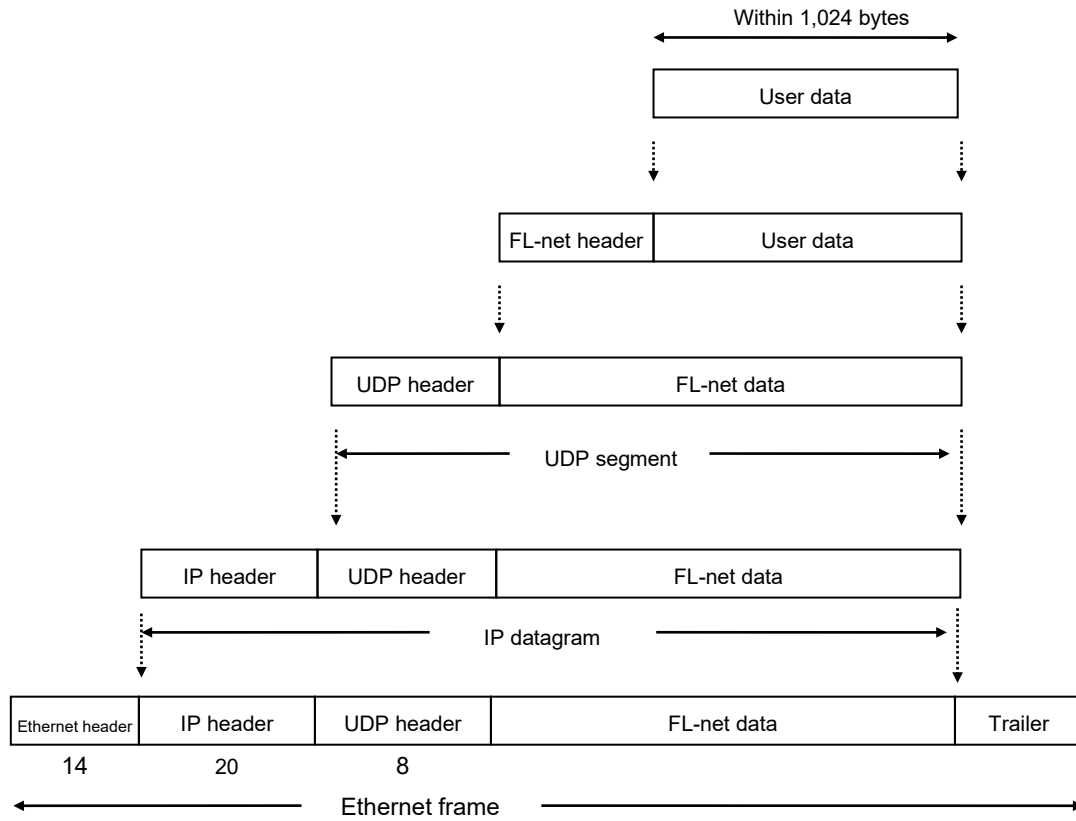


Figure A-3 Overview of the FL-net data format

Figure A-4 shows an example of one frame's worth of FL-net data observable over a communication line. In the example, 128 bytes of cyclic data is transferred.

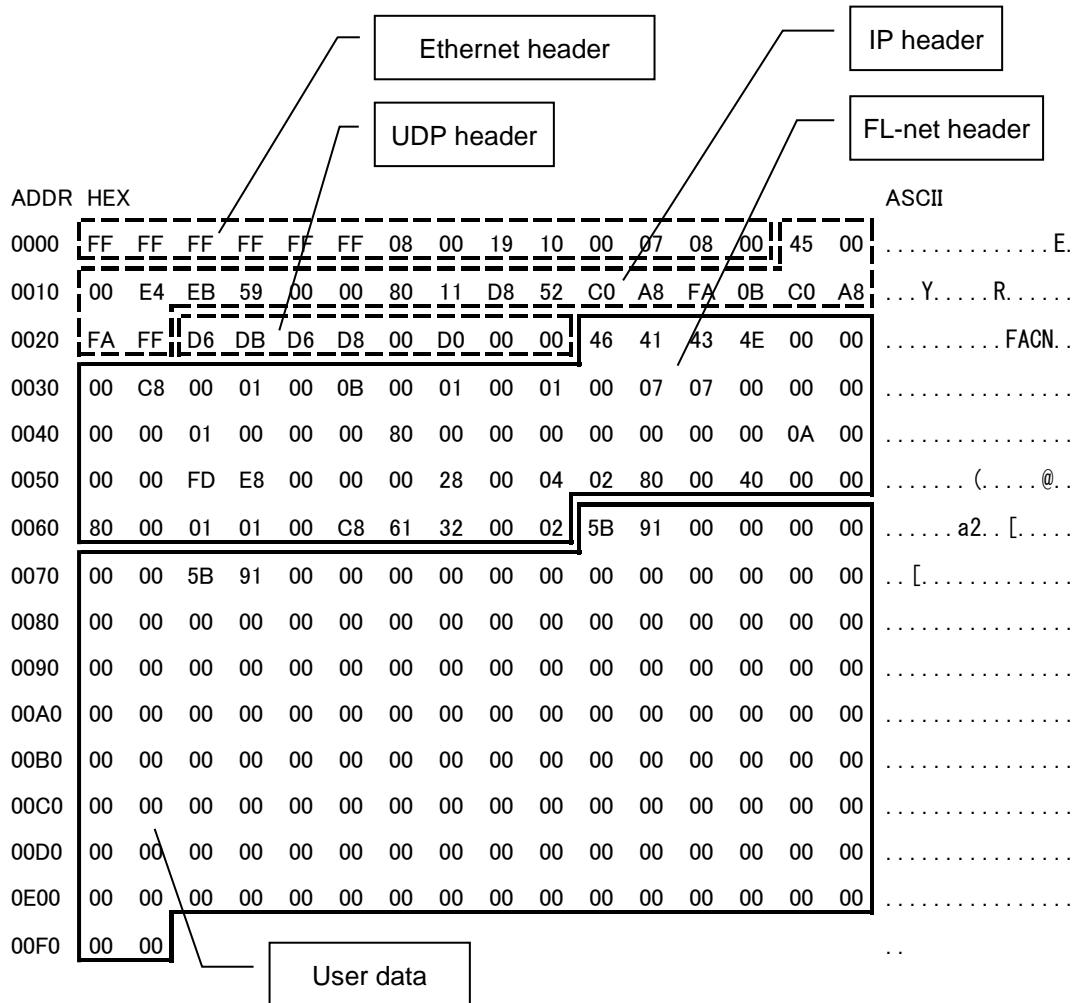


Figure A-4 Example of FL-net data (one frame)

(2) FL-net header format

The FL-net header consists of 64 to 96 bytes.

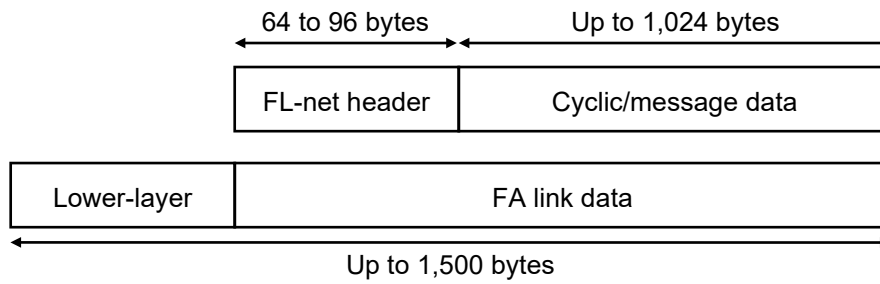


Figure A-5 FL-net header

The FL-net header is attached to the beginning of every frame related to the FL-net protocol.

A.1.9 FL-net transaction codes

FL-net provides the following message transmission services:

- Word block data read
- Word block data write
- Network parameter read
- Network parameter write
- Stop directive
- Run directive
- Profile read
- Log data read
- Log data clear
- Message return
- Transparent type

The header of each message contains a request transaction code (TCD) or response transaction code that identifies the message frame.

Table A-4 List of transaction codes

Transaction code	Description
0 to 9999	Reserve
10000 to 59999	Transparent message
60000 to 64999	Reserve
65000	Token frame
65001	Cyclic frame
65002	Join request frame header
65003	Byte block data read (request) (reserve)
65004	Byte block data write (request) (reserve)
65005	Word block data read (request)
65006	Word block data write (request)
65007	Network parameter read (request)
65008	Network parameter write (request)
65009	Stop directive (request)
65010	Run directive (request)
65011	Profile read (request) (reserve)
65012	Trigger header
65013	Log read (request)
65014	Log clear (request)
65015	For message return testing (request)
65016 to 65202	Reserve (for future extension)
65203	Byte block data read (response)
65204	Byte block data write (response)
65205	Word block data read (response)
65206	Word block data write (response)
65207	Network parameter read (response)
65208	Network parameter write (response)
65209	Stop directive (response) (reserve)
65210	Run directive (response) (reserve)
65211	Profile read (response)
65212	Reserve
65213	Log data read (response)
65214	Log data clear (response)
65215	For message return testing (response)
65216 to 65399	Reserve (for future extension)
65400 to 65535	Reserve

A.1.10 Transaction code receive operation at a UDP port

Table A-5 shows the operations performed upon receipt of frames at a UDP port defined in FL-net, according to the transaction code.

Table A-5 Processing of transaction code reception at a UDP port

Transaction code	For token frame or cyclic frame (UDP port = 55000)	For message frame (UDP port = 55001)	For trigger frame or join request frame (UDP port = 55002)
00000 to 09999	Discard	Process or discard	Discard
10000 to 59999	Discard	Process (for transparent message)	Discard
60000 to 64999	Discard	Discard	Discard
65000 to 65001	Process	Discard	Discard
65002	Discard	Discard	Process
65003 to 65011	Discard	Process	Discard
65012	Discard	Discard	Process
65013 to 65016	Discard	Process	Discard
65017 to 65199	Discard	Perform unsupported processing	Discard
65200 to 65202	Discard	Discard	Discard
65203 to 65211	Discard	Process	Discard
65212	Discard	Discard	Discard
65213 to 65216	Discard	Process	Discard
65217 to 65399	Discard	Process or Discard	Discard
65400 to 65535	Discard	Discard	Discard

<Remarks> The same transaction code is not defined twice, even if the UDP port number is different.

A.2 FL-net network management

A.2.1 FL-net token management

(1) Tokens

As a rule, a node can transmit data when it has the token.

A node without the token can transmit only two items: a token reissue request in the event of a token monitoring timeout, and a join request frame in order to join the network.

- The FA net circulates one token among the nodes.
- When a node receives the token, the node has permission to transmit data to the network until it passes the token to the next node.
- The token circulates among all nodes that have joined the FL-net network.
- The token is monitored by a timer so that if it stops circulating through the network for a predetermined time, a new token is reissued automatically.
- If two or more tokens exist in the network, a function works to ensure that only one token takes precedence.

(2) Token circulation

As a rule, there is only one token in a network.

If two or more tokens exist in the network, the token with the smallest destination node number takes precedence, and all other tokens are discarded.

A frame containing a token (token frame) contains the token's destination node number and transmission node number.

A node becomes the holder of a token when its node number matches the destination node number of the token frame it receives.

The order of token rotation is determined by node numbers.

The token is passed around in a rotation in ascending order, among the nodes registered in the participating node management table.

The node with the largest node number passes the token to the node with the smallest node number.

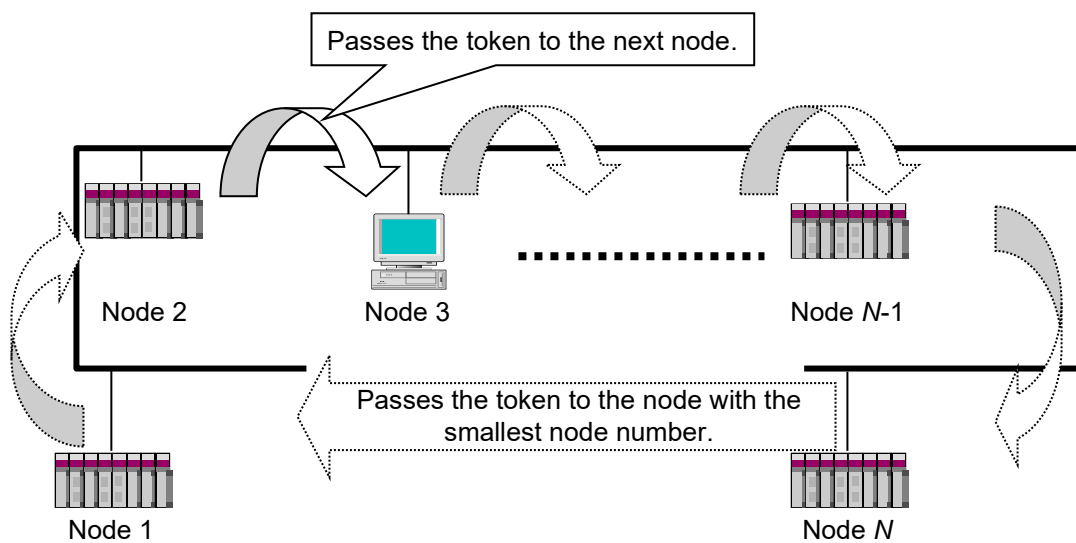
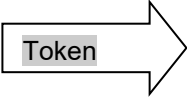
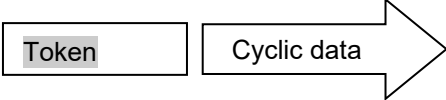
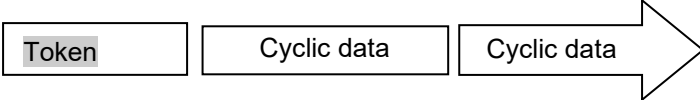
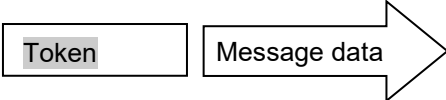
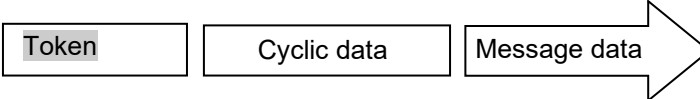
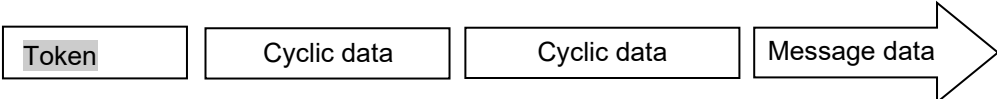


Figure A-6 Token circulation

(3) Tokens and data

Table A-6 shows the six different data patterns used for token transmission.

Table A-6 Tokens and data

Pattern	Description
When there is no data to be transmitted	Only the token is transmitted.
	
When only cyclic data is to be transmitted	The token is transmitted after cyclic data is transmitted.
	
When only cyclic data is to be transmitted after being divided into frames	The cyclic data is transmitted, and after the last frame, the token is transmitted.
	
When only message data is to be transmitted	The message data is transmitted, and then the token is transmitted.
	
When cyclic data and message data is to be transmitted	The message data and cyclic data is transmitted, and then the token is transmitted.
	
When cyclic data and message data is to be transmitted with the cyclic data divided into frames	The message data and cyclic data is transmitted, and after the last frame, a token is transmitted.
	

(4) Frame interval (minimum frame interval)

The frame interval is the time interval between the instant at which the self-node receives the token from another node and the instant at which the self-node starts frame transmission. The minimum frame interval is the minimum period of time a node must wait to initiate a frame transmission.

In an FL-net network, this minimum frame interval is shared across the whole network.

Every time a node joins or leaves the network, the maximum value of the minimum frame intervals set by nodes that have joined the network is calculated and updated for all nodes.

A.2.2 Joining and leaving an FL-net network

(1) Joining an FL-net network

When a node starts up, it monitors the line until a join token detection timeout occurs. If the node does not receive a token during this time, the node concludes that network startup is in progress, and joins the network as a new node. On the other hand, if the node receives a token, the node concludes that the network has already started, and joins the network midway.

(a) Joining as a new node

If a node does not receive a token before the join token detection timeout, the node prepares for trigger transmission and then starts transmission after a predetermined time (determined by dividing the node number by 8, and multiplying the remainder by 4 ms). If trigger reception occurs before trigger transmission, the node does not transmit a trigger. During the join request frame reception wait time (1,200 ms) after trigger reception, all nodes wait to transmit a join request frame while checking for duplicate node numbers or addresses and updating the participating node management table. When the join request frame transmission wait time (node number x 4 ms) has elapsed since trigger reception, the nodes transmit a join request frame. If a node recognizes a duplicate address in a join request frame of another node, the former node sets the common memory starting address and common memory size for areas 1 and 2 to 0 and does not transmit cyclic data. When a node recognizes a duplicate address, it sets the address duplication flag and resets the common memory data validity notification flag. When a join request frame reception wait timeout occurs, the participating node with the smallest node number transmits the token first, in accordance with the participating node management table. When a node recognizes a duplicate node number, it does not perform any transmission or reception.

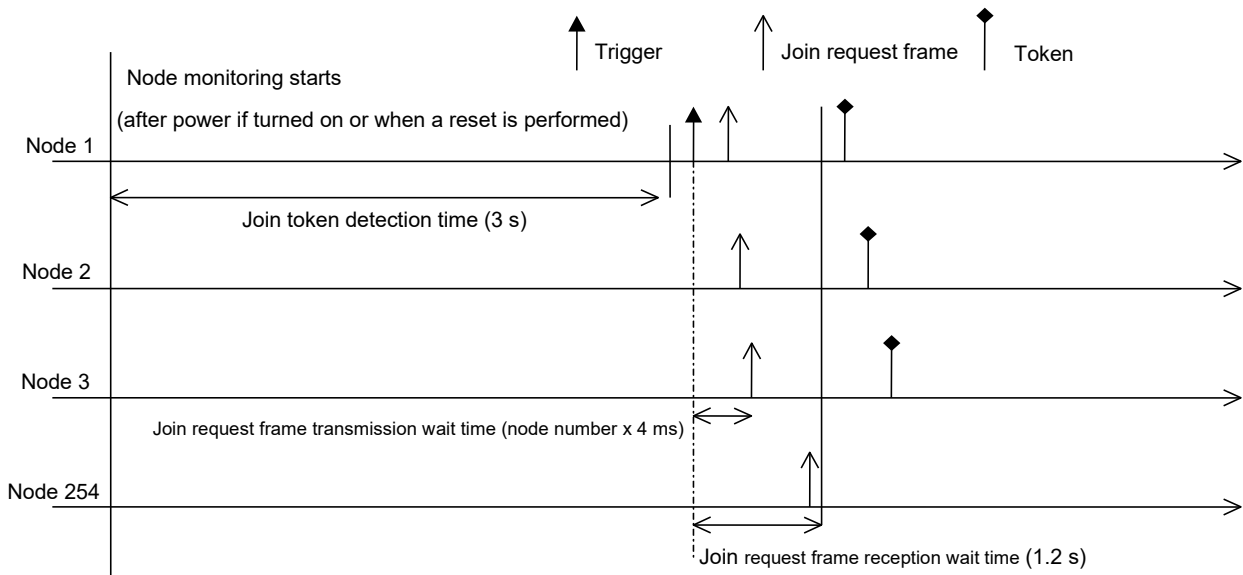


Figure A-7 Startup time chart 1

(b) Joining midway

If a node receives a token within the join token detection time, it concludes that a link has been established, and does not transmit a join request frame until the token makes three rounds. Frames received during this time are checked for duplicate values such as duplicate node numbers and addresses, and the participating node management table is updated accordingly. If a duplicate address is detected, the common memory starting address and common memory size of areas 1 and 2 are set to 0, and cyclic data transmission is not performed. If a node recognizes a duplicate address, it sets the address multiplexing flag and resets the common memory data validity notification flag. If the node number is normal, the node transmits a join request frame after the join request frame transmission wait time has elapsed. The join request frame is sent whether or not the node is holding the token. If a node recognizes a duplicate node number, it does not transmit a join request frame and does not join the network.

<Remarks>

Join token detection time: This is the time period for checking the network operation status.

Round: Rounds are determined based on the time at which the token addressed to the smallest node number is received.

Join request frame transmission wait time: A join request frame is transmitted after a predetermined time has elapsed (self-node number x 4 ms) in order to avoid conflicts with any other new nodes joining the network.

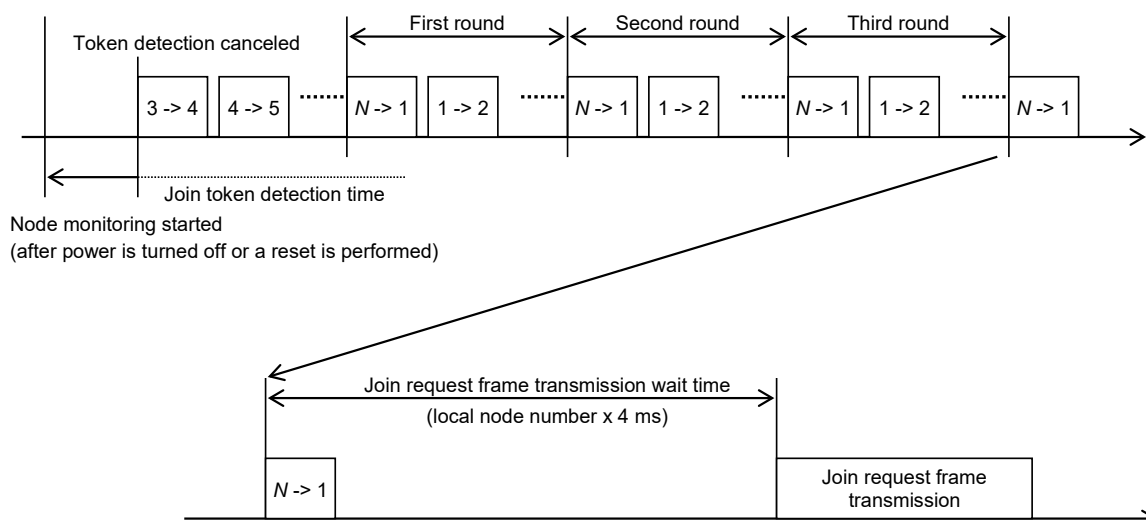


Figure A-8 Startup time chart 2

(2) Leaving an FL-net network

A node performs a node number check every time it receives a token frame. If three token frames are not received from a particular node in succession, that node is considered to have left the FL-net network (including cases where the node holding the token does not transmit the token even after the token monitoring time has elapsed).

When a node is considered to have left the network as explained in this section, information about that node is deleted from the management tables.

A.2.3 Node status management

Node status management is performed by using the self-node management table, participating node management table, and network management table. Table A-7 gives an overview of node status management.

Table A-7 Overview of the node status management table

Name	Description
Self-node management table	Manages the self-node settings
Participating node management table	Manages information on the nodes that have joined the network
Network management table	Manages information shared across the network

A.2.4 FL-net self-node management table

(1) Basic functions

This table manages data about the self-node settings. The following outlines the functions of this table:

- The table is used to read join request frames and network parameters.
- Management data is set by a higher FL-net layer at the time a node is started.
- The node name, and the starting address and size of the transmission area of the common memory can be set from the network.

(2) Management data

Table A-8 Self-node management table

Item	Length	Description
Node number	1 byte	1 to 254
Starting address of area 1 data for the common memory	2 bytes	Word address (0 to 0x1Ff)
Size of area 1 data for the common memory	2 bytes	Size (0 to 0x1Ff)
Starting address of area 2 data for the common memory	2 bytes	Word address (0 to 0x1Fff)
Size of area 2 data for the common memory	2 bytes	Size (0 to 0x1Fff)
Higher layer status	2 bytes	RUN/STOP/ALARM/WARNING/NORMAL
Token monitoring timeout	1 byte	In units of 1 ms
Minimum frame interval	1 byte	In units of 100 us
Vendor name	10 bytes	Vendor name
Manufacturer model	10 bytes	Manufacturer model, device name
Node name (equipment name)	10 bytes	User-defined node name
Protocol version	1 byte	Fixed at 0x80
FA link status	1 byte	Joined or left the network, etc.
Self-node status	1 byte	Duplicate node number detected, etc.

A.2.5 Participating node management table for an FL-net network

(1) Basic functions

The statuses of nodes that have joined the network are monitored by using each node's participating node management table. This table handles data for an individual node that has joined the network; that is, data that can be managed on a per-node basis. The following summarizes the basic functions of this table:

- When a node receives a token frame upon startup, its participating node management table and network management table are updated.
- Every time a node receives a token frame, it updates its participating node management table.
- When a join request frame is received from a new node, its participating node management table is updated.
- If a token frame is not received from a node or a timeout is detected for that node three times in succession, the node is deleted from the table.

(2) Management data

The token of each node is continuously monitored to create and manage its participating node management table (Table A-9).

Table A-9 Participating node management table

Item	Length	Description
Node number	1 byte	1 to 254
Higher layer status	2 bytes	RUN/STOP/ALARM/WARNING/NORMAL
Starting address of area 1 data for the common memory	2 bytes	Word address (0 to 0x1Ff)
Size of area 1 data for the common memory	2 bytes	Size (0 to 0x1FF)
Starting address of area 2 data for the common memory	2 bytes	Word address (0 to 0x1Fff)
Size of area 2 data for the common memory	2 bytes	Size (0 to 0x1Fff)
Refresh cycle time limit	2 bytes	In units of 1 ms
Token monitoring time	1 byte	In units of 1 ms
Minimum frame interval	1 byte	In units of 100 us
Link status	1 byte	Joined or left the network, etc.

0x1FFF is a hexadecimal number (1FFF hex).

<Remarks> This is contained in a received token frame.

A.2.6 FL-net status management

(1) Basic functions

Management is provided for parameters related to the network status.

(2) Management data

Table A-10 Network management table

Item	Length	Description
Number of node that has the token	1 byte	Node currently holding the token
Minimum frame interval	1 byte	In units of 100 us
Refresh cycle time limit	2 bytes	In units of 1 ms
Measured refresh cycle time (current)	2 bytes	In units of 1 ms
Measured refresh cycle time (maximum)	2 bytes	In units of 1 ms
Measured refresh cycle time (minimum)	2 bytes	In units of 1 ms

A.2.7 FL-net message sequence number management

(1) Basic functions

Management is provided for message transmission sequence numbers and sequence version numbers.

(2) Transmission management data

Table A-11 Transmission management data for message sequence number management

Item	Length	Description
Sequence number/version number	4 bytes	Sequence number/version number for message transmission
Sequence number (one-to- <i>N</i> transmission)	4 bytes	0x1 to 0xFFFFFFFF
Sequence number (one-to-one transmission)	4 bytes x 256	0x1 to 0xFFFFFFFF

0xFFFFFFFF is a hexadecimal number (FFFFFFFF hex).

(3) Reception management data

Table A-12 Reception management data for message sequence number management

Item	Length	Description
Sequence number/version number	4 bytes	0x1 to 0xffffffff
Sequence number (one-to-one reception)	4 bytes	0x1 to 0xffffffff
Sequence number (one-to- <i>N</i> reception)	4 bytes	0x1 to 0xffffffff

0xFFFFFFFF is a hexadecimal number (FFFFFFFF hex).

A.3 Profile of FL-net

A.3.1 Types of device communication information

In FL-net, information about the communications of devices connected to the network is classified into three types as shown in Figure A-9.

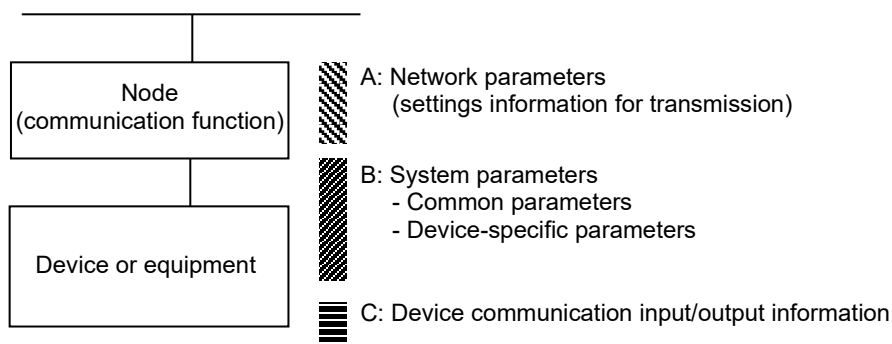


Figure A-9 Types of device communication information

- Network parameters (A) are settings information that is required for transmission.
- System parameters (B) are static parameters in management information that are for identifying the kinds of devices on the network. These parameters are divided into common parameters and device-specific parameters.
- Device communication input/output information (C) can be accessed by other devices on the network as needed for applications. This information includes dynamic information, which changes depending on application operations and device status.

A.3.2 Details of common parameters

Table A-13 shows details about the common parameters.

Table A-13 Details of common parameters

Parameter name	Name string (PrintableString type) Length, string	Data type (type)	Contents of parameter Length, contents
Device profile common specification version	6, "COMVER"	INTEGER	Example: 1, 1
System parameter identification string	2, "ID"	PrintableString	7, "SYSPARA"
System parameter revision number	3, "REV"	INTEGER	Example: 1, 0
System parameter revision date	7, "REVDATE"	[INTEGER], 2, (0001 to 9999), [INTEGER], 1, (01 to 12), [INTEGER], 1, (01 to 31)	Example: 2, 1998 1, 9 1, 30
Device category	10, "DVCATEGORY"	PrintableString	Example: 3, "PLC" #
Vendor name	6, "VENDOR"	PrintableString	Example: 4, "MSTC"
Product model	7, "DVMODEL"	PrintableString	Example: 3, "JOP"

#: The following describes the contents of the device category parameter:

"PC" or "PLC" ... Programmable controller

"NC" or "CNC" ... Computer numerical control

"RC" or "ROBOT" ... Robot controller

"COMPUTER" ... Personal computer, panel computer, workstation, display, or other computer

"SP-*.*)" ... To be specified by the vendor (*.): Single-byte alphanumeric characters)

"OTHER" ... Other

As a transfer syntax rule, entire system parameter sets, common parameter sets, system parameter revision dates, and entire device-specific parameter sets (optional) must be of a SEQUENCE structure type (device-specific inner structuring of the parameter is optional).

A.3.3 Details of device-specific parameters (when used)

Table A-14 Details of device-specific parameters

Parameter name	Name string	Data type	Contents of parameter
Device-specific parameter identification string	2, "ID"	PrintableString	7, "DEV PARA"
The vendor freely defines the contents for each device.			

A.3.4 Example system parameters (for PLC)

- (1) Example of system parameters for a PLC in tabular document notation

Table A-15 System parameters in tabular document notation (for PLC)

Parameter name	Name string (PrintableString type) Length, string	Data type (type)	Contents of parameter Length, contents
SysPara			
Device profile common specification version	6, "COMVER"	INTEGER	1, 1
System parameter identification string	2, "ID"	PrintableString	7, "SYSPARA"
System parameter revision number	3, "REV"	INTEGER	1, 0
System parameter revision date	7, "REVDATE"	[INTEGER], 2, (0001 to 9999), [INTEGER], 1, (01 to 12), [INTEGER], 1, (01 to 31)	2, 1998 1, 9 1, 30
Device category	10, "DVCATEGORY"	PrintableString	3, "PLC"
Vendor name	6, "VENDOR"	PrintableString	29, "MSTC-JOP Electric Corporation"
Product model	7, "DVMODEL"	PrintableString	5, "PLC-M"

Parameter name	Name string	Data type	Contents of parameter
PlcmPara			
Device-specific parameter identification string	2, "ID"	PrintableString	7, "DEVPARA"
CPU1 name	8, "CPU1NAME"	PrintableString	9, "PMSP35-5N"
CPU2 name	8, "CPU2NAME"	PrintableString	9, "PMSP25-2N"
CPU3 name	8, "CPU3NAME"	PrintableString	9, "PMSP25-2N"
CPU4 name	8, "CPU4NAME"	PrintableString	9, "PMBP20-0N"
Module 105 name	9, "IO105NAME"	PrintableString	9, "PMWD64-4N"
Module 106 name	9, "IO106NAME"	PrintableString	9, "PMLD01-0N"
Module 107 name	9, "IO107NAME"	PrintableString	9, "PMLE01-5N"

(2) Abstract syntax

<Type definition>

```

PlcmRecord ::=SEQUENCE
    {
        syspara          SysparaType,
        plcmpara         PlcmType
    }

SysparaType ::=SEQUENCE
    {
        nameCOMVER       NameType,
        paraCOMVER       INTEGER,
        nameID           NameType,
        paraID           NameType,
        nameREV          NameType,
        paraREV          INTEGER,
        nameREVDATE      NameType,
        paraREVDATE      DateType,
        nameDVCATEGORY   NameType,
        paraDVCATEGORY   NameType,
        nameVENDOR       NameType,
        paraVENDOR       NameType,
        nameDVMODEL      NameType,
        paraDVMODEL      NameType
    }

PlcmType ::=
SEQUENCE
    {
        nameID           NameType,
        paraID           NameType,
        module           SEQUENCE OF ModInfo
        DEFAULT { }
    }

NameType ::= PrintableString
DateType ::= SEQUENCE
    {
        year            INTEGER,
        month           INTEGER,
        day             INTEGER
    }

ModInfo ::= SEQUENCE
    {
        nameMODULE      NameType,
        paraMODULE      NameType
    }

```

<Value definition>

```

{
  syspara {
    nameCOMVER      "COMVER",
    paraCOMVER      1,
    nameID          "ID",
    paraID          "SYSPARA",
    nameREV         "REV",
    paraREV         0,
    nameREVDATE     "REVDATE",
    paraREVDATE     { year 1998,
                    month 9,
                    day 30 },
    nameDVCATEGORY "DVCATEGORY",
    paraDVCATEGORY "PLC",
    nameVENDOR      "VENDOR",
    paraVENDOR      "MSTC-JOP Electric Corporation",
    nameDVMODEL     "DVMODEL",
    paraDVMODEL     "PLC-M"
  }
  plcpara {
    nameID          "ID",
    paraID          "DEV PARA",
    module
    {
      { nameMODULE "CPU1NAME",
        paraMODULE "PMSP35-5N" },
      { nameMODULE "CPU2NAME",
        paraMODULE "PMSP25-2N" },
      { nameMODULE "CPU3NAME",
        paraMODULE "PMSP25-2N" },
      { nameMODULE "CPU4NAME",
        paraMODULE "PMBP20-0N" },
      { nameMODULE "IO105NAME",
        paraMODULE "PMWD64-4N" },
      { nameMODULE "IO106NAME",
        paraMODULE "PMLD01-0N" },
      { nameMODULE "IO107NAME",
        paraMODULE "PMLE01-5N" }
    }
  }
}

```


(3) Transfer syntax data array (encoding)

Identifier	Length	Contents
\$30	\$820133	
\$30	\$7D	
\$13	\$06	"COMVER"
\$02	\$01	1
\$13	\$02	"ID"
\$13	\$07	"SYSPARA"
\$13	\$03	"REV"
\$02	\$01	0
\$13	\$07	"REVDATE"
\$30	\$0A	
\$02	\$02	\$07CE
\$02	\$01	\$09
\$02	\$01	\$1E
\$13	\$0A	"DVCATEGORY"
\$13	\$03	"PLC"
\$13	\$06	"VENDOR"
\$13	\$1D	"MSTC-JOP Electric Corporation"
\$13	\$07	"DVMODEL"
\$13	\$05	"PLC-M"
\$30	\$81B1	
\$13	\$02	"ID"
\$13	\$07	"DEVPARA"
\$30	\$15	
\$13	\$08	"CPU1NAME"
\$13	\$09	"PMSP35-5N"
\$30	\$15	
\$13	\$08	"CPU2NAME"
\$13	\$09	"PMSP25-2N"
\$30	\$15	
\$13	\$08	"CPU3NAME"
\$13	\$09	"PMSP25-2N"
\$30	\$15	
\$13	\$08	"CPU4NAME"
\$13	\$09	"PMBP20-0N"
\$30	\$16	
\$13	\$09	"IO105NAME"
\$13	\$09	"PMWD64-4N"
\$30	\$16	
\$13	\$09	"IO106NAME"
\$13	\$09	"PMLD01-0N"
\$30	\$16	
\$13	\$09	"IO107NAME"
\$13	\$09	"PMLE01-5N"

(4) Arrangement of data on the circuit

The following shows the sequence of data transmitted over a circuit.

Data transmission begins from address (0) in the relative address 00 column in the following table. Data is transmitted, byte by byte, in the order indicated by the horizontal arrow. Transmission of data in the relative address 00 column is followed by transmission of data in the relative address 10 column, and so on in order of relative address.

Relative address	Data (hexadecimal notation)															
	(0)	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(A)	(B)	(C)	(D)	(E)	(F)
00	30	82	01	33	30	7D	13	06	"C"	"O"	"M"	"V"	"E"	"R"	02	01
10	01	13	02	"I"	"D"	13	07	"S"	"Y"	"S"	"P"	"A"	"R"	"A"	13	03
20	"R"	"E"	"V"	02	01	00	13	07	"R"	"E"	"V"	"D"	"A"	"T"	"E"	30
30	0A	02	02	07	CE	02	01	09	02	01	1E	13	0A	"D"	"V"	"C"
40	"A"	"T"	"E"	"G"	"O"	"R"	"Y"	13	03	"P"	"L"	"C"	13	06	"V"	"E"
50	"N"	"D"	"O"	"R"	13	1D	"M"	"S"	"T"	"C"	"_"	"J"	"O"	"P"	" "	"E"
60	"l"	"e"	"c"	"t"	"r"	"i"	"c"	" "	"C"	"o"	"r"	"p"	"o"	"r"	"a"	"t"
70	"I"	"o"	"n"	13	07	"D"	"V"	"M"	"O"	"D"	"E"	"L"	13	05	"P"	"L"
80	"C"	"_"	"M"	30	81	B1	13	02	"I"	"D"	13	07	"D"	"E"	"V"	"P"
90	"A"	"R"	"A"	30	15	13	08	"C"	"P"	"U"	"1"	"N"	"A"	"M"	"E"	13
A0	09	"P"	"M"	"S"	"P"	"3"	"5"	"_"	"5"	"N"	30	15	13	08	"C"	"P"
B0	"U"	"2"	"N"	"A"	"M"	"E"	13	09	"P"	"M"	"S"	"P"	"2"	"5"	"_"	"2"
C0	"C"	"N"	30	15	13	08	"P"	"U"	"3"	"N"	"A"	"M"	"E"	13	09	"P"
D0	"M"	"S"	"P"	"2"	"5"	"_"	"2"	"N"	30	15	13	08	"C"	"P"	"U"	"4"
E0	"N"	"A"	"M"	"E"	13	09	"P"	"M"	"B"	"P"	"2"	"0"	"_"	"0"	"N"	30
F0	16	13	09	"I"	"O"	"1"	"0"	"5"	"N"	"A"	"M"	"E"	13	09	"P"	"M"
100	"W"	"D"	"6"	"4"	"_"	"4"	"N"	30	16	13	09	"I"	"O"	"1"	"0"	"6"
110	"_"	"0"	"N"	30	16	13	09	"I"	"O"	"1"	"0"	"7"	"N"	"A"	"M"	"E"
120	13	09	"N"	"A"	"M"	"E"	13	09	"P"	"M"	"L"	"D"	"0"	"1"	"P"	"M"
130	"L"	"E"	"0"	"1"	"_"	"5"	"N"									

A.3.5 Example system parameters (for CNC)

Table A-16 shows examples of CNC system parameters.

Table A-16 System parameters in tabular document notation (for CNC)

Parameter name	Name string (PrintableString type) Length, string	Data type (type)	Contents of parameter Length, contents
SysPara			
Device profile common specification version	6, "COMVER"	INTEGER	1, 1
System parameter identification string	2, "ID"	PrintableString	7, "SYSPARA"
System parameter revision number	3, "REV"	INTEGER	1, 0
System parameter revision date	7, "REVDATE"	[INTEGER], 2, (0001 to 9999), [INTEGER], 1, (01 to 12), [INTEGER], 1, (01 to 31)	2, 1998 1, 9 1, 30
Device category	10, "DVCATEGORY"	PrintableString	3, "CNC"
Vendor name	6, "VENDOR"	PrintableString	9, "MSTCJ LTD"
Product model	7, "DVMODEL"	PrintableString	16, "MSTCJ Series 16a"

Parameter name	Name string	Data type	Contents of parameter
CncPara			
Device-specific parameter identification string	2, "ID"	PrintableString	7, "DEVPARA"
Model name	5, "MODEL"	PrintableString	8, "MS16a-MA"
Series	6, "SERIES"	PrintableString	4, "MSF1"
Revision	3, "REV"	INTEGER	1, 0
System	7, "System"	SEQUENCE	*
System information	7, "SysInfo"	SEQUENCE	*

*: This parameter is a Constructed type, and has the following data:

System			
Option configuration flag	5, "SFLAG"	BIT STRING	8, "00100101"
Number of control axes	4, "AXES"	INTEGER	2, 4

SysInfo			
Input virtual address	2, "IN"	OCTET STRING	6, "000000"
Output virtual address	3, "OUT"	OCTET STRING	6, "040000"

(1) Abstract syntax

<Type definition>

```

CncRecord ::= SEQUENCE
{
  SysPara          SysParaType,
  CncPara          CncParaType,
}

SysParaType ::= SEQUENCE
{
  nameCOMVER      NameType,
  paraCOMVER      INTEGER,
  nameID          NameType,
  paraID          NameType,
  nameREV         NameType,
  paraREV         INTEGER,
  nameREVDATE     NameType,
  paraREVDATE     DateType,
  nameDVCATEGORY  NameType,
  paraDVCATEGORY  NameType,
  nameVENDOR      NameType,
  paraVENDOR      NameType,
  nameDVMODEL     NameType,
  paraDVMODEL     NameType
}

CncParaType ::= SEQUENCE
{
  nameID          NameType,
  paraID          NameType,
  nameMODEL       NameType,
  paraMODEL       NameType,
  nameSERIES      NameType,
  paraSERIES      NameType,
  nameREV         NameType,
  paraREV         INTEGER,
  nameSystem      NameType,
  paraSystem      SystemType,
  nameSysInfo     NameType,
  paraSysInfo     SysInfoType
}

SystemType ::= SEQUENCE
{
  nameINPUT       NameType,
  paraINPUT       BIT STRING,
  nameAXES        NameType,
  paraAXES        INTEGER
}

SysInfoType ::= SEQUENCE
{
  nameIN          NameType,
  paraIN          OCTET STRING,
  nameOUT         NameType,
  paraOUT         OCTET STRING
}

NameType := PrintableString
DateType ::= SEQUENCE
{
  year            INTEGER,
  month           INTEGER,
  day             INTEGER
}

```

Appendix

<Value definition>

```

{
  SysPara  {
    nameCOMVER      "COMVER",
    paraCOMVER      1,
    nameID          "ID",
    paraID          "SYSPARA",
    nameREV         "REV",
    paraREV         0,
    nameREVDATE     "REVDATE",
    paraREVDATE     {   year  1998,
                       month  9,
                       day   30           },
    nameDVCATEGORY  "DVCATEGORY",
    paraDVCATEGORY  "CNC",
    nameVENDOR      "VENDOR",
    paraVENDOR      "MSTCJ LD",
    nameDVMODEL     "DVMODEL",
    paraDVMODEL     "MSTCJ Series 16a"
  }

  CncPara  {
    nameID          "ID",
    paraID          "DEVPARA",
    nameMODEL       "MODEL",
    paraMODEL       "MS16a-MA",
    nameSERIES      "SERIES",
    paraSERIES      "MSF1",
    nameREV         "REV",
    paraREV         0,
    nameSystem      "System",
    paraSystem      {   nameINPUT  "SFLAG",
                       paraINPUT  `00100101'B,
                       nameAXES   "AXES",
                       paraAXES   4           },
    nameSysInfo     "SysInfo",
    paraSysInfo     {   nameIN     "IN",
                       paraIN     `000000000000'H
                       nameOUT    "OUT",
                       paraOUT    `000400000000'H   }
  }
}

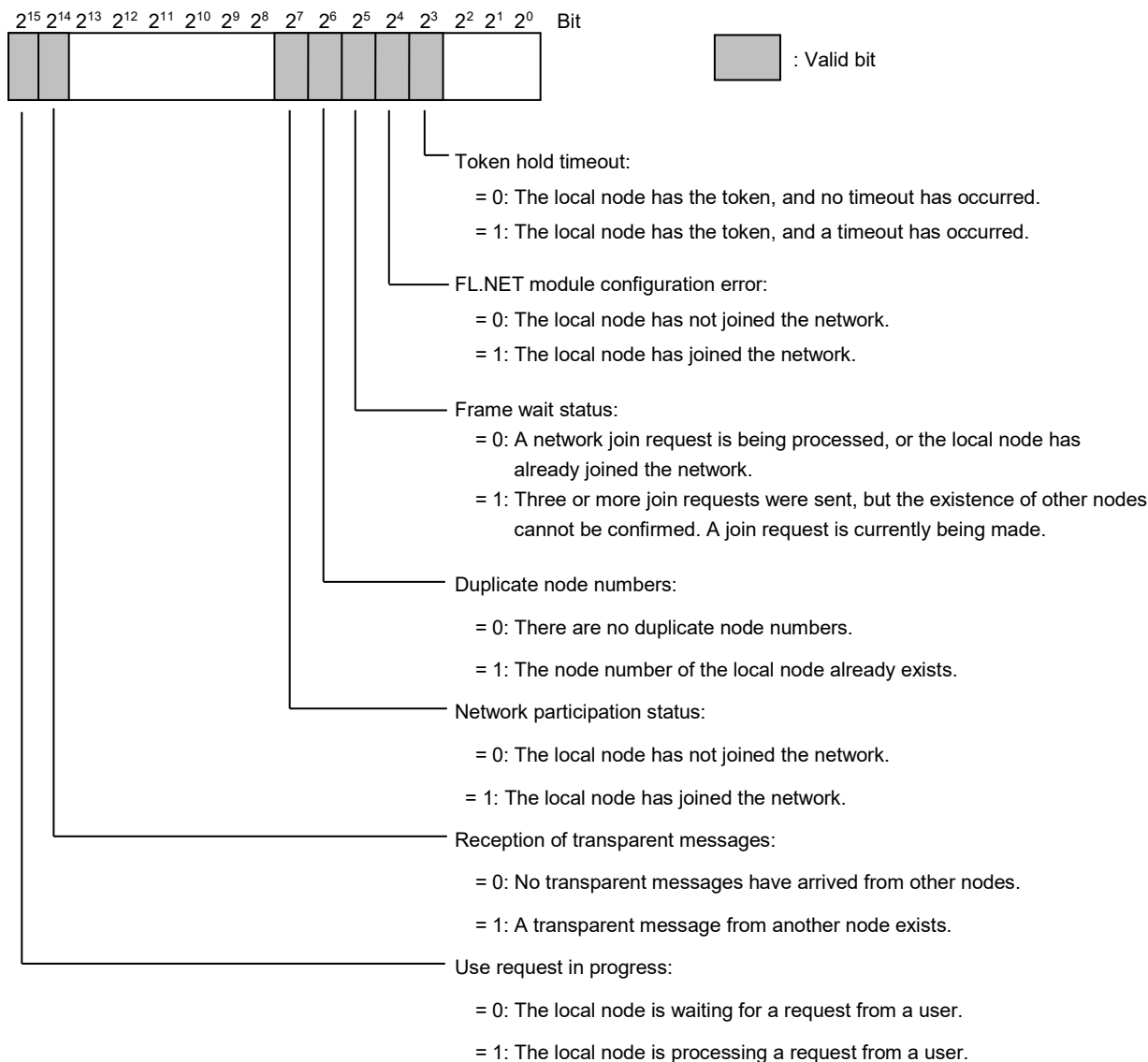
```

A.4 Contents of each status flag

This section describes the contents of the self-node status flag, the FA link status flag, and the higher layer status flag.

(1) Contents of the self-node status flag

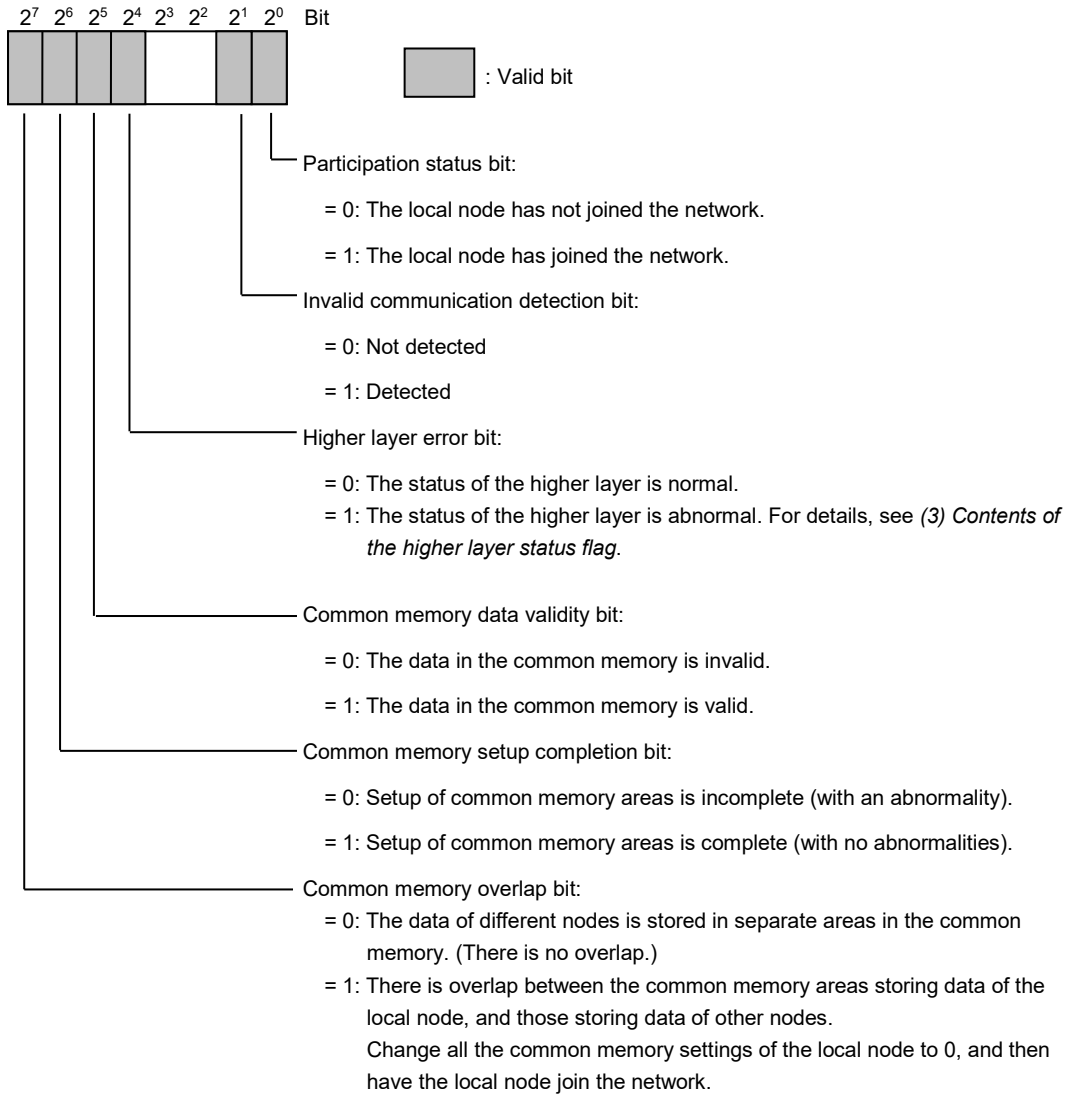
The following shows the bit allocation of the self-node status flag for the FL.NET module:



Note: In the **Self-node information** tab (Figure 5-77), which is opened from the FL.NET SYSTEM/S10VE setup tool, you can only reference the lower byte (bits 2⁷ to 2⁰).

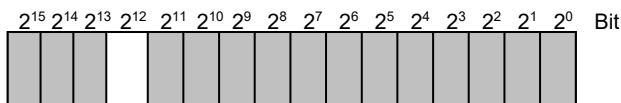
(2) Contents of the FA link status flag

The following shows the bit allocation in the FA link status flag:



(3) Contents of the higher layer status flag

The following shows the bit allocation of the higher layer status flag:



Higher layer error code (consisting of 12 bits):
 This shows the error code of the error that occurred in the higher layer.
 The following table lists the error codes for this FL.NET module:

Error code	Description
0x000	The higher layer (S10VE) is in a normal state.
0x0FF	The CPU is down in the higher layer (S10VE).

Higher layer error status bit (consisting of 2 bits):
 This shows the status of the higher layer when an error has occurred.

Bit 2 ¹⁴	Bit 2 ¹³	Description
OFF	OFF	The higher layer is in a normal state.
OFF	ON	An abnormality occurred in the higher layer, but operations of the higher layer can still continue (<i>ALARM</i> status). Transmission of cyclic data and message data is assured.
ON	OFF	An abnormality occurred in the higher layer, and operations of the higher layer cannot continue (<i>WARNING</i> status).
ON	ON	Cyclic data and message data are not assured to be correct.

For this FL.NET module, bits 2¹⁴ and 2¹³ are set to ON if an abnormality occurs in the higher layer.

Higher layer operation status bit:
 = 0: The higher layer (S10VE) is in the *STOP* status (inactive).
 = 1: The higher layer (S10VE) is in the *RUN* status (active).

This page is intentionally left blank.