HITACHI

USER'S MANUAL

OPTION

FL.NET

（LQE702）

S10V

Programmable Controller

SVE-1-130（C）

USER'S MANUAL

OPTION

FL.NET

（LQE702）

S10V

Programmable Controller

First Edition, February 2010, SVE-1-130(A) (out of print)
Second Edition, July 2011, SVE-1-130(B) (out of print)
Third Edition, January 2013, SVE-1-130(C)

BI<IC> (FL-MW2007, AI10)

SAFETY PRECAUTIONS

- Before installation, operation, maintenance, and/or inspection of this product, be sure to read through carefully this manual and other related manuals. Do not use this product until 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 so that users of this product may easily reach it.
- This manual contains information on potential hazards that is intended as a guide for safe use of this product. The potential hazards listed in the manual are divided into four hazard levels of danger, warning, caution, and notice, according to the level of their severity. The following are definitions of the safety labels containing the corresponding signal words DANGER, WARNING, CAUTION, and NOTICE.

⚠ DANGER : This safety label identifies precautions that, if not heeded, will result in death or serious injury.

⚠ WARNING : Identifies precautions that, if not heeded, could result in death or serious injury.

⚠ CAUTION : Identifies precautions that, if not heeded, could result in minor or moderate injury.

NOTICE : This safety label without a safety alert symbol identifies precautions that, if not heeded, could result in property damage or loss not related to personal injury.

Failure to observe any of the ⚠CAUTION and *NOTICE* statements used in this manual could also lead to a serious consequence, depending on the situation in which this product is used. Therefore, be sure to observe all of those statements without fail.

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

Serious injury: Is an injury that requires hospitalization for medical treatment, has aftereffects, and/or requires long-term follow-up care. Examples of serious injuries are as follows: vision loss, burn (caused by dry heat or extreme cold), electric-shock injury, broken bone, poisoning, etc.

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 are as follows: burn, electric-shock injury, etc.

Property damage or loss not related to personal injury: Is a damage to or loss of personal property. Examples of property damages or losses not related to personal injury are as follows: damage to this product or other equipment or their breakdown, loss of useful data, etc.

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 planned 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 unique needs. A variety of industry standards are available to establish such safety rules and regulations.

1. ⚠ General Safety Guidelines

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

- Follow all the operating procedures provided in this manual.
- Pay special attention to and follow all the hazard warnings on the machine and in the manual. Failure to do so can cause injury to yourself or damage to the machine.
- Do not perform any operation or action in any way other than as provided in this manual. When in doubt, call the designated field engineer. Keep in mind that the hazard warnings in this manual or on the machine cannot cover every possible case, as it is impossible to predict and evaluate all circumstances beforehand.
  Be alert and use your common sense.
- Do not install, wire, handle, modify, or use maintenance parts in any manner not described in this manual. Such a practice may result in breakdown of this equipment or peripherals, injury or even death. Hitachi will not be responsible for any accident or failure resulting from such mishandling.

Read the following safety guidelines carefully and follow them when you conduct maintenance of the machine.

Before starting maintenance

- Maintenance of the machine must be done only by trained and qualified field engineers.
- Read and follow the safety guidelines and procedures in this manual and the related manuals.
- In this manual and on the machine, hazard warnings are provided to aid you in preventing or reducing the risk of death, personal injury, or product damage. Understand and follow these hazard warnings fully.
- Keep in mind that the hazard warnings in this manual or on the machine cannot cover every possible case, as it is impossible to predict and evaluate all circumstances beforehand.
  Be alert and use your common sense.

During work

- For each procedure, follow the given sequence of steps.
- Use the special tools and instruments, specified for the work in the manual or commercially available tools and instruments which fit the purpose.
- Use measurement instruments and powered tools which are properly calibrated or periodically inspected.
- Keep the maintenance area neat and tidy.
- Always put away parts, materials or tools when not in use.
- Wear an eye protector where anything may fly about.
- When using sharp objects or cutting tools, make sure that no part of your body lies in the path of the blade bit, or point.
- Before finishing your work, make sure that all parts removed during maintenance have been installed back in their original positions in the machine.
  Make sure that no tool or foreign material is left in the machine.

Prevention of electric shocks

- Before starting work, make sure that, unless otherwise specifically instructed, there is no potential electric hazard in the maintenance area such as insufficient grounding or a wet floor.
- Before starting work, note where the emergency power-off switches are located and make sure you know how to operate them.
- Unless otherwise specifically instructed, cut off all power sources to the machine before starting maintenance. Just switching off the machine power supplies is usually not enough.

  When power is fed from a wall or floor outlet, unplug the power supply cord, or turn off the switch on the power distribution panel or board. Attach a notice on the panel or board prohibiting the use of the switch.

  If the energy isolating device such as the switch on the power distribution panel or board accepts a lockout device, turn off the power, lock out the energy isolating device, and bring the key with you. When you take over the work and the key for the lockout device if applicable, do not assume that the power is off. Make sure yourself that the above-mentioned conditions such as switches are satisfied. If necessary, use a measurement tool to ensure that the power is off.
- Do not touch any uninsulated conductor or surface, where so instructed, which remains charged for a limited time after the external power supply to the machine is disconnected.
- 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 close to a hazardously energized part, do not work alone; work with another person who can immediately turn off the power in an emergency.
- Do not wear any metallic item such as a wrist watch with a metallic surface, or metallic accessories.

  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.
- Unless otherwise specifically instructed, use only one hand when it is necessary to work near an exposed live electric circuit.

  This prevents the completion of the circuit through your heart even 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 it 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.

Procedures in an emergency

For 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.
  Use the emergency power-off switch, if there is one, or, otherwise, a 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.
- Then, 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 should be performed by a trained and qualified person.

For outbreak of fire

- First, shut off all the power from the machine using the emergency power-off switch, if there is one, or the normal power-off switch.
- If the fire continues burning after the power is shut off, take suitable actions including the use of a fire extinguisher or a call for the fire department.

2. ⚠ Hazard Warning Statements

The following are the hazard warning statements contained in this manual.

2.1 ⚠ CAUTION Statement

(chapter 6, page 6-2)

⚠ **CAUTION**
Before replacing the module, switch it off to avoid electrical shock hazards and also to prevent it from being damaged or malfunctioning.

2.2 NOTICE Statements

(chapter 1, page 1-5)

NOTICE
Users of this product must have an adequate knowledge of the Windows® environment and user interface. This system conforms to the Windows® standard, and this manual is prepared for those users who are familiar with the basic Windows® operating procedures.

(chapter 2, page 2-8)

NOTICE
Switch off the power supply before operating the MAIN/SUB setting switch. If you operate while the power supply is applied, it may result in a malfunction.

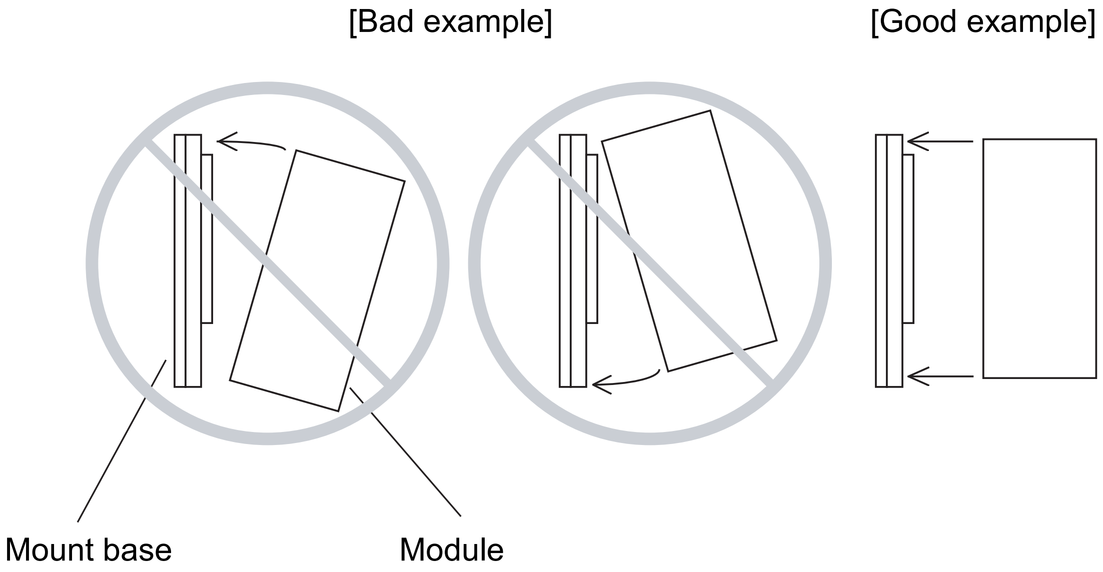
(chapter 3, page 3-2)

NOTICE
● There are no specific rules about the mounting position or unoccupied slots. ● This model-LQE702 FL.NET module may not be mounted on the same mount base in conjunction with any other FL.NET module of models LQE500 or LQE502 and any EQ.LINK module of model LQE701.

(chapter 3, page 3-3)

NOTICE

- Dust or other foreign matter might accumulate on the connector, resulting in poor contact. Immediately after the module is unpacked, perform the mounting and wiring procedures.
- To prevent the module from being damaged, observe the following precautions when you mount or demount the module:
  - Before mounting the module to the mount base connector, check that the connector pins are properly aligned and not bent, broken, or soiled with dirt or the like.
  - Ensure that the module is parallel to the mount base vertical surface as shown below when mounting. If you connect a module to or disconnect it from its connector while it is tilted, the connector pins may become damaged.
  - If the mount base is positioned overhead due to the employed enclosure structure, use a stepladder or the like and mount the module squarely. If you mount the module obliquely, the connector may become damaged.



(chapter 3, page 3-4)

NOTICE

Switch off the power supply before operating the MAIN/SUB setting switch. If you operate while the power supply is applied, it may result in a malfunction.

(chapter 4, page 4-2)

NOTICE

- This hardware unit may malfunction if it is connected poorly or has a broken line. After connecting the connector, check whether the locking.
- Do not touch the connector during power on. Otherwise, the system may malfunction due to static electricity, etc.

(chapter 4, page 4-5)

NOTICE

The only port setting that is supported by the FL.NET module (model LQE702) is auto-negotiation. Do not use 100-Mbps full-duplex setting for the port of the switching hub. Disregarding this rule may result in a failure of data communication when the load on the communication line builds up.

(chapter 4, page 4-6)

NOTICE

- Ground the FG (frame ground) terminal as follows: Connect the FG terminal on each module provided with external terminals to the grounding terminal on the mount base. Perform Class D grounding for the grounding terminal on the mount base.
- Use ground lines whose size is 2 mm^2 or more.

(chapter 5, page 5-39)

NOTICE

- If the LPU memory 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 from the number of words specified by the LPU memory area word count setting.
- If any node is left without being set, the FL.NET module cannot transfer to the S10V the data received from that node into the common memory.

(chapter 5, page 5-41)

NOTICE

- When the self-node status area is set up, the self-node status flag is transferred as 1-word data.
- When the FA link status area is set up, the FA link status flag for the self-node is transferred to the low-order byte of the specified area.
- If the area for self-node status or FA link status information is not specified, the FL.NET module cannot transfer the status information to the S10V.

(chapter 5, page 5-43)

NOTICE

- The register/extension memory address range applicable to the higher layer status area and FA link status area is the same as for the LPU memory allocation area.
- When the higher layer status area is set up, the contents of the higher layer status flag are transferred as 1-word data.
- When the FA link status area is set up, the contents of the FA link status flag for the node are transferred to the low-order byte of the specified area.
- The higher layer status/FA link status area setting must not be a duplicate of the LPU memory allocation area setting for a remote node (reception setting) or the self-node or of the setting for any other area.
- If any node is omitted from the node settings given, the FL.NET module cannot transfer the higher layer status or FA link status of that node to the S10V.

(chapter 5, page 5-46)

NOTICE

Do not mix to use C mode handler and mathematical/logical function.
Please be sure to unify either C mode handler or mathematical/logical function at FL.NET module unit.

(chapter 5, page 5-49)

NOTICE

- When issuing a request to the FL.NET module, do not simultaneously use the C mode handler and mathematical/logical function.
- You cannot specify an index in a parameter (PI/O address.)

(chapter 5, page 5-55)

NOTICE

The server feature of the FL.NET module returns an abnormal response if, on reception of a word block read request, the virtual size is found exceeding 512 words. The error code used in such an event is 0xFFFFFFFF.

(chapter 5, page 5-57)

NOTICE

The server feature of the FL.NET module returns an abnormal response if, on reception of a word block write request, the virtual size is found exceeding 512 words or conflicting with the data size. The error code used in such an event is 0xFFFFFFFF.

(chapter 5, page 5-64)

NOTICE

- When rewriting the network parameters, take extreme care not to cause any setting duplication to the common memory.
- The server feature of the FL.NET module returns an abnormal response if, upon reception of a network parameter write request, the parameter selection flag is found to be not between 1 and 3. The error code used in such an event is 0xFFFFFFFF.

(chapter 5, page 5-65)

NOTICE

The FL.NET module does not support the server feature for a stop directive. If a stop request is issued to the FL.NET module, it returns a response to indicate that it does not support the stop directive.

(chapter 5, page 5-67)

NOTICE

The FL.NET module does not support the server feature for a run directive. If a run request is issued to the FL.NET module, it returns a response to indicate that it does not support the run directive.

(chapter 5, page 5-78)

NOTICE

Exercise due care when using the above transparent message request. If the task number is erroneously specified, the CPU or CMU of the remote node (S10V) may become inoperative.

(chapter 5, page 5-90)

NOTICE

- When an offset size is specified for a specific node number, it is stored in memory until the FL.NET module is turned OFF or reset. Note, however, that the user cannot reference the offset setting within the FL.NET module after completion of offset setup. The offset setting must be managed by the user.
- The above feature is designed for use in cases where the available memory space of the S10V is insufficient. It is recommended that you do not use the feature when the memory space of the S10V is adequate.
- When the address limit of common memory area 1 or 2 is exceeded by the application of an offset, the resulting excess portion of address data will not transfer to the S10V memory. Exercise care not to exceed the limits of the common memory areas.

(chapter 5, page 5-115)

NOTICE

- The S10V BASE SYSTEM is required for operating the S10V FL.NET. If it is not installed, you cannot install the S10V FL.NET.
- Before installing the S10V FL.NET, be sure to exit all the currently open Windows®-based programs. Do not forget to exit anti-virus software and other memory-resident programs. If you install the FL.NET without exiting such programs, an error may occur during installation. If such an error occurs, first uninstall the S10V FL.NET as directed in “5.4.2 Uninstalling,” exit all the Windows®-based programs, and then install the S10V FL.NET again.

(chapter 5, page 5-117)

NOTICE

If Windows® opens a window during the uninstall process to display the question “Remove Shared File?”, click the No button to retain shared files.

(chapter 6, page 6-2)

NOTICE

Static electricity could cause damage to the module. Before handling the module allow static charges on the human body to discharge.

WARRANTY AND SERVICING

Unless a special warranty contract has been arranged, the following warranty is applicable to this product.

1. Warranty period and scope

   Warranty period

   The warranty period for this product is for one year after the product has been delivered to the specified delivery site.

   Scope

   If a malfunction should occur during the above warranty period while using this product under normal product specification conditions as described in this manual, please deliver the malfunctioning part of the product to the dealer or Hitachi Engineering & Services Co., Ltd. The malfunctioning part will be replaced or repaired free of charge. If the malfunctioning is shipped, however, the shipment charge and packaging expenses must be paid for by the customer.

   This warranty is not applicable if any of the following are true.

   - 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 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 the operation of this product or from malfunctions of this product. This warranty is valid only in Japan and is not transferable.

2. 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 schools.
   - Examinations and repairs after the warranty period is concluded.
   - Even if the warranty is valid, examination of malfunctions that are caused by reasons outside the above warranty scope.

This Page Intentionally Left Blank

This manual provides information for the following hardware and program product:

<Hardware product>
FL.NET (LQE702)

<Program product>
S-7895-30, S10V FL.NET SYSTEM, 01-03

Revision record

Revision No.	Revision record (revision details and reason for revision)	Month, Year	Remarks
A	First edition	February 2010	
B	Product safety labeling is revised to include additional safety level information	July 2011	
C	Windows® 7 (32-bit) operating system is newly supported.	January 2013	

In addition to the above changes, all the unclear descriptions and typographical errors found are also corrected without prior notice.

PREFACE

Thank you for purchasing the FL.NET module, which is an option for use with the S10V.
This manual, named "USER'S MANUAL OPTION FL.NET," describes how to use the FL.NET module. For proper use of the CPU link module, it is requested that you thoroughly read this manual.

The S10V product is available in two types: standard model and environmentally resistant model. The environmentally resistant model has thicker platings and coatings than those for the standard model.
The model number of the environmentally resistant model is marked by adding the suffix "-Z" to the model number of the standard model.

(Example) Standard model: LQE702
Environmentally resistant model: LQE702-Z

This manual is applicable to both the standard model and environmentally resistant models. Although the descriptions contained in this manual are based on the standard model, follow the instructions set forth in this manual for proper use of the product even if you use the environmentally resistant model.

<Precautions on the FL-net protocol version>
For the FL-net protocol, Ver. 1.00 and Ver. 2.00 are available. These versions are not compatible and a Ver. 100 device and a Ver. 2.00 device cannot be connected to each other. Note the protocol version that the FL.NET module can support varies depending on the model. The FL-net protocol version for each model is shown below.

Model	LQE500	LQE502/LQE702
FL-net protocol version	Ver. 1.00	Ver. 2.00

Do not connect an FL-net protocol Ver. 1.00 device to a Ver. 2.00 device. The FL-net protocol Ver. 2.00 device has a function that prevents participation in the data link when an FL-net protocol Ver. 100 device is detected on the network. Accordingly, the Ver. 1.00 device and Ver. 2.00 device cannot configure a data link.

To determine whether device versions are compatible, see below.

Self-device Other device	Ver. 1.00 device	Ver. 2.00 device
Ver. 1.00 device	√	nc
Ver. 2.00 device	nc	√

√: Connectable, nc: Not connectable

Ver. 1.00: FL-net protocol version of the device manufactured on the standard of the FA Control Network [FL-net (OPCN-2)] - Protocol Specification JEM 1479 2000-edition issued by Japan Electrical Manufacturers' Association (JEMA)

Ver. 2.00: FL-net protocol version of the device manufactured on the standard of the FA Control Network [FL-net (OPCN-2)] - Protocol Specification JEM 1479 2002-revision issued by Japan Electrical Manufacturers' Association (JEMA)

The existing parameter setting software (FL.NET system) shown in the following table can be used for LQE702 regardless of the FL-net protocol version.

Name	Model	Remarks
FL.NET system for S10V	S-7895-30J	For LQE500, LQE502, and LQE702

<Trademarks>

- Microsoft® Windows® operating system, Microsoft® Windows® 2000 operating system, Microsoft® Windows® XP operating system, Microsoft® Windows® 7 (32-bit) operating system are registered trademarks of Microsoft Corporation in the United States and/or other countries.
- Ethernet® is registered trademark of Xerox, Corp.

<Note for storage capacity calculations>

- Memory capacities and requirements, file sizes and storage requirements, etc. 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 equals signs).
  1 KB (kilobyte) = 1,024 bytes
  1 MB (megabyte) = 1,048,576 bytes
  1 GB (gigabyte) = 1,073,741,824 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

CONTENTS

FIGURES

TABLES

This Page Intentionally Left Blank

1 OVERVIEW

1.1 What is FL-net

FL-net is an open FA network that the FA Open Systems Promotion Group (FAOP) standardized in the Manufacturing Science and Technology Center (MSTC), which is an organization affiliated to the Ministry of International Trade and Industry.　The Japan Electrical Manufacturers' Association (JEMA) prepared the specification.

As shown in Figure 1-1, this network can interconnect programmable controllers (PLCs), computer numerical control (CNC) devices, and various other factory automation controllers and personal computers of many different brands to exercise control and monitoring functions.

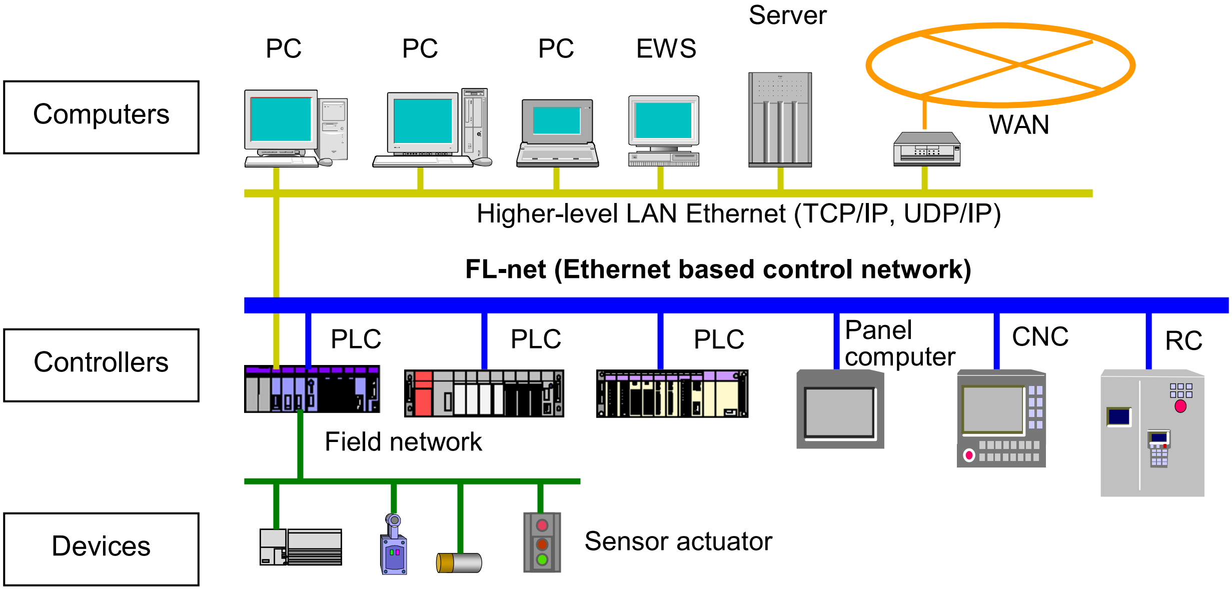


Figure 1-1　FA Control Network Configuration Example

1.2 FL-net Features

FL-net has the following features:

- An open system can be implemented.
- A multivendor environment can be established.
- Programmable controllers (PLCs), computer numerical control (CNC) devices, and various other factory automation controllers and personal computers of many different brands can be interconnected to exercise control and monitoring functions.

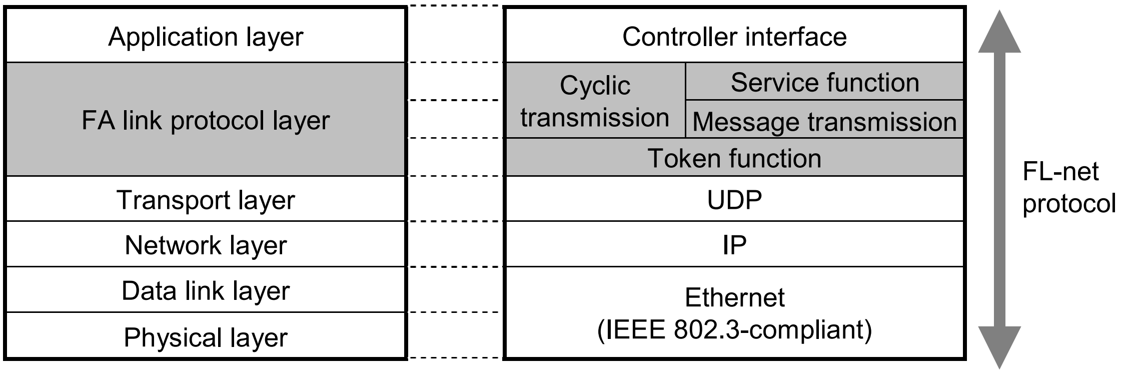


Figure 1-2 FL-net Protocol Basis Structure

<A widely accepted standard is complied with.>

FL-net is based on Ethernet, which is a standard for OA (office automation) equipment. Further, standard UDP/IP is used to achieve a high degree of communication efficiency.

- Low price

  Since common communication devices can be used as components, the price is decreased.
- Use of widespread network equipment

  Transceivers, hubs, cables, personal computer LAN cards, and various other widely accepted Ethernet devices can be used.
- Future speed increase

  In the future, you can expect that the speed of transfer will increase from 100 Mbps to 1 Gbps.
- Introduction of fiber-optic communication

  Optical repeaters and other widespread devices for Ethernet can be used to replace relevant portions with fiber-optic lines. The introduction of such fiber-optic lines will permit transmission over a distance of 500 m or longer, provide increased noise immunity, and protect against a lightning surge in outdoor wiring.

<The functions required for FA controllers are supported.>

Since the initial objective of FL-net is to meet user requests, it offers various features required for FA (factory automation).

- Large-scale network
  Up to 254 units of devices (nodes) can be connected.
- Selective use of two different communication features for various purposes
  Both the common memory feature and message communication feature are supported. The former permits all nodes to share the same data through cyclic communications. The latter allows necessary information to be exchanged only when it is needed.
- Large-capacity common memory
  The common memory has a capacity as large as 8k bits + 8k words.
- Fast response
  When FL-net-compliant modules are interconnected, the performance varies with the connected devices. However, when the S10V FL.NET modules (Ver. 2.00: Model LQE702) are interconnected, the response speed is as high as 42 ms/32 nodes (in 2k bit + 2k word mode).
- High reliability based on the use of a masterless system
  Since no master exists, individual nodes can participate and depart without affecting the communications maintained by the other nodes. As a result, any nodes can be freely switched on/off and serviced.

1.3 System Software Specifications

1.3.1 System overview

To use the FL.NET module, you must register various items of module information in the module. This can be accomplished by using the system software (tools) listed below. The operating procedures for these tools are similar to those for general Windows® applications.

Table 1-1 Types of System Software (Tools)

Package name	Model (For S10V)	Version	Supply style
FL.NET system	S-7895-30	01-03 or later	Option

1.3.2 Required hardware and software

The following hardware and software items are required for the use of the FL.NET module system software:

- Personal computer (main unit) containing a Pentium 300 MHz or faster CPU, or a 1 GHz or faster CPU (when Windows® 7 (32-bit version) is used)
- Display having a resolution of 800 × 600 dots (SVGA) or higher
- Microsoft® Windows® 2000 operating system, Microsoft® Windows® XP operating system or Microsoft® Windows® 7 (32-bit) operating system
- At least 64 MB of RAM (when Windows® 2000 is used)
- At least 128 MB of RAM (when Windows® XP is used)
- At least 1 GB of RAM (when Windows® 7 (32-bit) is used)
- At least 10 MB of free hard disk space
- Cable for connecting the personal computer to the LPU unit (RS-232C cross cable with D-sub 9-pin connectors) or cable for connecting the personal computer to the CMU or ET.NET module (10BASE-T or 100BASE-T twisted pair cross cable with RJ-45 modular connectors)

NOTICE

Users of this product must have an adequate knowledge of the Windows® environment and user interface. This system conforms to the Windows® standard, and this manual is prepared for those users who are familiar with the basic Windows® operating procedures.

This Page Intentionally Left Blank

2 FL.NET MODULE

2.1 System Configuration

The S10V FL.NET module (LQE702) is Hitachi's module that is compliant with the FL-net protocol.

This is hereafter described as the FL.NET module.

Figure 2-1 shows a typical system configuration for the use of FL-net communications.

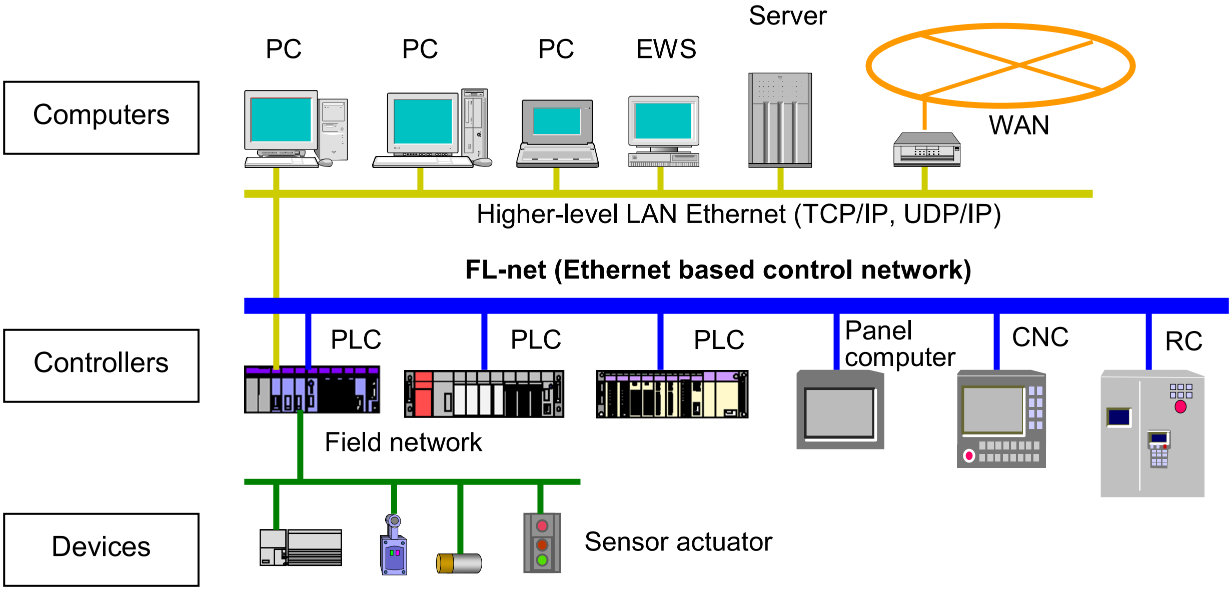


Figure 2-1 FA Control Network Configuration Example

Table 2-1 General Specifications

Item	Specifications
Operating ambient temperature	0 to 55°C
Storage ambient temperature	-20 to 75°C
Operating atmosphere	Dust: 0.1 mg/m^3 maximum; free from corrosive gas
Operating ambient humidity	30 to 90% RH (non-condensing)
Storage ambient humidity	10 to 90% RH (non-condensing)
Vibration resistance	JIS C 0040-compliant
Shock resistance	JIS C 0041-compliant

2.1.1　Functionality and performance specifications

Table 2-2　Functionality and Performance Specifications

Item	Specifications
Model	LQE702
Maximum number of mountable modules	2 modules per LPU (Installing from the leftmost slot is not required.)
Transfer speed	100 Mbps
Electrical interface	IEEE 802.3-compliant (CSMA/CD-compliant)
Transmission protocol	UDP/IP, FA link protocol Ver. 2.00
Number of connectable units	Up to 254 units per network
Connector	RJ45 connector (10BASE-T/100BASE-TX)
Module outside dimensions	34 (W) × 130 (H) × 100.2 (D) mm
Weight	180 g
Transfer word count	Cyclic: 8.5k words maximum (node) Message: 1,024 bytes maximum (node)
Cable length	Twisted-pair cable: 100 m maximum/segment (category 5 or higher)

2.1.2　Support tool specification

For the use of the FL.NET module, you must set the node numbers, common memory, and various other items with the setup tool named [S10V FL.NET SYSTEM].
For the detailed specifications for the setup tool, see “5.4　Installing and Starting up the System” and “5.5　Operating Method.”

2.1.3 Link data specifications

(1) Common memory area
The FL.NET module allows you to set up a common memory area for each node.
For the common memory area setup procedure, see "5.3.1 Link parameter setup procedure." For the procedure for reserving a common memory area in the LPU's memory, see "5.3.2 LPU memory allocation procedure."

Table 2-3 Registers Available for Common Memory Area Allocation in LPU Memory

No.	Available register/address	Remarks
1	XW000 to XWFF0	External input
2	YW000 to YWFF0	External output
3	JW000 to JWFF0	Transfer register
4	QW000 to QWFF0	Receive register
5	GW000 to GWFF0	Global link register
6	RW000 to RWFF0	Internal register
7	EW400 to EWFF0	Event register
8	MW000 to MWFF0	Internal register
9	DW000 to DWFFF	Function data register
10	FW000 to FWBFF	Function work register
11	LBW0000 to LBWFFF0	Work register
12	LWW0000 to LWWFFFF	Word work register
13	LXW0000 to LXW3FFF	Word work register

(2) Virtual address space and physical memory

Table 2-4 Virtual Address Space and Physical Memory

<table>
<tr><th>Item</th><th colspan="2">Description</th></tr>
<tr><td>Area name</td><td colspan="2">Memory address (0x000000 to 0xFFFFFF)</td></tr>
<tr><td>Access unit</td><td colspan="2">Word</td></tr>
<tr><td>Area size</td><td colspan="2">16,777,216 bytes</td></tr>
<tr><td>Access attribute</td><td colspan="2">Read/Write
Conditions: A write operation cannot be performed on some ROM areas and OS areas.</td></tr>
<tr><td rowspan="5">Correspondence to virtual address (word block) (*)</td><td>Vendor-unique notation</td><td>Virtual address</td></tr>
<tr><td>0x000000, 0x000001</td><td>0x00000000</td></tr>
<tr><td>0x000002, 0x000003</td><td>0x00000001</td></tr>
<tr><td>≀ ≀</td><td>≀</td></tr>
<tr><td>0xFFFFFE, 0xFFFFFF</td><td>0x007FFFFF</td></tr>
<tr><td>Data arrangement</td><td colspan="2">A 2-byte area corresponds to one word of a word block.
MSB LSB
| Address n | Address n + 1 |</td></tr>
<tr><td>Other access method</td><td colspan="2">Nothing in particular</td></tr>
</table>

(*) To support the correspondence to virtual addresses, a byte block is not supported.

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

Table 2-5 Virtual Address Spaces and Symbol Names

Register name	Symbol name	Memory address	Virtual address space
External input	X000 to XFFF	0x240000 to 0x241FFE	0x120000 to 0x120FFF
External output	Y000 to YFFF	0x242000 to 0x243FFE	0x121000 to 0x121FFF
Internal register	R000 to RFFF	0x0AC000 to 0x0ADFFE	0x00056000 to 0x00056FFF
Global register	G000 to GFFF	0x0A8000 to 0x0A9FFE	0x00054000 to 0x00054FFF
On-delay timer	T000 to T1FF	0x0B3000 to 0x0B33FE	0x00059800 to 0x000599FF
One-shot timer	U000 to U0FF	0x0B5000 to 0x0B51FE	0x0005A800 to 0x0005A8FF
Counter	C000 to C0FF	0x0B7000 to 0x0B71FE	0x0005B800 to 0x0005B8FF
Keep	K000 to KFFF	0x0B0000 to 0x0B1FFE	0x00058000 to 0x00058FFF
System register	S000 to SBFF	0x0BE800 to 0x0BFFFE	0x0005F400 to 0x0005FFFF
Z-register	Z000 to Z3FF	0x0BE000 to 0x0BE7FE	0x0005F000 to 0x0005F3FF
Internal register	M000 to MFFF	0x0AE000 to 0x0AFFFE	0x00057000 to 0x00057FFF
Transfer register	J000 to JFFF	0x0A2000 to 0x0A3FFE	0x00051000 to 0x00051FFF
Receive register	Q000 to QFFF	0x0A6000 to 0x0A7FFE	0x00053000 to 0x00053FFF
External input (word)	XW000 to XWFF0	0x414000 to 0x4141FE	0x20A000 to 0x20A0FE
External output (word)	YW000 to YWFF0	0x414200 to 0x4143FE	0x20A100 to 0x20A1FE
Internal register (word)	RW000 to RWFF0	0x0E0C00 to 0x0E0DFE	0x00070600 to 0x000706FF
Global register (word)	GW000 to GWFF0	0x0E0800 to 0x0E09FE	0x00070400 to 0x000704FF
On-delay timer (word)	TW000 to TW1F0	0x0E1300 to 0x0E133E	0x00070980 to 0x0007099F
One-shot timer (word)	UW000 to UW0F0	0x0E1500 to 0x0E151E	0x00070A80 to 0x00070A8F
Counter (word)	CW000 to CW0F0	0x0E1700 to 0x0E171E	0x00070B80 to 0x00070B8F
Keep (word)	KW000 to KWFF0	0x0E1000 to 0x0E11FE	0x00070800 to 0x000708FF
System register (word)	SW000 to SWBF0	0x0E1E80 to 0x0E1FFE	0x00070F40 to 0x00070FFF
Z-register (word)	ZW000 to ZW3F0	0x0E1E00 to 0x0E1E7E	0x00070F00 to 0x00070F3F
Internal register (word)	MW000 to MWFF0	0x0E0E00 to 0x0E0FFE	0x00070700 to 0x000707FF
Transfer register (word)	JW000 to JWFF0	0x0E0200 to 0x0E03FE	0x00070100 to 0x000701FF
Receive register (word)	QW000 to QWFF0	0x0E0600 to 0x0E07FE	0x00070300 to 0x000703FF
Function data register	DW000 to DWFFF	0x061000 to 0x062FFE	0x00030800 to 0x000317FF
Function work register	FW000 to FWBFF	0x0E2000 to 0x0E37FE	0x00071000 to 0x00071BFF
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

(3) Error memory area

The FL.NET module offers tables for recording error information.

Table 2-6 Error Memory Area for Link Data

Item	Specifications
Error message data table	Stores the error messages generated by the other nodes.

For detailed information about the tables, see “7 TROUBLESHOOTING.”

(4) Status memory area

For the self-node status of the FL.NET module, FA link status, and Higher layer status you can refer to the setup tool [S10V FL.NET SYSTEM].

2.1.4 Link parameter setup area

The link parameters for the FL.NET module can be referenced with the setup tool named S10V FL.NET SYSTEM (for details, see “5.3 Using the FL.NET Module”).

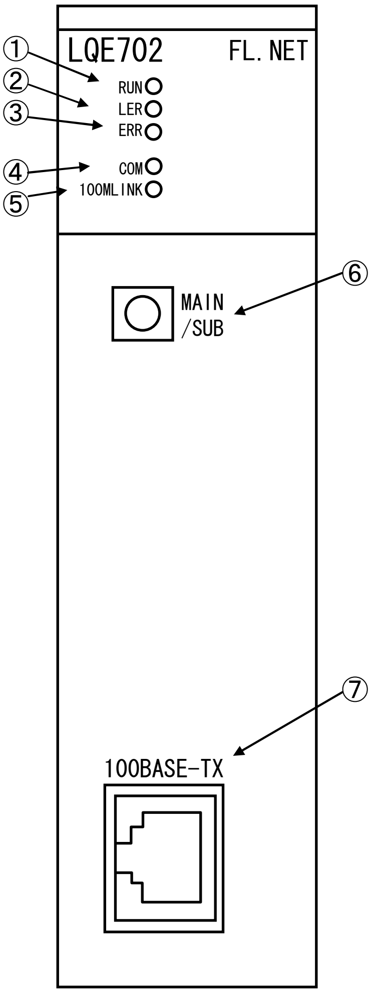
2.1.5 Profile system parameter setup area

The profile system parameters for the FL.NET module can be referenced with the setup tool named S10V FL.NET SYSTEM (for details, see “5.3 Using the FL.NET Module”).
For the FL.NET module, the profile system parameters for the vendor name, manufacturer model, and protocol version are fixed data. The FL.NET module automatically changes the FA link status and self-node status in accordance with the communication status and other conditions.
The user can merely perform node name setting with the setup tool.

Table 2-7 Profile System Parameter Setup Area (Description example)

Item	Length	Data	Description
Vendor name	10 bytes	“HITACHI”	Vendor name
Maker form	10 bytes	“LQE702”	Maker form/device name
Node name (equipment name)	10 bytes	————	User-defined node name
Protocol version	1 byte	————	Fixed at 0x80.
FA link status	1 byte	————	Entering/leaving, etc.
Self-node status	1 byte	————	Node number duplication detection, etc.

2.2 FL.NET Module Component Names and Functions



No.	Name	Function
①	RUN LED	This LED indicates when the module is up and running normally.
②	LER LED	This LED indicates when a unconnected FA link, or FA link communication failure.
③	ERR LED	This LED indicates when a fatal fault occurs.
④	COM LED	This LED indicates while data flows along a transmission path (a carrier is detected).
⑤	100MLINK LED	This LED indicates when the transmission line is connected for communication at 100 Mbps.
⑥	MAIN/SUB selector switch	Used to define the module as the main or submodule. The allowable set value is one of the following: Switch setting: 0 — Indicates that it is defined as the main module. 1 — Indicates that it is defined as the submodule. 2 to B — Prohibited C to F — Prohibited (used for maintenance purposes)
⑦	RJ45 connector	Used for data communication through the 100BASE-TX interface.

Switch setting	Explanation
0	Indicates that it is defined as the main module.
1	Indicates that it is defined as the submodule.
2 to B	Prohibited
C to F	Prohibited (used for maintenance purposes)

NOTICE

Switch off the power supply before operating the MAIN/SUB setting switch. If you operate while the power supply is applied, it may result in a malfunction.

3 FL.NET MODULE INSTALLATION

3.1 Mount Base

The FL.NET module can be mounted in the mount bases shown in Table 3-1.

Table 3-1 Mount Bases Applicable to the FL.NET Module

Name	Model	Specification
4-slot mount base	HSC-1540	Power supply + LPU + 4 slots (option, for I/O)
8-slot mount base	HSC-1580	Power supply + LPU + 8 slots (option, for I/O)

3.2 Mounting the Module

Mount the option module in an option slots (slot number 0 through 7) on the mount base as shown below.

NOTICE

- There are no specific rules about the mounting position or unoccupied slots.
- This model-LQE702 FL.NET module may not be mounted on the same mount base in conjunction with any other FL.NET module of models LQE500 or LQE502 and any EQ.LINK module of model LQE701.

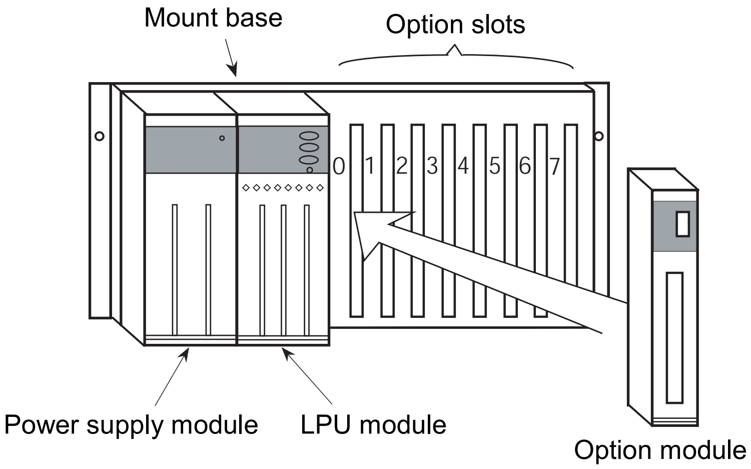
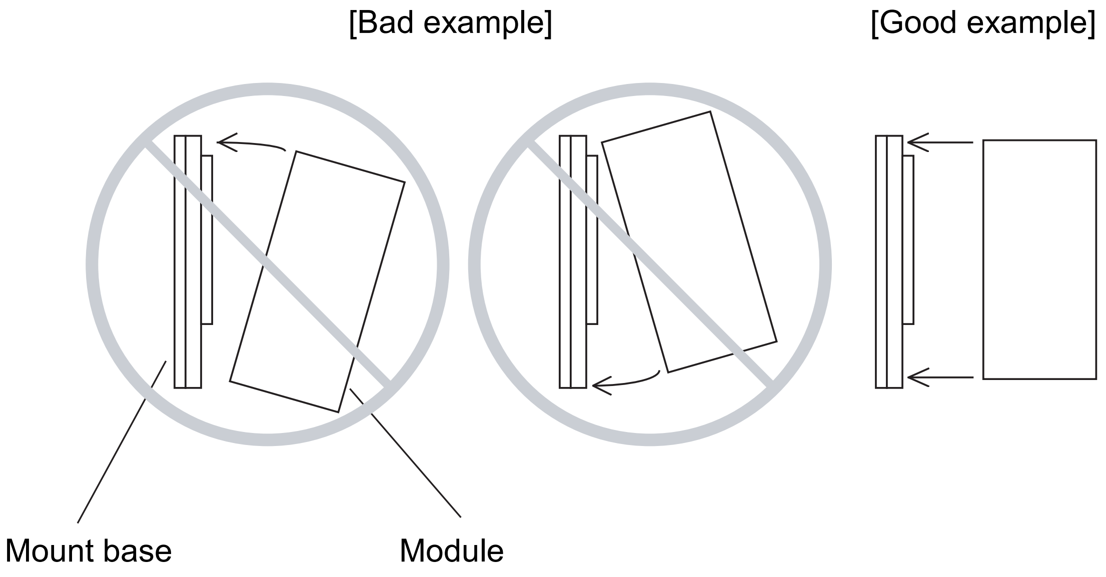


Figure 3-1 Mounting the Option Module

NOTICE

- Dust or other foreign matter might accumulate on the connector, resulting in poor contact. Immediately after the module is unpacked, perform the mounting and wiring procedures.
- To prevent the module from being damaged, observe the following precautions when you mount or demount the module:
  - Before mounting the module to the mount base connector, check that the connector pins are properly aligned and not bent, broken, or soiled with dirt or the like.
  - Ensure that the module is parallel to the mount base vertical surface as shown below when mounting. If you connect a module to or disconnect it from its connector while it is tilted, the connector pins may become damaged.
  - If the mount base is positioned overhead due to the employed enclosure structure, use a stepladder or the like and mount the module squarely. If you mount the module obliquely, the connector may become damaged.



3.3 Setting the MAIN/SUB Selector Switch

When using the FL-net module, it is necessary to perform main module/submodule setup. The main module/submodule setup can be accomplished by setting the MAIN/SUB selector switch either in “0” position (main module) or “1” position (submodule). If only one FL.NET module is mounted in the mount base, set its MAIN/SUB selector switch in “0” position.

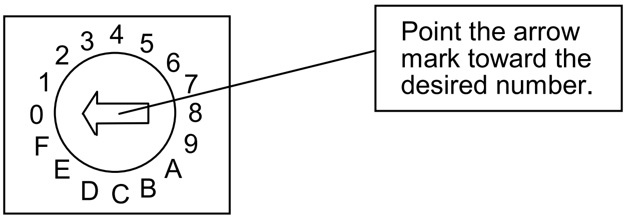

2 3 4 5 6 7 8 9 A B C D E F 0 1
Point the arrow mark toward the desired number.

Figure 3-2 Setting the MAIN/SUB Selector Switch

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

Table 3-2 Details of MAIN/SUB Selector Switch

MAIN/SUB		Meaning of setting
Main module	Submodule	
0	1	100BASE-TX communication.
2	3	These settings are not supported by the FL.NET module. If the FL.NET module is given any of these settings, it will encounter a MAIN/SUB switch setting error during operation, which makes data communication impossible.
4	5	
6	7	
8	9	
A	B	
C	D	These settings are reserved for maintenance activities, so do not use them during regular operation.
E	F	

NOTICE

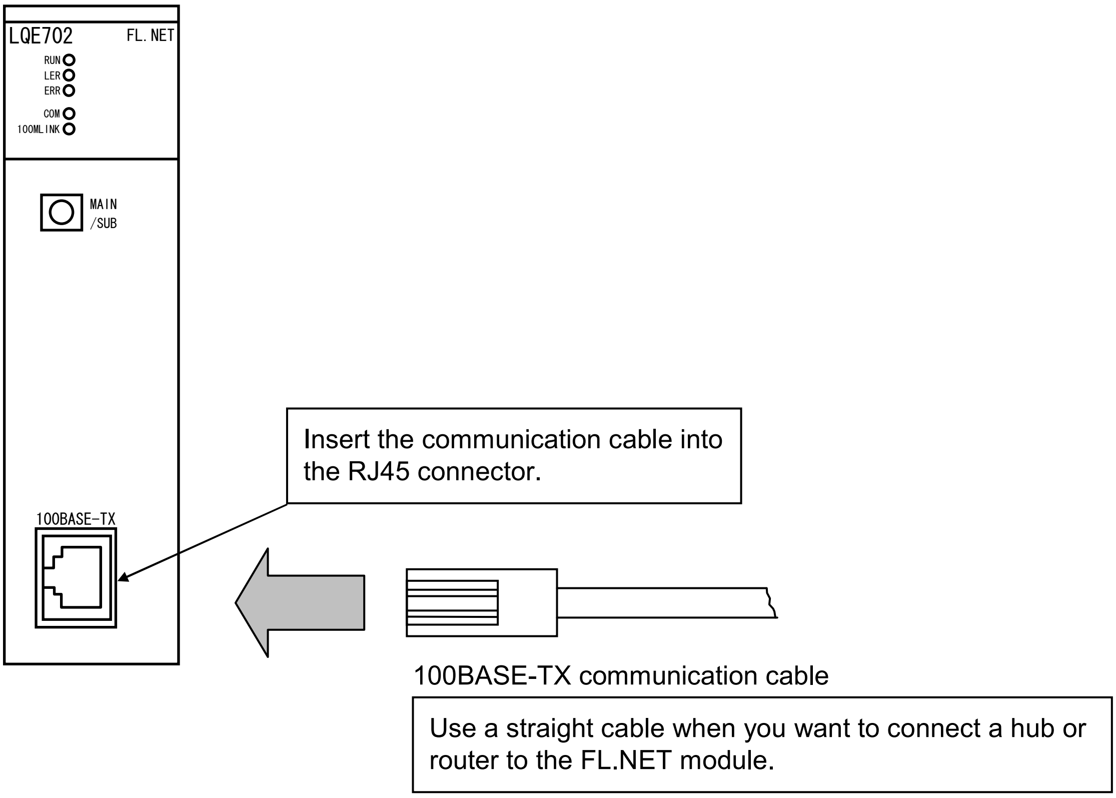
Switch off the power supply before operating the MAIN/SUB setting switch. If you operate while the power supply is applied, it may result in a malfunction.

4 FL.NET MODULE WIRING PROCEDURES

4.1 Connecting the Communication Cable

The module uses the 100BASE-TX interface connector to establish communication with the other modules.

Ensure that the cable is connected as shown in Figure 4-1.



Insert the communication cable into the RJ45 connector.

100BASE-TX communication cable

Use a straight cable when you want to connect a hub or router to the FL.NET module.

Figure 4-1 100BASE-TX Communication Cable Wiring

NOTICE

- This hardware unit may malfunction if it is connected poorly or has a broken line. After connecting the connector, check whether the locking.
- Do not touch the connector during power on. Otherwise, the system may malfunction due to static electricity, etc.

4.2 Recommended Network Components

The LQE702 is a standard product conformed to the global standard of IEEE802.3. It may happen, however, that the LQE702 does not function successfully when used in conjunction with certain network components conforming to the same standard. To avoid this inconvenience, use network components of the make recommended by us to connect to the LQE702.
Table 4-1 give listings of the kinds of network components recommended by us.

Table 4-1 Network Component List

No.	Product name	Manufacturer	Model	Remarks
①	FL.NET	Hitachi, Ltd.	LQE702	
②	Hub	Hitachi, Ltd.	H-7612-90	Switching hub
③	Twisted-pair cable	Hitachi Cable, Ltd.	HUTP-CAT5E-4P***	*** is the cable length.

4.3 System Configuration

Data communication between Ethernet network devices becomes possible if you connect such devices to a hub, as shown in Figure 4-2. To connect Ethernet devices to the hub, use twisted-pair cables.

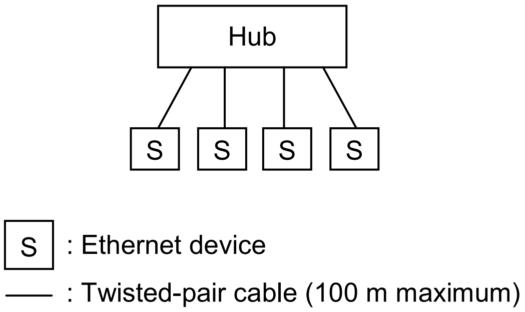


Figure 4-2 Hub and Connected Devices

■ Configurations using 100-Mbps hubs

- If a 100-Mbps hub of class 1 is used to connect Ethernet devices, it may not be connected in a multi-stage form. In this case, connect the hub and Ethernet devices in the following way:

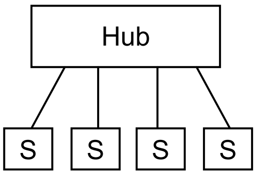


Figure 4-3 A Sample Configuration Using a 100-Mbps Hub of Class 1

- If a 100-Mbps hub of class 2 is used to connect Ethernet devices, it may be connected in a multi-stage form, although the maximum number of 100-Mbps hubs of the same class that can be connected together is 2. In this case, connect the hubs and Ethernet devices in the following way (the maximum allowable length of cable connecting the two hubs together is 5 m):

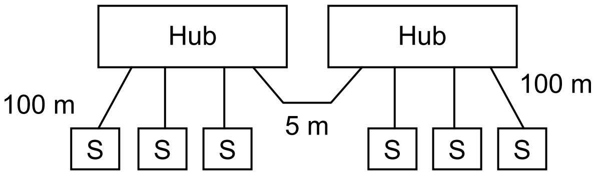


Figure 4-4 A Sample Configuration Using 100-Mbps Hubs of Class 2

- If 100-Mbps switching hubs are used to connect Ethernet devices together in a multi-stage cascade form, there is basically no limitation on the number of stages used in the multi-stage connection. In this case, consult the switching hub manual and connect them according to the instructions given in the manual.

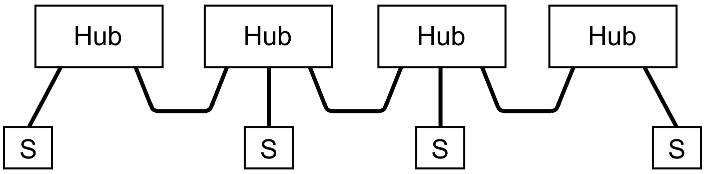


Figure 4-5 A Sample Configuration Using 100-Mbps Switching Hubs

NOTICE

The only port setting that is supported by the FL.NET module (model LQE702) is auto-negotiation. Do not use 100-Mbps full-duplex setting for the port of the switching hub. Disregarding this rule may result in a failure of data communication when the load on the communication line builds up.

4.4 Ground Wiring

The FL.NET module requires no ground wiring.

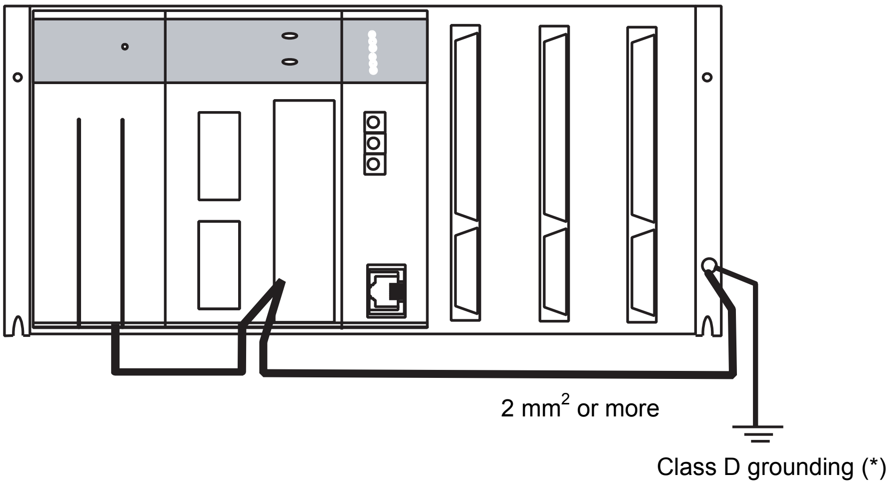


Figure 4-6 Unit Ground Wiring

* Class D grounding is defined in the Technical Standard for Electrical Facilities of Japan. This standard states that the grounding resistance must be 100 ohms or less for equipment operating on 300 VAC or less, and 500 ohms or less for devices that shut down automatically within 0.5 seconds when shorting occurs in low tension lines.

NOTICE

- Ground the FG (frame ground) terminal as follows: Connect the FG terminal on each module provided with external terminals to the grounding terminal on the mount base. Perform Class D grounding for the grounding terminal on the mount base.
- Use ground lines whose size is 2 mm^2 or more.

5 USER GUIDE

5.1 FL-net

5.1.1 FL-net overview

(1) FL-net concept

The FL-net is an Ethernet-based factory-automation control network. It has a cyclic transmission function and message transmission function.

The basic ideas incorporated in the FL-net are as follows:

- Ethernet is used as the media (physical level, data link) of communication between FA (factory automation) controllers.
- UDP/IP prevalent throughout Ethernet is used to implement a basic means of data transmission.
- Transmission over a predetermined period of time is assured by managing/controlling communication media access of networked nodes (for collision avoidance) while making use of the above-mentioned basic means of data transmission.

The FL-net is designed for use in an FA (factory automation) control network that provides data exchanges between programmable controllers (PLCs), robot controllers (RCs), computer numerical control (CNC) devices, and various other production system controllers and control personal computers.

Figure 5-1 shows the role of the FL-net.

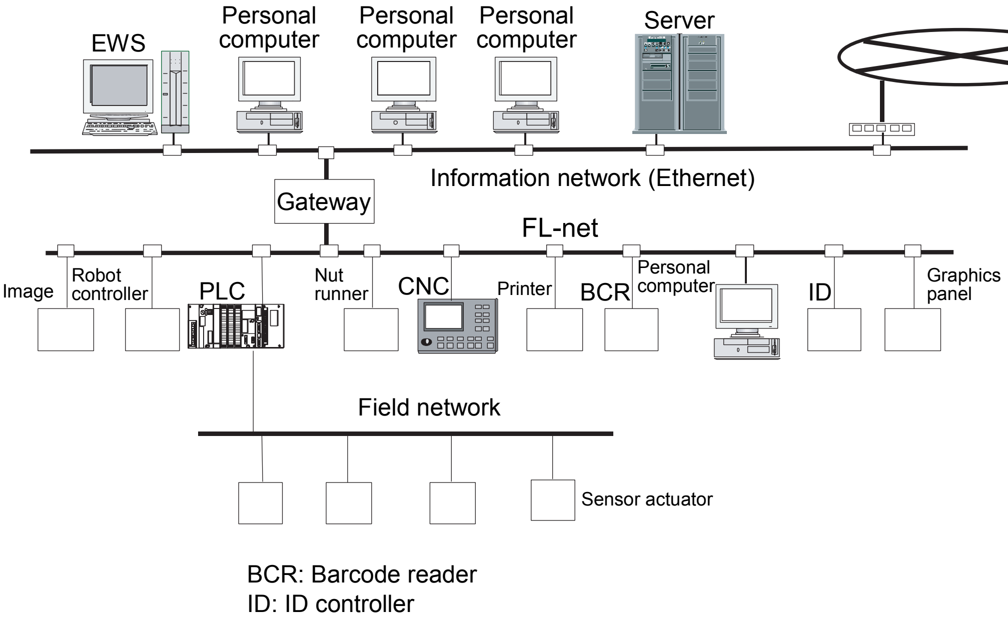


Figure 5-1 FL-net Concept

(2) FL-net protocol

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

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

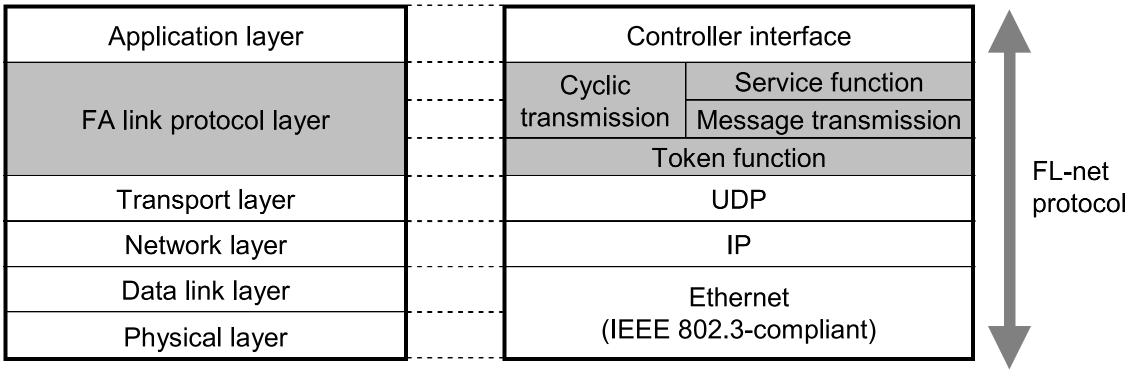


Figure 5-2 FA Link Protocol

(3) FL-net transmission system features

The features of the FL-net FA link protocol layer are outlined below:

- Masterless, token-based transmission management is exercised to avoid collisions.
- A token can be circulated at fixed intervals to define the refresh cycle time.
- A predetermined token is transmitted together with cyclic data.
- At startup, a token is transmitted from a node having the lowest node number.
- If a token is not transmitted for a predetermined period of time, the next node transmits a node.
- Thanks to the use of a masterless, token-based system, network inoperativeness can be avoided even when some nodes become faulty.
- Operating mode (RUN/STOP), hardware error (ALARM), and other information management tables are available for referencing the operating status of the other nodes.

(4) FL-net IP address

The IP address is an address that is used in IP(Internet Protocol)-based transmission to indicate a specific node (station). It is therefore necessary to set and manage IP addresses so as to avoid address duplication.

For each FL-net node, an IP address needs to be set. It is recommended that you use Class C IP addresses in accordance with the FL-net protocol.

The IP address default value for the FL-net is 192.168.250.***, where the *** portion is a node number. (The default value is recommended by the FL-net protocol.)

Network address	Host number (node number)
192.168.250.	n (n: 1 to 254)

Figure 5-3 FL-net IP Address

5.1.2 Connection capacity and node numbers

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

Node number: 1 to 249	For normal FL-net devices.
Node number: 250 to 254	For FL-net maintenance.
Node number: 255	Used within the FL-net (used for global address broadcasting) and not available to the user.
Node number: 0	Used within the FL-net and not available to the user.

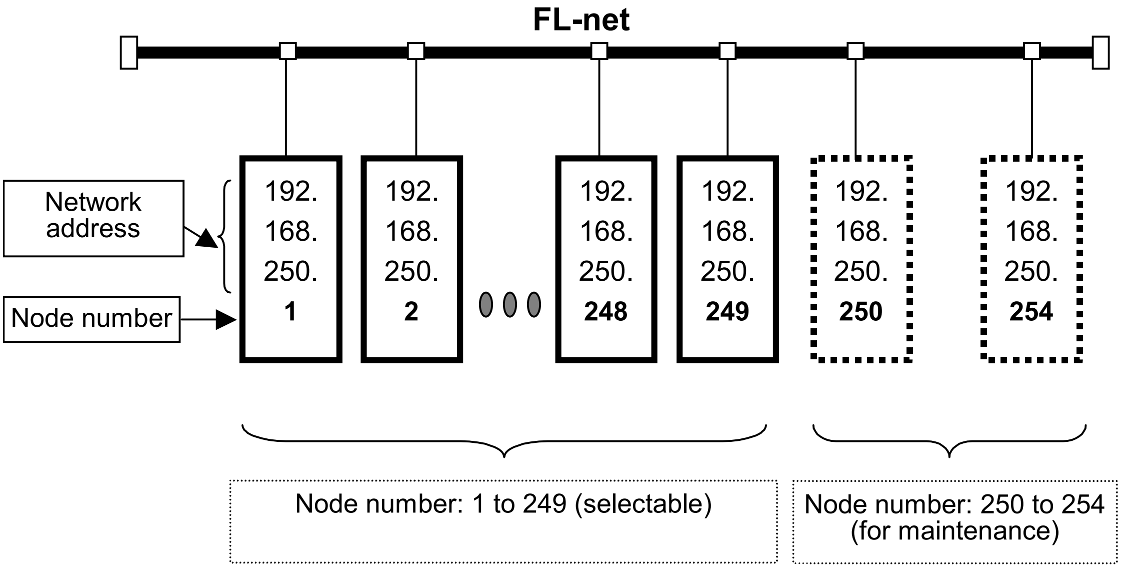
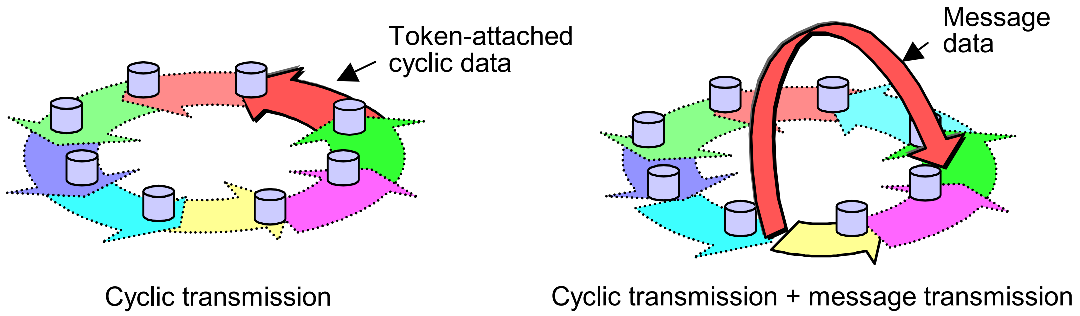


Figure 5-4 FL-net Connection Capacity and Node Numbers

5.1.3　Supported data communications

Data communications supported by the FL-net are cyclic transmission and message transmission.



Cyclic transmission

Cyclic transmission + message transmission

Figure 5-5　Data Communication Types Supported by FL-net

(1)　Cyclic transmission

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

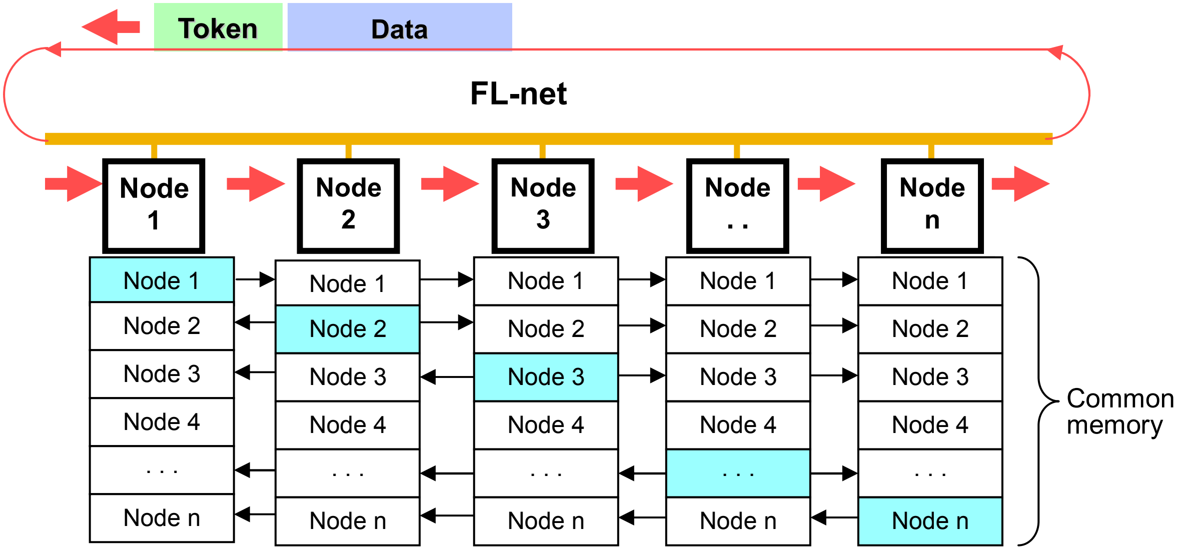


Figure 5-6　Typical Common Memory and Cyclic Transmission

(2) Message transmission

Message transmission provides nonperiodic transmission of data. Under normal conditions, communication is transmitted to a specific node upon request.

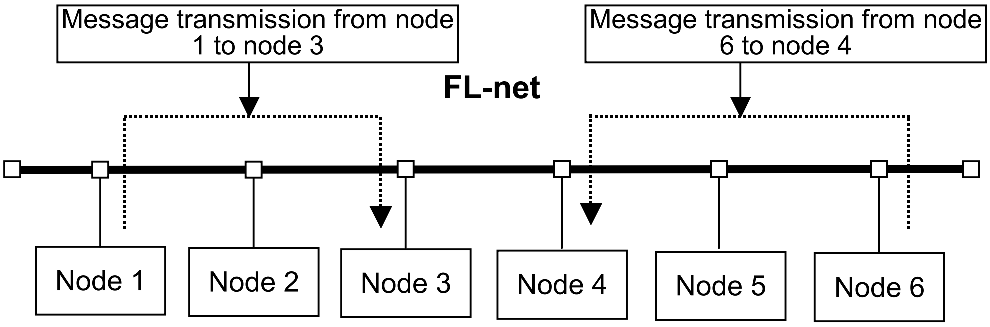

Message transmission from node 1 to node 3
Message transmission from node 6 to node 4
FL-net
Node 1
Node 2
Node 3
Node 4
Node 5
Node 6

Figure 5-7 Typical Message Transmission

5.1.4 Amount of data transmission

(1) Cyclic transmission

The entire network has 8.5k words (= 8k bits + 8k words). The maximum permissible amount of transmission is 8.5k words per node. Note that each word consists of two bytes.

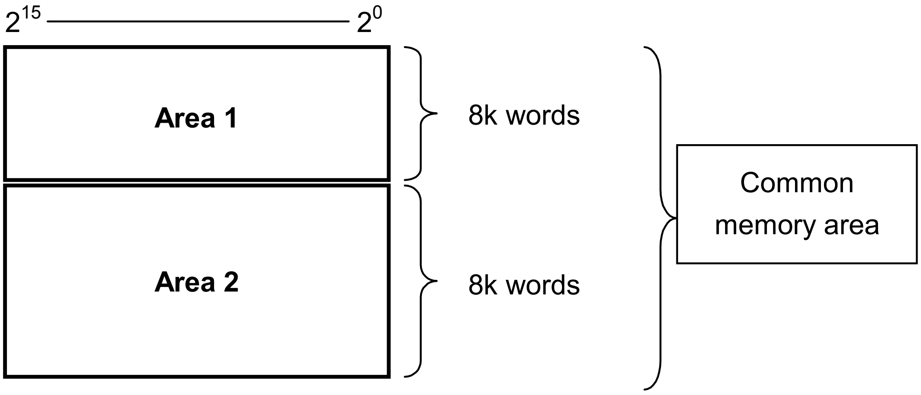


Figure 5-8 Data Amount of Cyclic Transmission

(2) Message transmission

The permissible amount of one message frame is maximum 1,024 bytes (excluding the header).

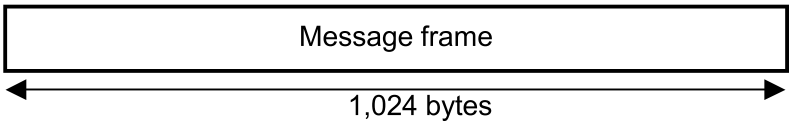


Figure 5-9 Data Amount of Message Transmission

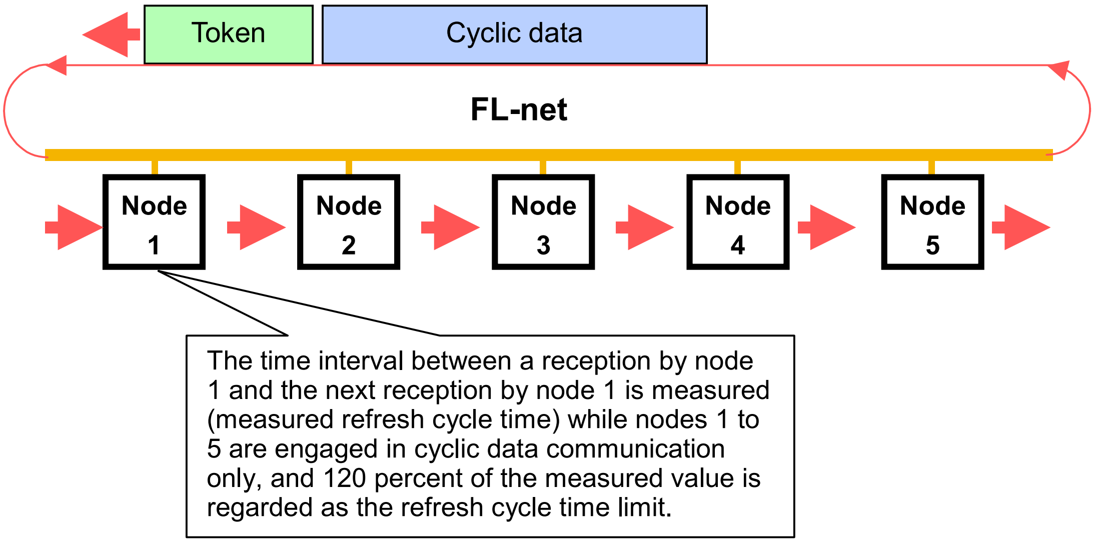
5.1.5 Transfer cycle and monitoring

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

Each node constantly monitors message communication frames flowing within the network during the time interval between the reception of a token addressed to itself and the reception of the next token addressed to itself. When no message communication frame flows within the network during such one cycle, 120 percent of the duration of one such cycle is used as the refresh cycle time limit.

Thanks to the above monitoring process, the refresh cycle time limit is dynamically determined by the number of participating nodes within the network.

Example: When only cyclic data communication is maintained by five nodes (no message communication is effected by any nodes)



The time interval between a reception by node 1 and the next reception by node 1 is measured (measured refresh cycle time) while nodes 1 to 5 are engaged in cyclic data communication only, and 120 percent of the measured value is regarded as the refresh cycle time limit.

Figure 5-10 Typical Refresh Cycle Time Limit

For the calculation procedure, see “5.3.7 FL-net module communication performance.”

5.1.6 Data area and memory

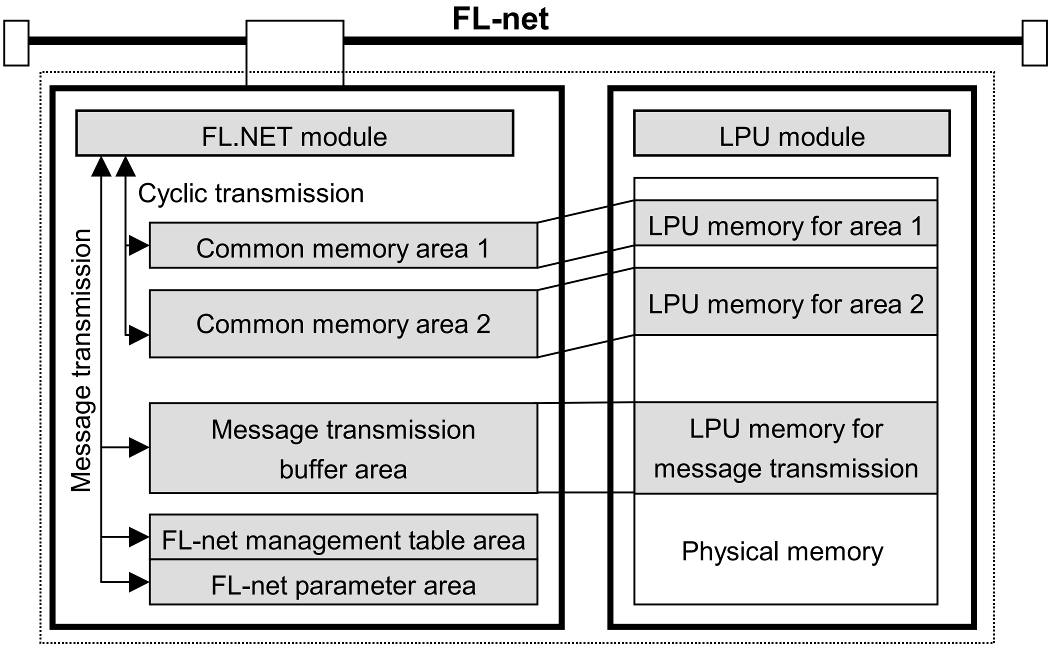


Figure 5-11 Data Area and Memory

5.1.7 Communication management tables

Node status management is provided by means of the self-node management table, participating node management table, and network management table.

(1) Self-node management table

The self-node management table is used to manage the self-node setup.

Table 5-1 Self-node Management Table

Name	Length	Description/data range
Node number	1 byte	1 to 254
Area 1 data starting address for common memory	2 bytes	Word address (0 to 0x1FF)
Area 1 data size for common memory	2 bytes	Size (0 to 0x1FF)
Area 2 data starting address for common memory	2 bytes	Word address (0 to 0x1FFF)
Area 2 data size for common memory	2 bytes	Size (0 to 0x1FFF)
Higher layer status	2 bytes	RUN/STOP/ALARM/WARNING/NORMAL
Token surveillance timeout	1 byte	Variable in 1ms units
Minimum frame interval	1 byte	Variable in 100μs units
Vendor name	10 bytes	Name of the vendor
Maker form	10 bytes	Maker form/device name
Node name (equipment name)	10 bytes	User-defined node name
Protocol version	1 byte	Fixed at 0x80
FA link status	1 byte	Entering/leaving, etc.
Self-node status	1 byte	Node number duplication detection, etc.

(2) Participating node management table

The participating node management table is used to manage the information about nodes that have participated in 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
Area 1 data starting address for common memory	2 bytes	Word address (0 to 0x1FF)
Area 1 data size for common memory	2 bytes	Size (0 to 0x1FF)
Area 2 data starting address for common memory	2 bytes	Word address (0 to 0x1FFF)
Area 2 data size for common memory	2 bytes	Size (0 to 0x1FFF)
Refresh cycle time	2 bytes	Variable in 1ms units
Token surveillance timeout	1 byte	Variable in 1ms units
Minimum frame interval	1 byte	Variable in 100 µs units
FA link status	1 byte	Entering/leaving information, etc.

(3) Network management table

The network management table manages the information common in the network.

Table 5-3 Network Management Table

Name	Length	Description/data range
Token maintenance node number	1 byte	Node that currently holds the token
Minimum frame interval	1 byte	Variable in 100 µs units
Refresh cycle time	2 bytes	Variable in 1ms units
Refresh cycle measurement time (current)	2 bytes	Variable in 1ms units
Refresh cycle measurement time (maximum)	2 bytes	Variable in 1ms units
Refresh cycle measurement time (minimum)	2 bytes	Variable in 1ms units

5.1.8 Cyclic transmission and area

(1) Cyclic transmission overview

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

- Transmits data when the node holds the token.
- Recognizes nodes conducting no cyclic transmission as far as they participate in the network.
- Ensures that all the cyclic data to be transmitted are transmitted when the token is held.

Token: Only one token basically exists within the network. If two or more tokens should exist within the network, the one with the lowest destination node number takes precedence with the others discarded.

Token frame: A token frame (frame that contains the token) consists of a token destination node number and token transmission node number.
A node becomes a token holder when it agrees with the destination node number for the token in a token frame it receives.

Token sequence: The order of token rotation is determined by a node number. Token rotation occurs in the ascending order of nodes registered in the participating node management table. A node having the maximum node number passes the token to a node having the minimum node number.

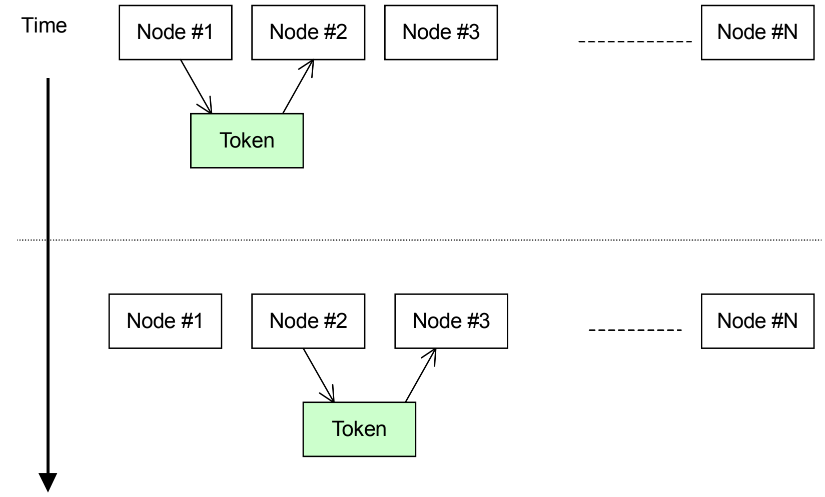


Figure 5-12 Token Rotation and Cyclic Transmission 1

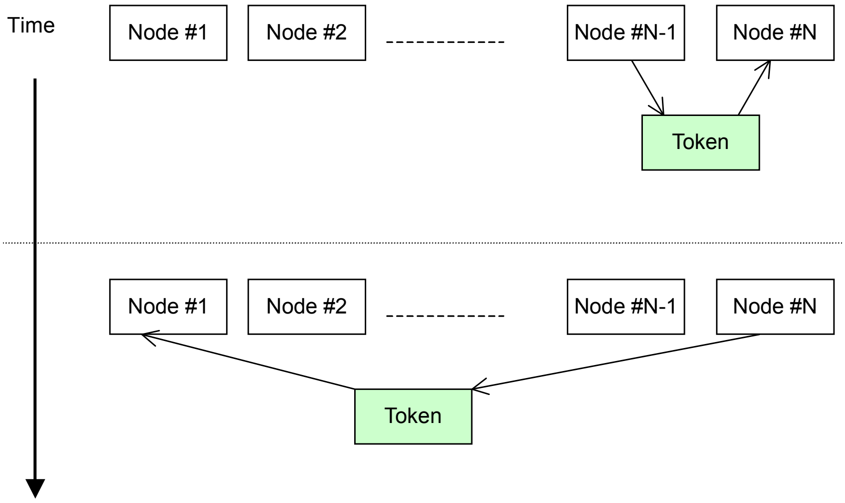


Figure 5-13 Token Rotation and Cyclic Transmission 2

(2) Common memory

The ideas of the common memory are described below.

- Furnishes the nodes conducting cyclic transmission with functions that can be handled as the functions of the common memory.
- Two areas (Area 1 and 2) can be allocated for one node.
- Two or more frames are used for data transmission when the 1,024-byte transmission size limit per frame is exceeded by an area transmitted by one node.
- In the reception of data divided into two or more frames, the common memory does not refresh until all the frames coming from a node are completely received.
- The common memory size furnished by the communication section of one node is fixed at 8.5k words (= 8k bits + 8k words).
- As the common memory's transmission area for one node, Area 1 and 2 can be both set up as desired as far as the maximum permissible size is not exceeded.
- The same data can be shared by the entire system when individual nodes broadcast data at fixed intervals. Nodes on the FL-net 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 one node is the reception area for another node.

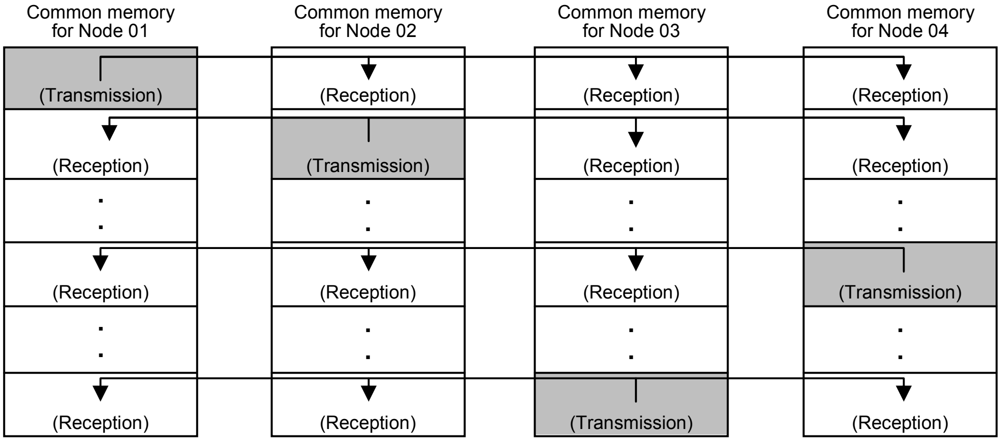


Figure 5-14 Cyclic Transmission Common Memory Area Example 1

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

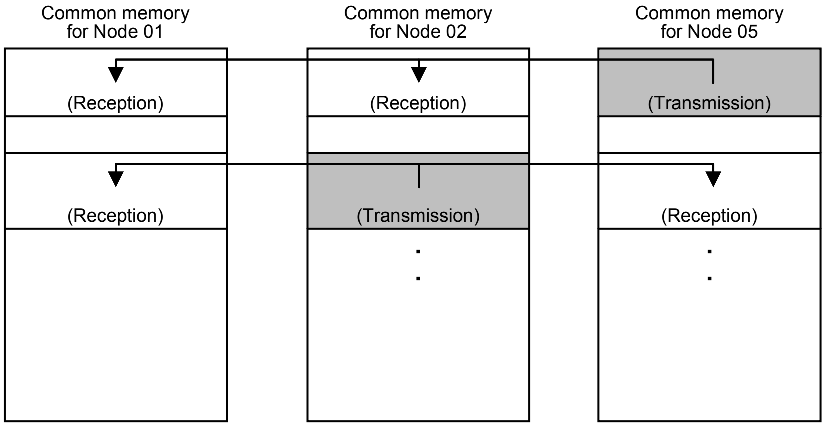

Common memory for Node 01
Common memory for Node 02
Common memory for Node 05
(Reception)
(Reception)
(Transmission)
(Reception)
(Transmission)
(Reception)

Figure 5-15 Cyclic Transmission Common Memory Area Example 2

(3) Area 1 and 2

One node can allocate two data areas (Area 1 and 2) in the common memory. Transmission area setup is performed in accordance with the area starting address and size.

For area access, a word address is used. Area 1 consists of 0.5k words and Area 2, 8k words.

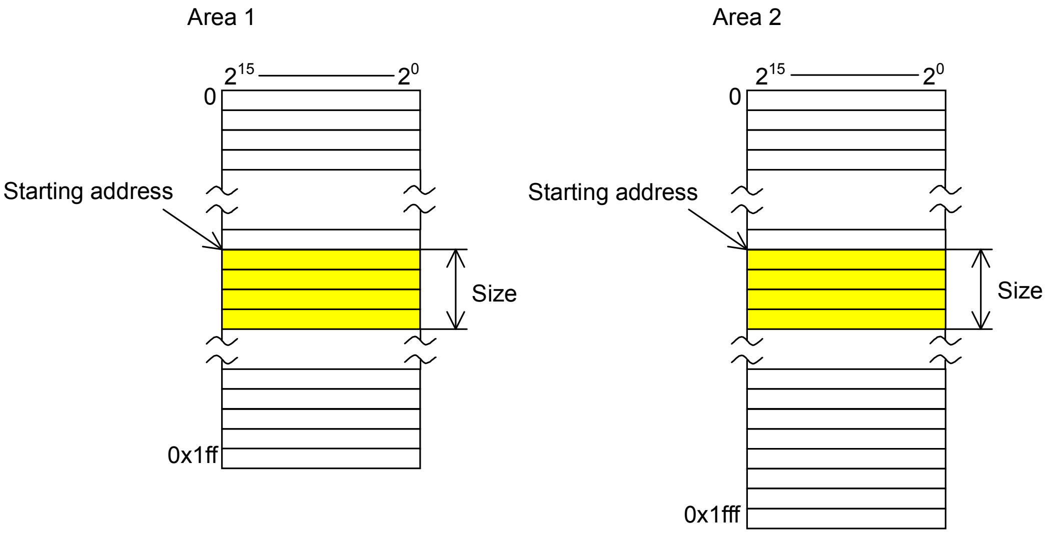


Figure 5-16 Common Memory Area 1 and 2

(4) Data synchronicity assurance

In cyclic transmission, the data to be transmitted is divided into frames depending its size. The following procedures are used to assure common memory (internal memory such as the FL.NET module) synchronicity on an individual 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 sequentially transmitted. If the size of the data retained by a transmitting node is larger than the maximum permissible size limit for the transmission of one frame, the buffered data is divided into two or more frames for transmission.

- Reception sequence refresh timing

  When a receiving node completely receives cyclic data from another node, it refreshes the associated area while maintaining the synchronicity with a higher layer.
  Even when cyclic data is transmitted after being divided into two or more frames, the associated area refreshes when all the frames sent from a node are completely received.
  If all such transmitted frames are not received, the entire data received from the transmitting node is discarded.

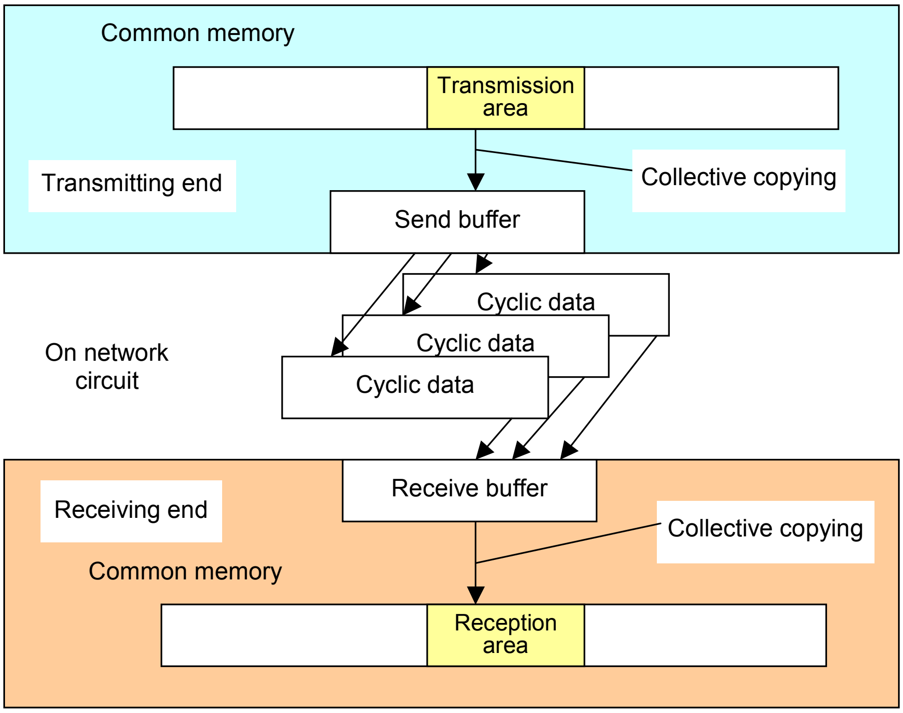


Figure 5-17 Data Synchronicity Assurance

5.1.9 Message transmission

(1) Message transmission overview

The message transmission function supports asynchronous data exchanges that are made between nodes.

The fundamentals of the message transmission function are described below.

- When a node receives the token, it can transmit up to one frame of data before cyclic frame transmission.
- The maximum amount of data that can be transmitted in one session is 1,024 bytes.
- A special algorithm is employed to ensure that the refresh cycle time limit for cyclic transmission is not exceeded.
- One-to-one transmission for sending data to a preselected remote node and one-to-n transmission for sending data to all nodes are both achievable.
- In one-to-one message transmission, the delivery verification feature is available to check whether transmitted data is properly received by a recipient.

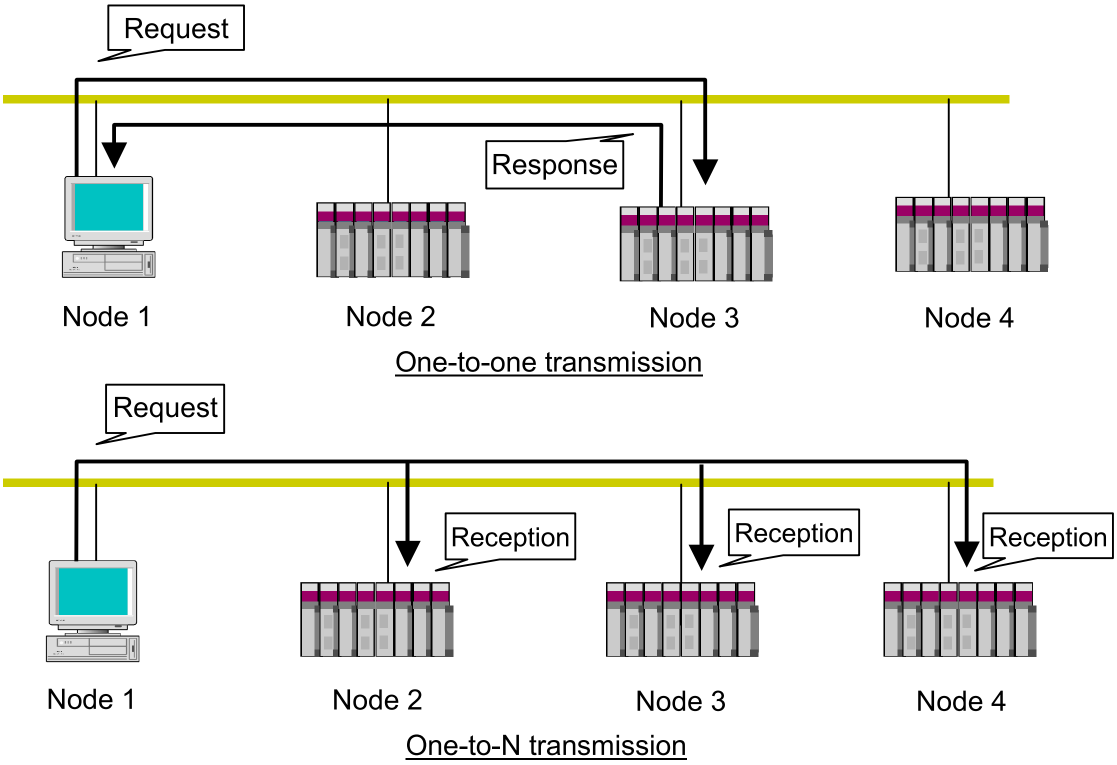


Figure 5-18 Message Transmission Overview

(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	ns	ns
Byte block write	ns	ns
Word block read	√	√
Word block write	√	√
Network parameter read	√	√
Network parameter write	√	√
Run/stop directive	√	ns
Profile read	ns	√
Communication log data read	√	√
Communication log data clear	√	√
Message return	√	√
Transparent message	√	√

√: Supported ns : Not supported

(3) Message function description

- Byte block read

  This message function reads a remote 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.

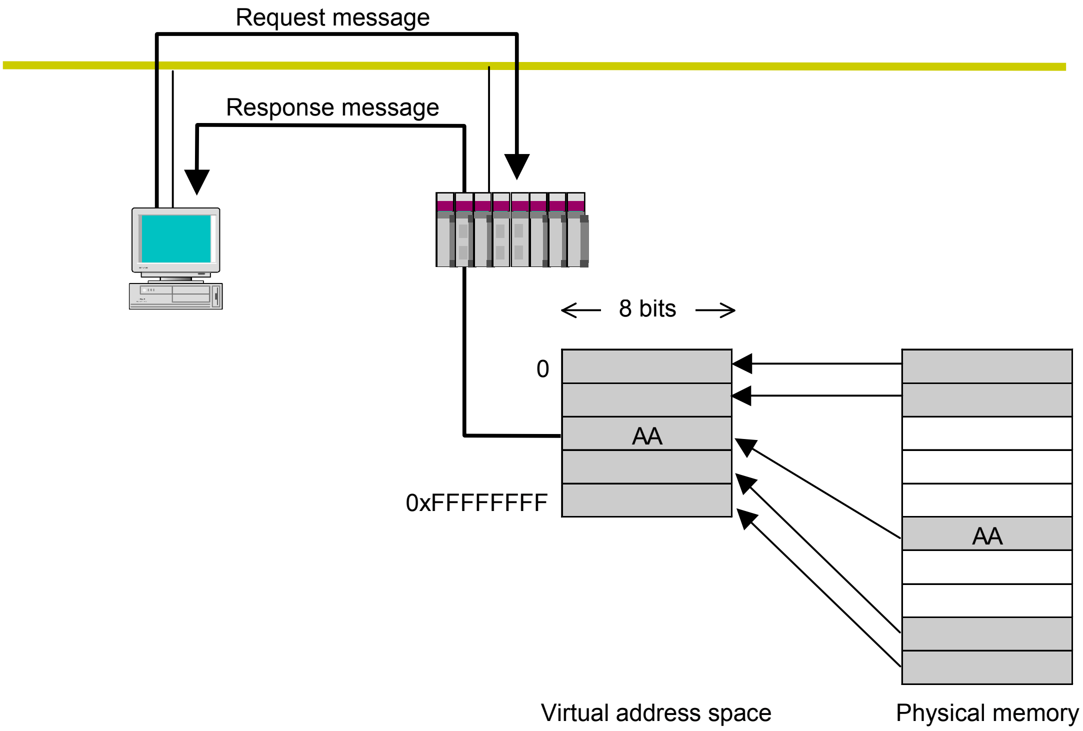


Figure 5-19 Byte Block Read

- Byte block write

  This message function writes into a remote 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.

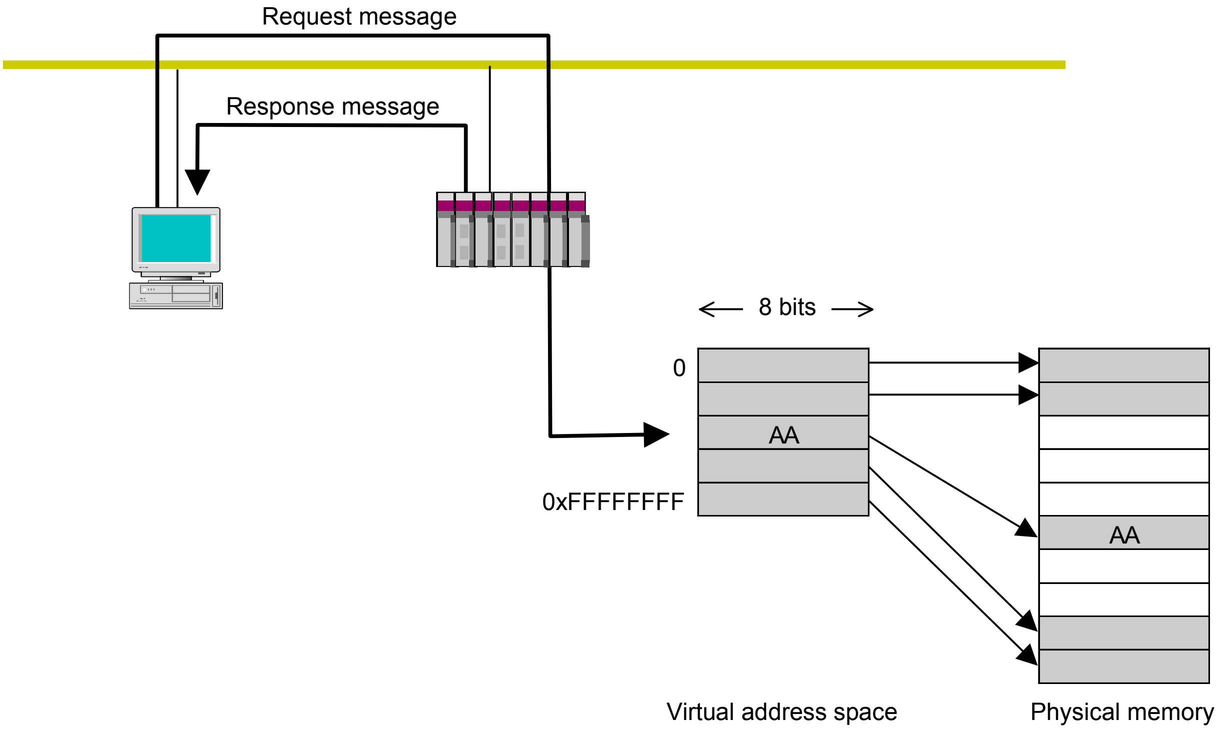


Figure 5-20　Byte Block Write

- Word block read
  This message function reads a remote 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."

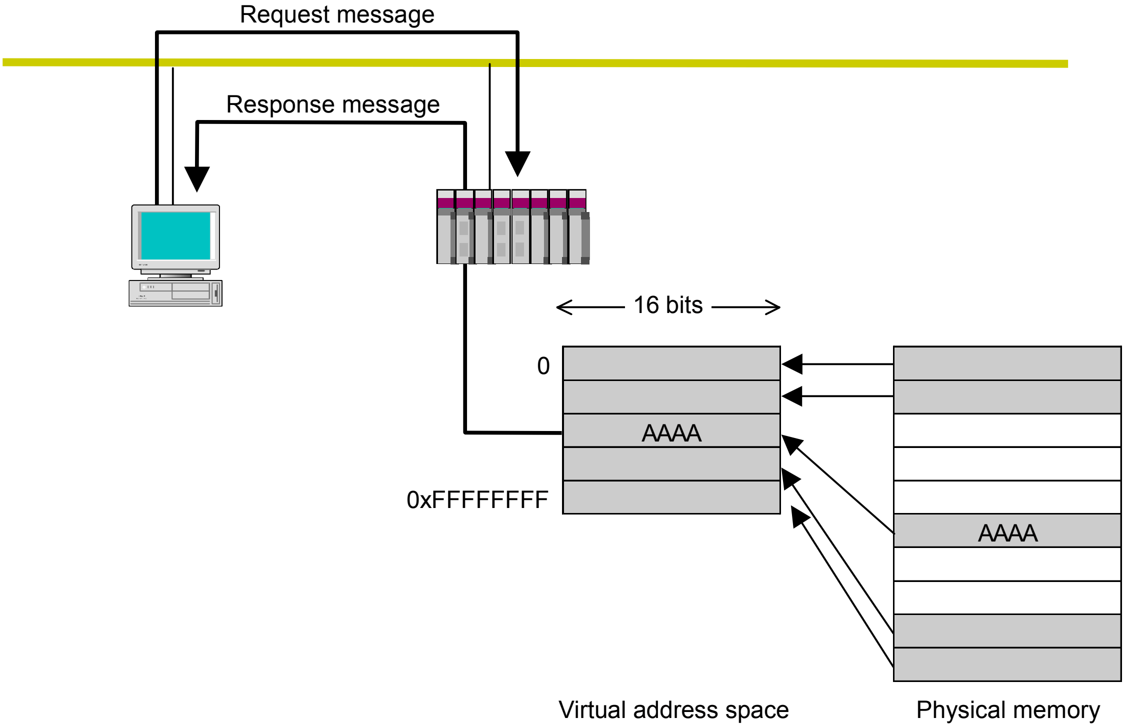


Figure 5-21 Word Block Read

- Word Block Write

  This message function writes into a remote 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.”

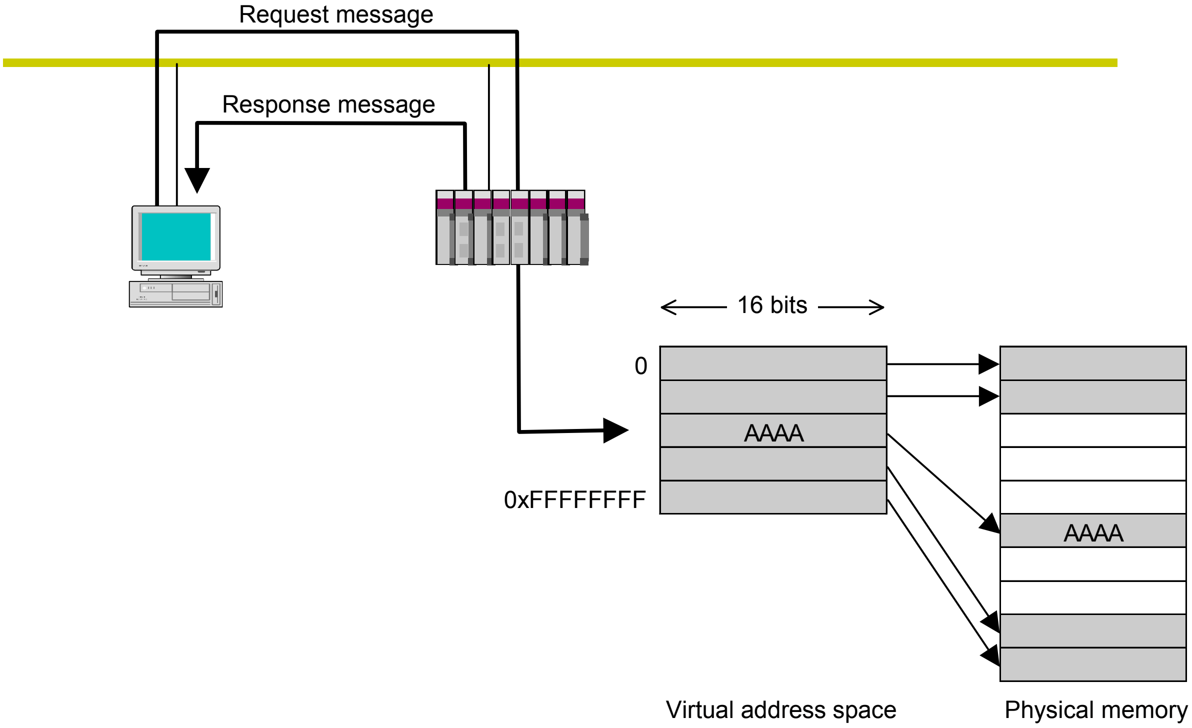


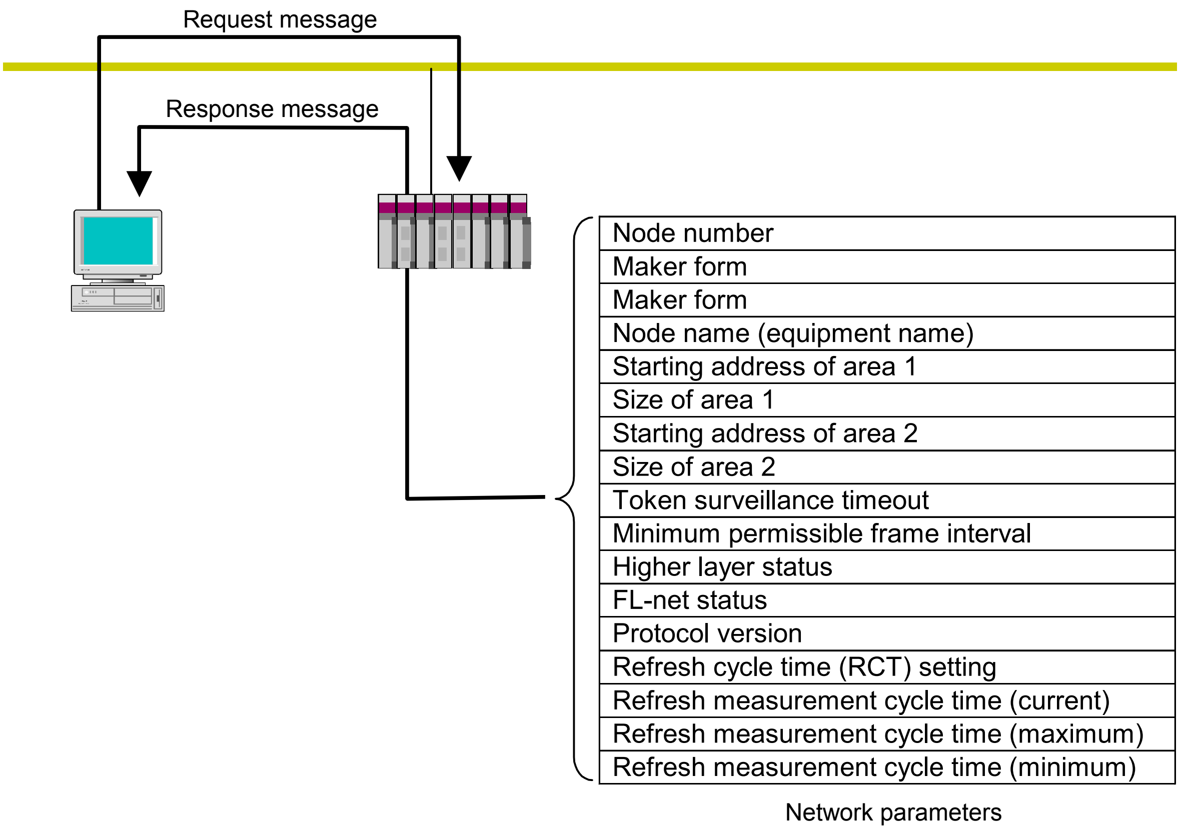
Figure 5-22　Word Block Write

- Network parameter read

  This function reads a remote node's network parameter information from the network. The table 5-5 lists the items of information that this function reads. For use instructions, see "5.3 Using the FL.NET Module."

Table 5-5 Network Parameter Information

Node number
Vendor name
Maker form
Node name (equipment name)
Common memory address and size
Token surveillance timeout
Refresh cycle time
Measured refresh cycle time (measured value)
Minimum frame interval
Higher layer status
FL-net status
Protocol version



Node number
Maker form
Maker form
Node name (equipment name)
Starting address of area 1
Size of area 1
Starting address of area 2
Size of area 2
Token surveillance timeout
Minimum permissible frame interval
Higher layer status
FL-net status
Protocol version
Refresh cycle time (RCT) setting
Refresh measurement cycle time (current)
Refresh measurement cycle time (maximum)
Refresh measurement cycle time (minimum)

Figure 5-23 Network Parameter Read

- Network parameter write

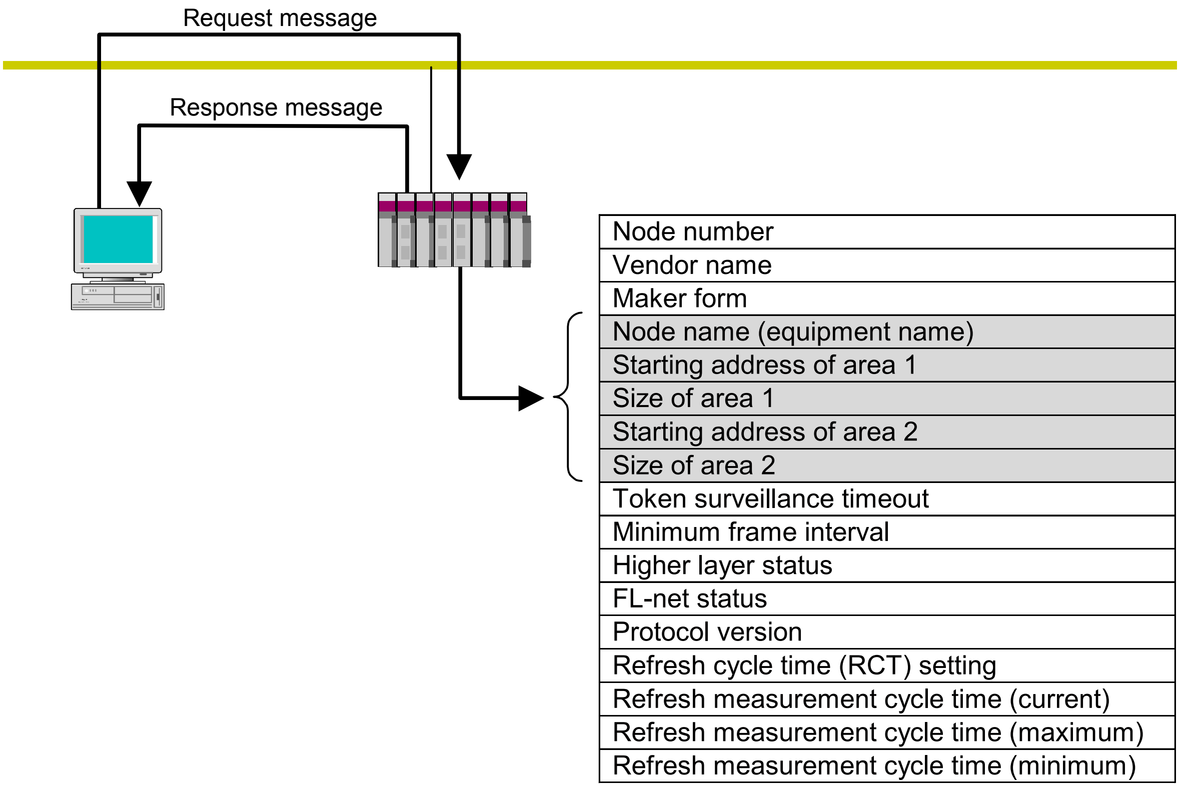
  This function edits a remote node's network parameter information from the network. The following items of information can be edited.　For use instructions, see "5.3　Using the FL.NET Module."

  • Node name (equipment name)

  • Common memory area address and size

  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 common memory address or size is changed, the remote node departs from the network and then participates in it again.　If only the node name is changed, the remote node does not depart from the network.



Node number
Vendor name
Maker form
Node name (equipment name)
Starting address of area 1
Size of area 1
Starting address of area 2
Size of area 2
Token surveillance timeout
Minimum frame interval
Higher layer status
FL-net status
Protocol version
Refresh cycle time (RCT) setting
Refresh measurement cycle time (current)
Refresh measurement cycle time (maximum)
Refresh measurement cycle time (minimum)

Network parameters

Figure 5-24　Network Parameter Write

- Run/stop directive

  This function allows the network to remotely run or stop the operation of a device connected to the FL-net. Only the associated request is supported by the FL.NET module. For use instructions, see “5.3 Using the FL.NET Module.”

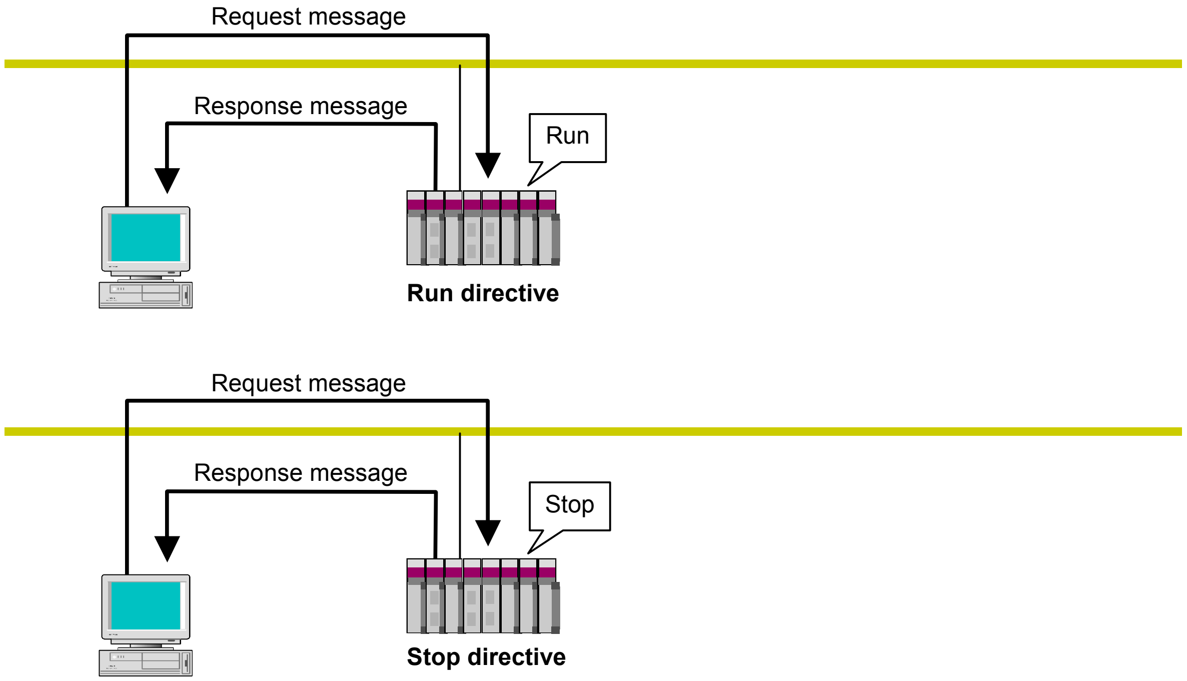


Figure 5-25 Run/Stop Directive

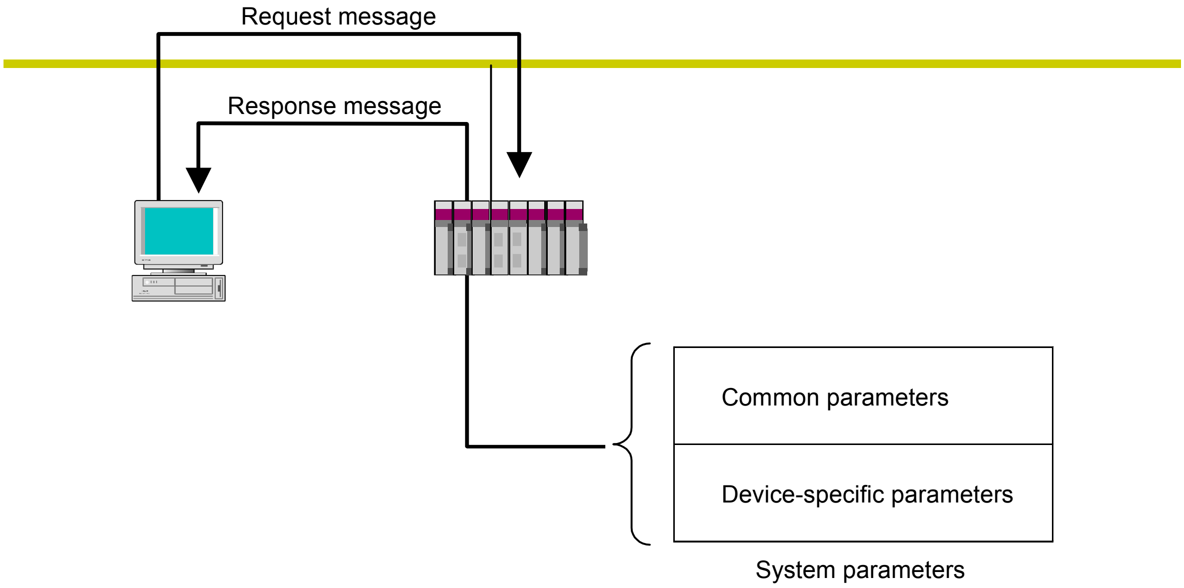
- Profile read

  This function reads from the network the device profile system parameters that represent the information about a remote node. Only the associated response is supported by the FL.NET module.

  The system parameters represent the following items of information:

  • Common parameters (essential)

  • Device-specific parameters (optional)



For the details of device profile system parameters, see “8 APPENDIXES.”

Figure 5-26 Profile Read

- Communication log data read

  This function reads a remote node's log information from the network. For use instructions, see "5.3 Using the FL.NET Module."

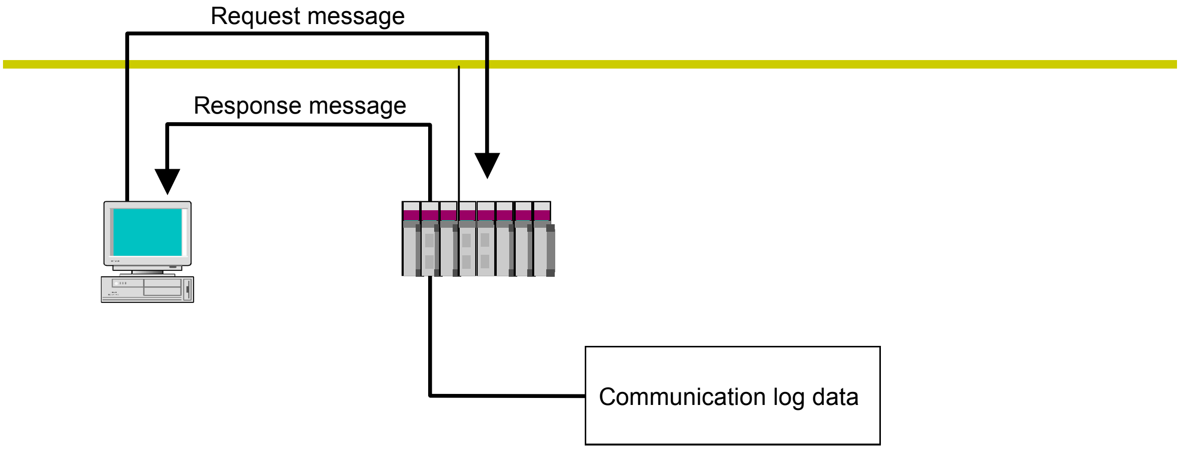


Figure 5-27 Communication Log Data Read

- Communication log data clear

  This function clears a remote node's log information from the network. For use instructions, see "5.3 Using the FL.NET Module."

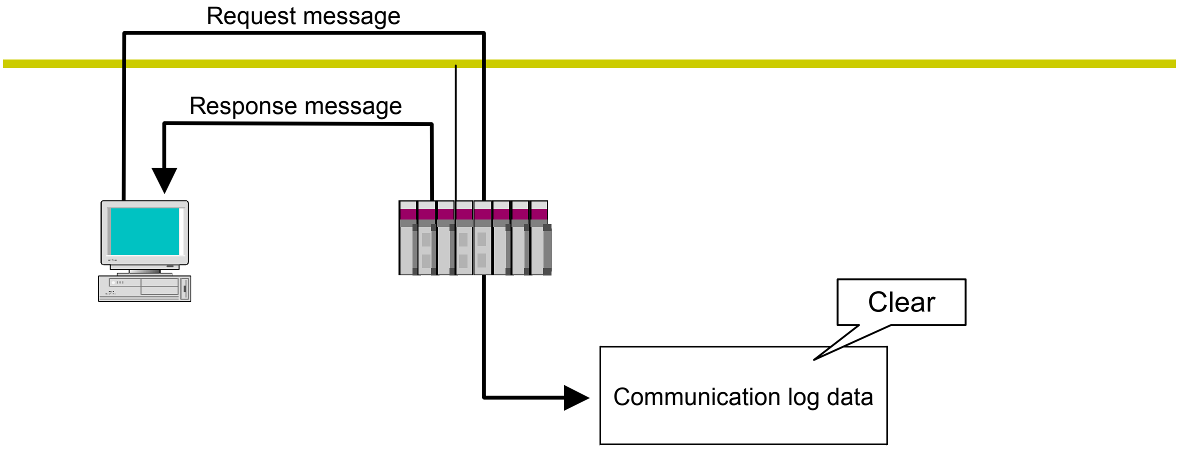


Figure 5-28 Communication Log Data Clear

- Message return

  This function returns a received message.

  A message return automatically takes place within the FL.NET module.

  For use instructions, see “5.3 Using the FL.NET Module.”

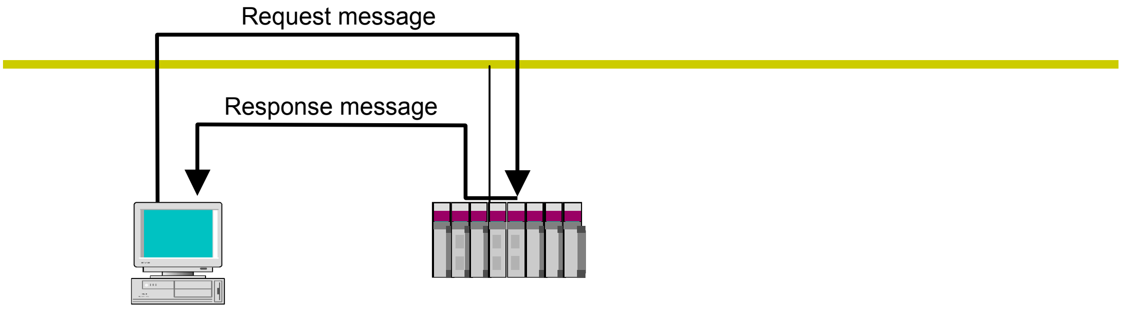


Figure 5-29 Message Return Function

- Transparent message transmission

  This function offers a transparent service to an FL-net higher layer.

  This service notifies an FL-net higher layer of a received message. Upon receipt of such a notification, the FL-net higher layer sends the same notification to the user interface level. When the notification is sent to 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.”

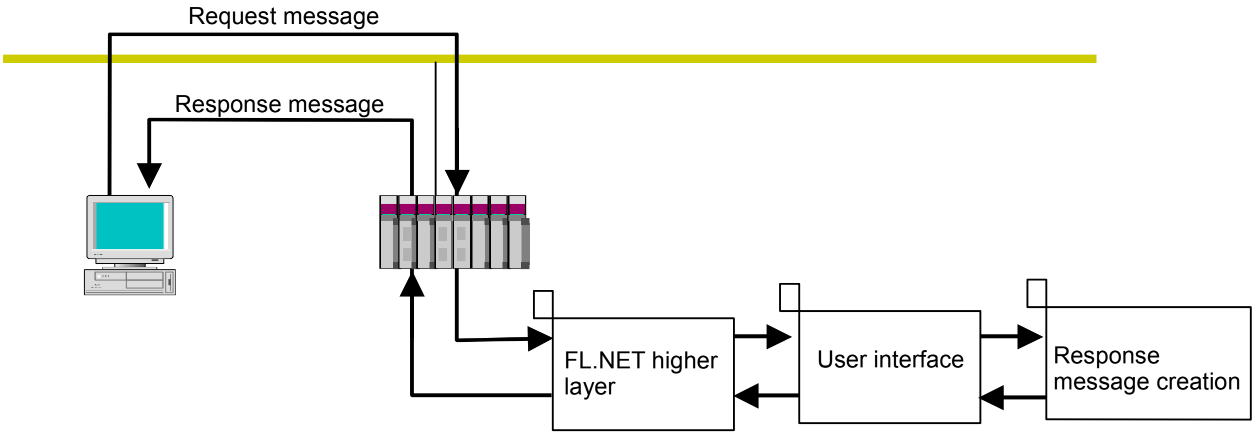
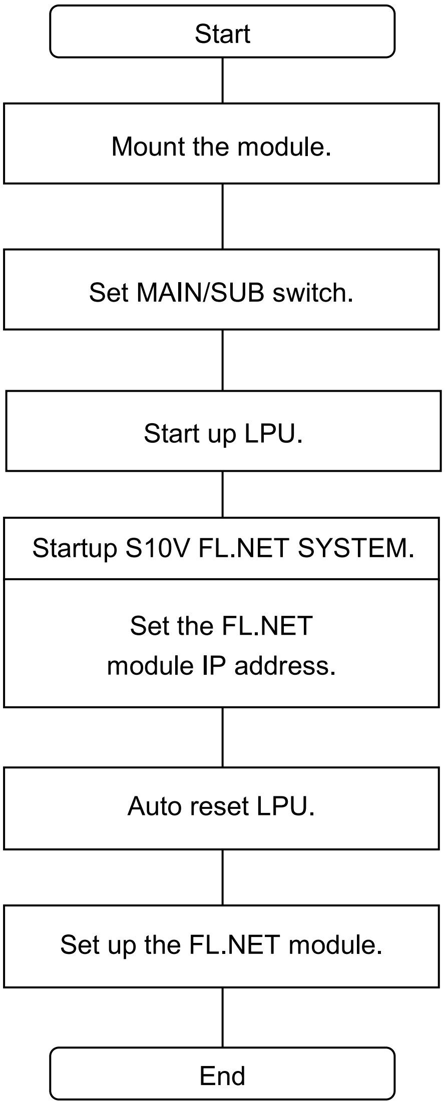


Figure 5-30 Transparent Message Transmission

5.2 FL.NET Module Setup Procedures

5.2.1 Startup procedure

The startup procedure for the FL.NET module is described below:


Start
Mount the module.
Set MAIN/SUB switch.
Start up LPU.
Startup S10V FL.NET SYSTEM.
Set the FL.NET module IP address.
Auto reset LPU.
Set up the FL.NET module.
End

① Switch OFF the LPU unit and then install the FL.NET module.

② Set the MAIN/SUB selector switch on the FL.NET module (See “3.3 Setting the MAIN/SUB Selector Switch.”).

③ Switch ON the LPU unit.

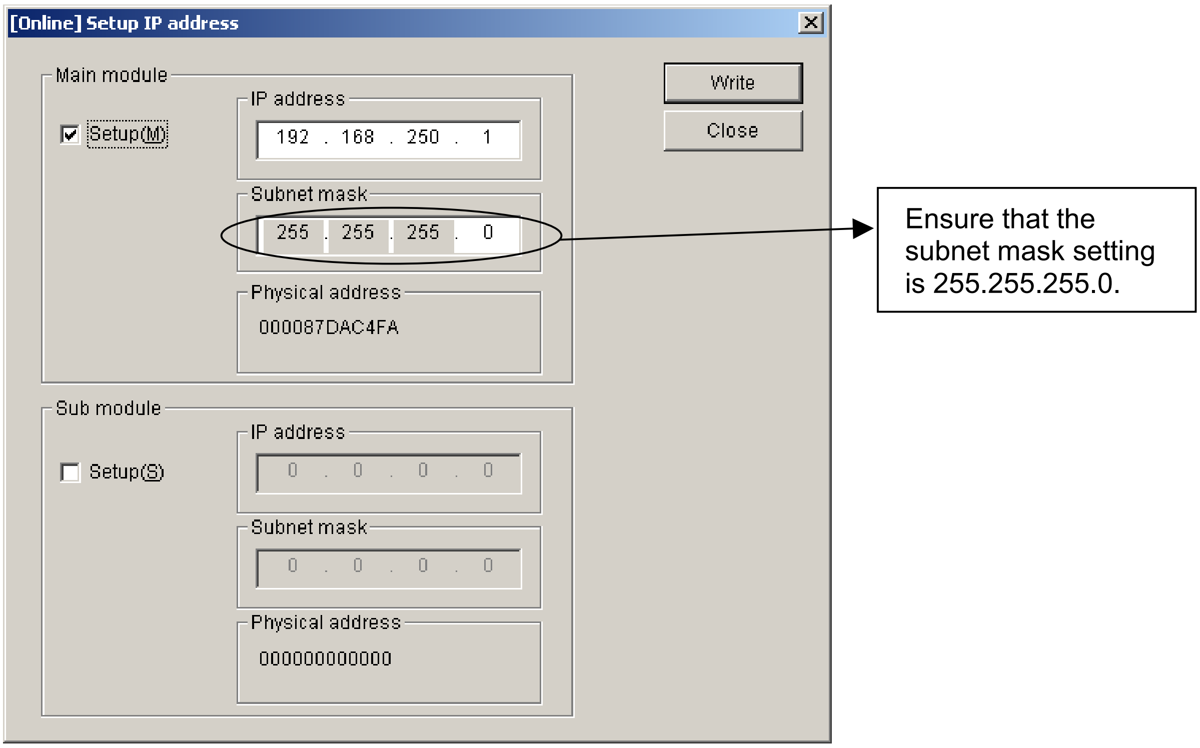
④ Connect the LPU unit to the Windows® personal computer via an RS-232C interface cable or Ethernet. Start the setup tool named “S10V FL.NET SYSTEM” (See “5.4 Installing and Starting up the System.”).

⑤ Set the IP address and subnet mask for the FL.NET module. (For IP address and subnet mask setup procedures, see item (1), “IP address/subnet setup precautions,” on the next page.)

⑥ Set up the FL.NET module link parameter (See “5.3.1 Link parameter setup procedure.”).

(1) IP address/subnet setup precautions

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 Class C for IP address setup and set the network address to 192.168.250. Also, be sure that the subnet mask setting is 255.255.255.0.



[Online] Setup IP address

Main module

☑ Setup(M)

IP address: 192 . 168 . 250 . 1

Subnet mask: 255 . 255 . 255 . 0

Physical address: 000087DAC4FA

Sub module

□ Setup(S)

IP address: 0 . 0 . 0 . 0

Subnet mask: 0 . 0 . 0 . 0

Physical address: 000000000000

Ensure that the subnet mask setting is 255.255.255.0.

Figure 5-31 IP Address Setup Window

Note 1: If all the IP address entries are set to 0 or 255, an input error occurs.

Note 2: If all the host number entries are set to /0 or /F, the FL.NET module performs the same process as in cases where IP address setup is not completed.

Note 3: If the FL.NET module is not installed, the IP address setup window of the setup tool named [S10V FL.NET SYSEM] indicates the physical address /FFFFFFFFFFFF. If the FL.NET module is installed but IP address setup is not completed, the window shows the physical address /000000000000.

Note 4: When referencing the physical address, install 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 installed.

Note 5: If the IP address is not set, with the LER LED on the FL.NET module blinking.

5.2.2 Module installation and switch setup

For the procedures for module installation and switch setup, see “3 FL.NET MODULE INSTALLATION.”

5.2.3 Tool connection procedure

For the tool connection procedure, See “5.4 Installing and Starting Up the System.”

5.2.4 Tool startup procedure

For the tool startup procedure, See “5.4 Installing and Starting Up the System.”

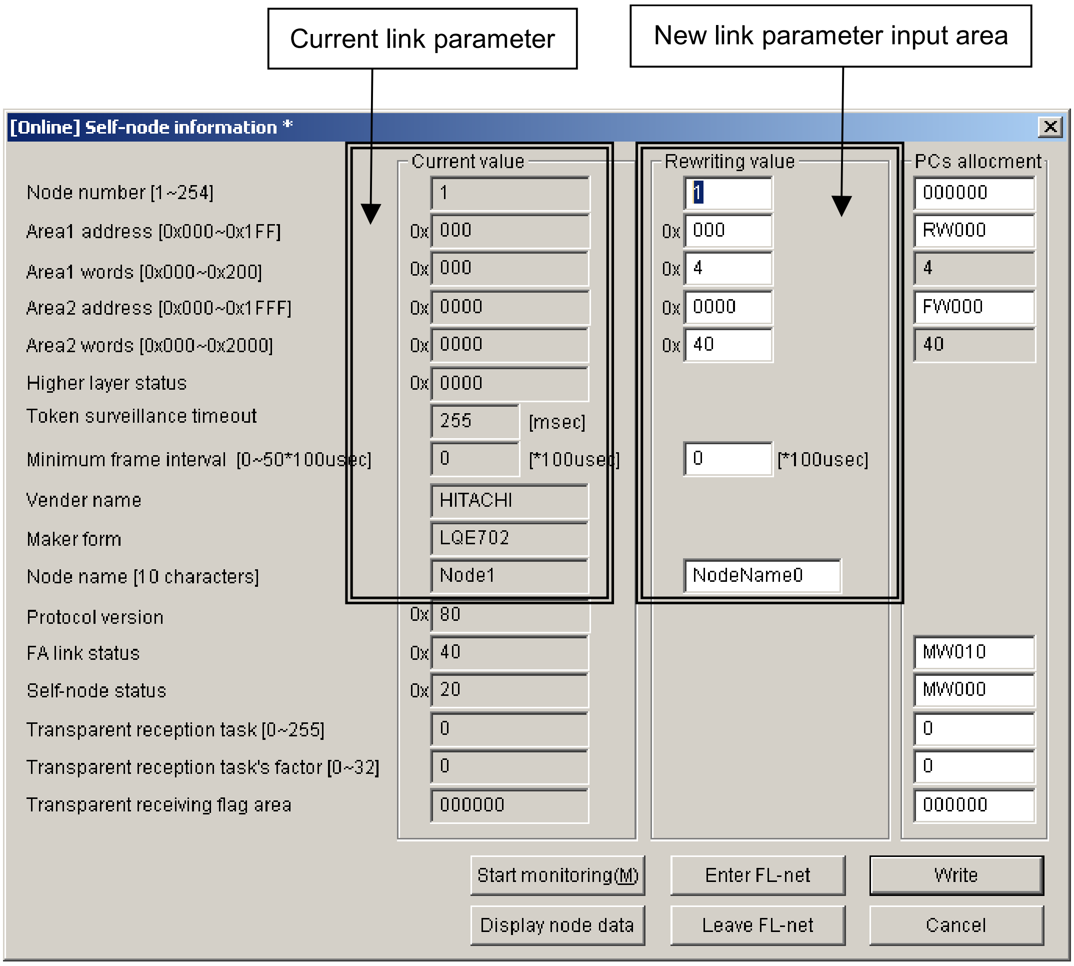
5.3 Using the FL.NET Module

5.3.1 Link parameter setup procedure

The link parameters are used to set up the FL.NET module common memory area for each node. Use the setup tool named [S10V FL.NET SYSTEM] to perform link parameter setup. The self-node setup screen is shown below.

The "Current value" section shows the current settings for the FL.NET module.

To change the settings, enter desired values in the "Rewriting value" section and click the Write button. The newly entered link parameter settings are then saved in the FL.NET module.



Current link parameter

New link parameter input area

	Current value	Rewriting value	PCs allocment
Node number [1~254]	1	1	000000
Area1 address [0x000~0x1FF]	0x 000	0x 000	RW000
Area1 words [0x000~0x200]	0x 000	0x 4	4
Area2 address [0x000~0x1FFF]	0x 0000	0x 0000	FW000
Area2 words [0x000~0x2000]	0x 0000	0x 40	40
Higher layer status	0x 0000		
Token surveillance timeout	255 [msec]		
Minimum frame interval [0~50*100usec]	0 [*100usec]	0 [*100usec]	
Vender name	HITACHI		
Maker form	LQE702		
Node name [10 characters]	Node1	NodeName0	
Protocol version	0x 80		
FA link status	0x 40		MW010
Self-node status	0x 20		MW000
Transparent reception task [0~255]	0		0
Transparent reception task's factor [0~32]	0		0
Transparent receiving flag area	000000		000000

Figure 5-32 Link Parameter Setup Window

The table below shows the acceptable input value range for each setting item.

Input item	Description	Acceptable setting range
Node number	This number is used to identify the nodes for FL-net communication purposes.	1 to 254
Token surveillance timeout	This value represents the period of time (ms) during which the token addressed to the self-node can be held.	Fixed at 255 (cannot be edited by the user)
Minimum frame interval	This value (variable in 100 μs units) 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 words	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 words	This value specifies the size of the area 2 transmission area. Enter a hexadecimal number.	0 to 0x2000 (8192)
Node name	This entry represents the self-node name.	String of no more than 10 single-byte alphanumeric characters

For detailed procedure descriptions, See “5.5 Operating Method.”

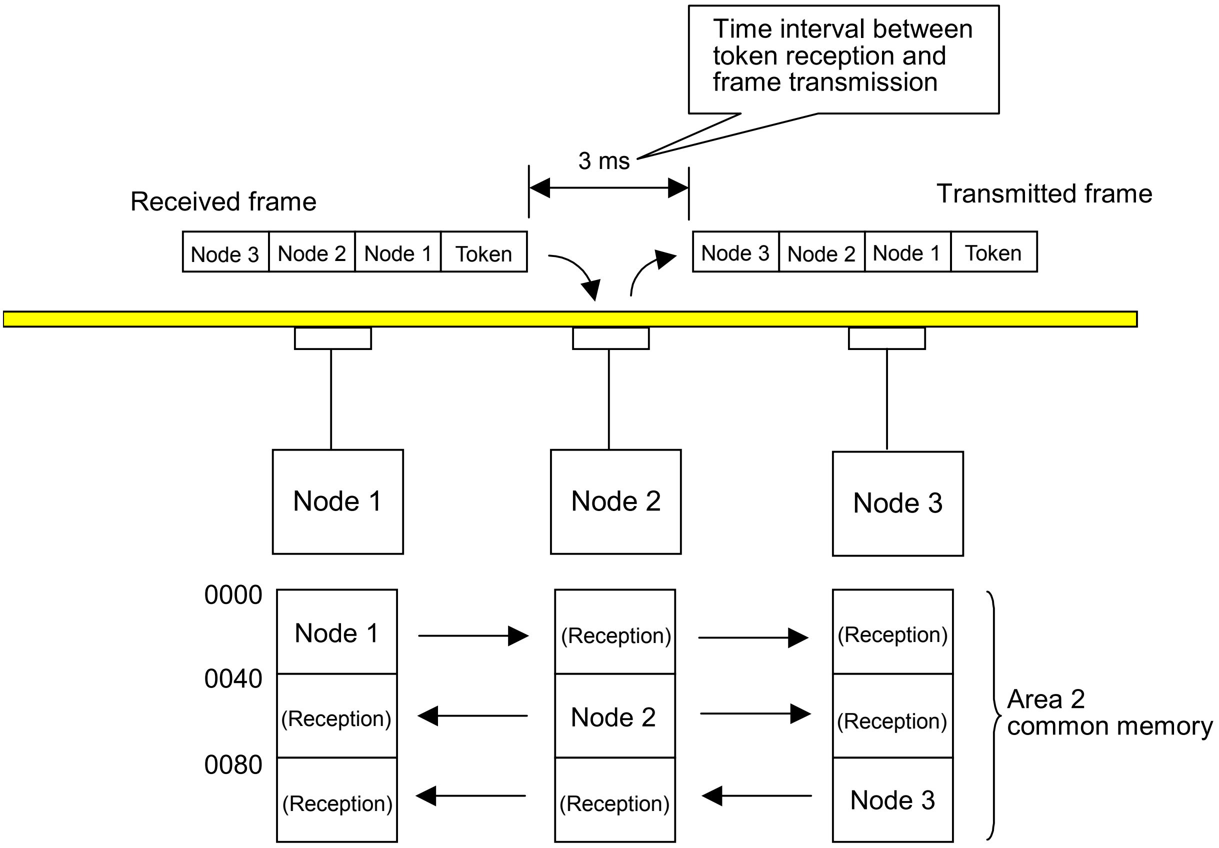
Note 1: If the minimum frame interval is set to “0”, it will be automatically changed to its minimum value used with each of the FL.NET module and other FL-net compliant modules connected to the FL-net network.

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

Example: When the following link parameter settings are employed, the FL-net module operates as indicated below:

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

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



Data is exchanged using the area 2 common memory area as shown above.

5.3.2 LPU memory allocation procedure

The FL.NET module allocates the common memory area in the LPU memory. The data received by the FL.NET module is transferred to the resulting LPU memory.

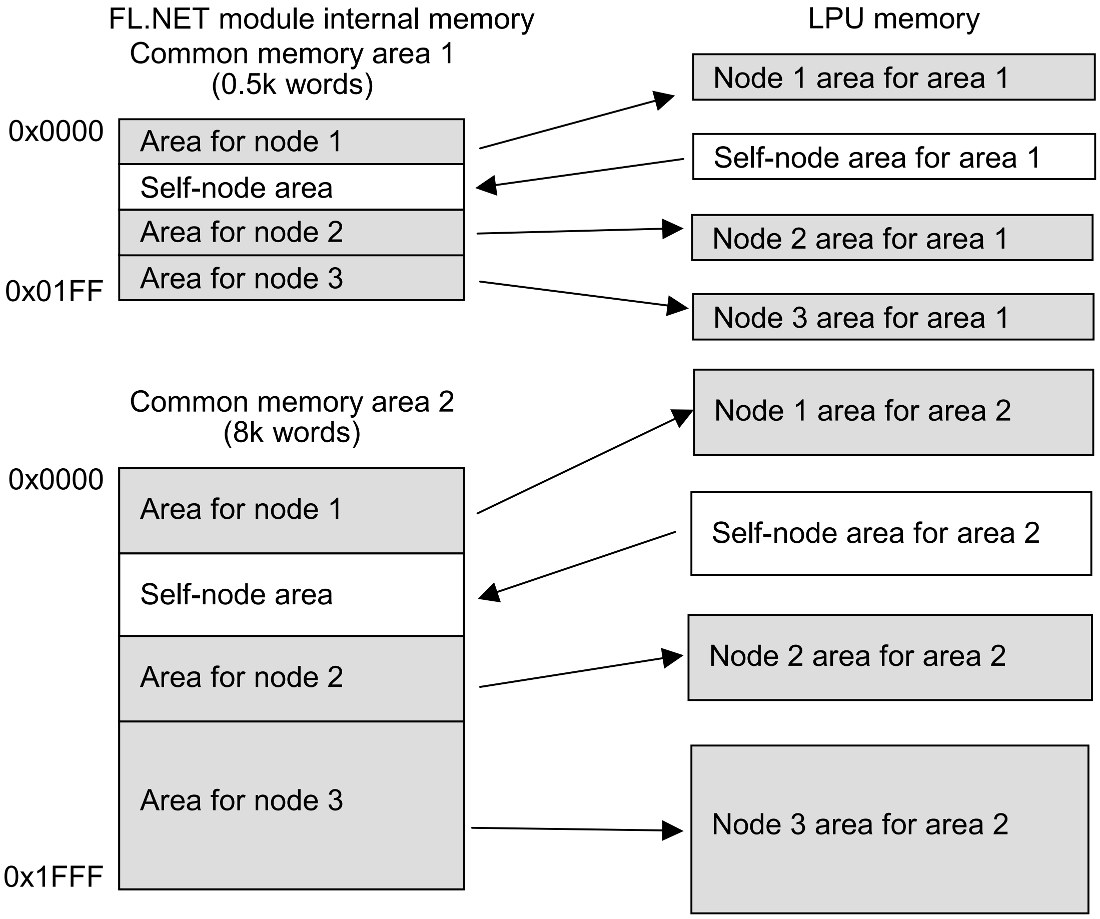


Figure 5-33 Common Memory Data Transfer Image of FL.NET Module

Setup for common memory area allocation in the LPU memory is to be performed with the setup tool named [S10V FL.NET SYSTEM].
For the details of the procedures to be performed from various setup windows, See “5.5 Operating Method.”

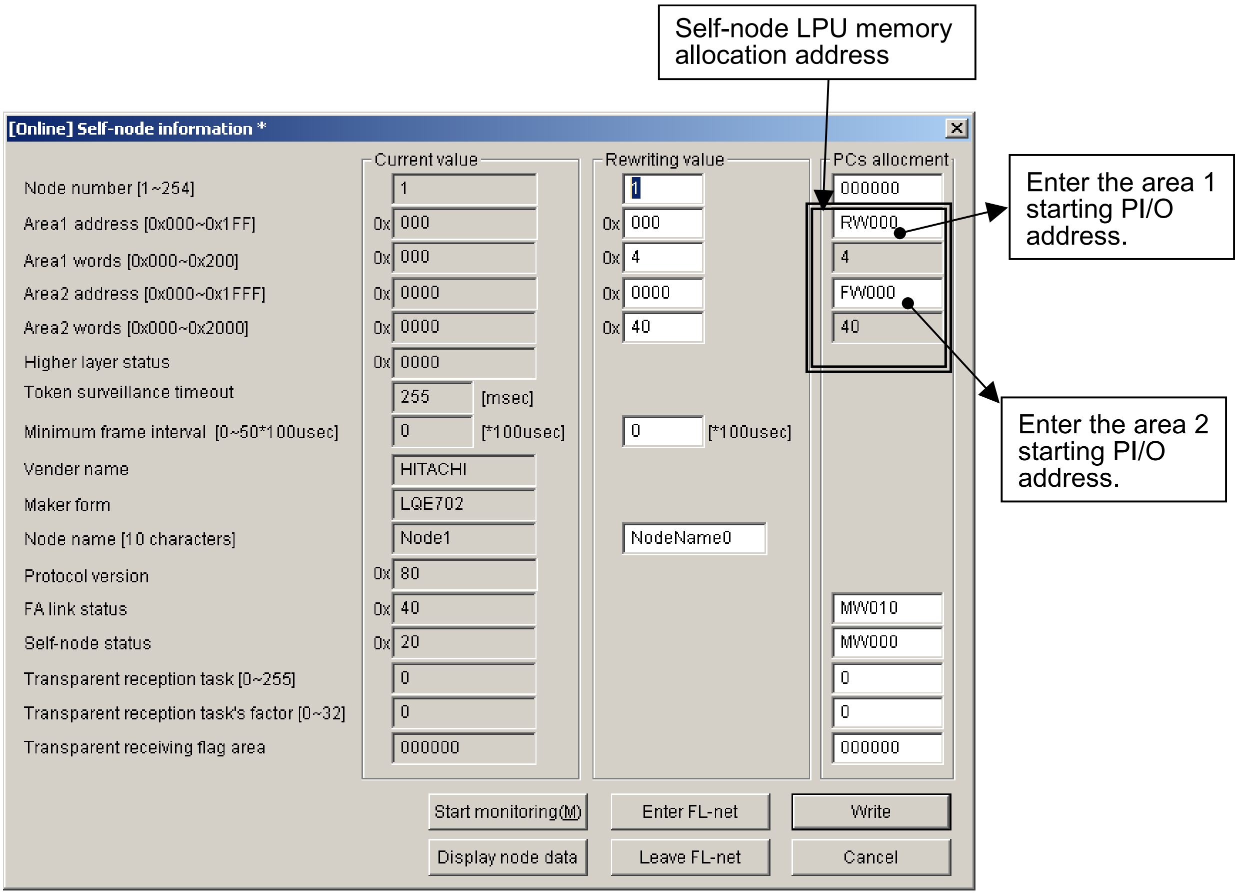
Table 5-6 shows the addresses that are available for common memory area allocation in the LPU memory.

Table 5-6 Address Ranges for Common Memory Area Allocation in LPU Memory

Selectable register/address	Remarks
XW000 to XWFF0	External input
YW000 to YWFF0	External output
JW000 to JWFF0	Transfer register
QW000 to QWFF0	Receive register
GW000 to GWFF0	Global link register
RW000 to RWFF0	Internal register
EW400 to EWFF0	Event register
MW000 to MWFF0	Internal register
DW000 to DWFFF	Function data register
FW000 to FWBFF	Function work register
LBW0000 to LBWFFF0	Work register
LWW0000 to LWWFFFF	Word work register
LXW0000 to LXW3FFF	Word work register

(1) Allocating the self-node LPU memory

With the setup tool named [S10V FL.NET SYSTEM], open the [self-node information] window. In the "PCs allocment" section, enter the addresses for the PI/O to be allocated as the self-node LPU memory area. After completion of input, click the Write button. The entered settings are then saved in the FL.NET module.



	Current value	Rewriting value	PCs allocment
Node number [1~254]	1	1	000000
Area1 address [0x000~0x1FF]	0x 000	0x 000	RW000
Area1 words [0x000~0x200]	0x 000	0x 4	4
Area2 address [0x000~0x1FFF]	0x 0000	0x 0000	FW000
Area2 words [0x000~0x2000]	0x 0000	0x 40	40
Higher layer status	0x 0000		
Token surveillance timeout	255 [msec]		
Minimum frame interval [0~50*100usec]	0 [*100usec]	0 [*100usec]	
Vender name	HITACHI		
Maker form	LQE702		
Node name [10 characters]	Node1	NodeName0	
Protocol version	0x 80		
FA link status	0x 40		MW010
Self-node status	0x 20		MW000
Transparent reception task [0~255]	0		0
Transparent reception task's factor [0~32]	0		0
Transparent receiving flag area	000000		000000

Self-node LPU memory allocation address

Enter the area 1 starting PI/O address.

Enter the area 2 starting PI/O address.

Figure 5-34 Self-node LPU Memory Allocation Window

Input item	Description	Acceptable selection range
Area 1 address (PCs allocation)	Set the starting address for the allocation of area 1 common memory and LPU memory.	See Table 5-6, "Address Ranges for Common Memory Area Allocation in LPU Memory. "
Area 2 address (PCs allocation)	Set the starting address for the allocation of area 2 common memory and LPU memory.	See Table 5-6, "Address Ranges for Common Memory Area Allocation in LPU Memory. "

Note 1: The area 1 word count (area 2 word count) for PCs allocation is the same as the link parameter area 1 word count (area 2 word count).

Note 2: The allocation address setting for the self-node LPU memory must not be a duplicate of any one of those for the other nodes.

Note 3: LPU memory allocation cannot be performed over more than one PI/O range.

Example: An unacceptable setup example is given below:

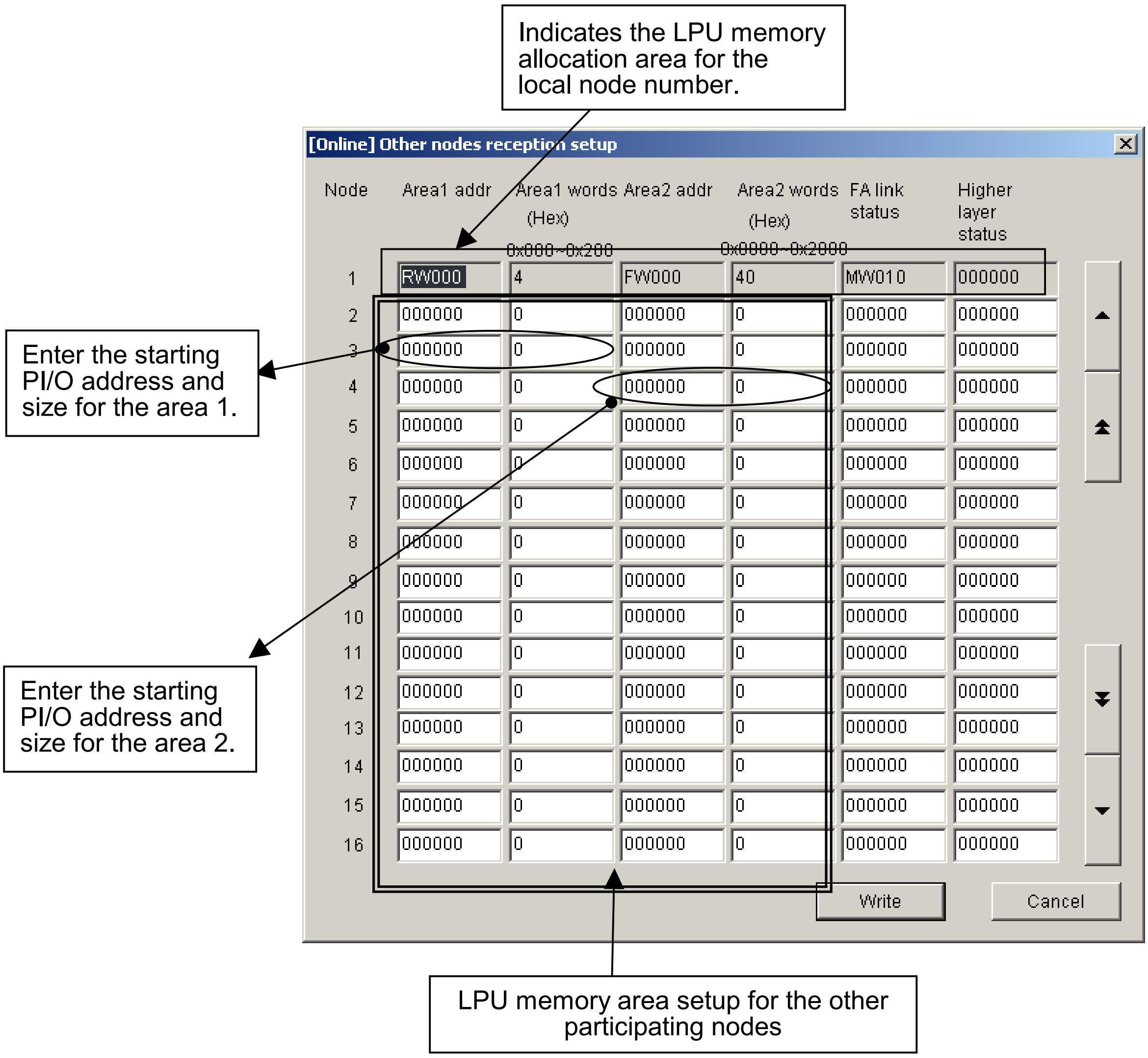
When the word count (size) setting for area 1 is 3 words, “RWFF0” cannot be specified for LPU memory allocation (because the RW area limit would be exceeded).

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 PCs allocation cannot be set.

(2) Setting the LPU memory area for the other nodes

Perform reception setup for the other nodes with the setup tool named [S10V FL.NET SYSTEM].

The data received from various nodes are transferred to the area allocated in the LPU memory. If the LPU memory 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 from the number of words specified by the LPU memory area word count setting. To acquire the entire data received by the common memory, ensure that the size setting made is equal to the common memory setting (link parameter setting) for the other nodes.



Indicates the LPU memory allocation area for the local node number.

Enter the starting PI/O address and size for the area 1.

Enter the starting PI/O address and size for the area 2.

LPU memory area setup for the other participating nodes

Node	Area1 addr	Area1 words (Hex) 0x000~0x200	Area2 addr	Area2 words (Hex) 0x0000~0x2000	FA link status	Higher layer status
1	RW000	4	FW000	40	MW010	000000
2	000000	0	000000	0	000000	000000
3	000000	0	000000	0	000000	000000
4	000000	0	000000	0	000000	000000
5	000000	0	000000	0	000000	000000
6	000000	0	000000	0	000000	000000
7	000000	0	000000	0	000000	000000
8	000000	0	000000	0	000000	000000
9	000000	0	000000	0	000000	000000
10	000000	0	000000	0	000000	000000
11	000000	0	000000	0	000000	000000
12	000000	0	000000	0	000000	000000
13	000000	0	000000	0	000000	000000
14	000000	0	000000	0	000000	000000
15	000000	0	000000	0	000000	000000
16	000000	0	000000	0	000000	000000

Figure 5-35 Common Memory Area Setup Window for Other Participating Nodes

The table below shows the acceptable input ranges for various input items.

Input item	Description	Acceptable selection range
Area 1 address	Specify the starting address of the area 1 reception area for the specified node (having the node number indicated at the left end).	See Table 5-6, “Address Ranges for Common Memory Area Allocation in LPU Memory.”
Area 1 word count	Specify the size of the area 1 reception area for the specified node (having the node number indicated at the left end). Enter a hexadecimal number.	0 to 0x200 (512)
Area 2 address	Specify the starting address of the area 2 reception area for the specified node (having the node number indicated at the left end).	See Table 5-6, “Address Ranges for Common Memory Area Allocation in LPU Memory.”
Area 2 word count	Specify the size of the area 2 reception area for the specified node (having the node number indicated at the left end). Enter a hexadecimal number.	0 to 0x2000 (8192)

Note 1: The area selected for reception setup for an other (remote) node must not be a duplicate of the area allocated in the self-node LPU memory area.

Note 2: LPU memory N allocation cannot be performed over more than one PI/O range.

Example: An unacceptable setup example is given below:

When the word count (size) setting for area 1 is 3, “RWFF0” cannot be specified as the area 1 address (because the RW area limit would be exceeded).

NOTICE

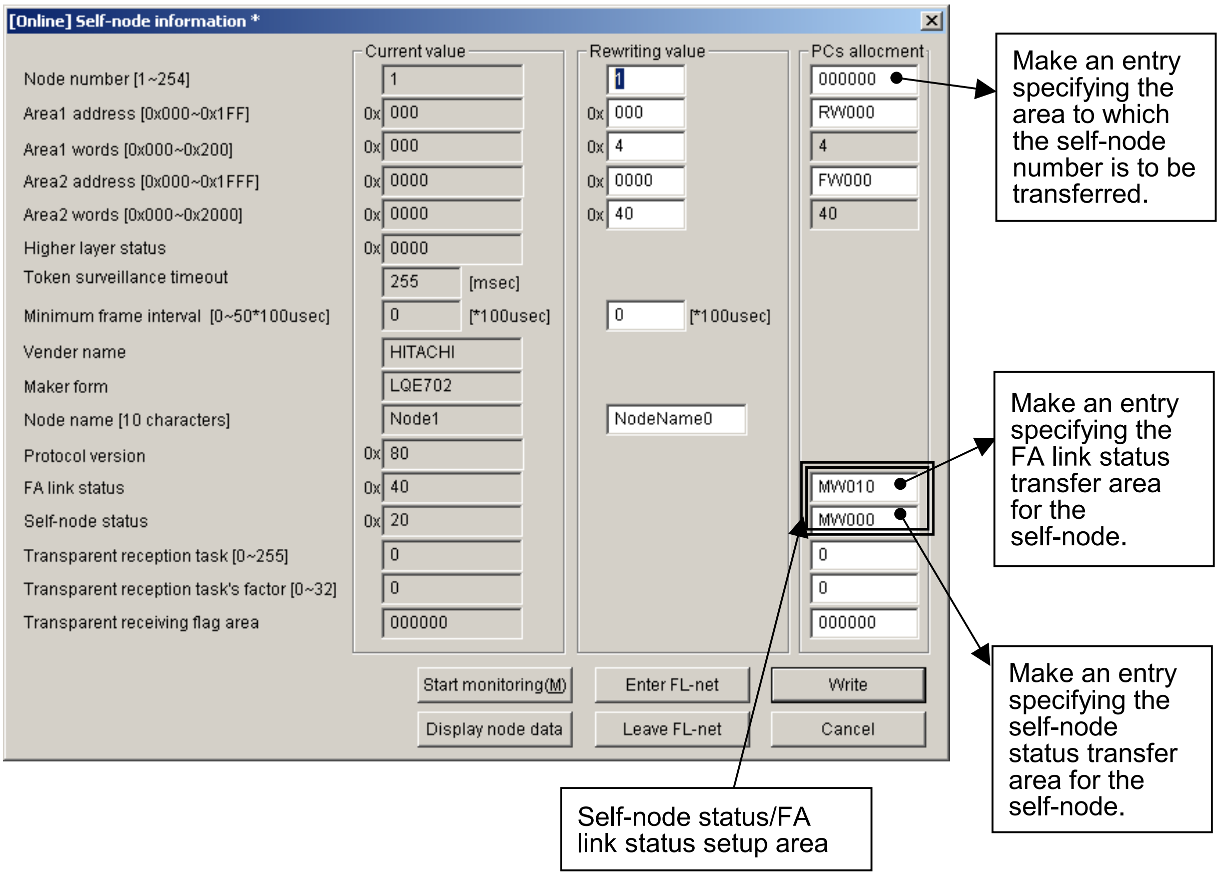
- If the LPU memory area word count differs from the common memory size setting (link parameter area word count), the FL.NET module transfers data from the common memory. The amount of this data transfer is determined from the number of words specified by the LPU memory area word count setting.
- If any node is left without being set, the FL.NET module cannot transfer to the S10V the data received from that node into the common memory.

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

The self-node status/FA link status of the FL.NET module, the FA link status of each node, and a higher layer status flag can be referenced with the setup tool named [S10V FL.NET SYSTEM].

When various items of status information on the S10V need to be referenced, the status of participating nodes and the like can be grasped by setting the transfer area for each flag.

To set up the self-node status/FA link status transfer area for the self-node, use the [Self-node information] window.



[Online] Self-node information *

	Current value	Rewriting value	PCs allocment
Node number [1~254]	1	1	000000
Area1 address [0x000~0x1FF]	0x 000	0x 000	RW000
Area1 words [0x000~0x200]	0x 000	0x 4	4
Area2 address [0x000~0x1FFF]	0x 0000	0x 0000	FW000
Area2 words [0x000~0x2000]	0x 0000	0x 40	40
Higher layer status	0x 0000		
Token surveillance timeout	255 [msec]		
Minimum frame interval [0~50*100usec]	0 [*100usec]	0 [*100usec]	
Vender name	HITACHI		
Maker form	LQE702		
Node name [10 characters]	Node1	NodeName0	
Protocol version	0x 80		
FA link status	0x 40		MW010
Self-node status	0x 20		MW000
Transparent reception task [0~255]	0		0
Transparent reception task's factor [0~32]	0		0
Transparent receiving flag area	000000		000000

Make an entry specifying the area to which the self-node number is to be transferred.

Make an entry specifying the FA link status transfer area for the self-node.

Make an entry specifying the self-node status transfer area for the self-node.

Self-node status/FA link status setup area

Figure 5-36 Setup Window for Setting Various Status Flag Areas for the Self-node

For the on-screen information (bit allocation) concerning the self-node status and FA link status, see "5.3.6 Using the management tables."

The table below shows the acceptable input ranges for various input items.

Input item	Description	Acceptable input range
Node number (PCs allocation)	Specify the area to which the self-node number is to be transferred (the area size is 1 word).	See Table 5-6, "Address Ranges for Common Memory Area Allocation in LPU Memory."
FA link status (PCs allocation)	Specify the area to which the FA link status of the self-node is to be transferred (the area size is 1 word).	See Table 5-6, "Address Ranges for Common Memory Area Allocation in LPU Memory."
Self-node status (PCs allocation)	Specify the area to which the self-node status is to be transferred (the area size is 1 word).	See Table 5-6, "Address Ranges for Common Memory Area Allocation in LPU Memory."

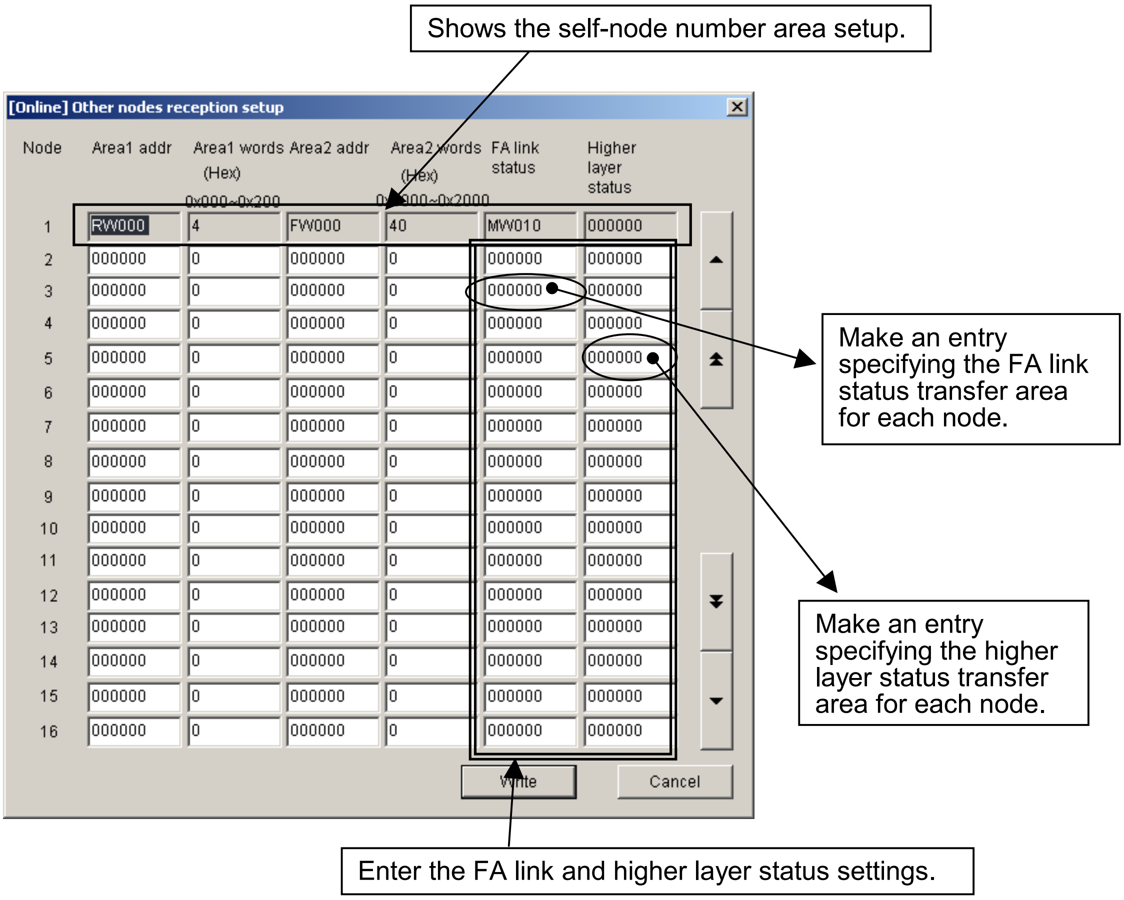
Note 1: The register/extension memory address range applicable to the self-node status area and FA link status area is the same as for the LPU memory allocation area.

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

NOTICE

- When the self-node status area is set up, the self-node status flag is transferred as 1-word data.
- When the FA link status area is set up, the FA link status flag for the self-node is transferred to the low-order byte of the specified area.
- If the area for self-node status or FA link status information is not specified, the FL.NET module cannot transfer the status information to the S10V.

To set up the FA link status/higher layer flag area, use the setup window for the other nodes reception setup.



Shows the self-node number area setup.

[Online] Other nodes reception setup

Node	Area1 addr	Area1 words (Hex) 0x000~0x200	Area2 addr	Area2 words (Hex) 0x000~0x2000	FA link status	Higher layer status
1	RW000	4	FW000	40	MW010	000000
2	000000	0	000000	0	000000	000000
3	000000	0	000000	0	000000	000000
4	000000	0	000000	0	000000	000000
5	000000	0	000000	0	000000	000000
6	000000	0	000000	0	000000	000000
7	000000	0	000000	0	000000	000000
8	000000	0	000000	0	000000	000000
9	000000	0	000000	0	000000	000000
10	000000	0	000000	0	000000	000000
11	000000	0	000000	0	000000	000000
12	000000	0	000000	0	000000	000000
13	000000	0	000000	0	000000	000000
14	000000	0	000000	0	000000	000000
15	000000	0	000000	0	000000	000000
16	000000	0	000000	0	000000	000000

Make an entry specifying the FA link status transfer area for each node.

Make an entry specifying the higher layer status transfer area for each node.

Enter the FA link and higher layer status settings.

Figure 5-37 Setup Window for Setting the Status Flag Areas for the Other Nodes Reception Setup

The table below shows the acceptable input ranges for the input items.

Input item	Description	Acceptable input range
FA link status	Specify the area to which the FA link status of the node having the node number indicated at the left end is to be transferred (the area size is 1 word).	See Table 5-6, "Address Ranges for Common Memory Area Allocation in LPU Memory."
Higher layer status	Specify the area to which the higher layer status of the node having the node number indicated at the left end is to be transferred (the area size is 1 word).	See Table 5-6, "Address Ranges for Common Memory Area Allocation in LPU Memory."

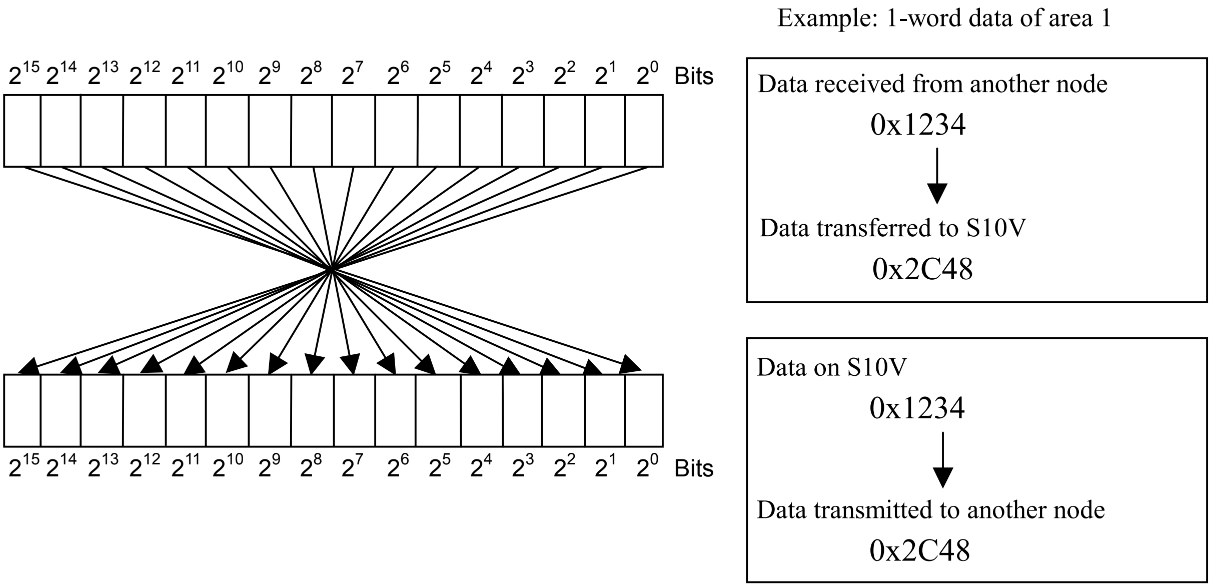
For the on-screen information (bit allocation) concerning the FA link status and higher layer status, see "5.3.6 Using the management tables."

NOTICE

- The register/extension memory address range applicable to the higher layer status area and FA link status area is the same as for the LPU memory allocation area.
- When the higher layer status area is set up, the contents of the higher layer status flag are transferred as 1-word data.
- When the FA link status area is set up, the contents of the FA link status flag for the node are transferred to the low-order byte of the specified area.
- The higher layer status/FA link status area setting must not be a duplicate of the LPU memory allocation area setting for a remote node (reception setting) or the self-node or of the setting for any other area.
- If any node is omitted from the node settings given, the FL.NET module cannot transfer the higher layer status or FA link status of that node to the S10V.

5.3.3 Using bit data

The FL.NET module subjects the common memory area 1 data to bit conversion at the time of transmission/reception.


2^{15} 2^{14} 2^{13} 2^{12} 2^{11} 2^{10} 2^{9} 2^{8} 2^{7} 2^{6} 2^{5} 2^{4} 2^{3} 2^{2} 2^{1} 2^{0} Bits
2^{15} 2^{14} 2^{13} 2^{12} 2^{11} 2^{10} 2^{9} 2^{8} 2^{7} 2^{6} 2^{5} 2^{4} 2^{3} 2^{2} 2^{1} 2^{0} Bits

Example: 1-word data of area 1

Data received from another node
0x1234
Data transferred to S10V
0x2C48

Data on S10V
0x1234
Data transmitted to another node
0x2C48

Figure 5-38 Bit Conversion of Area 1 Data

5.3.4 Using word data

The FL.NET module does not perform bit conversion when transmitting or receiving the data of the common memory area 2. It transfers the data to the network without changing its arrangement on the S10V, so the data received from the other nodes is transferred as it is.

5.3.5 Using message communications

(1) Message communication (server side)

When the FL.NET module receives a message request from another node, it processes the message within itself. The user does not have to pay attention to it.

However, when the FL.NET module receives a transparent message, it notifies the user of the transparent message reception via the self-node status flag and transparent reception flag.

For details, see item (4), "Transparent message reception."

When a transparent message is received, use a C mode program or mathematical/logical function to fetch it from the FL.NET module.

If the FL.NET module is filled with transparent messages, it may not be able to process any more request messages.

(2) Message communication (client side)

When requesting a message for another node (or receiving a transparent message), it is necessary to issue a request to the FL.NET module from a C mode program or mathematical/logical function.

The FL.NET module offers a C mode handler and mathematical/logical function as a means of issuing a message request.

The procedures for using a C mode program or ladder program to issue a message request are explained below:

Note: For message communication, an environment for permitting a user program to reference the self-node status flag is required.

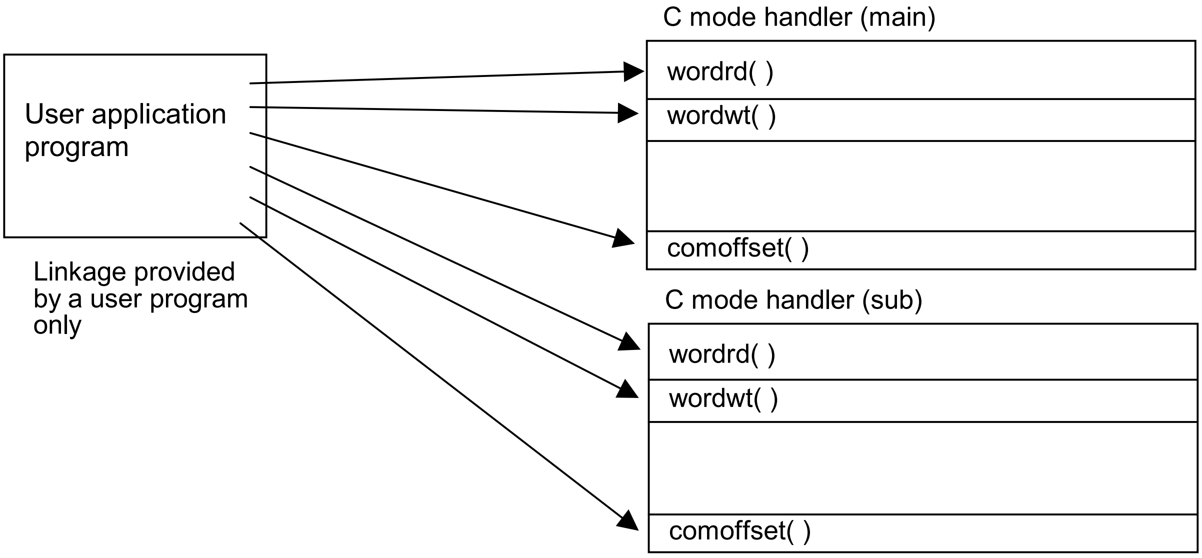
Set up a self-node status flag area from the self-node setup window of the setup tool named [S10V FL.NET SYSTEM]. For detailed procedure descriptions, See "5.5 Operating Method."

(3) C mode handler or mathematical/logical function

(a) Message transmission request by C mode handler

A C mode handler is called as a C function. It issues a message request to the FL.NET module to exchange data in place of a user program. Various C mode handlers are available for use with all message types.

A C mode handler must be called by specifying its entry address. A user program cannot be created (linked) in such a manner that a C mode handler is contained in it.



It should also be pointed out in this connection that the C mode handler for the FL.NET module merely issues a request to the FL.NET module and then terminates its whole process immediately.

When the return code fed from the C mode handler is -1 (0xFFFFFFFF), it means that a parameter abnormality exists or that the handler is busy processing another message. Check the data at an error code storage address specified at C mode handler startup.

When the return code is 0, the C mode handler indicates that a request has been processed normally.

When return code is -2 (0xFFFFFFFE), it means that an address on CMU was specified at parameter which specifies an address, such as an error code storage address.

After receipt of a request, the FL.NET module turns ON (sets) the “user request processing in progress” bit in the self-node status flag, and turns it OFF (resets it) at the end of the processing.

To verify the end of message processing after issuance, check the self-node status flag. The status indicates the end of the processing.

After a request is issued by the C mode handler, do not make another message request until the "user request processing in progress" bit is reset. If any message request is issued before the bit is reset, it is not processed because another message is being processed (the "message processing already in progress" error code is set at an error code storage address).
After the "user request processing in progress" bit is reset, enter the status prevailing after the end of the processing at an error code storage address specified at C mode handler startup.
To check whether the requested processing is ended normally, note the code that is set at the error code storage address.

Table 5-7 C Mode Handler List

Name	Subroutine call address		Functionality description
	Main	Sub	
wordrd( )	/D60020	/DE0020	Issues a word block read request.
wordwt( )	/D60040	/DE0040	Issues a word block write request.
parard( )	/D60060	/DE0060	Issues a network parameter read request.
parawt( )	/D60080	/DE0080	Issues a network parameter write request.
reqstop( )	/D600A0	/DE00A0	Issues a stop request.
reqrun( )	/D600C0	/DE00C0	Issues a run request.
logrd( )	/D600E0	/DE00E0	Issues a communication log read request.
logclr( )	/D60100	/DE0100	Issues a communication log clear request.
mesret( )	/D60120	/DE0120	Issues a message return request.
reqmacro( )	/D60140	/DE0140	Issues a specified-task control request (Hitachi's unique transparent type of support).
toukaread( )	/D60160	/DE0160	Issues a transparent message reception request (Hitachi's unique transparent type of support).
toukasend( )	/D60180	/DE0180	Issues a transparent message transmission request (Hitachi's unique transparent type of support).
comoffset( )	/D601A0	/DE01A0	Issues a common memory offset feature request.

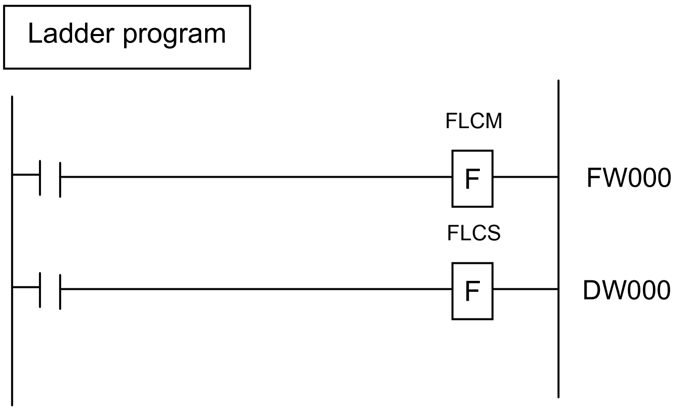
Note: For bit allocation for the self-node status flag, see "5.3.6 Using the management tables."

NOTICE

Do not mix to use C mode handler and mathematical/logical function.
Please be sure to unify either C mode handler or mathematical/logical function at FL.NET module unit.

(b) Message transmission request by mathematical/logical function

A mathematical/logical function is called from a ladder program to issue a message request to the FL.NET module for data exchange purposes. The mathematical/logical function for the main module/submodule is available. The parameter specified at mathematical/logical function startup dictates the message processing to be requested. For the procedure for registering the FL.NET module mathematical/logical function, See "5.5 Operating Method."



As is the case with the C mode handler, the mathematical/logical function merely issues a request to the FL.NET module and then terminates the processing.

After receipt of a request, the FL.NET module sets the "user request processing in progress" bit in the self-node status flag, and resets it at the end of the processing.

To verify the end of the message processing after issuance, check the self-node status flag. The status indicates the end of the processing.

After a request is issued by the mathematical/logical function, do not make another message request until the "user request processing in progress" bit is resets. If any message request is issued before the bit is reset, it is not processed because another message is being processed (the "message processing already in progress" error code is set at an error code storage address).

After the "user request processing in progress" bit is reset, enter the status prevailing after the end of the processing at an error code storage address specified at mathematical/logical function startup.

To check whether the requested processing is ended normally, note the code that is set at the error code storage address.

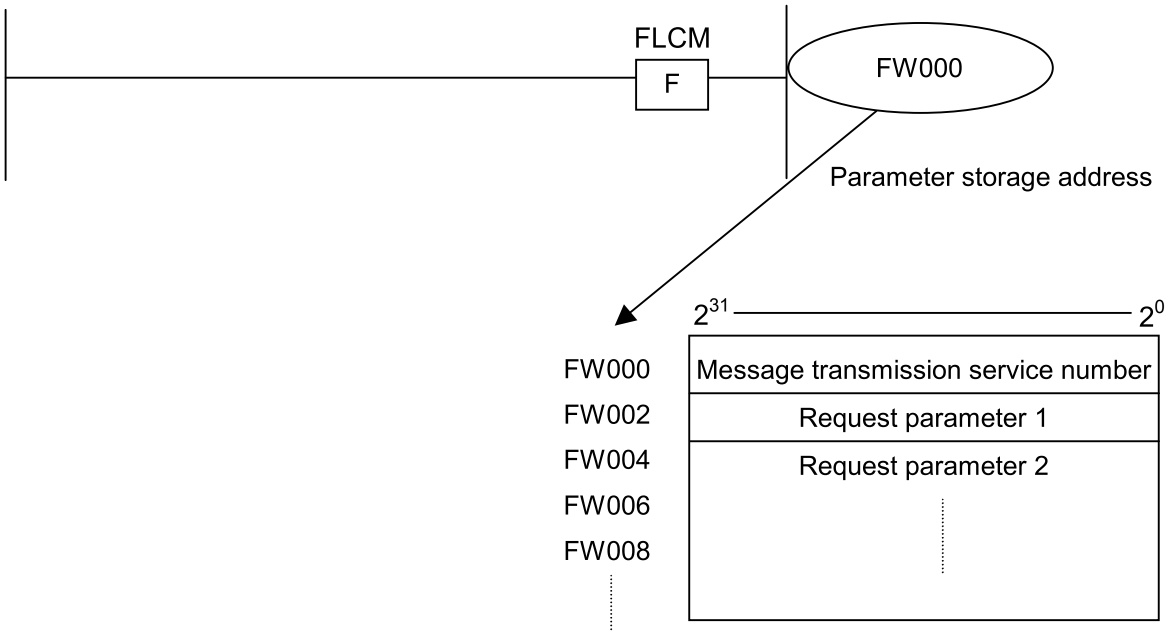
Table 5-8 Mathematical/Logical Function List

Name		Mathematical/logical function registration address		Functionality description
Main	Sub	Main	Sub	
FLCM	FLCS	/D60000	/DE0000	Issues various message transmission requests.

<Process request to FL.NET module by mathematical/logical function>

Create a ladder program with the mathematical/logical function for the FL.NET module. Write a request parameter at an address specified for the mathematical/logical function and then execute the mathematical/logical function.

When making a message transmission request from the mathematical/logical function, you must use a message transmission service number to specify the type of the message to request. For details on various parameters, see "(5) Parameters for various message requests."



Whenever specifying the parameters for use with a mathematical/logical function, use a long word length.

Also, selectively use an appropriate mathematical/logical function depending on the employed module setup. (When issuing a request to a main module, use the main module mathematical/logical function FLCM. To issue a request to a submodule, use the submodule mathematical/logical function FLCS.)

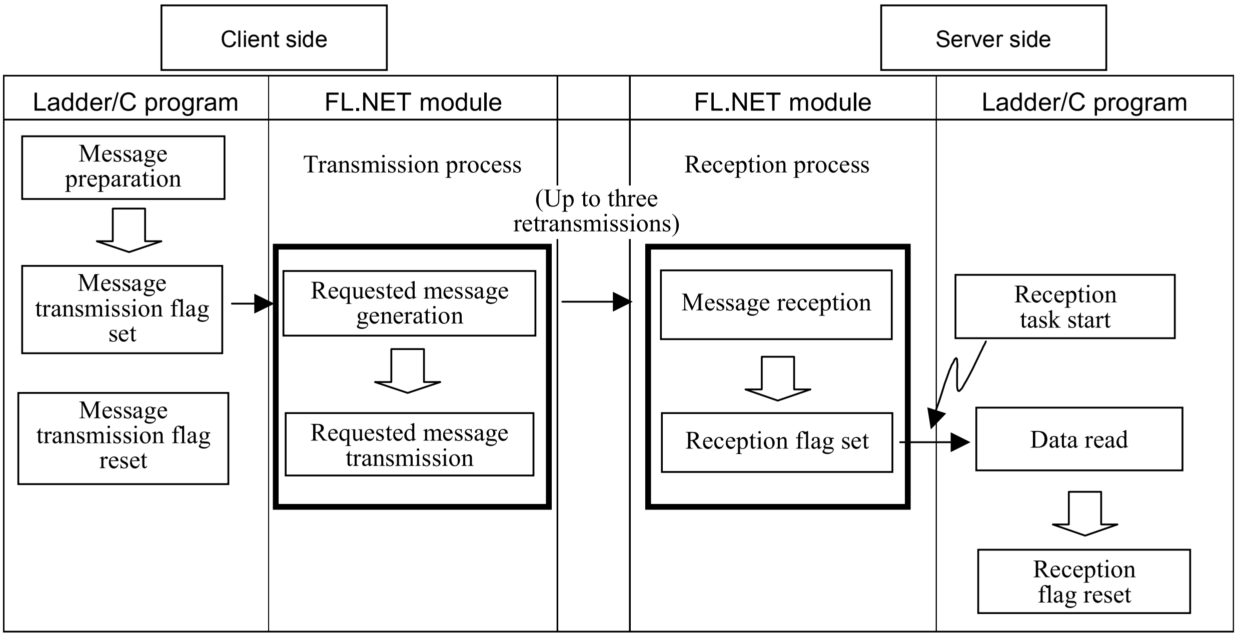
Note: For bit allocation for the self-node status flag, see "5.3.6 Using the management tables."

NOTICE

- When issuing a request to the FL.NET module, do not simultaneously use the C mode handler and mathematical/logical function.
- You cannot specify an index in a parameter (PI/O address.)

(4) Transparent message reception

The figure below outlines the operation performed for transparent messages.



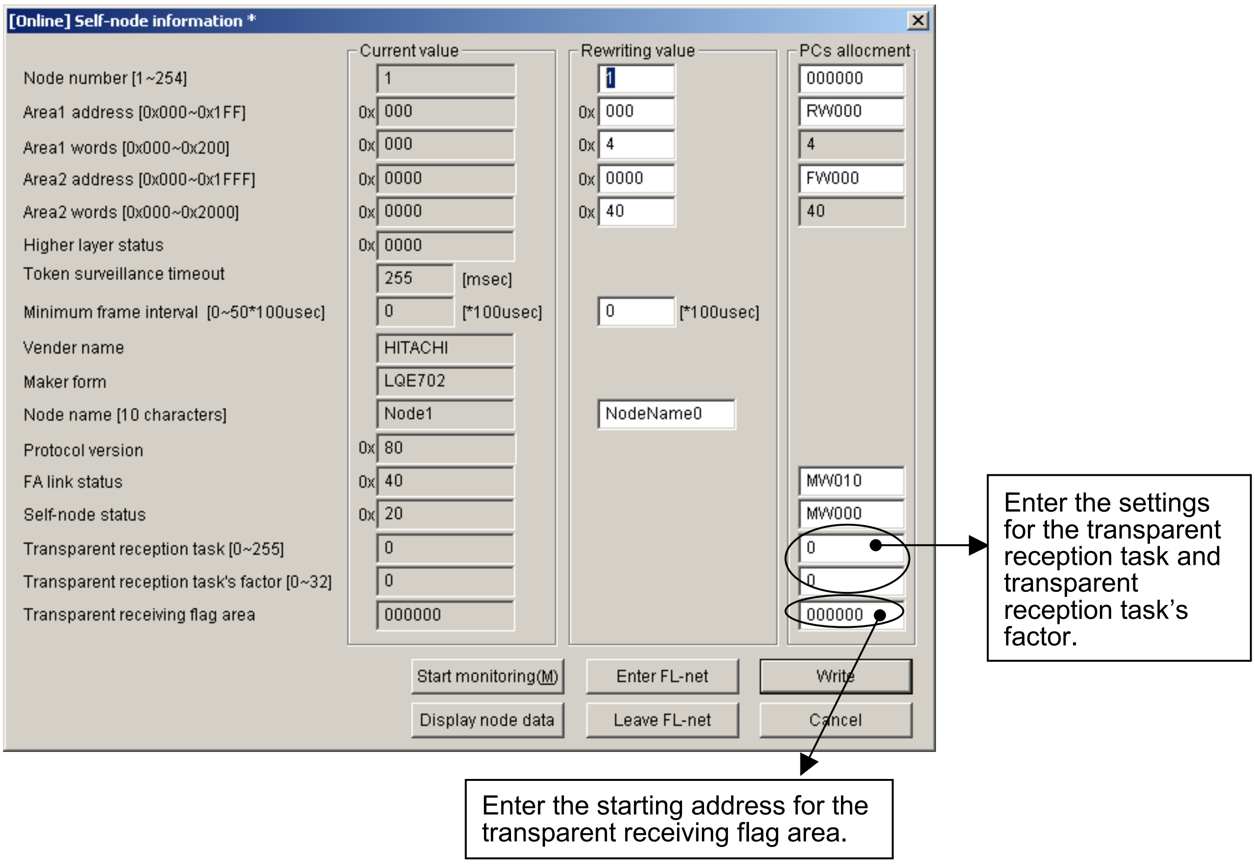
The FL.NET module has special functions concerning the reception of transparent messages.

<Transparent reception task>

When a transparent message is received in situations where a transparent reception task is registered in the FL.NET module, task queuing takes place with the start factor (transparent reception task's factor) that is set for the task number of the registered task.
If the registered task is not released, however, it does not start.

<Transparent receiving flag>

When a transparent receiving flag is set for the FL.NET module, a 16-word area beginning with the associated selected address is used as a transparent-type reception flag area.
When the FL.NET module receives a transparent message, the transparent receiving flag area indicates the node from which the message has been transmitted.
A total of eight cases of transparent message reception buffer are readied.
When a transparent message is received, it is first stored in the reception buffer. When it is stored, its transmission source node number is checked, and the bit corresponding to the transmission source node number for the transparent receiving flag area is set.
After the transparent message is fetched by the user, a check is conducted to determine whether messages having the same transmission source node number as for the message delivered to the user remain within the transparent message reception buffer.
When there are no more messages having the same transmission source node number, the bit corresponding to the transmission source node number for the transparent receiving flag area is reset.
However, when a received transparent message is for a TCD that is related to a function (preselected task control and preselected subroutine control) uniquely supported by the FL.NET module, it is not handled as a transparent message.
A transparent receiving task and transparent receiving flag can be set up with the setup tool named [S10V FL.NET SYSTEM]. For details on the procedure, See "5.5 Operating Method."



	Current value	Rewriting value	PCs allocment
Node number [1~254]	1	1	000000
Area1 address [0x000~0x1FF]	0x 000	0x 000	RW000
Area1 words [0x000~0x200]	0x 000	0x 4	4
Area2 address [0x000~0x1FFF]	0x 0000	0x 0000	FW000
Area2 words [0x000~0x2000]	0x 0000	0x 40	40
Higher layer status	0x 0000		
Token surveillance timeout	255 [msec]		
Minimum frame interval [0~50*100usec]	0 [*100usec]	0 [*100usec]	
Vender name	HITACHI		
Maker form	LQE702		
Node name [10 characters]	Node1	NodeName0	
Protocol version	0x 80		
FA link status	0x 40		MW010
Self-node status	0x 20		MW000
Transparent reception task [0~255]	0		0
Transparent reception task's factor [0~32]	0		0
Transparent receiving flag area	000000		000000

Enter the settings for the transparent reception task and transparent reception task's factor.

Enter the starting address for the transparent receiving flag area.

Figure 5-39 Transparent Receiving Flag Area and Transparent Reception Task Setup Window

The table below shows the acceptable input ranges for the setup items.

Input item	Description	Acceptable setting range
Transparent reception task	Task number of the task that is to start upon receipt of a transparent message.	The acceptable setting ranges from 1 to 255. Because 230 to 255 are used by the system, they are not started.
Transparent reception task's factor	Factor for starting a transparent reception task.	0 to 32
Transparent receiving flag area	This area reports a transmission source node number when a transparent message is received (the area size is 16 words).	See Table 5-6, "Address Ranges for Common Memory Area Allocation in LPU Memory."

Note: The symbols selectable for a transparent receiving flag area are the same as for a LPU memory area. Also, note that a transparent receiving flag area setting must not be a duplicate of any of those for the other set areas.

Table 5-9 shows the bit allocation for the transparent receiving flag area.

Table 5-9 Transparent Receiving Flag Allocation

Address	Bit number															
	2^{15}	2^{14}	2^{13}	2^{12}	2^{11}	2^{10}	2^{9}	2^{8}	2^{7}	2^{6}	2^{5}	2^{4}	2^{3}	2^{2}	2^{1}	2^{0}
0x0000		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0x0002	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0x0004	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
0x0006	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63
0x0008	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79
0x000A	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95
0x000C	96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111
0x000E	112	113	114	115	116	117	118	119	120	121	122	123	124	125	126	127
0x0010	128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143
0x0012	144	145	146	147	148	149	150	151	152	153	154	155	156	157	158	159
0x0014	160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175
0x0016	176	177	178	179	180	181	182	183	184	185	186	187	188	189	190	191
0x0018	192	193	194	195	196	197	198	199	200	201	202	203	204	205	206	207
0x001A	208	209	210	211	212	213	214	215	216	217	218	219	220	221	222	223
0x001C	224	225	226	227	228	229	230	231	232	233	234	235	236	237	238	239
0x001E	240	241	242	243	244	245	246	247	248	249	250	251	252	253	254	

The numerical values in the above table represent transmission source node numbers for transparent messages. Note that a blank bit location is not used.

(Indication example)

The transparent receiving flag area is set to MW000.

When the self-node receives a transparent message from node number 17, the MW011 (the node 17 bit) is turned ON (set).

PI/O	Bit data
MW000	0x0000
MW010	0x4000
:	:
MW0F0	0x0000

(5) Parameters for various message requests

The rest of this section describes the parameters that the C mode handler and mathematical/logical function use to issue various message requests.

Word block read request: wordrd( )

[Linking procedure]

C language	
Main	Sub
Struct wordr_p {	Struct wordr_p {
long node ;	long node ;
unsigned short *Erradr ;	unsigned short *Erradr ;
unsigned short *Setadr ;	unsigned short *Setadr ;
long Readadr ;	long Readadr ;
long Readsz ;	long Readsz ;
} ;	} ;
≀	≀
long (*wordrd)( ) ;	long (*wordrd)( ) ;
long rtn ;	long rtn ;
struct wordr_p *padr ;	struct wordr_p *padr ;
≀	≀
wordrd = (long(*)( ))0xD60020 ;/* S10V */	wordrd = (long(*)( ))0xDE0020 ;/* S10V */
≀	≀
rtn = (*wordrd)(padr) ;	rtn = (*wordrd)(padr) ;

[Parameters for mathematical/logical function]

+0x0000	Message transmission service number (3)
+0x0004	Remote node number (1 to 254)
+0x0008	Error code storage address
+0x000C	Read data storage address
+0x0010	Virtual address
+0x0014	Virtual size

[Parameters]

```
padr                                : Input parameter storage starting address
Struct wordr_p {
                long node ;         : Remote node number (1 to 254)
     unsigned short *Erradr ;       : Error code storage address
     unsigned short *Setadr ;       : Read data storage address
                long Readadr ;      : Virtual address
                long Readsz ;       : Virtual size (1 to 512 words)
} ;
```

Be sure to specify an even-numbered address in padr.

Parameter	Input range
Message transmission service number	3 (only when a mathematical/logical function is used)
Remote node number	1 to 254
Error code storage address	Specify a real address (even-numbered address) on the LPU. (*)
Read data storage address	
Virtual address	0 to 0xFFFFFFFF
Virtual size	1 to 512 (variable in 1-word units)

(*) For the relationship between real address on LPU (memory address) and symbol name such as XW000, see "Table 2-5 Virtual Address Spare and Symbol Names." "Memory address" in table is the address to be specified.

[Return code] (for C mode handler only)

0 (0x00000000): Normal termination

-1 (0xFFFFFFFF): Parameter abnormality exists or the handler is busy processing another message. Error code is stored in area which specified error code storage address. For content and measure of error code, see "Table 7-3 List of Detectable Codes."

-2 (0xFFFFFFFE): Specifies real address (0x01000000 to 0xFFFFFFFE) on CMU at error code storage address or read data storage address. Please review the specified address.

[Function]

- A word block read request using the specified virtual address and virtual size will be issued to a remote node. When a response code is received, the data is transferred to a read data storage address.
- The virtual address varies from one FL.NET module to another. For the virtual address specifications for the FL.NET module, see “Table 2-4 Virtual Address Space and Physical Memory.”

NOTICE

The server feature of the FL.NET module returns an abnormal response if, on reception of a word block read request, the virtual size is found exceeding 512 words. The error code used in such an event is 0xFFFFFFFF.

Word block write request: wordwt( )

[Linking procedure]

C language	
Main	Sub

Struct wordw_p {
long node ;
unsigned short *Erradr ;
unsigned short *Setadr ;
long writeadr ;
long writesz ;
}
≀
long (*wordwt)( ) ;
long rtn ;
struct wordw_p *padr ;
≀
wordwt = (long(*)( ))0xD60040 ;/* S10V */
≀
rtn = (*wordwt)(padr) ;
	Struct wordw_p {
	long node ;
	unsigned short *Erradr ;
	unsigned short *Setadr ;
	long writeadr ;
	long writesz ;
	}
	≀
	long (*wordwt)( ) ;
	long rtn ;
	struct wordw_p *padr ;
	≀
	wordwt = (long(*)( ))0xDE0040 ;/* S10V */
	≀
	rtn = (*wordwt)(padr) ;

[Parameters for mathematical/logical function]

+0x0000	Message transmission service number (4)
+0x0004	Remote node number (1 to 254)
+0x0008	Error code storage address
+0x000C	Write data storage address
+0x0010	Virtual address
+0x0014	Virtual size

[Parameters]

```
padr                                  : Input parameter storage starting address
Struct wordrw_p {
              long node ;             : Remote node number (1 to 254)
    unsigned short *Erradr ;          : Error code storage address
    unsigned short *Setadr ;          : Write data storage address
              long Writeadr ;         : Virtual address
              long Writesz ;          : Virtual size (1 to 512 words)
} ;
```

Be sure to specify an even-numbered address in padr.

Parameter	Input range
Message transmission service number	4 (only when a mathematical/logical function is used)
Remote node number	1 to 254
Error code storage address	Specify a real address (even-numbered address) on the LPU. (*)
Write data storage address	
Virtual address	0 to 0xFFFFFFFF
Virtual size	1 to 512 (variable in 1-word units)

(*) For the relationship between real address on LPU (memory address) and symbol name such as XW000, see “Table 2-5　Virtual Address Spare and Symbol Names.”　“Memory address” in table is the address to be specified.

[Return code] (for C mode handler only)

0 (0x00000000): Normal termination

-1 (0xFFFFFFFF): Parameter abnormality exists or the handler is busy processing another message.　Error code is stored in area which specified error code storage address.　For content and measure of error code, see “Table 7-3　List of Detectable Codes.”

-2 (0xFFFFFFFE): Specifies real address (0x01000000 to 0xFFFFFFFE) on CMU at error code storage address or write data storage address.　Please review the specified address.

[Function]

- A word block write request using the specified virtual address, virtual size, and data at the write data storage address will be issued to a remote node.
- The virtual address varies from one FL.NET module to another.　For the virtual address specifications for the FL.NET module, see “Table 2-4　Virtual Address Space and Physical Memory.”

NOTICE

The server feature of the FL.NET module returns an abnormal response if, on reception of a word block write request, the virtual size is found exceeding 512 words or conflicting with the data size.　The error code used in such an event is 0xFFFFFFFF.

Network parameter read request: parard( )

[Linking procedure]

C language	
Main	Sub
Struct parar_p { long node ; unsigned short *Erradr ; unsigned char *Setadr ; } ; ≀ long (*parard)( ) ; long rtn ; struct parar_p *padr ; ≀ parard = (long(*)( ))0xD60060 ;/* S10V */ ≀ rtn = (*parard)(padr) ;	Struct parar_p { long node ; unsigned short *Erradr ; unsigned char *Setadr ; } ; ≀ long (*parard)( ) ; long rtn ; struct parar_p *padr ; ≀ parard = (long(*)( ))0xDE0060 ;/* S10V */ ≀ rtn = (*parard)(padr) ;

[Parameters for mathematical/logical function]

+0x0000	Message transmission service number (5)
+0x0004	Remote node number (1 to 254)
+0x0008	Error code storage address
+0x000C	Read parameter data storage address

[Parameters]

padr	: Input parameter storage starting address
Struct parar_p {	
long node ;	: Remote node number (1 to 254)
unsigned short *Erradr ;	: Error code storage address
unsigned char *Setadr ;	: Read parameter data storage address
} ;	

Be sure to specify an even-numbered address in padr.

Parameter	Input range
Message transmission service number	5 (only when a mathematical/logical function is used)
Remote node number	1 to 254
Error code storage address	Specify a real address (even-numbered address) on the CMU.
Read parameter data storage address	

(*) For the relationship between real address on LPU (memory address) and symbol name such as XW000, see "Table 2-5 Virtual Address Spare and Symbol Names." "Memory address" in table is the address to be specified.

[Return code] (for C mode handler only)

0 (0x00000000): Normal termination

-1 (0xFFFFFFFF): Parameter abnormality exists or the handler is busy processing another message. Error code is stored in area which specified error code storage address. For content and measure of error code, see "Table 7-3 List of Detectable Codes."

-2 (0xFFFFFFFE): Specifies real address (0x01000000 to 0xFFFFFFFE) on CMU at error code storage address or read parameter data storage address. Please review the specified address.

[Function]

A network parameter read request will be issued to a remote node.

When a response code is received, the network parameter information about the specified node transfers to the read parameter data storage address.

The data transfers to the read parameter data storage address in the format shown below.

The node number does not transfer to the S10V.

Read parameter data storage address →

	2^{15}	2^{0}
0x0000	Node name (equipment name)	
0x000A	Vendor name	
0x0014	Maker format	
0x001E	Area 1 starting address	
0x0020	Area 1 size	
0x0022	Area 2 starting address	
0x0024	Area 2 size	
0x0026	Reserve	Token surveillance timeout
0x0028	Reserve	Minimum frame interval
0x002A	Reserve	Link status
0x002C	Reserve	Protocol version
0x002E	Higher layer status	
0x0030	Refresh cycle time (RCT) setting	
0x0032	Refresh cycle measurement value (current)	
0x0034	Refresh cycle measurement value (maximum)	
0x0036	Refresh cycle measurement value (minimum)	

Network parameter write request: parawt( )

[Linking procedure]

C language	
Main	Sub
Struct paraw_p { long node ; unsigned short *Erradr ; unsigned char *Dataadr ; } ; ∫ long (*parawt)( ) ; long rtn ; struct paraw_p *padr ; ∫ parawt = (long(*)( ))0xD60080 ;/* S10V */ ∫ rtn = (*parawt)(padr) ;	Struct paraw_p { long node ; unsigned short *Erradr ; unsigned char *DataPadr ; } ; ∫ long (*parawt)( ) ; long rtn ; struct paraw_p *padr ; ∫ parawt = (long(*)( ))0xDE0080 ;/* S10V */ ∫ rtn = (*parawt)(padr) ;

[Parameters for mathematical/logical function]

+0x0000	Message transmission service number (6)
+0x0004	Remote node number (1 to 254)
+0x0008	Error code storage address
+0x000C	Write parameter data storage address

[Parameters]

```
padr                                  : Input parameter storage starting address
Struct paraw_p {
              long node ;             : Remote node number (1 to 254)
    unsigned short *Erradr ;          : Error code storage address
    unsigned char  *Setadr ;          : Write parameter data storage address
} ;
```

Be sure to specify an even-numbered address in padr.

Parameter	Input range
Message transmission service number	6 (only when a mathematical/logical function is used)
Remote node number	1 to 254
Error code storage address	Specify a real address (even-numbered address) on the LPU. (*)
Write parameter data storage address	

(*) For the relationship between real address on LPU (memory address) and symbol name such as XW000, see "Table 2-5 Virtual Address Spare and Symbol Names." "Memory address" in table is the address to be specified.

[Return code] (for C mode handler only)

0 (0x00000000): Normal termination

-1 (0xFFFFFFFF): Parameter abnormality exists or the handler is busy processing another message. Error code is stored in area which specified error code storage address. For content and measure of error code, see "Table 7-7 List of Detectable Codes."

-2 (0xFFFFFFFE): Specifies real address (0x01000000 to 0xFFFFFFFE) on CMU at error code storage address or write parameter data storage address. Please review the specified address.

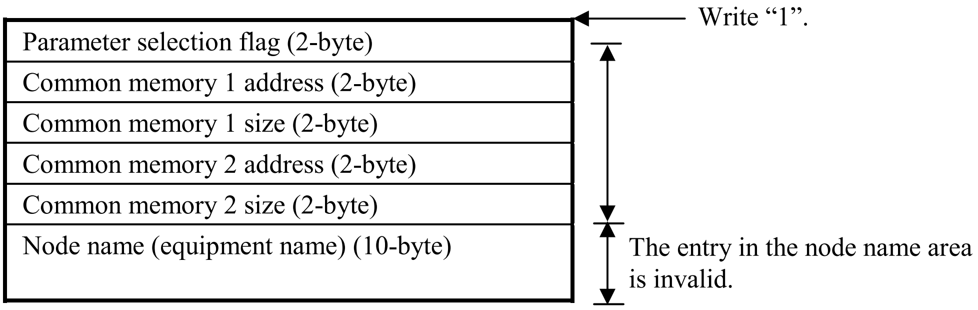
[Function]

A network parameter write request using the parameters at the write parameter data storage address will be issued to a remote node.

[Write parameter data structure] (contiguous areas)

Create the data as shown below in the area specified by the write parameter data storage address.

1. When rewriting the common memory address and common memory size only

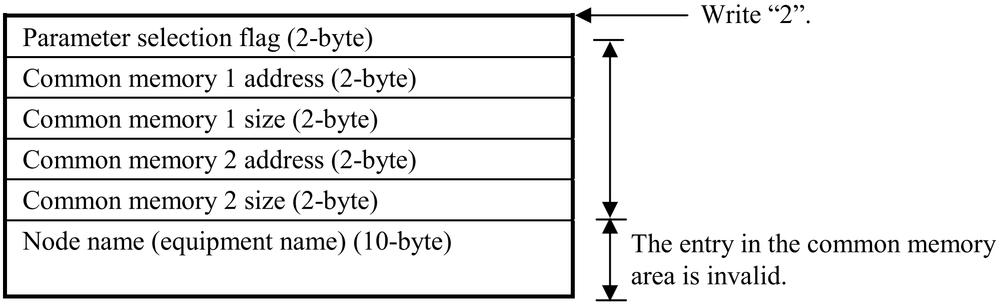


Parameter selection flag (2-byte)
Common memory 1 address (2-byte)
Common memory 1 size (2-byte)
Common memory 2 address (2-byte)
Common memory 2 size (2-byte)
Node name (equipment name) (10-byte)

Write “1”.

The entry in the node name area is invalid.

2. When rewriting the node name (equipment name) only

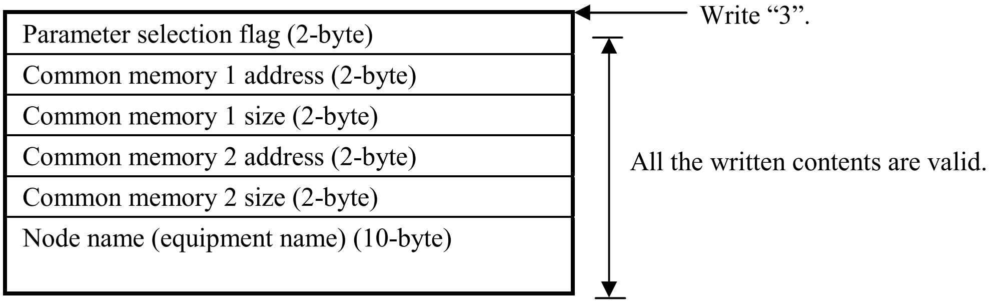


Parameter selection flag (2-byte)
Common memory 1 address (2-byte)
Common memory 1 size (2-byte)
Common memory 2 address (2-byte)
Common memory 2 size (2-byte)
Node name (equipment name) (10-byte)

Write “2”.

The entry in the common memory area is invalid.

3. When rewriting the common memory address, common memory size, and node name (equipment name)



Parameter selection flag (2-byte)
Common memory 1 address (2-byte)
Common memory 1 size (2-byte)
Common memory 2 address (2-byte)
Common memory 2 size (2-byte)
Node name (equipment name) (10-byte)

Write “3”.

All the written contents are valid.

NOTICE

- When rewriting the network parameters, take extreme care not to cause any setting duplication to the common memory.
- The server feature of the FL.NET module returns an abnormal response if, upon reception of a network parameter write request, the parameter selection flag is found to be not between 1 and 3. The error code used in such an event is 0xFFFFFFFF.

Stop request: reqstop( )

[Linking procedure]

C language	
Main	Sub

Struct reqs_p {
        long node ;
    unsigned short *Erradr ;
} ;
        ⟆
long (*ReqStop)( ) ;
long rtn ;
struct reqs_p *padr ;
        ⟆
ReqStop = (long(*)( ))0xD600A0 ;/* S10V */
        ⟆
rtn = (*ReqStop)(padr) ;
	Struct reqs_p {
	        long node ;
	    unsigned short *Erradr ;
	} ;
	        ⟆
	long (*ReqStop)( ) ;
	long rtn ;
	struct reqs_p *padr ;
	        ⟆
	ReqStop = (long(*)( ))0xDE00A0 ;/* S10V */
	        ⟆
	rtn = (*ReqStop)(padr) ;

[Parameters for mathematical/logical function]

+0x0000	Message transmission service number (7)
+0x0004	Remote node number (1 to 254)
+0x0008	Error code storage address

[Parameters]

```
padr                                : Input parameter storage starting address
Struct reqs_p {
                long node ;         : Remote node number (1 to 254)
     unsigned short *Erradr ;       : Error code storage address
} ;
```

Be sure to specify an even-numbered address in padr.

Parameter	Input range
Message transmission service number	7 (only when a mathematical/logical function is used)
Remote node number	1 to 254
Error code storage address	Specify a real address (even-numbered address) on the LPU. (*)

(*) For the relationship between real address on LPU (memory address) and symbol name such as XW000, see "Table 2-5　Virtual Address Spare and Symbol Names."　"Memory address" in table is the address to be specified.

[Return code] (for C mode handler only)

0 (0x00000000): Normal termination

-1 (0xFFFFFFFF): Parameter abnormality exists or the handler is busy processing another message.　Error code is stored in area which specified error code storage address.　For content and measure of error code, see "Table 7-3　List of Detectable Codes."

-2 (0xFFFFFFFE): Specifies real address (0x01000000 to 0xFFFFFFFE) on CMU at error code storage address.　Please review the specified address.

[Function]

A stop request will be issued to a remote node.

NOTICE

The FL.NET module does not support the server feature for a stop directive.　If a stop request is issued to the FL.NET module, it returns a response to indicate that it does not support the stop directive.

Run request: reqrun( )

[Linking procedure]

C language	
Main	Sub
Struct reqr_p { long node ; unsigned short *Erradr ; } ; ∫ long (*ReqRun)( ) ; long rtn ; struct reqr_p *padr ; ∫ ReqRun = (long(*)( ))0xD600C0 ;/* S10V */ ∫ rtn = (*ReqRun)(padr) ;	Struct reqr_p { long node ; unsigned short *Erradr ; } ; ∫ long (*ReqRun)( ) ; long rtn ; struct reqr_p *padr ; ∫ ReqRun = (long(*)( ))0xDE00C0 ;/* S10V */ ∫ rtn = (*ReqRun)(padr) ;

[Parameters for mathematical/logical function]

+0x0000	Message transmission service number (8)
+0x0004	Remote node number (1 to 254)
+0x0008	Error code storage address

[Parameters]

```
padr                            : Input parameter storage starting address
Struct reqr_p {
          long node ;           : Remote node number (1 to 254)
   unsigned short *Erradr ;     : Error code storage address
} ;
```

Be sure to specify an even-numbered address in padr.

Parameter	Input range
Message transmission service number	8 (only when a mathematical/logical function is used)
Remote node number	1 to 254
Error code storage address	Specify a real address (even-numbered address) on the LPU. (*)

(*) For the relationship between real address on LPU (memory address) and symbol name such as XW000, see "Table 2-5 Virtual Address Spare and Symbol Names." "Memory address" in table is the address to be specified.

[Return code] (for C mode handler only)

0 (0x00000000): Normal termination

-1 (0xFFFFFFFF): Parameter abnormality exists or the handler is busy processing another message. Error code is stored in area which specified error code storage address. For content and measure of error code, see "Table 7-3 List of Detectable Codes."

-2 (0xFFFFFFFE): Specifies real address (0x01000000 to 0xFFFFFFFE) on CMU at error code storage address. Please review the specified address.

[Function]

A run request will be issued to a remote node.

NOTICE

The FL.NET module does not support the server feature for a run directive. If a run request is issued to the FL.NET module, it returns a response to indicate that it does not support the run directive.

Profile read

The FL.NET module does not support the client feature of profile read for another node. However, it responds to a request from another node.

The FL.NET module's response to a profile read request from another node is indicated below:

Identifier	Length					
0x30	0x816A					
		Identifier	Length			
		0x30	0x68			
				Identifier	Length	Description
				0x13	0x06	"COMVER"
				0x02	0x01	"0x01"
				0x13	0x02	"ID"
				0x13	0x07	"SYSPARA"
				0x13	0x03	"REV"
				0x02	0x01	"0x01"
				0x13	0x07	"REVDATE"
				0x30	0x0A	
				0x02	0x02	"0x07D9 (2009) "
				0x02	0x01	"0x05(5) "
				0x02	0x01	"0x1A(26) "
				0x13	0x0A	"DVCATEGORY"
				0x13	0x03	"PLC"
				0x13	0x06	"VENDOR"
				0x13	0x07	"HITACHI"
				0x13	0x07	"DVMODEL"
				0x13	0x06	"LQE702"

Communication log data read request: logrd( )

[Linking procedure]

C language	
Main	Sub
Struct logr_p { long node ; unsigned short *Erradr ; unsigned char *logadr ; } ; ∫ long (*Logrd)( ) ; long rtn ; struct logr_p *padr ; ∫ Logrd = (long(*)( ))0xD600E0 ;/* S10V */ ∫ rtn = (*Logrd)(padr) ;	Struct logr_p { long node ; unsigned short *Erradr ; unsigned char *logadr ; } ; ∫ long (*Logrd)( ); long rtn ; struct logr_p *padr ; ∫ Logrd = (long(*)( ))0xDE00E0 ;/* S10V */ ∫ rtn = (*Logrd)(padr) ;

[Parameters for mathematical/logical function]

+0x0000	Message transmission service number (10)
+0x0004	Remote node number (1 to 254)
+0x0008	Error code storage address
+0x000C	Log data storage address

[Parameters]

```
padr                                  : Input parameter storage starting address
Struct logr_p {
                long node ;           : Remote node number (1 to 254)
    unsigned short *Erradr ;          : Error code storage address
    unsigned char  *logadr ;          : Log data storage address
} ;
```

Be sure to specify an even-numbered address in padr.

Parameter	Input range
Message transmission service number	10 (only when a mathematical/logical function is used)
Remote node number	1 to 254
Error code storage address	Specify a real address (even-numbered address) on the LPU. (*)
Log data storage address	

(*) For the relationship between real address on LPU (memory address) and symbol name such as XW000, see "Table 2-5 Virtual Address Spare and Symbol Names." "Memory address" in table is the address to be specified.

[Return code] (for C mode handler only)

0 (0x00000000): Normal termination

-1 (0xFFFFFFFF): Parameter abnormality exists or the handler is busy processing another message. Error code is stored in area which specified error code storage address. For content and measure of error code, see "Table 7-3 List of Detectable Codes."

-2 (0xFFFFFFFE): Specifies real address (0x01000000 to 0xFFFFFFFE) on CMU at error code storage address or log data storage address. Please review the specified address.

[Function]

- A communication log data read request will be issued to a remote node. When a response code returns, the communication log data within the response code transfers to the log data storage address.
- The communication log data to be transferred consists of 512 bytes. Each data set consists of 4-byte data. For details on communication log data, see "5.3.8 Using the communication log."

Communication log data clear request: logclr( )

[Linking procedure]

C language	
Main	Sub
Struct logclr_p { long node ; unsigned short *Erradr ; } ; ≀ long (*Logclr)( ) ; long rtn ; struct logclr_p *padr ; ≀ Logclr = (long(*)( )) 0xD60100 ;/* S10V */ ≀ rtn = (*Logclr)(padr) ;	Struct logclr_p { long node ; unsigned short *Erradr ; } ; ≀ long (*Logclr)( ) ; long rtn ; struct logclr_p *padr ; ≀ Logclr = (long(*)( ))0xDE0100 ;/* S10V */ ≀ rtn = (*Logclr)(padr) ;

[Parameters for mathematical/logical function]

+0x0000	Message transmission service number (11)
+0x0004	Remote node number (1 to 255)
+0x0008	Error code storage address

[Parameters]

```
padr                              : Input parameter storage starting address
Struct logr_p {
             long node ;          : Remote node number (1 to 255)
    unsigned short *Erradr ;      : Error code storage address
} ;
```

Be sure to specify an even-numbered address in padr.

Parameter	Input range
Message transmission service number	11 (only when a mathematical/logical function is used)
Remote node number	1 to 255
Error code storage address	Specify a real address (even-numbered address) on the LPU. (*)

(*) For the relationship between real address on LPU (memory address) and symbol name such as XW000, see "Table 2-5 Virtual Address Spare and Symbol Names." "Memory address" in table is the address to be specified.

[Return code] (for C mode handler only)

0 (0x00000000): Normal termination

-1 (0xFFFFFFFF): Parameter abnormality exists or the handler is busy processing another message. Error code is stored in area which specified error code storage address. For content and measure of error code, see "Table 7-3 List of Detectable Codes."

-2 (0xFFFFFFFE): Specifies real address (0x01000000 to 0xFFFFFFFE) on CMU at error code storage address. Please review the specified address.

[Function]

- A communication log data clear request will be issued to a remote node.
- When the remote node number is set to 255, a one-to-N transmission occurs.

Message return request: mesret( )

[Linking procedure]

C language	
Main	Sub
Struct mesreq_p { long node ; unsigned short *Erradr ; unsigned char *SendData ; long Sendsiz ; unsigned char *Recvadr ; } ; ≀ long (*Mesret)( ) ; long rtn ; struct mesreq_p *padr ; ≀ Mesret = (long(*)( ))0xD60120 ;/* S10V */ ≀ rtn = (*Mesret)(padr) ;	Struct mesreq_p { long node ; unsigned short *Erradr ; unsigned char *SendData ; long Sendsiz ; unsigned char *Recvadr ; } ; ≀ long (*Mesret)( ) ; long rtn ; struct mesreq_p *padr ; ≀ Mesret = (long(*)( ))0xDE0120 ;/* S10V */ ≀ rtn = (*Mesret)(padr) ;

[Parameters for mathematical/logical function]

+0x0000	Message transmission service number (12)
+0x0004	Remote node number (1 to 254)
+0x0008	Error code storage address
+0x000C	Message data storage starting address
+0x0010	Message data size (1 to 1,024)
+0x0014	Return message storage address

[Parameters]

```
padr                                 : Input parameter storage starting address
Struct mesreq_p {
                long node ;          : Remote node number (1 to 254)
     unsigned short *Erradr ;        : Error code storage address
     unsigned char  *SendData ;      : Message data storage starting address
                long Sendsiz ;       : Message data size (1 to 1,024)
     unsigned char  *Recvadr ;       : Return message storage address
} ;
```

Be sure to specify an even-numbered address in padr.

Parameter	Input range
Message transmission service number	12 (only when a mathematical/logical function is used)
Remote node number	1 to 254
Error code storage address	Specify a real address (even-numbered address) on the LPU. (*)
Message data storage starting address	
Message data size	1 to 1,024
Return message storage address	Specify a real address (even-numbered address) on the LPU. (*)

(*) For the relationship between real address on LPU (memory address) and symbol name such as XW000, see “Table 2-5 Virtual Address Spare and Symbol Names.” “Memory address” in table is the address to be specified.

[Return code] (for C mode handler only)

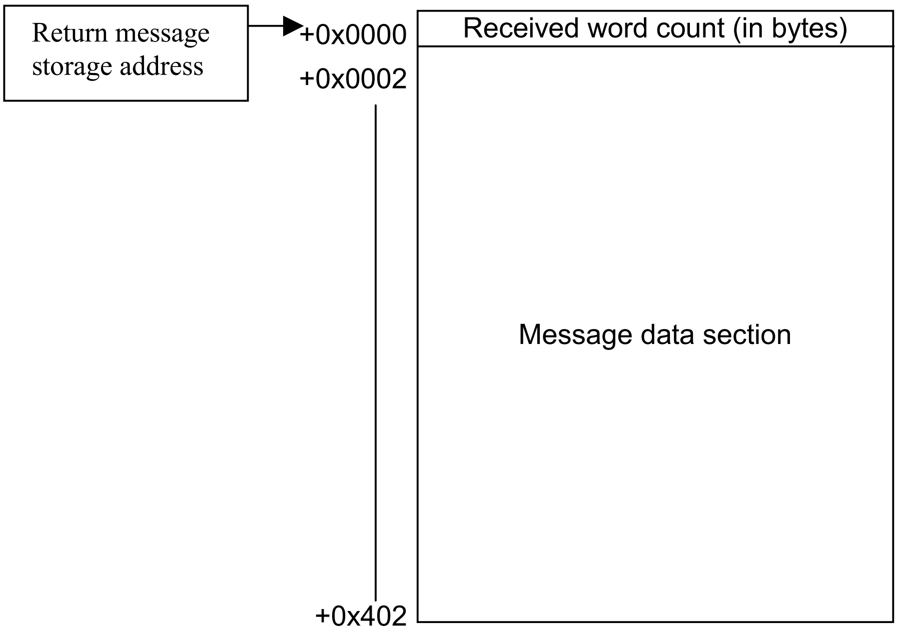
0 (0x00000000): Normal termination

-1 (0xFFFFFFFF): Parameter abnormality exists or the handler is busy processing another message. Error code is stored in area which specified error code storage address. For content and measure of error code, see “Table 7-3 List of Detectable Codes.”

-2 (0xFFFFFFFE): Specifies real address (0x01000000 to 0xFFFFFFFE) on CMU at error code storage address, message data storage starting address or return message storage address. Please review the specified address.

[Function]

- A message return request using the data at the message data storage starting address will be issued to a remote node. When a response message returns, the message data within the response code transfers to the return message storage address.
- The data transfers to the return message storage address in the following format.


Return message storage address
+0x0000
+0x0002
+0x402
Received word count (in bytes)
Message data section

Specified task control request (Hitachi's unique transparent type of support): reqmacro( )

[Operation performed]

Exercises control (abort/release/queue) over a user-specified task number.

[Linking procedure]

C language	
Main	Sub
Struct Reqmacro_p { long node unsigned short *Erradr ; unsigned long *Retadr ; long ParaCnt ; unsigned long Para[3] ; } ; ≀ long (*ReqMacro)( ) ; long rtn ; struct Reqmacro_p *padr ; ≀ ReqMacro = (long(*)( ))0xD60140 ;/* S10V */ ≀ rtn = (*ReqMacro)(padr) ;	Struct Reqmacro_p { long node unsigned short *Erradr ; unsigned long *Retadr ; long ParaCnt ; unsigned long Para[3] ; } ; ≀ long (*ReqMacro)( ) ; long rtn ; struct Reqmacro_p *padr ; ≀ ReqMacro = (long(*)( ))0xDE0140 ;/* S10V */ ≀ rtn = (*ReqMacro)(padr) ;

[Parameters for mathematical/logical function]

+0x0000	Message transmission service number (13)
+0x0004	Remote node number (1 to 254)
+0x0008	Error code storage address
+0x000C	CPMS macro execution result storage address
+0x0010	CPMS macro parameter count
+0x0014	CPMS macro parameter 1
+0x0018	CPMS macro parameter 2
+0x001C	CPMS macro parameter 3

[Parameters]

```
padr                                : Input parameter storage starting address
Struct Reqmacro_p {
                    long node ;     : Remote node number (1 to 254)
       unsigned short *Erradr ;     : Error code storage address
        unsigned long *Retdadr ;    : CPMS macro execution result storage address
                    long ParaCnt ;  : CPMS macro parameter count
        unsigned long Para[3] ;     : CPMS macro parameter
} ;
```

Be sure to specify an even-numbered address in padr.

Parameter	Input range
Message transmission service number	13 (only when a mathematical/logical function is used)
Remote node number	1 to 254
Error code storage address	Specify a real address (even-numbered address) on the LPU. (*)
CPMS macro execution result storage address	
CPMS macro parameter count	See [Function]
CPMS macro parameter	

(*) For the relationship between real address on LPU (memory address) and symbol name such as XW000, see “Table 2-5 Virtual Address Spare and Symbol Names.” “Memory address” in table is the address to be specified.

[Return code] (for C mode handler only)

0 (0x00000000): Normal termination

-1 (0xFFFFFFFF): Parameter abnormality exists or the handler is busy processing another message. Error code is stored in area which specified error code storage address. For content and measure of error code, see “Table 7-3 List of Detectable Codes.”

-2 (0xFFFFFFFE): Specifies real address (0x01000000 to 0xFFFFFFFE) on CMU at error code storage address or CPMS macro execution result storage address. Please review the specified address.

[Function]

A task abort, release, or queue execution request can be issued to a remote node.

When a response message is received, the task control execution result (log value) is written at the CPMS macro execution result storage address.

Specify the parameters as indicated below:

<To execute a task abort>

CPMS macro parameter count = 2
CPMS macro parameter [0] = 1
CPMS macro parameter [2] = task number (1 to 255)

<To execute a task release>

CPMS macro parameter count = 2
CPMS macro parameter [0] = 2
CPMS macro parameter [1] = task number (1 to 255)

<To execute a task queue>

CPMS macro parameter count = 3
CPMS macro parameter [0] = 3
CPMS macro parameter [1] = task number (1 to 255)
CPMS macro parameter [2] = task number (0 to 32)

Note: Task numbers 230 to 255 are used by the system. So they are not started even if specified.

NOTICE

Exercise due care when using the above transparent message request. If the task number is erroneously specified, the CPU or CMU of the remote node (S10V) may become inoperative.

Transparent message reception request (Hitachi's unique transparent type of support): toukaread( )

[Linking procedure]

C language	
Main	Sub
Struct ToukaRead_p { long node ; unsigned short *Erradr ; unsigned char *dataadr ; unsigned long datasiz ; } ; ⟆ long (*ToukaRead)( ) ; long rtn ; struct ToukaRead_p *padr ; ⟆ ToukaRead = (long(*)( ))0xD60160 ;/* S10V */ ⟆ rtn = (*ToukaRead)(padr) ;	Struct ToukaRead_p { long node ; unsigned short *Erradr ; unsigned char *dataadr ; unsigned long datasiz ; } ; ⟆ long (*ToukaRead)( ) ; long rtn ; struct ToukaRead_p *padr ; ⟆ ToukaRead = (long(*)( ))0xDE0160 ;/* S10V*/ ⟆ rtn = (*ToukaRead)(padr) ;

[Parameters for mathematical/logical function]

+0x0000	Message transmission service number (17)
+0x0004	Reception destination node number (0 to 254)
+0x0008	Error code storage address
+0x000C	Reception data storage address
+0x0010	Data word count (0 to 1,024)

[Parameters]

```
padr                              : Input parameter storage starting address
Struct ToukaRead_p {
            long node ;           : Reception destination node number (0 to 254)
    unsigned short *Erradr ;      : Error code storage address
    unsigned char  *dataadr ;     : Reception data storage address
    unsigned long  datasiz ;      : Data word count (0 to 1,024)
} ;
```

Be sure to specify an even-numbered address in padr.

Parameter	Input range
Message transmission service number	17 (only when a mathematical/logical function is used)
Reception destination node number	0 to 254
Error code storage address	Specify a real address (even-numbered address) on the LPU. (*)
Reception data storage address	
Data word count	0 to 1,024 (variable in 1-byte units)

(*) For the relationship between real address on LPU (memory address) and symbol name such as XW000, see "Table 2-5 Virtual Address Spare and Symbol Names." "Memory address" in table is the address to be specified.

[Return code] (for C mode handler only)

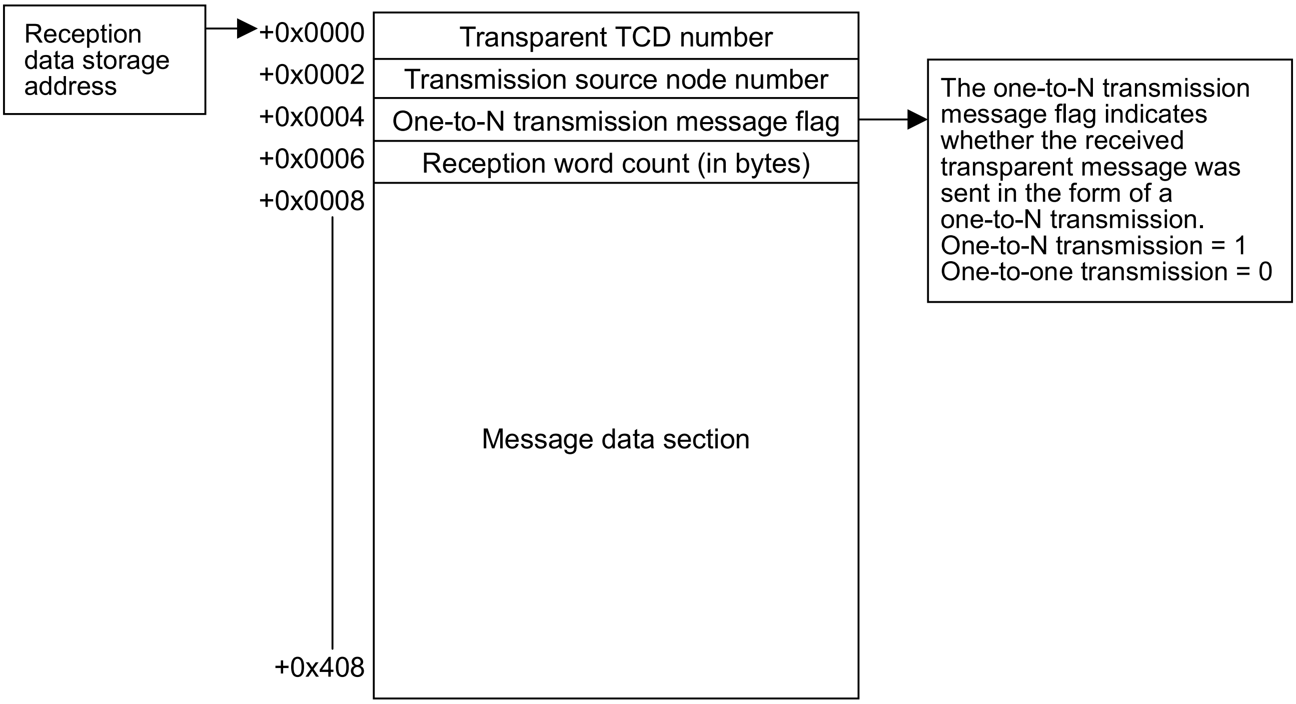
0 (0x00000000): Normal termination

-1 (0xFFFFFFFF): Parameter abnormality exists or the handler is busy processing another message. Error code is stored in area which specified error code storage address. For content and measure of error code, see "Table 7-3 List of Detectable Codes."

-2 (0xFFFFFFFE): Specifies real address (0x01000000 to 0xFFFFFFFE) on CMU at error code storage address or reception data storage address. Please review the specified address.

[Function]

- Reception word count will be transferred to the area which is specified received transparent message by reception data storage address
- If there is no transparent message from a reception destination node number, a parameter error occurs.
- When the reception destination node number is 0, the first-retrieved data within a transparent message queue transfers to the reception data storage address without regard to the transmission source node number (this transfer does not take place in order of reception). If no transparent message is received, a parameter error occurs.
- When the user-specified data word count is lower than the actually reception word count, only as many words of message data as specified by the user will be transferred.
- A transparent message transfers to the reception data storage address in the following format:


Reception data storage address
+0x0000
+0x0002
+0x0004
+0x0006
+0x0008
+0x408
Transparent TCD number
Transmission source node number
One-to-N transmission message flag
Reception word count (in bytes)
Message data section
The one-to-N transmission message flag indicates whether the received transparent message was sent in the form of a one-to-N transmission.
One-to-N transmission = 1
One-to-one transmission = 0

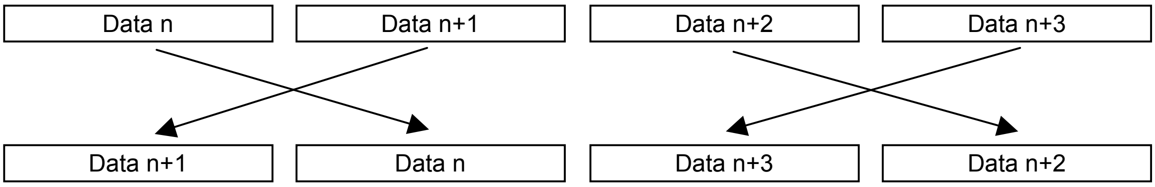
Note: When the user-specified data word count is expressed by an odd number of bytes, the 1-byte data "0x00" is added to the finally-transferred reception data.

- FL.NET module's unique feature

  When the received transparent message TCD number is between 0 and 999, the reception message data is handled as word data in little-endian format. However, if the reception message data word count is expressed by an odd number of bytes, the data does not convert normally into little-endian format. If the specified TCD number is other than the above, the reception data is transferred as it is.

Data conversion effected when a transparent message having a TCD number between 0 and 999 is received

Example: When 4-byte data is received



Byte-swapped on an individual word basis and then received.

Transparent message transmission request (Hitachi's unique transparent type of support): toukasend( )

[Linking procedure]

C language	
Main	Sub
Struct ToukaSend_p {	Struct ToukaSend_p {
long node ;	long node ;
unsigned short *Erradr ;	unsigned short *Erradr ;
unsigned char *dataadr ;	unsigned char *dataadr ;
unsigned long datasiz ;	unsigned long datasiz ;
unsigned long TcdNo ;	unsigned long TcdNo ;
} ;	} ;
≀	≀
long (*ToukaSend)( ) ;	long (*ToukaSend)( ) ;
long rtn ;	long rtn ;
struct ToukaSend_p *padr ;	struct ToukaSend_p *padr ;
≀	≀
ToukaSend = (long(*)( ))0xD60180 ;/* S10V */	ToukaSend = (long(*)( ))0xDE0180 ;/* S10V */
≀	≀
rtn = (*ToukaSend)(padr) ;	rtn = (*ToukaSend)(padr) ;

[Parameters for mathematical/logical function]

+0x0000	Message transmission service number (18)
+0x0004	Transmission destination node number (1 to 255)
+0x0008	Error code storage address
+0x000C	Transmission data storage address
+0x0010	Data word count (0 to 1,024)
+0x0014	Transmission message TCD (0 to 59999)

[Parameters]

padr	: Input parameter storage starting address
Struct ToukaSend_p {	
long node ;	: Transmission destination node number (1 to 255)
unsigned short *Erradr ;	: Error code storage address
unsigned char *dataadr ;	: Transmission data storage address
unsigned long datasiz ;	: Data word count (0 to 1,024)
unsigned long TcdNo ;	: Transmission message TCD (0 to 59999)
} ;	

Be sure to specify an even-numbered address in padr.

Parameter	Input range
Message transmission service number	18 (only when a mathematical/logical function is used)
Transmission destination node number	1 to 255
Error code storage address	Specify a real address (even-numbered address) on the LPU. (*)
Transmission data storage address	
Data word count	0 to 1,024 (specified in bytes)
Transmission message TCD	0 to 59999 (with the exception of the following) 1000, 1001, 1200, 1201

(*) For the relationship between real address on LPU (memory address) and symbol name such as XW000, see “Table 2-5　Virtual Address Spare and Symbol Names.”　“Memory address” in table is the address to be specified.

[Return code] (for C mode handler only)

0 (0x00000000): Normal termination

-1 (0xFFFFFFFF): Parameter abnormality exists or the handler is busy processing another message.　Error code is stored in area which specified error code storage address.　For content and measure of error code, see “Table 7-3　List of Detectable Codes.”

-2 (0xFFFFFFFE): Specifies real address (0x01000000 to 0xFFFFFFFE) on CMU at error code storage address or transmission data storage address.　Please review the specified address.

[Function]

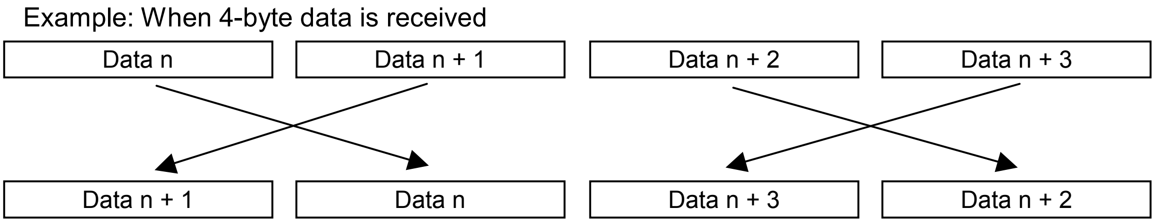
- As many data words as specified in bytes will be transferred from the area specified by a transmission data storage address to the remote node having a specified node number as a transparent message having the TCD number specified by a transmission message TCD.
- When the transmission destination node number is set to 255, a one-to-N transmission takes place.
- When the data word count is set to 0, a transmission takes place with no data added.
- For a transmission message TCD, you cannot specify the following TCD numbers that are used by the transparent message-unique support feature (specified task control/specified subroutine control).

<TCD numbers that cannot be specified>
1000, 1001, 1200, 1201

● FL.NET module-unique feature

When a TCD number between 0 and 999 is specified, the transmission data is sent as data in little-endian format. If the transmission word count is expressed by an odd number of bytes, the transmission data is converted into little-endian format with the 1-byte data "0x00" added. If the specified TCD number is other than the above, the transmission data is sent as is.

Data conversion effected when the specified TCD number is between 0 and 999



Byte-swapped on an individual word basis and then transmitted.

Common memory offset feature: comoffset( )

[Operation performed]

When this handler is executed, the starting address for data transfer is set at a position that is shifted by a specified offset amount from the beginning of a specified node’s common memory area.

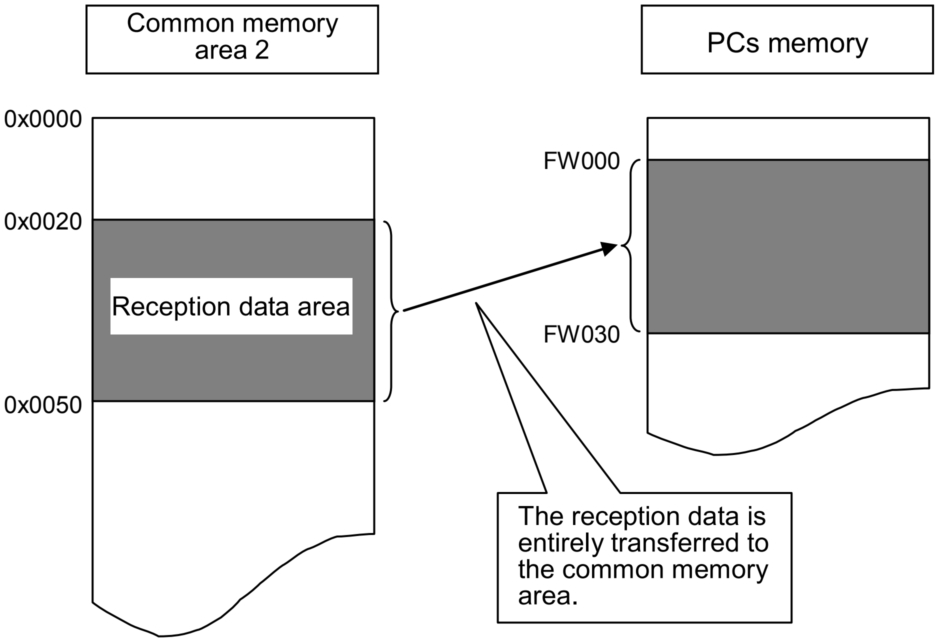
For each node, the offset size can be variously set for common memory area 1 and 2.

This handler is effective when only part of the received data is to be used.

The examples given below indicate the difference between a normal operation (no offset used) and an operation performed with the offset specified.

Example: When a normal operation is performed (with no offset specified)

Common memory area 2 address: 0x0020
Common memory area 2 word count: 0x0030
PCs memory address: FW000
PCs memory word count: 30



The reception data is entirely transferred to the common memory area.

When an offset is specified

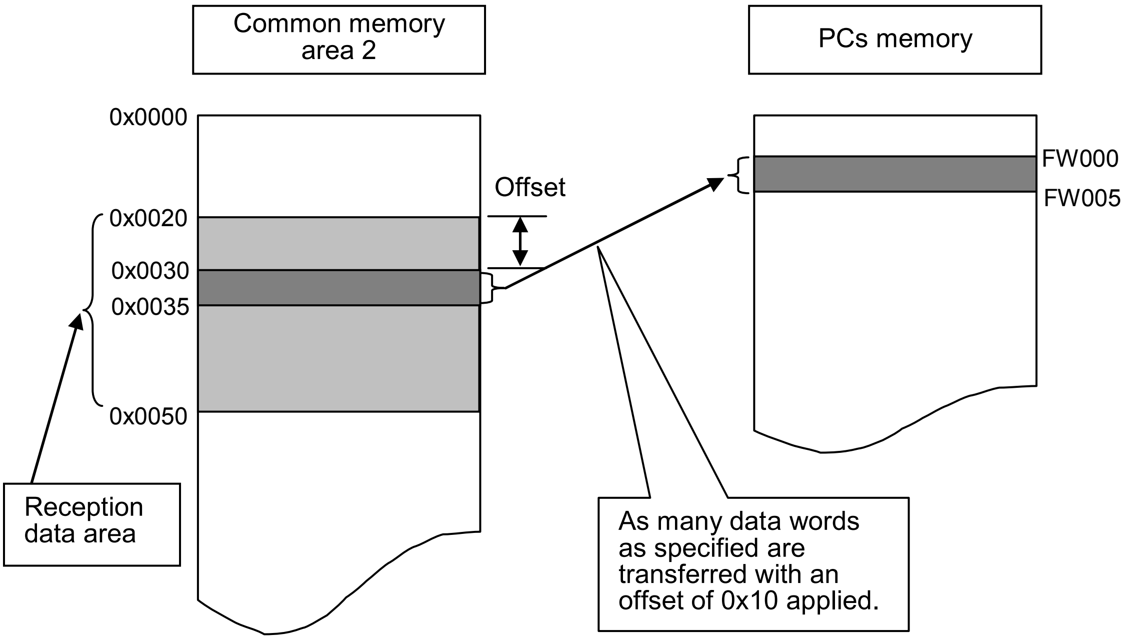
Common memory area 2 address: 0x0020

Common memory area 2 word count: 0x0030

PCs memory address: FW000

PCs memory word count: 5

Offset size: 0x10



[Linking procedure]

C language	
Main	Sub
Struct CommonOffset_p { long node ; unsigned short *Erradr ; unsigned long com1offset ; unsigned long com2offset ; } ; ⌇ long (*comoffset)( ) ; long rtn ; struct CommonOffset_p *padr ; ⌇ comoffset= (long(*)( ))0xD601A0 ;/* S10V */ ⌇ rtn = (*comoffset)(padr) ;	Struct CommonOffset_p { long node ; unsigned short *Erradr ; unsigned long com1offset ; unsigned long com2offset ; } ; ⌇ long (*comoffset)( ) ; long rtn ; struct CommonOffset_p *padr ; ⌇ comoffset= (long(*)( ))0xDE01A0 ;/* S10V */ ⌇ rtn = (*comoffset)(padr) ;

[Parameters for mathematical/logical function]

+0x0000	Message transmission service number (19)
+0x0004	Specified node number (1 to 254)
+0x0008	Error code storage address
+0x000C	Common memory area 1 offset size
+0x0010	Common memory area 2 offset size

[Parameters]

```
padr                                 : Input parameter storage starting address
Struct CommonOffset_p {
                long node ;          : Specified node number (1 to 254)
    unsigned short *Erradr ;         : Error code storage address
    unsigned long  com1offset ;      : Common memory area 1 offset size
    unsigned long  com2offset ;      : Common memory area 2 offset size
} ;
```

Be sure to specify an even-numbered address in padr.

Parameter	Input range
Message transmission service number	19 (only when a mathematical/logical function is used)
Specified node number	1 to 254
Error code storage address	Specify a real address (even-numbered address) on the LPU. (*)
Common memory area 1 offset size	0 to 0x1FF (variable in 1-word units)
Common memory area 2 offset size	0 to 0x1FFF (variable in 1-word units)

(*) For the relationship between real address on LPU (memory address) and symbol name such as XW000, see "Table 2-5 Virtual Address Spare and Symbol Names." "Memory address" in table is the address to be specified.

[Return code] (for C mode handler only)

0 (0x00000000): Normal termination

-1 (0xFFFFFFFF): Parameter abnormality exists or the handler is busy processing another message. Error code is stored in area which specified error code storage address. For content and measure of error code, see "Table 7-3 List of Detectable Codes."

-2 (0xFFFFFFFE): Specifies real address (0x01000000 to 0xFFFFFFFE) on CMU at error code storage address. Please review the specified address.

[Function]

- When the node having a specified node number is to be subjected to common memory transfer, an offset size will be reported to the FL.NET module.
- If the specified node number coincides with the self-node number, a parameter error occurs.
- If the maximum size (0x200) of common memory area 1 is exceeded by the offset size specified for common memory area 1, a parameter error occurs.
- If the maximum size (0x2000) of common memory area 2 is exceeded by the offset size specified for common memory area 2, a parameter error occurs.
- If you specify the offset size during message processing, a “message processing already in progress” error occurs.

NOTICE

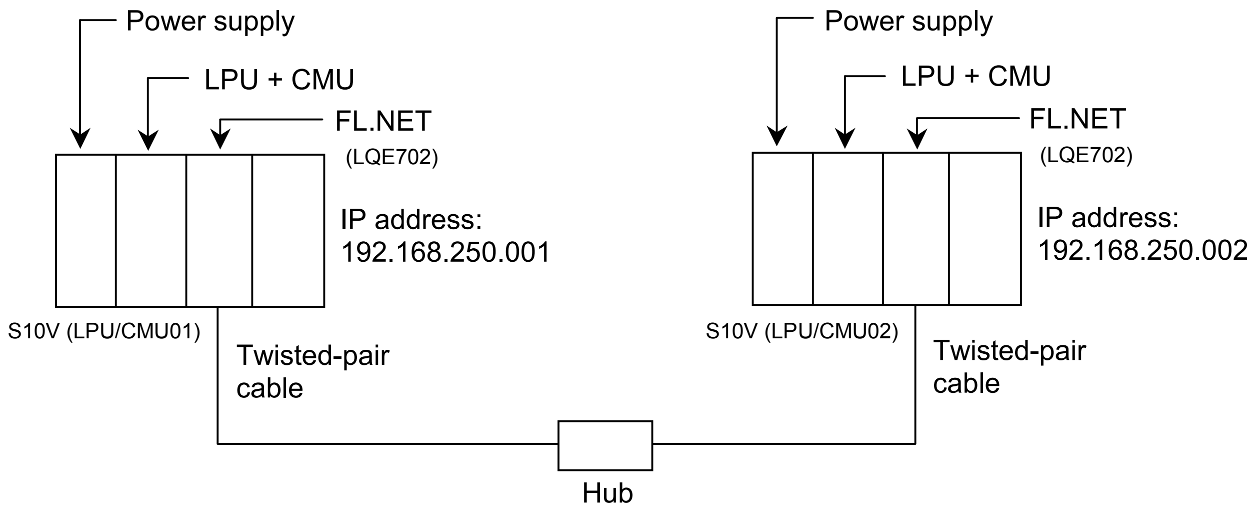
- When an offset size is specified for a specific node number, it is stored in memory until the FL.NET module is turned OFF or reset. Note, however, that the user cannot reference the offset setting within the FL.NET module after completion of offset setup. The offset setting must be managed by the user.
- The above feature is designed for use in cases where the available memory space of the S10V is insufficient. It is recommended that you do not use the feature when the memory space of the S10V is adequate.
- When the address limit of common memory area 1 or 2 is exceeded by the application of an offset, the resulting excess portion of address data will not transfer to the S10V memory. Exercise care not to exceed the limits of the common memory areas.

(6) Program examples

Program examples are shown below. These examples permit two units of the FL.NET module to transmit/receive transparent messages.

(a) Transmission/reception by C mode handler

[System configuration]



For using the C mode handler in the S10V, the LPU module and the CMU module are required.

Ensure that the FL.NET module MAIN/SUB selector switches for LPU/CMU01 and LPU/CMU02 are set to 0.

System component list

Product name	Model	Quantity	Remarks
Power supply	LQV000	2	
LPU + CMU	LQP510 + LQP520	2	
FL.NET	LQE702	2	
Mount base	HSC-1540, 1580	2	Select a model among those shown at left.
Twisted-pair cable	HUTP-CAT5E-4P	2	Maker: Hitachi Cable, Ltd.

[FL.NET module settings]

The FL.NET module settings for LPU/CMU01 and LPU/CMU02 are shown below:

Self-node settings for the FL.NET modules

Setup item	LPU/CMU01 setting	LPU/CMU02 setting
Node number	1	2
Area 1 address (setting)	0x000	0x004
Area 1 words	0x004	0x004
Area 1 address (PCs allocation)	RW000	RW040
Area 2 address (setting)	0x0000	0x0040
Area 2 words	0x0040	0x0040
Area 2 address (PCs allocation)	DW000	DW040
Self-node status (PCs allocation)	RW080	RW080
Transparent reception task	—	—
Transparent reception task's factor	—	—
Transparent receiving flag	—	RW100

Remote node settings for FL.NET modules

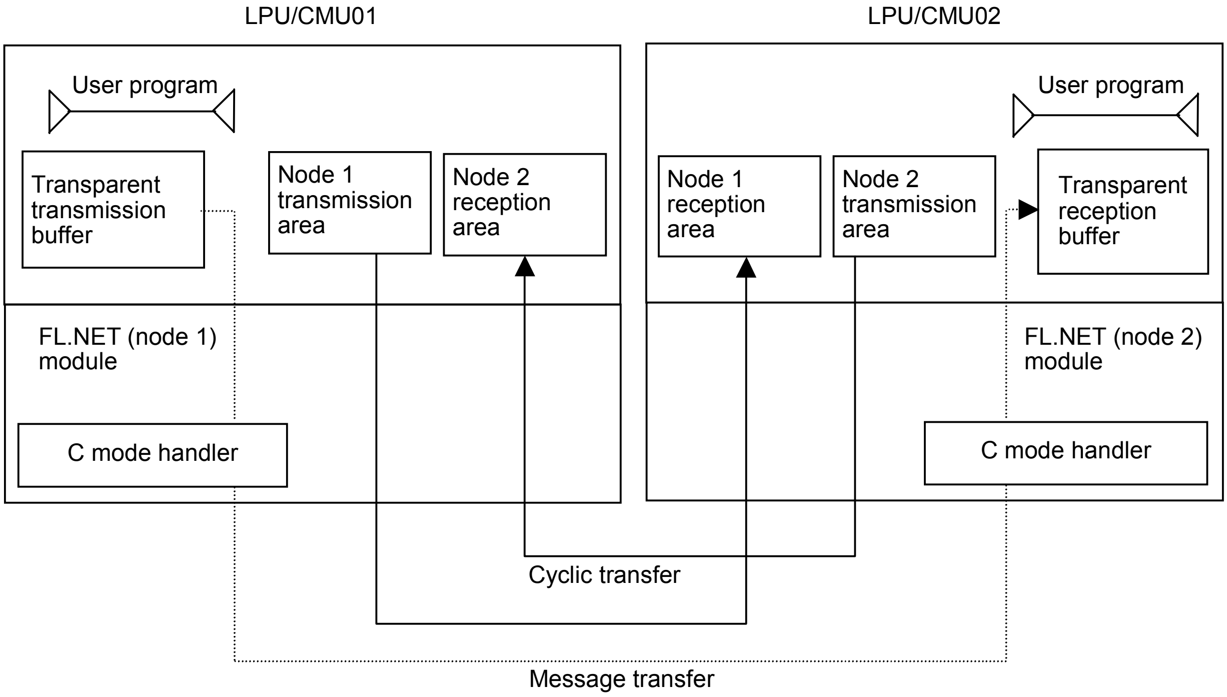
Setup item	LPU/CMU01 setting		LPU/CMU02 setting	
	Node 1	Node 2	Node 1	Node 2
Area 1 address	No setting is required because node 1 is the self-node.	RW040	RW000	No setting is required because node 2 is the self-node.
Area 1 words		0x004	0x004	
Area 2 address		DW040	DW000	
Area 2 words		0x0040	0x0040	
FA link status		—	—	
Higher layer status		—	—	

[Program structure]

The program structure is shown below. With the module having the node number 1 (FL.NET module for LPU/CMU01) and the module having the node number 2 (FL.NET module for LPU/CMU02) interconnected, cyclic transfer operations are performed between the FL-net modules. (The FL.NET modules perform cyclic transfer operations. The user does not have to pay attention to the operations.)

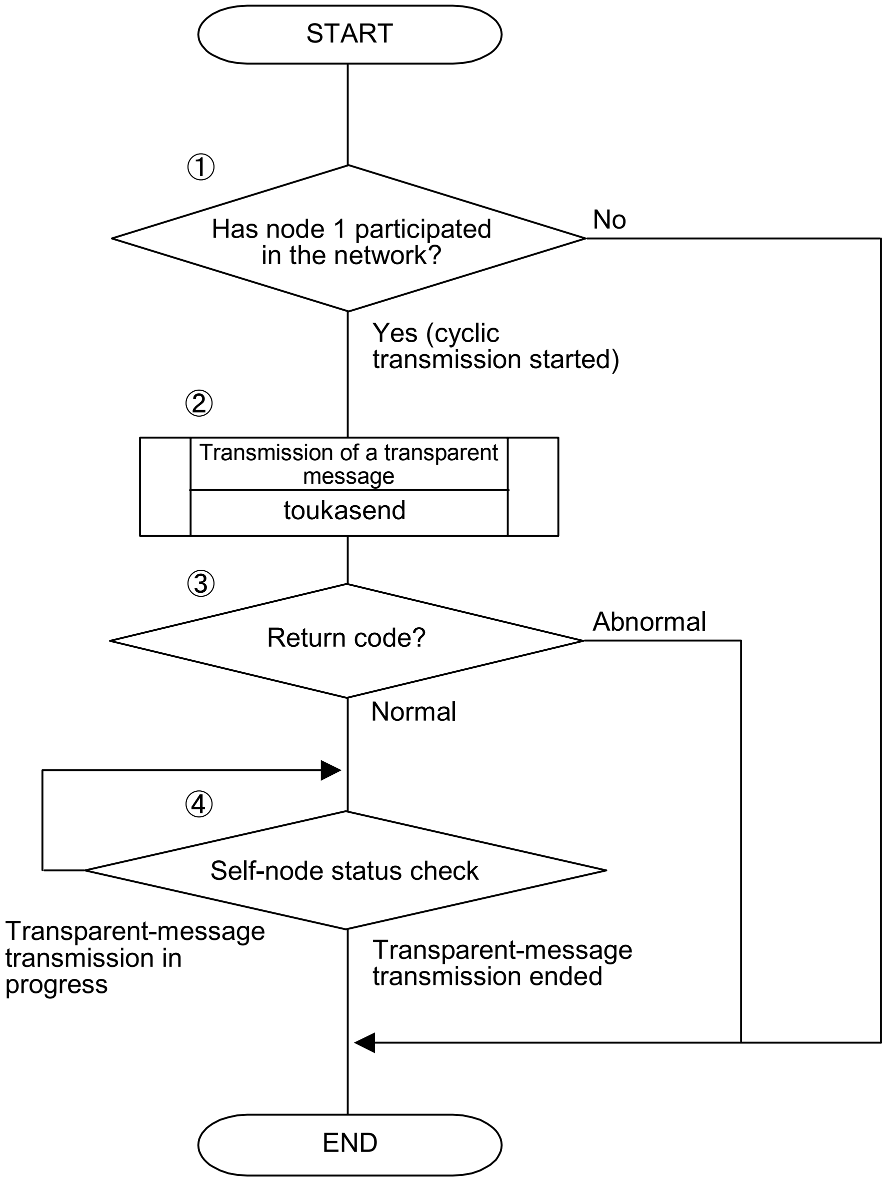
Subsequently, the node 1 module transmits a 1,024-byte transparent message (TCD number = 11000) to the node 2 module. The node 2 module receives the transparent message from node 1. This program example operates in this manner.

When running this user program, be sure to start it from LPU/CMU01.



Item \ LPU/CMU		LPU/CMU01	LPU/CMU02
Function		Transparent message transmission.	Transparent message reception.
Transmission buffer	Address	0x450000	——
	Number of bytes	1,024	——
Reception buffer	Address	——	0x450000
	Number of bytes	——	1,024
C mode handler starting address	toukaread( )	0xD60160	0xD60160
	toukasend( )	0xD60180	0xD60180

[LPU/CMU01-side program flowchart]


START
①
Has node 1 participated in the network?
No
Yes (cyclic transmission started)
②
Transmission of a transparent message
toukasend
③
Return code?
Abnormal
Normal
④
Self-node status check
Transparent-message transmission in progress
Transparent-message transmission ended
END

① The self-node status transfer area is checked. When the contents of the checked area are 0x0080, the system concludes that the self-node has participated in the network.

② A request for the transmission of a transparent message is issued to the node 2 module.

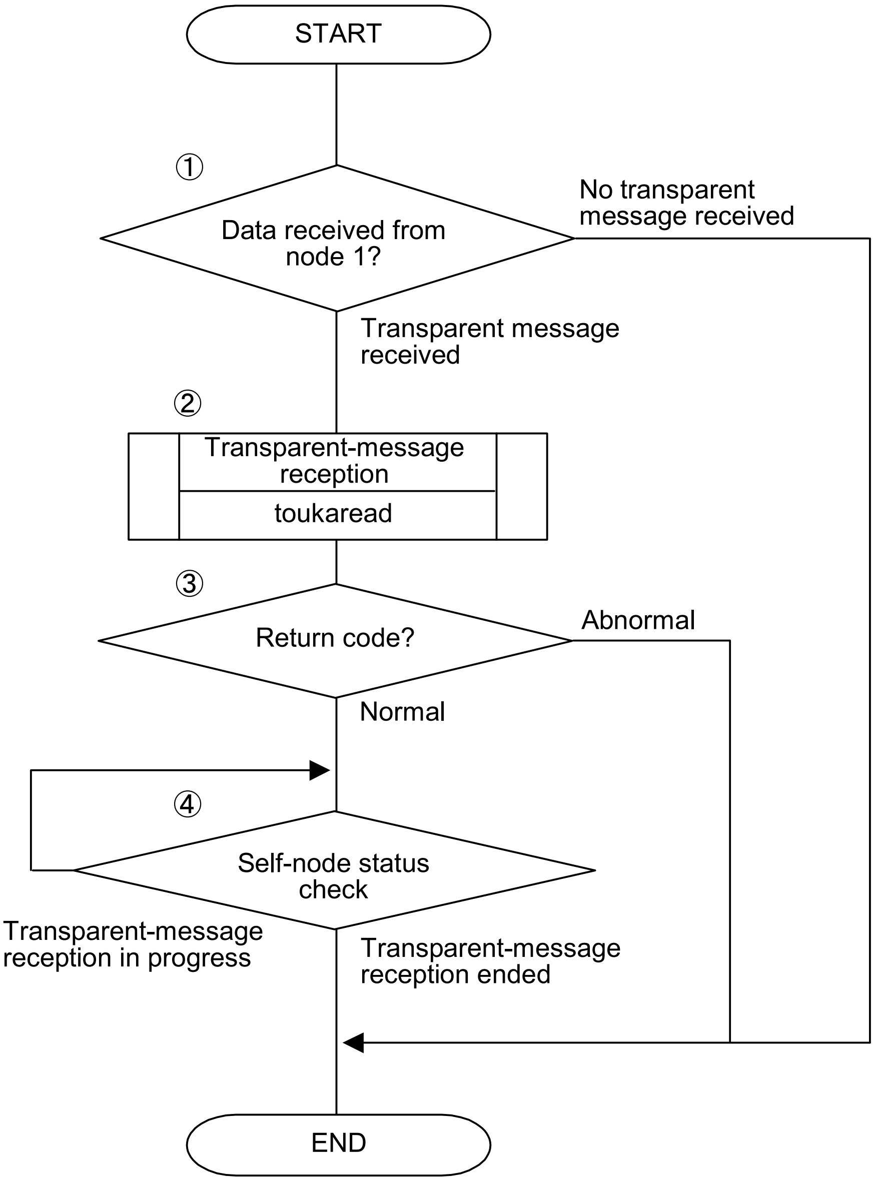
③ The C mode handler checks a return code to determine whether the request is accepted normally. (The request is accepted normally if the return code is 0 or not accepted normally if the return code is -1.)

④ The self-node status transfer area is checked to wait until bit 2^{15} (0x8000) turns OFF. (Bit 2^{15} of the self-node status flag is the “user request processing in progress” bit.)

[LPU/CMU01-side C program example]

```
#define TOUKA_SEND    0xD60180L    /* toukasend() starting address (main) */
#define SBUFADR       0x450000L    /* Transmission buffer address */
#define PARADDR       0x452000L    /* Input parameter storage starting address */
#define RW080         0x0E0C10L    /* Self-node status transfer area (RW080) */
#define RW090         0x0E0C12L    /* Transparent-message transmission error code area */

struct ToukaSend_p {
    long            node;          /* Transmission destination node number */
    unsigned short  *Erradr;       /* Error code storage address */
    unsigned char   *dataadr;      /* Transmission data storage starting address */
    unsigned long   datasiz;       /* Transmission data byte count */
    unsigned long   TcdNo;         /* Transparent-message TCD number */
};

/******************************************/
/* task2:Transmission(LPU/CMU01) */
/******************************************/

main()
{
    register long   ( *toukasend )();
    long            rtn;
    struct          ToukaSend_p     *send;
    unsigned short  *nodeflg;

    nodeflg   = ( unsigned short *)RW080;
    toukasend = ( long(*)( ))TOUKA_SEND;
    send      = ( struct ToukaSend_p *)PARADDR;
    if( !( *nodeflg & 0x0080)){               /* Self-node status check */
        return;
    }

    send->node    =  0x00000002;              /* Transmission destination node number */
    send->Erradr  = (unsigned short*)RW090;  /* Error code storage address */
    send->dataadr = (unsigned char*)SBUFADR  /* Transmission data storage starting address */
    send->datasiz = 1024;                     /* Transmission data byte count */
    send->TcdNo   = 11000;                    /* Transparent-message TCD number */
    rtn = ( toukasend )( send );              /* Transparent-message transmission */

    if( rtn != 0){                            /* Return code check   */
        return;
    }

    while( 1 ){
        if( !( *nodeflg & 0x8000)){          /* Wait for message processing termination */
            break;
        }
    }
}
```

[LPU/CMU02 side program flowchart]


START
①
Data received from node 1?
No transparent message received
Transparent message received
②
Transparent-message reception
toukaread
③
Return code?
Abnormal
Normal
④
Self-node status check
Transparent-message reception in progress
Transparent-message reception ended
END

① The transparent-type reception flag area is checked. When the contents of the checked area are 0x4000 and bit 2^{14} within the self-node status transfer area is ON (set), the system concludes that there is a transparent message from node 1. (Starting address bit 2^{14} of the transparent-type reception flag is the node 1 reception bit.)

② A request for the reception of a transparent message is issued.

③ The C mode handler checks the return code to determine whether the request is accepted normally. (The request is accepted normally if the return code is 0 or not accepted normally if the return code is -1.)

④ The self-node status transfer area is checked to wait until bit 2^{15} (0x8000) is turned OFF (reset). (Bit 2^{15} of the self-node status flag is the “user request processing in progress” bit.)

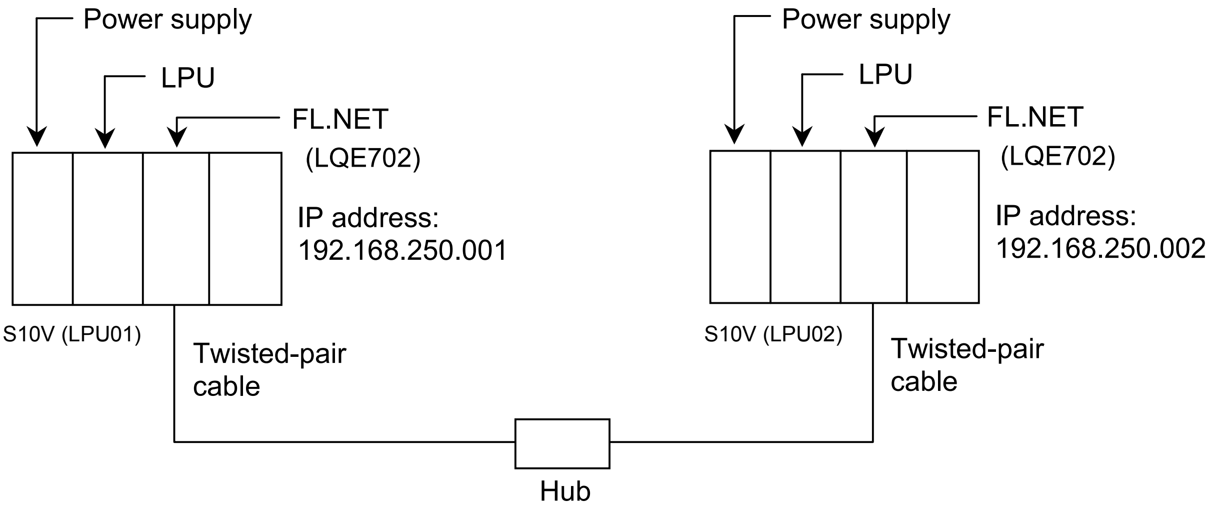
[LPU/CMU02-side C program example]

```
#define TOUKA_READ    0xD60160L    /* toukaread() starting address (main) */
#define RBUFADR       0x450000L    /* Reception buffer address */
#define PARADDR       0x452000L    /* Input parameter storage starting address */
#define RW080         0x0E0C10L    /* Self-node status transfer area (RW080) */
#define RW090         0x0E0C12L    /* Transparent-message transmission error code area */
#define RW100         0x0E0C20L    /* Transparent receiving flag area */

struct ToukaRead_p {
    long            node;          /* Transmission destination node number */
    unsigned short  *Erradr;       /* Error code storage address */
    unsigned char   *dataadr;      /* Reception data storage starting address */
    unsigned long   datasiz;       /* Received data byte count */
};

/****************************************/
/* task3:Reception (LPU/CMU02) */
/****************************************/

main()
{
    register long   ( *toukaread )();
    long            rtn;
    struct          ToukaRead_p     *read;
    unsigned short  *nodeflg, *recvarea;

    recvarea  = ( unsigned short *)RW100;        /* Transparent receiving flag area */
    nodeflg   = ( unsigned short *)RW080;
    toukaread = ( long(*)( ))TOUKA_READ;
    read      = ( struct ToukaRead_p *)PARADDR;
    if( ( *recvarea != 0x4000) ||                /* Transparent receiving flag area check */
       !( *nodeflg & 0x4000)) {                  /* Self-node status check */
        return;
    }

    send->node    =  0x00000001;                /* Transmission soure node number */
    send->Erradr  = (unsigned short*)RW090;     /* Error code storage address */
    send->dataadr = (unsigned char*)RBUFADR     /* Reception data storage starting address */
    send->datasiz = 1024;                       /* Received data byte count */
    rtn = ( toukaread )( read );                /* Transparent-message reception */

    if( rtn != 0){                              /* Return code check */
        return;
    }

    while( 1 ){
        if( !( *nodeflg & 0x8000)){            /* Wait for message processing termination */
            break;
        }
    }
}
```

Note: The user program examples presented on the previous page are created to facilitate understanding. In a practical program, it is necessary to check an error code that is written to an error code storage address after a "user request processing in progress" bit is turned OFF (reset) subsequently to message issuance.

(b) Transmission/reception by mathematical/logical function

[System configuration]



Ensure that the FL.NET module MAIN/SUB selector switches for LPU01 and LPU02 are set to 0.

System component list

Product name	Model	Quantity	Remarks
Power supply	LQV000	2	
LPU	LQP510	2	
FL.NET	LQE702	2	
Mount base	HSC-1540, 1580	2	Select a model among those shown at left.
Twisted-pair cable	HUTP-CAT5E-4P	2	Manufacturer: Hitachi Cable, Ltd.

[FL.NET module settings]

The FL.NET module settings for LPU01 and LPU02 are shown below:

Self-node settings for the FL.NET modules

Setup item	LPU01 setting	LPU02 setting
Node number	1	2
Area 1 address (setting)	0x000	0x004
Area 1 words	0x004	0x004
Area 1 address (PCs allocation)	RW000	RW040
Area 2 address (setting)	0x0000	0x0040
Area 2 words	0x0040	0x0040
Area 2 address (PCs allocation)	DW000	DW040
Self-node status (PCs allocation)	RW080	RW080
Transparent reception task	———	———
Transparent reception task's factor	———	———
Transparent receiving flag	———	RW100

Remote node settings for FL.NET modules

Setup item	LPU01 setting		LPU02 setting	
	Node 1	Node 2	Node 1	Node 2
Area 1 address	No setting is required because node 1 is the self-node.	RW040	RW000	No setting is required because node 2 is the self-node.
Area 1 words		0x004	0x004	
Area 2 address		DW040	DW000	
Area 2 words		0x0040	0x0040	
FA link status		———	———	
Higher layer status		———	———	

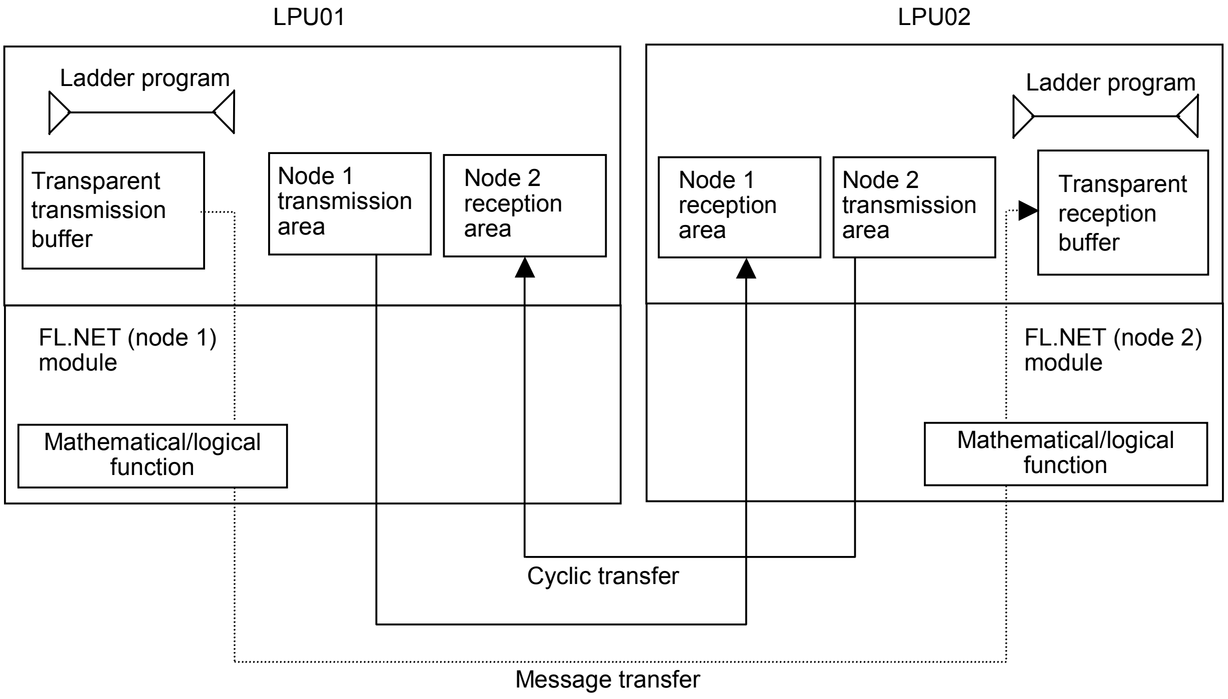
[Program structure]

The program structure is shown below. With the module having the node number 1 (FL.NET module for LPU01) and the module having the node number 2 (FL.NET module for LPU02) interconnected, cyclic transfer operations are performed between the FL.NET modules. (The FL.NET modules perform cyclic transfer operations. The user does not have to pay attention to the operations.)

When R0A0 startup is detected in LPU01 side ladder program subsequently, the node 1 module transmits a 1,024-byte transparent message (TCD number = 11000) to the node 2 module. The node 2 module receives the transparent message from node 1 module.

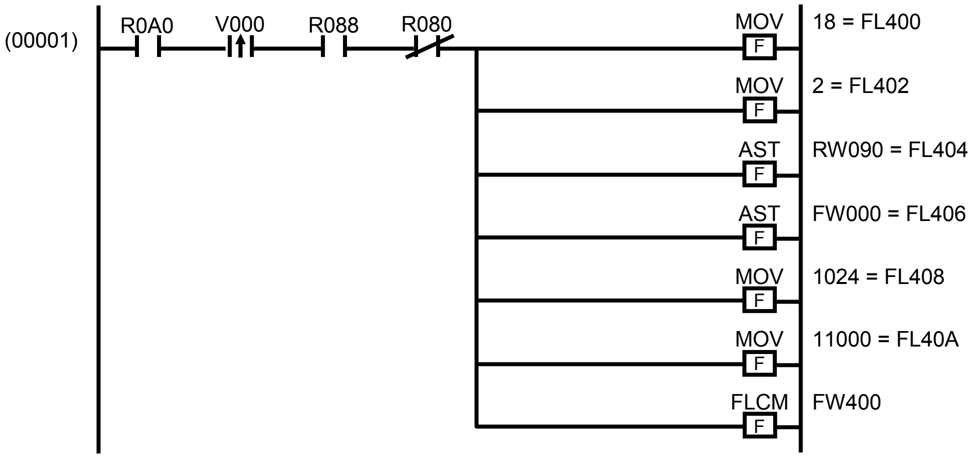
This program example is a ladder program, which runs in this manner.

When using this ladder program, be sure to place the S10V in the RUN state.



Item \ LPU		LPU01	LPU02
Function		Transmits transparent messages.	Receives transparent messages.
Send buffer	Address	FW000 to FW1FF	——
	Number of bytes	1,024 bytes	——
Receive buffer	Address	——	FW000 to FW203
	Number of bytes	——	1,024 bytes + 4 words
Mathematical/logical function	FLCM	0xD60000	0xD60000

[LPU01 side ladder program]



When the R0A0 turns ON, the program runs and checks the network participation status bit (R088) and user-requested-processing-in-progress bit (R080) in the self-node status flag (message processing can be performed when the network participation status bit is set with the user-requested-processing-in-progress bit reset.)
The parameters for transparent-message transmission are specified by mathematical/logical functions.

- MOV 18 = FL400
  The message transmission service number is set to 18. A transparent-message transmission request is specified.

- MOV 2 = FL402
  The remote station (LPU02) is specified.

- AST RW090 = FL404
  The error code storage address (RW090 real address) is specified.

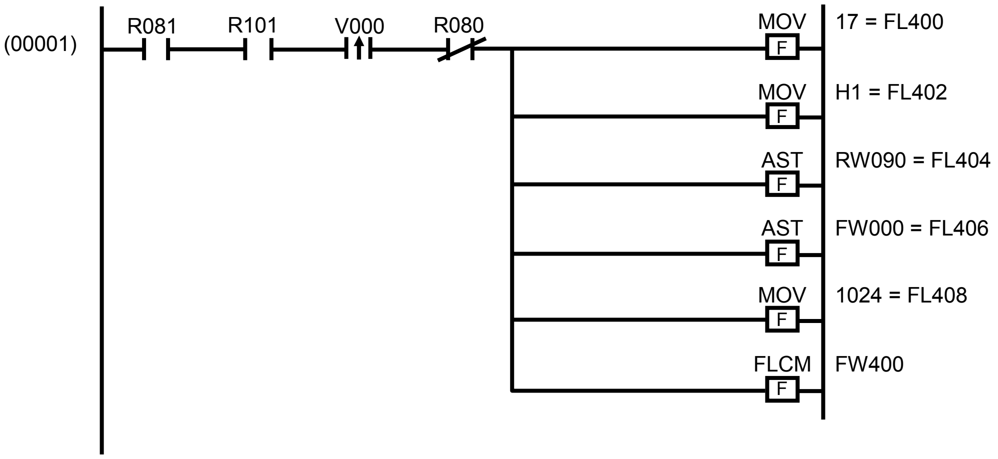
- AST FW000 = FL406
  The transmission buffer address (FW000 real address) is specified.

- MOV 1024 = FL408
  The transmission word count is specified.

- MOV 11000 = FL40A
  The TCD number (11000) is specified for the transparent message to be transmitted.

- FLCM FW400
  The parameter storage address (FW400 real address) is specified for the mathematical/logical function (FLCM). A message request (transparent-message transmission) is issued to the FL.NET module.

[LPU02-side ladder program]



The program runs when the transparent-message reception bit (R081) in the self-node status flag is set and the node 1 reception bit (R101) in the transparent-type reception flag area is also set. The program then checks the user-requested-processing-in-progress bit (R080) in the self-node status flag (message processing can be performed when the user-requested-processing-in-progress bit is reset.)
The parameters for transparent-message transmission are specified by mathematical/logical functions.

- MOV 17 = FL400
  The message transmission service number is set to 17. A transparent-message reception request is specified.

- MOV H1 = FL402
  A transparent message from the remote station (LPU01) is specified.

- AST RW090 = FL404
  The error code storage address (RW090 real address) is specified.

- AST FW000 = FL406
  The reception buffer address (FW000 real address) is specified.

- MOV 1024 = FL408
  The reception word count is specified.

- FLCM FW400
  The parameter storage address (FW400 real address) is specified for the mathematical/logical function (FLCM). A message request (transparent-message transmission) is issued to the FL.NET module.

Note: The ladder program examples presented here are created to facilitate understanding. In a practical program, it is necessary to check an error code that is written at an error code storage address after a “user request processing in progress” bit is reset subsequently to message issuance.

5.3.6 Using the management tables

The FL.NET module uses various management tables to manage the status of communications with remote nodes.

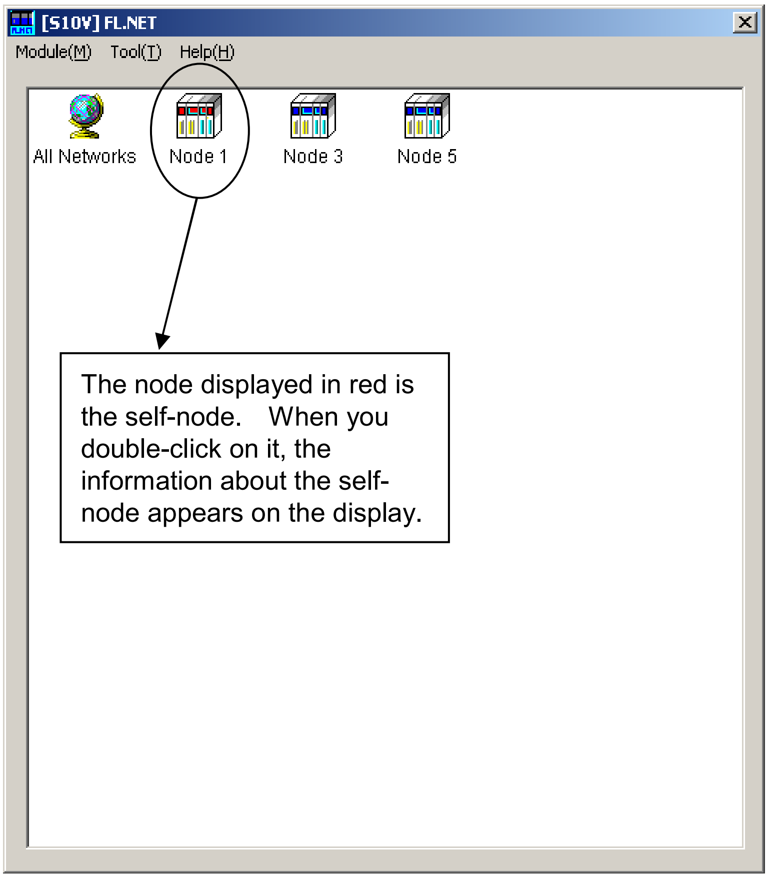
You can determine the status of communications with the remote nodes by referencing these management tables.

Note, however, that the setup tool named [S10V FL.NET SYSTEM] must be used to reference the management tables within the FL.NET module.

For details on the procedures to be performed from various windows, See "5.5 Operating Method."

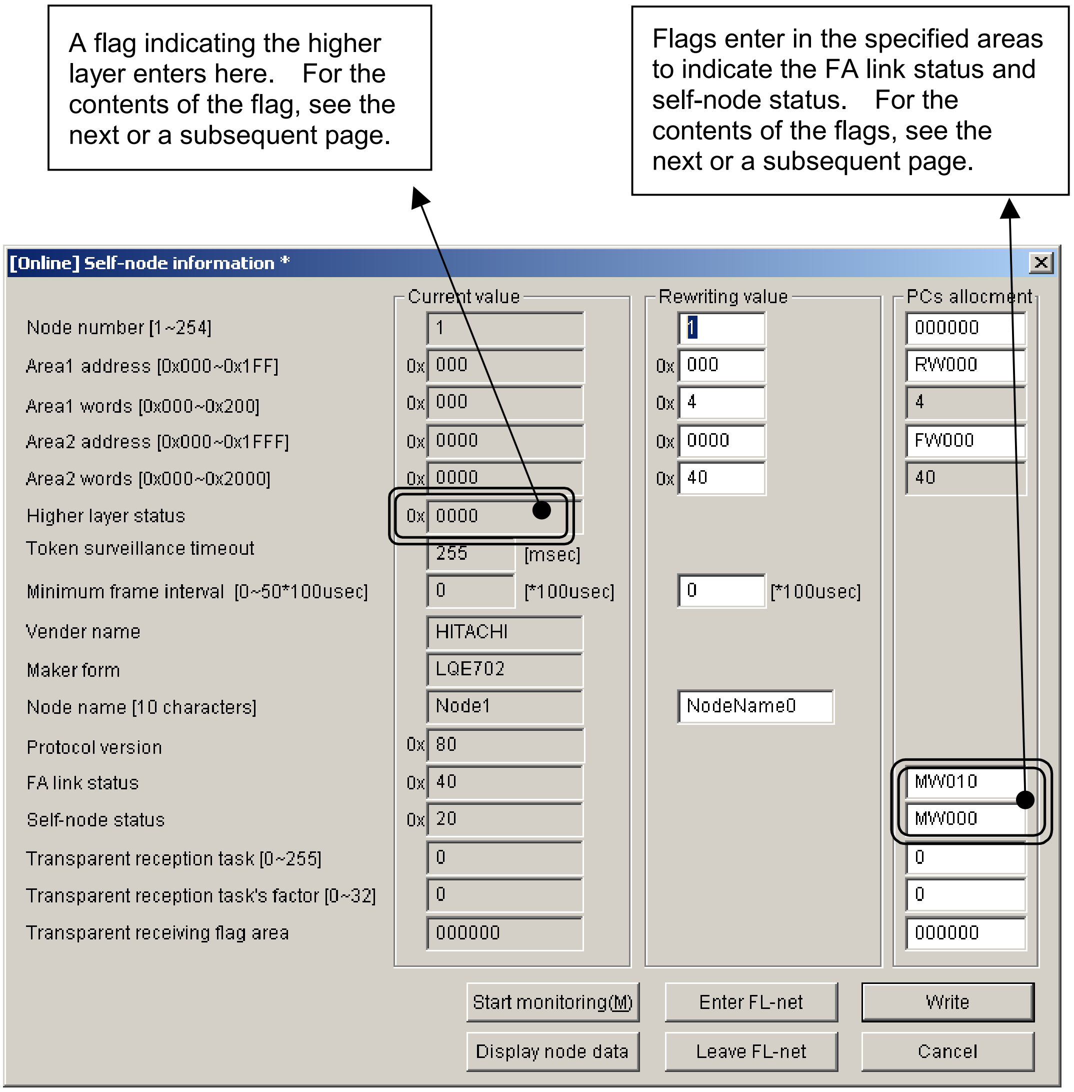
(1) Referencing the self-node management table

To view the self-node management table, use the setup tool [S10V FL.NET SYSTEM] to display the information about participating nodes. When a list of the numbers of the nodes that have participated in the network appears on the display, select the on-window icon for the self-node.



The node displayed in red is the self-node. When you double-click on it, the information about the self-node appears on the display.

Figure 5-40 Participating Node Number List Window 1



A flag indicating the higher layer enters here. For the contents of the flag, see the next or a subsequent page.

Flags enter in the specified areas to indicate the FA link status and self-node status. For the contents of the flags, see the next or a subsequent page.

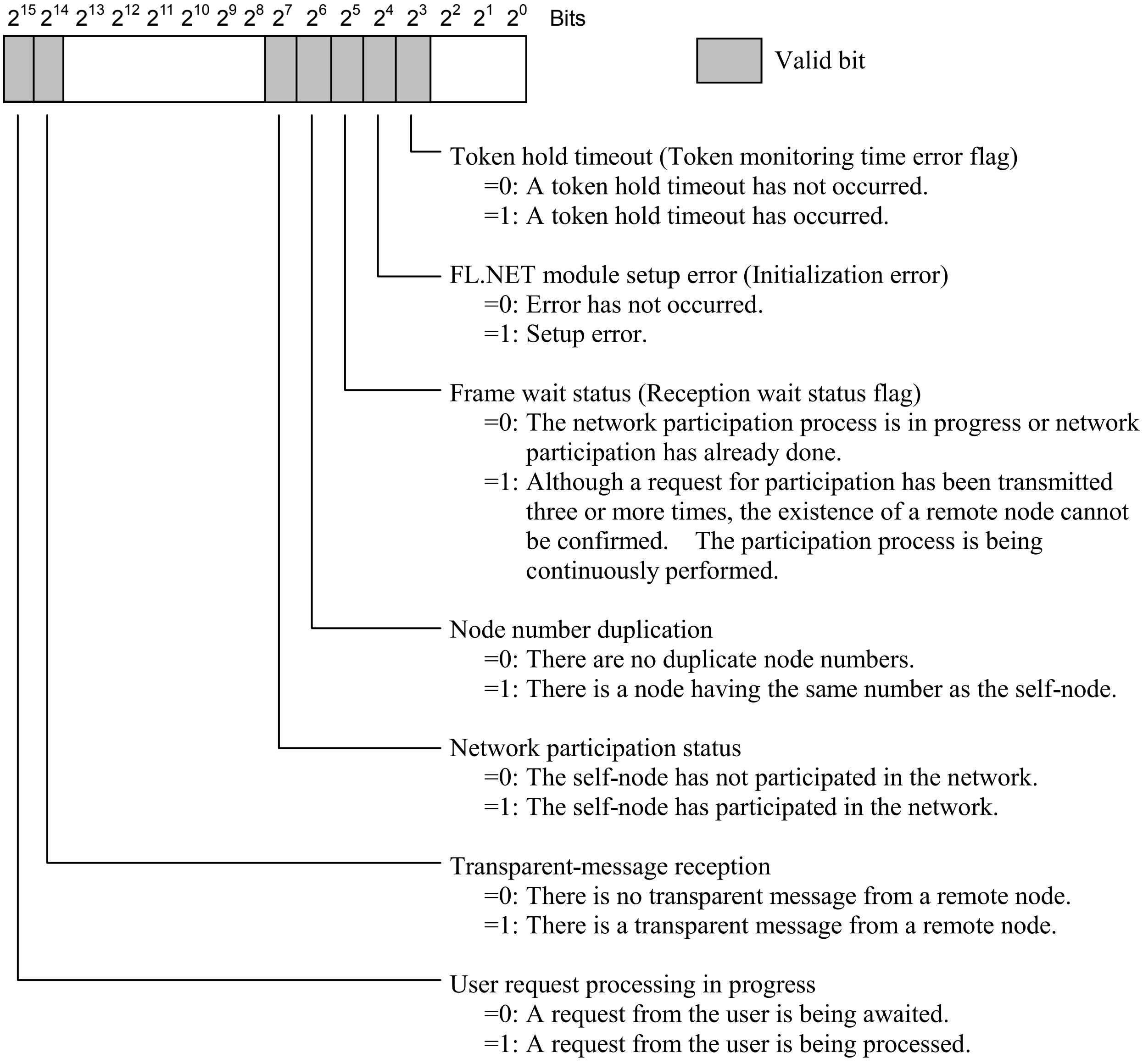
[Online] Self-node information *

	Current value	Rewriting value	PCs allocment
Node number [1~254]	1	1	000000
Area1 address [0x000~0x1FF]	0x 000	0x 000	RW000
Area1 words [0x000~0x200]	0x 000	0x 4	4
Area2 address [0x000~0x1FFF]	0x 0000	0x 0000	FW000
Area2 words [0x000~0x2000]	0x 0000	0x 40	40
Higher layer status	0x 0000		
Token surveillance timeout	255 [msec]		
Minimum frame interval [0~50*100usec]	0 [*100usec]	0 [*100usec]	
Vender name	HITACHI		
Maker form	LQE702		
Node name [10 characters]	Node1	NodeName0	
Protocol version	0x 80		
FA link status	0x 40		MW010
Self-node status	0x 20		MW000
Transparent reception task [0~255]	0		0
Transparent reception task's factor [0~32]	0		0
Transparent receiving flag area	000000		000000

Figure 5-41 Self-node Information (Self-node Management Table) Window

[Contents of self-node status flag]

In the FL.NET module, the bit allocation for the self-node status flag is as indicated below:



2^{15} 2^{14} 2^{13} 2^{12} 2^{11} 2^{10} 2^9 2^8 2^7 2^6 2^5 2^4 2^3 2^2 2^1 2^0 Bits
Valid bit

Token hold timeout (Token monitoring time error flag)
=0: A token hold timeout has not occurred.
=1: A token hold timeout has occurred.

FL.NET module setup error (Initialization error)
=0: Error has not occurred.
=1: Setup error.

Frame wait status (Reception wait status flag)
=0: The network participation process is in progress or network participation has already done.
=1: Although a request for participation has been transmitted three or more times, the existence of a remote node cannot be confirmed. The participation process is being continuously performed.

Node number duplication
=0: There are no duplicate node numbers.
=1: There is a node having the same number as the self-node.

Network participation status
=0: The self-node has not participated in the network.
=1: The self-node has participated in the network.

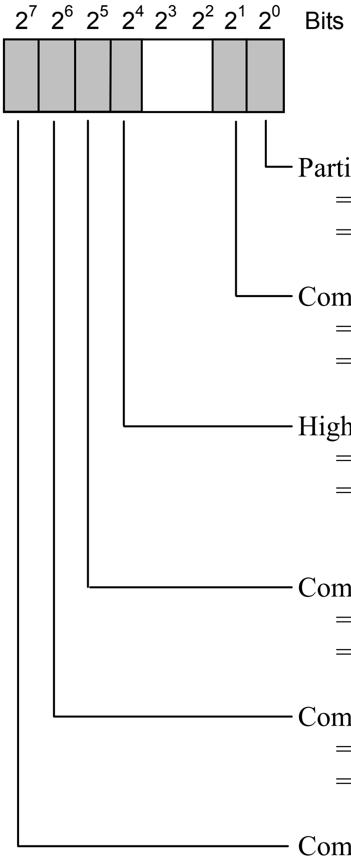
Transparent-message reception
=0: There is no transparent message from a remote node.
=1: There is a transparent message from a remote node.

User request processing in progress
=0: A request from the user is being awaited.
=1: A request from the user is being processed.

Note: The above self-node status represents the contents that are transferred to the S10V. The self-node information dialog box of the setup tool [S10V FL.NET SYSTEM] allows you to reference the low-order byte (bits 2^7 through 2^0) only.

[Contents of FA link status flag (FA link status)]

The bit allocation for the FA link status flag is as indicated below:



▨ Valid bit

Participation status bit (Participation leaving flag)
=0: The self-node has not participated in the network.
=1: The self-node has participated in the network.

Communication invalidity detection bit
=0: No detection
=1: Detection

Higher layer bit
=0: The higher layer is normal.
=1: The higher layer is abnormal. For details, refer to the higher layer status flag.

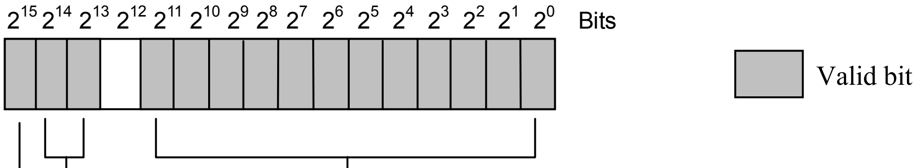
Common memory data validity bit
=0: The common memory data is invalid.
=1: The common memory data is valid.

Common memory setup completion bit
=0: Common memory setup is not complete (an error exists).
=1: Common memory setup is completed (no error exists).

Common memory duplication bit (Address duplication error flag)
=0: There is no duplicate common memory setting.
=1: There is a duplicate common memory setting for a remote node. All the common memory settings for the self-node will be changed to 0 to participate in the network.

[Contents of higher-layer status flag]

The bit allocation for the higher-layer status flag is as indicated below:


2^15 2^14 2^13 2^12 2^11 2^10 2^9 2^8 2^7 2^6 2^5 2^4 2^3 2^2 2^1 2^0 Bits
Valid bit

Higher-layer error code (12-bit)
The code for an error existing in the higher layer is displayed. The following table shows the error codes for the FL.NET module.

Error code	Description
0x000	The higher layer (S10V) is normal.
0x0FF	The LPU down unit in the higher layer (S10V) is inoperative.

Higher-layer error status bits (2 bits)
These bits describe a higher-layer error when it occurs.

Bit 2^{14}	Bit 2^{13}	Description
OFF	OFF	The higher layer is normal.
OFF	ON	An error exists in the higher layer but permits the continuation of operation (ALARM state). The cyclic data and message data are guaranteed.
ON	OFF	An error exists in the higher layer and does not permit the continuation of operation (WARNING state). The cyclic data and message data are not guaranteed.
ON	ON	

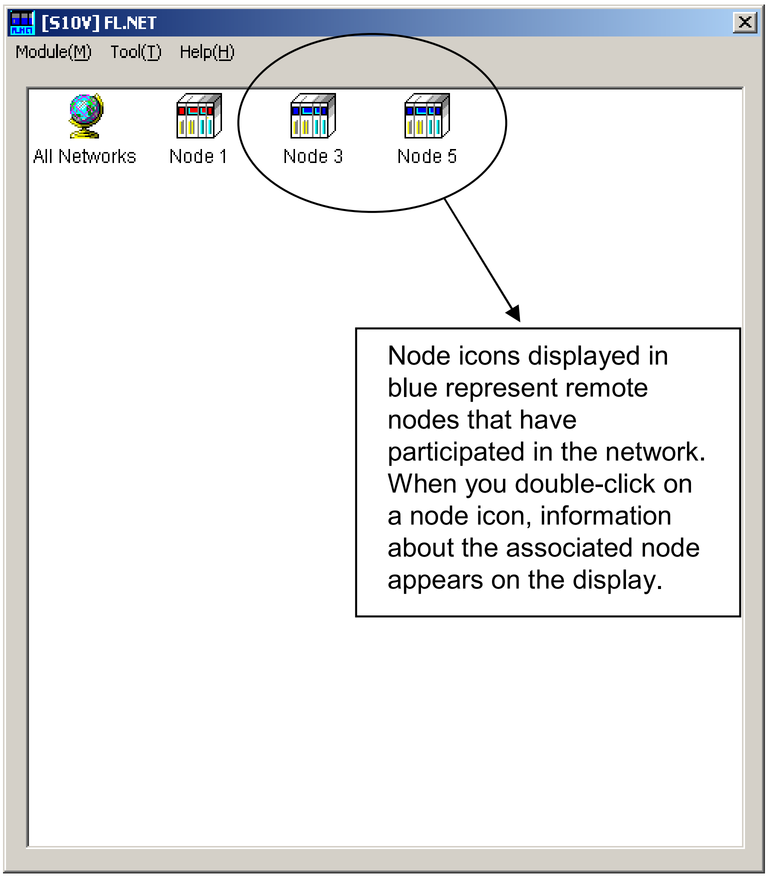
When a higher-layer error occurs, the FL.NET module turns ON (sets) bits 2^{14} and 2^{13}.

Higher-layer operation status bit (RUN/STOP)
=0: The higher layer (S10V) is halted (in the STOP state).
=1: The higher layer (S10V) is running (in the RUN state).

(2) Referencing the participating node management table

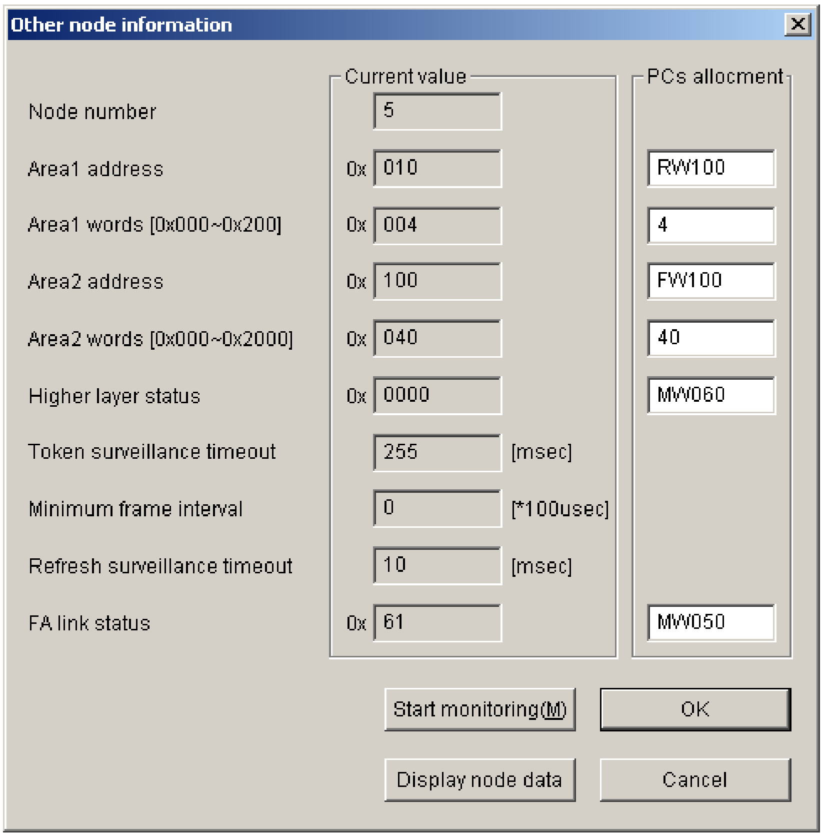
When you display information about participating nodes with the setup tool [S10V FL.NET SYSTEM], you can view the [Other node information] dialog box, which lists the node numbers of the nodes that have participated in the network and the participating node management table contents concerning the participating node numbers.

Note, however, that the setup tool [S10V FL.NET SYSTEM] merely allows you to view the contents of the participating node management table concerning the nodes that have participated in the network.



Node icons displayed in blue represent remote nodes that have participated in the network. When you double-click on a node icon, information about the associated node appears on the display.

Figure 5-42 Participating Node Number List Window 2



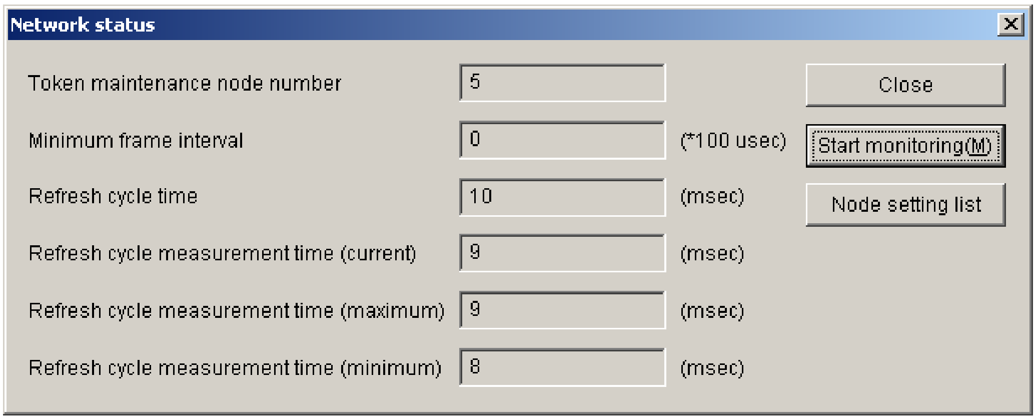
Other node information

	Current value		PCs allocment
Node number	5		
Area1 address	0x 010		RW100
Area1 words [0x000~0x200]	0x 004		4
Area2 address	0x 100		FW100
Area2 words [0x000~0x2000]	0x 040		40
Higher layer status	0x 0000		MW060
Token surveillance timeout	255	[msec]	
Minimum frame interval	0	[*100usec]	
Refresh surveillance timeout	10	[msec]	
FA link status	0x 61		MW050

Figure 5-43 Participating Other Node Link Information Window

For the bit allocation for the higher layer status and FA link status about individual nodes, refer to the respective node manuals.

(3) Referencing the network management table
The network status window of the setup tool [S10V FL.NET SYSTEM] allows you to view the contents of the network management table.



Network status

Token maintenance node number	5	
Minimum frame interval	0	(*100 usec)
Refresh cycle time	10	(msec)
Refresh cycle measurement time (current)	9	(msec)
Refresh cycle measurement time (maximum)	9	(msec)
Refresh cycle measurement time (minimum)	8	(msec)

Figure 5-44 Network Status (Network Management Table) Window

The contents of the above window are explained below:

On-screen item	Description
Token maintenance node number	This is the node number of the node that currently holds the token.
Minimum frame interval	This value denotes the minimum frame interval that is allowed on the network. It represents the maximum value of the nodes that compose the network.
Refresh cycle time	This value is 1.2 times the time required for a token circulation through the network.
Refresh cycle measurement time (current)	These values are the current value, maximum value, or minimum value of the measured time required for a token circulation through the network.
Refresh cycle measurement time (maximum)	
Refresh cycle measurement time (minimum)	

5.3.7 FL.NET module communication performance

The method of calculating the communication performance of the FL-NET module is shown below. However, the method varies depending on the destination of the connection and number of words. Accordingly, the worst case scenario is shown below on the supposition of communication performance between FL-NET modules.
The meanings of symbols used in the expression are shown below.

S: Number of transmitted words of the self-node (words)
R: Number of received words sent from the previous node to the self-node (words)
F: Quotient resulting from dividing (S - 1) by 512
(Example: F = 0 when S = 512, F = 1 when S = 513)
Fn: F of each node. n is the node number.
RCT: Refresh cycle time (ms)
SCT: Sequence cycle time (ms)
MS: Transmitted message word count (bytes)
MR: Received message word count (bytes)
MFT: Minimum frame interval (ms)
W: Total number of transmitted words of all nodes (words)
n: Number of connected nodes

- Token hold time (ms): Token holding time of the self-node

Token hold time = 1.1 + 0.001 × S + 0.001 × R + (MFT + 0.2) × F + MFT

- Refresh cycle time (ms): Sum of the token hold time values of all the nodes.

Refresh cycle time = 1.1 × n + 0.002 × W + (MFT + 0.2) × (F1 + F2 + F3 ...... + Fn) + MFT × n

- Cyclic transfer throughput (ms)

  Time from setting data in the memory of the transmitting side CPU till calculating by using calculation by the ladder program of the receiving side CPU.

Cyclic transfer throughput = RCT + SCT .... [For ladder processing]

Cyclic transfer throughput = RCT .... [For C mode processing]

- Word block read throughput (ms)

  Time from starting the operational function for word block reading till transmitting to transmission of ACK to the opposite part from the self-node.

Word block read throughput = 8 + 3RCT + Token holding time

= 9.1 + 0.001 × S + 0.001 × R + (MFT + 0.2) × F + MFT

- Transparent message (ms)

Message creating time (ms) = 0.02 + (0.0007 × MS)............. [When message creating time > MFT]

Message creating time (ms) = MFT .... [When message creating time ≤ MFT]

Message transmit processing time (ms) = (0.00096 × MS)

Message receive processing time (ms) = (0.0013 × MR)

1:1 transmission request processing time = 1.272 + 2RCT + Token holding time of the self-node + Message creating time + Message transmit processing time + Message receive processing time

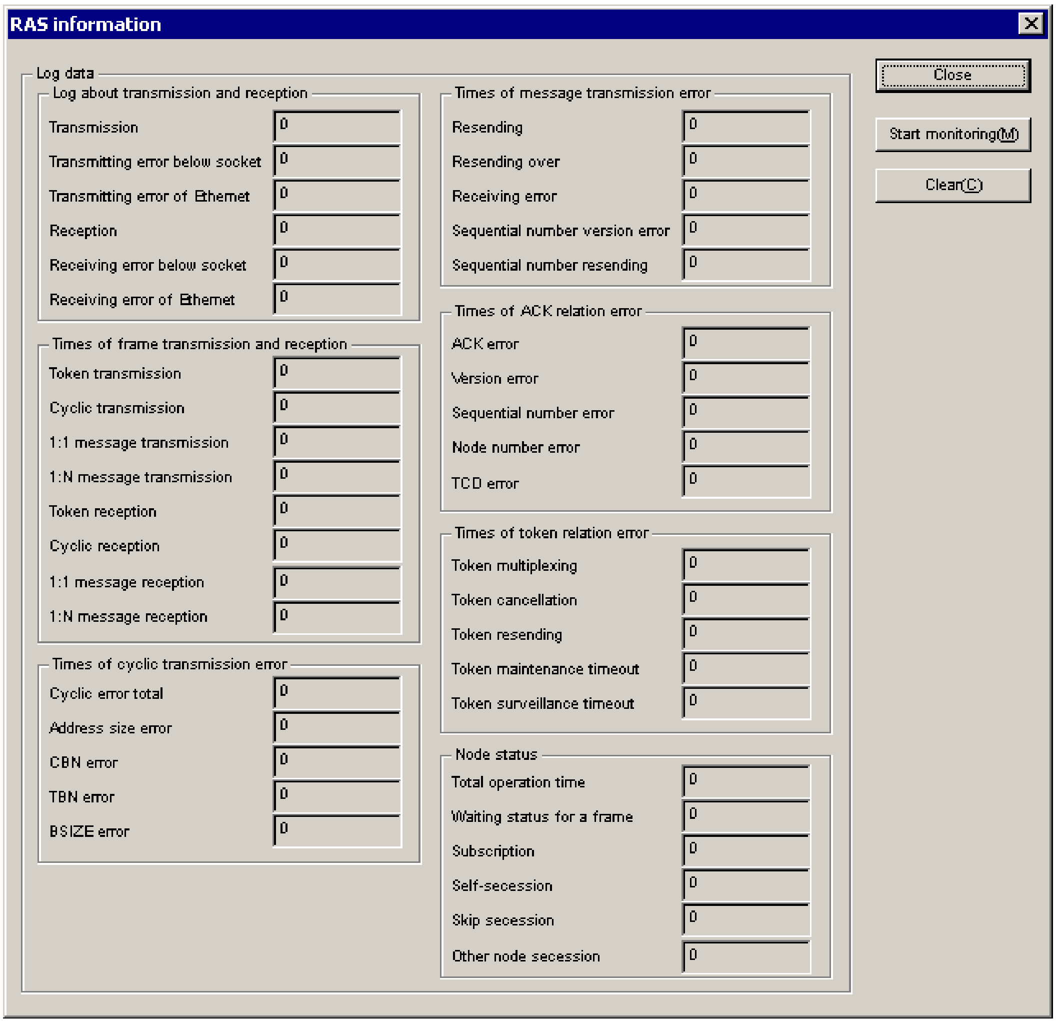
= 3.992 + 2RCT + 1.1 + 0.001× S + 0.001 × R + (MFT + 0.2) × F + MFT + (0.00166 × MS) + (0.0013 × MR)

1:N transmission request processing time = 0.742 + RCT + Message creating time + Message transmit processing time

= 0.762 + RCT + (0.00166 × MS)

5.3.8 Using the communication log

With the setup tool [S10V FL.NET SYSTEM], you can view the RAS information (communication log) maintained in the FL.NET module.



RAS information

Log data

Log about transmission and reception	
Transmission	0
Transmitting error below socket	0
Transmitting error of Ethernet	0
Reception	0
Receiving error below socket	0
Receiving error of Ethernet	0

Times of frame transmission and reception	
Token transmission	0
Cyclic transmission	0
1:1 message transmission	0
1:N message transmission	0
Token reception	0
Cyclic reception	0
1:1 message reception	0
1:N message reception	0

Times of cyclic transmission error	
Cyclic error total	0
Address size error	0
CBN error	0
TBN error	0
BSIZE error	0

Times of message transmission error	
Resending	0
Resending over	0
Receiving error	0
Sequential number version error	0
Sequential number resending	0

Times of ACK relation error	
ACK error	0
Version error	0
Sequential number error	0
Node number error	0
TCD error	0

Times of token relation error	
Token multiplexing	0
Token cancellation	0
Token resending	0
Token maintenance timeout	0
Token surveillance timeout	0

Node status	
Total operation time	0
Waiting status for a frame	0
Subscription	0
Self-secession	0
Skip secession	0
Other node secession	0

Figure 5-45 [RAS Information] Dialog Box

The table below describes the on-screen log information (RAS information).

Item		On-screen information description
Log about transmission and reception	Transmission	Number of frames whose transmissions were requested
	Transmitting error below socket	Number of transmission errors in the socket section
	Transmitting error of Ethernet	Unused
	Reception	Number of frames received from the socket section
	Receiving error below socket	Number of reception errors and abnormal frame receptions in the socket section
	Receiving error of Ethernet	Unused
Times of frame transmission and reception	Token transmission	Number of token transmissions
	Cyclic transmission	Number of cyclic frame transmissions without a token
	One-to-one message transmission	Number of one-to-one message transmissions
	One-to-N message transmission	Number of one-to-N message transmissions
	Token reception	Number of token receptions
	Cyclic reception	Number of cyclic frame receptions without a token
	One-to-one message reception	Number of one-to-one message receptions
	One-to-N message reception	Number of one-to-N message receptions
Times of cyclic transmission error	Cyclic error total	Number of errors in cyclic transmission reception
	Address size error	Unused
	CBN error	Number of errors concerning frame arrangement
	TBN error	Number of errors concerning frame segment count
	BSIZE error	Number of errors concerning frame size
Times of message transmission error	Resending	Number of message transmission retries made
	Resending over	Number of message transmission retries exceeded errors
	Receiving error	Number of message transmission reception errors
	Sequential number version error	Number of sequence number/version error recognitions in message receptions
	Sequential number resending	Number of message transmission retry recognitions in message receptions
Times of ACK relation error	ACK error	Number of ACK errors
	Version error	Number of ACK sequence number/version mismatches
	Sequential number error	Number of ACK sequence number mismatches
	Node number error	Unused
	TCD error	Unused
Times of token error	Token multiplexing	Number of recognitions of multiplexed tokens
	Token cancellation	Number of token cancellations
	Token sending	Number of token reissues
	Token maintenance timeout	Number of token hold timeout occurrences
	Token surveillance timeout	Number of token monitoring timeout occurrences
Node status	Total operating time	Unused
	Waiting status for a frame	Number of frame waits due to the lack of remote nodes on the network
	Subscription	Number of subscriptions to the network
	Self secession	Number of occurrences of three consecutive token hold timeouts or departures due to the lack of remote nodes on the network
	Skip secession	Number of departures due to the skip of a token addressed to the self-node
	Other node secession	Number of times other distant nodes that are connected with the FL-NET module via the FL-net network have collectively seceded from that network.

5.4 Installing and Starting Up the System

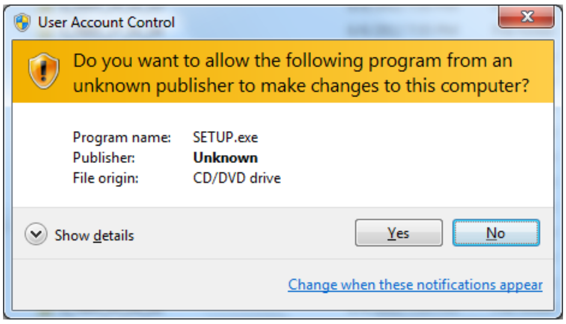
5.4.1 Installing

To install the S10V FL.NET, you must execute the setup program that is stored in the S10V FL.NET DISK1 folder on the CD.
Double-click “setup.exe” that is stored in the DISK1 folder on the S10V FL.NET system CD.
Since no window opens upon completion of installation, attach a shortcut to the desktop as needed.
Click the Start button and choose [(All) Programs] – [Hitachi S10V] – [S10V FL.NET SYSTEM] – [S10V FL.NET SYSTEM] from the [Start] menu on the Windows® screen. Click and hold the right mouse button on the [S10V FL.NET SYSTEM] and move the pointer to the desktop. Then, choose [Copy Here] from the pop-up menu.

NOTICE
● The S10V BASE SYSTEM is required for operating the S10V FL.NET. If it is not installed, you cannot install the S10V FL.NET. ● Before installing the S10V FL.NET, be sure to exit all the currently open Windows®-based programs. Do not forget to exit anti-virus software and other memory-resident programs. If you install the FL.NET without exiting such programs, an error may occur during installation. If such an error occurs, first uninstall the S10V FL.NET as directed in “5.4.2 Uninstalling,” exit all the Windows®-based programs, and then install the S10V FL.NET again.

<Notes on installing in Windows® 7 (32-bit)>

Installing the S10V FL.NET system in Windows® 7 (32-bit) operating system requires prior logging onto the operating system with an appropriate Administrator account, which is the Administrator account first created in the initial condition of your personal computer. When you have so logged on, you can then double-click "setup.exe" that is stored in the DISK 1 folder on the S10V FL.NET System CD. When "setup.exe" is started, the dialog box as shown below will appear. Click the Yes button to continue the execution of the setup program.



User Account Control

Do you want to allow the following program from an unknown publisher to make changes to this computer?

Program name: SETUP.exe
Publisher: **Unknown**
File origin: CD/DVD drive

The S10V FL.NET system cannot be installed on a per-user basis. To install the S10V FL.NET system successfully, the user must first log onto the operating system with an appropriate Administrator account, which is the Administrator account first created in the initial condition of your personal computer.
The S10V FL.NET system may not be installed properly in any of the following cases: 1) administrator permission is acquired by using User Account Control(*) with a standard user account and 2) logon is made with an Administrator account that has been created using User Account Control with a standard user account.
If you make a logon with a user account that is different from the one you have used for the installation of the S10V FL.NET system, the installed program may be missing from the program menu displayed. In this case, you should perform the following series of steps: 1) make a logon again with the Administrator account first created in the initial condition of your personal computer; 2) uninstall the installed program; and 3) install the program again.
When you want to create a new account, be sure to make a logon with an Administrator account. Do not use User Account Control at that time.

(*) User Account Control is a Microsoft Windows feature that temporarily grants administrative rights to standard user accounts.

A message reporting a read-only file detected may be displayed during the reinstallation of the S10V FL.NET system. In this case, click the Yes button to set off overwriting.

5.4.2 Uninstalling

The existing FL.NET System needs to be uninstalled when, for instance, you want to upgrade it. The procedure required for uninstalling it is as follows:

(1) Uninstalling from Windows® 2000
Click on Start button on your Windows desktop and choose [Settings] – [Control Panel]. When the Control Panel opens, double-click on [Add/Remove Programs]. Then, choose "S10V FL.NET SYSTEM" in the [Change or Remove Programs] tab and click the Change/Remove button. When the [Confirm File Deletion] dialog box appears, click the Yes button.

(2) Uninstalling from Windows® XP
Click on Start button on your Windows desktop and choose ([Settings] –)[Control Panel]. When the Control Panel opens, double-click on [Add/Remove Programs]. Then, choose "S10V FL.NET SYSTEM" in the [Change or Remove Programs] tab and click the Change/Remove button. When the [Confirm File Deletion] dialog box appears, click the Yes button.

(3) Uninstalling from Windows® 7 (32-bit)
Click on Start button on your Windows desktop and choose [Control Panel]. When the Control Panel opens, click [Programs and features]. Then, select "S10V FL.NET SYSTEM" and click Uninstall/Change button. When the [Confirm File Deletion] dialog box appears, click the Yes button.

NOTICE

If Windows® opens a window during the uninstall process to display the question "Remove Shared File?", click the No button to retain shared files.

5.4.3 Starting up the system

To start up the FL.NET system, perform the following procedure:

- S10V FL.NET system startup procedure to start it up in online mode

(1) To start up the S10V FL.NET system from the Windows® desk top, double-click the "S10V FL.NET SYSTEM" icon. The S10V FL.NET system can also be started up from the Start button. To accomplish this, choose [(All) Programs] – [Hitachi S10V] – [S10V FL.NET SYSTEM] – [S10V FL.NET SYSTEM] from the Start menu.
The [[S10V] FL.NET] window shown below will then appear. At this stage of the procedure, the FL.NET system is not connected with the PCs yet.

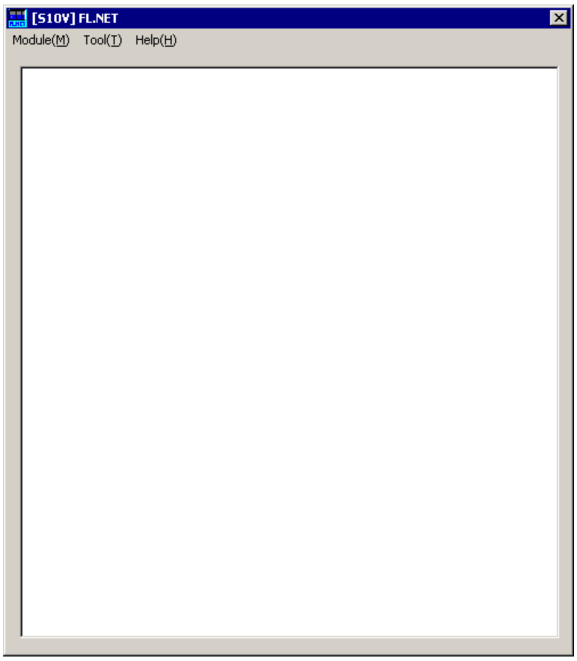


Figure 5-46 [[S10V] FL.NET] Window

(2) By choosing [Change connection] from the [Tool] menu, display the [Connection type] window (Figure 5-47) on-screen. When the [Connection type] window appears, specify the desired destination of connection and click the OK button (see "5.4.4 Changing connections" for details on the communication type). If you need not change the current connection destination setting, click the Cancel button instead.

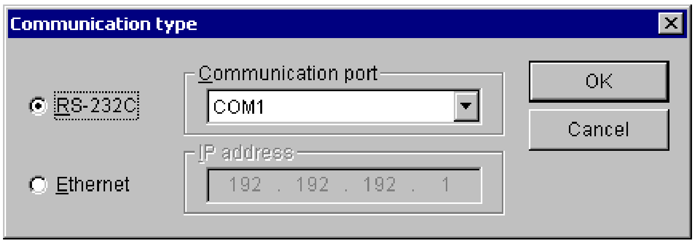


Figure 5-47 The [Communication type] Window

- S10V FL.NET system startup procedure to start it up in offline mode
  The procedure described below enables you to create a setup information file for the FL.NET and edit it, all in offline mode, even if the actual target machine is not present in your user system. To send the setup information file prepared this way to the target machine, first put the FL.NET system into online mode and then send it by choosing [Main] – [Send setups], or [Sub] – [Send setups], from the [Module] menu.

(1) Take the same action as specified in Step (1) under "● S10V FL.NET system startup procedure to start it up in online mode." The FL.NET system's main window will then appear.

(2) Choose [Offline] from the [Module] menu. The file selection dialog box as shown below will then appear. Choose the desired FL.NET setup information file you want to edit in offline mode.

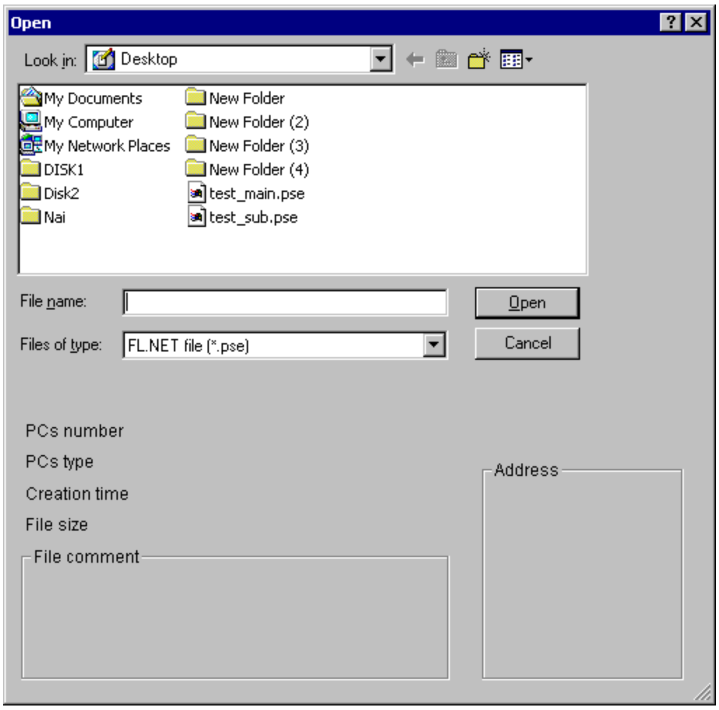


Figure 5-48 File Selection Dialog Box

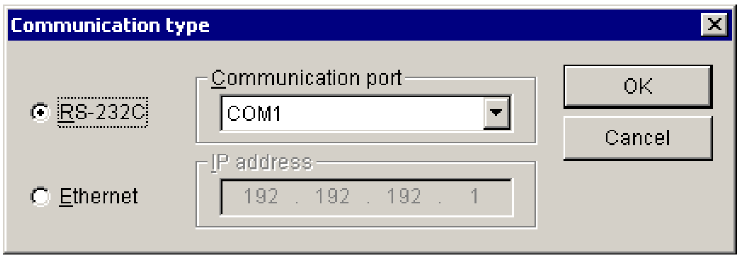
5.4.4 Changing connections

Function: Set the communication type between the PCs and the PC.

Operation: The procedure is shown below.

(1) From the [Tool] menu of the [[S10V] FL.NET] window, click the Change connection button.

(2) The [Communication type] window is displayed.



Communication type

● RS-232C

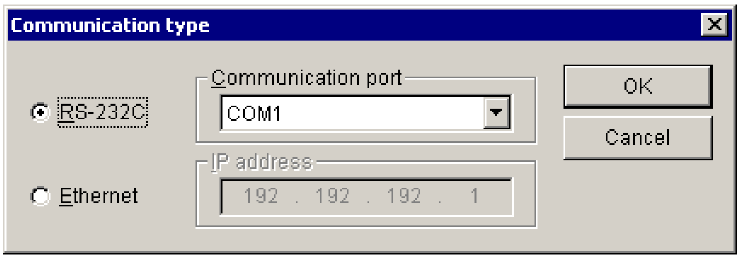
○ Ethernet

Communication port: COM1

IP address: 192 . 192 . 192 . 1

(The [Communication type] window will also be displayed automatically when the operation mode is switched from offline to online by choosing [Online] from the [Module] menu.)

(3) When the communication type is RS-232C, click the [RS-232C] radio button and then select "communication port."



Communication type

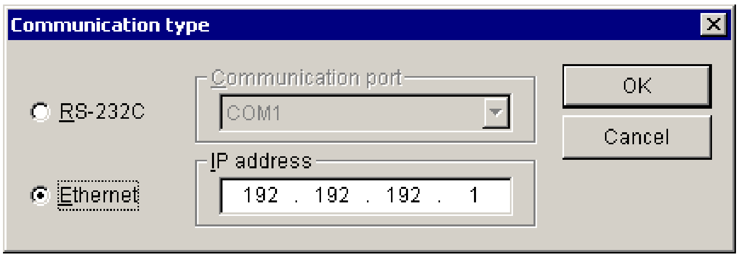
● RS-232C

○ Ethernet

Communication port: COM1

IP address: 192 . 192 . 192 . 1

(4) When the communication type is Ethernet, click the [Ethernet] radio button and then enter the "IP address" of the connection destination.



Communication type

○ RS-232C

● Ethernet

Communication port: COM1

IP address: 192 . 192 . 192 . 1

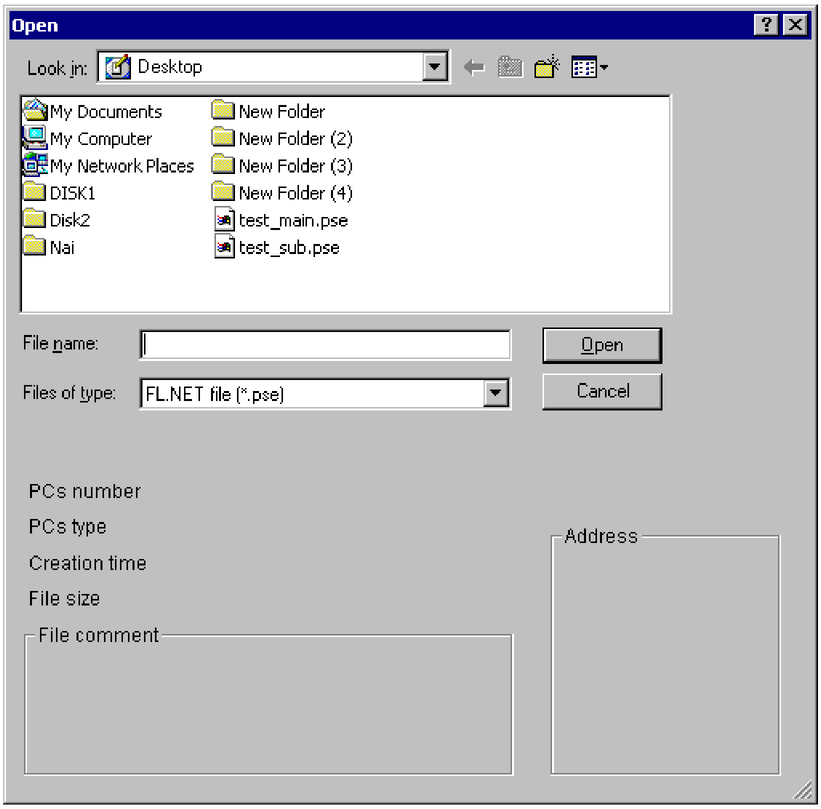
5.4.5 Selecting an edition file

Function: The function of this action is to choose a file you want to edit in offline mode. The files that you can choose for editing are those setup info files which have already been saved or prepared. You can also create a new file by specifying a non-existing file name.

Operation: The procedure used is described below.

(1) Select [Offline] from the [Module] menu. Switch the operation mode from online to offline, the [Open] window as shown below will then appear.

(2) In offline mode, if the currently selected file needs to be changed to another file, choose [Edit File Select] from the [Tool] menu. The [Open] window as shown below will then appear.

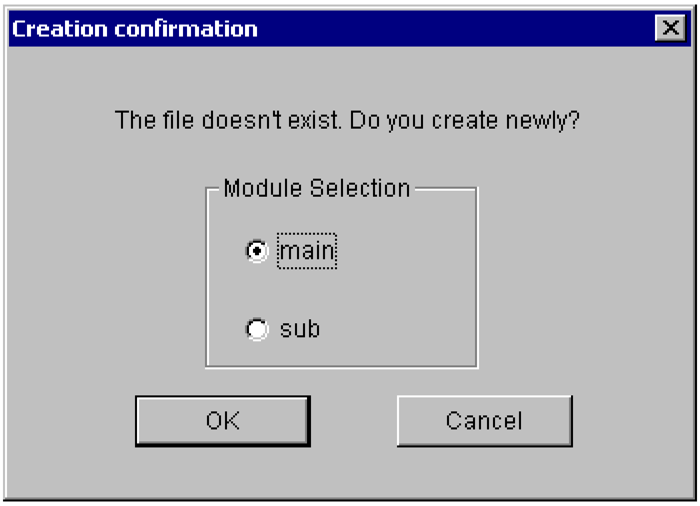


Look in: Desktop

Files of type: FL.NET file (*.pse)

Figure 5-49 The [Open] Window -- an Example

(3) If you want to create an FL.NET setup information file and edit its content, enter a non-existing file name into the “File name” box and click the Open button. The [Create confirmation] dialog box will then appear. Choose between “main” or “sub” and click the OK button.

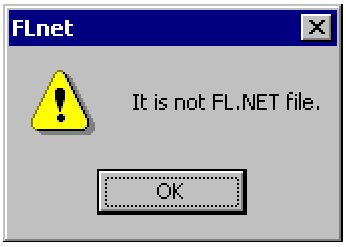


Creation confirmation

The file doesn't exist. Do you create newly?

Figure 5-50 The [Create confirmation] Dialog Box

(4) If you want to edit an already created FL.NET setup information file, choose it in the [Open] window. Then, the "PCs number" (in decimal), "PCs type" (in hexadecimal), and other information will be displayed. When they are displayed, click the Open button. If the selected file is not an FL.NET setup information file or is an invalid file, the following error message dialog box will appear:



FLnet

It is not FL.NET file.

Figure 5-51 The Error Message Dialog Box Displayed in the Event of a Non-Existent or Invalid FL.NET Setup Info File Specified

When a valid FL.NET setup info file is selected, you can now edit the setup information as you do while the system is running in online mode.

5.4.6 Closing the system

On the [[S10V] FL.NET] window, click the ☒ button or select [Exit] from the [Module] menu.

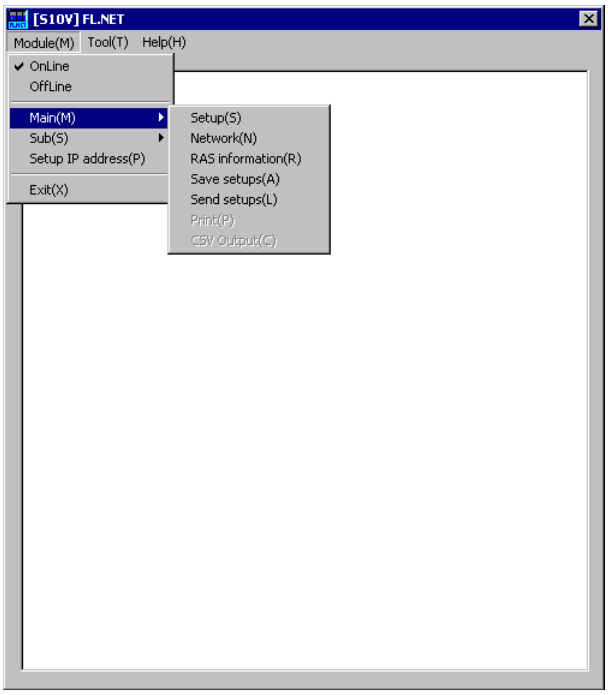
5.5 Operating Method

5.5.1 Self-node information

Function: Set the parameters required for the FL.NET module and set common memory allocation. And monitor the FL.NET module information.

Operation: The procedure is shown below.

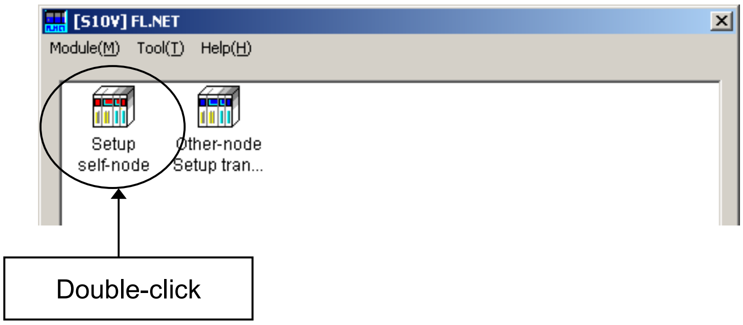
(1) Select [Main] or [Sub] from the [Module] menu of the [[S10V] FL.NET] window.



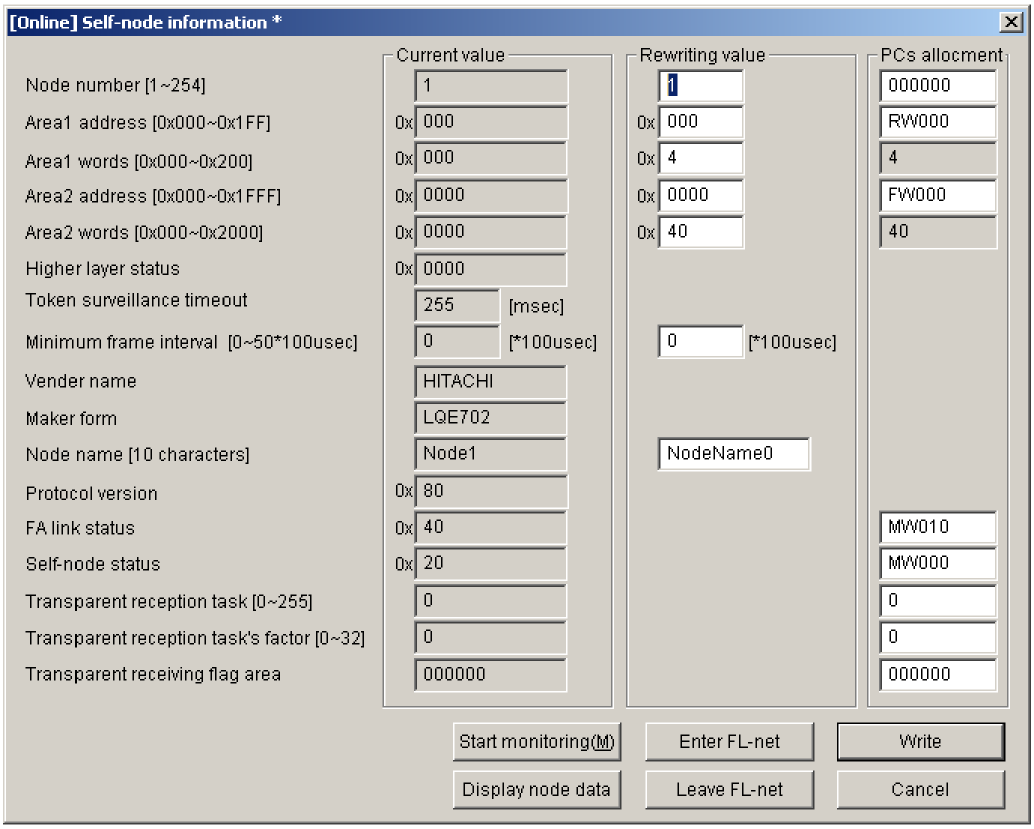
(2) Select the [Setup] menu.

(3) The [Setup self-node] icon (red) and [Other-node setup tran…] icon (blue) are displayed.

(4) Double-click the [Setup self-node] icon (red).



(5) The [[Online] Self-node information] window is displayed.



[Online] Self-node information *

	Current value	Rewriting value	PCs allocment
Node number [1~254]	1	1	000000
Area1 address [0x000~0x1FF]	0x 000	0x 000	RW000
Area1 words [0x000~0x200]	0x 000	0x 4	4
Area2 address [0x000~0x1FFF]	0x 0000	0x 0000	FW000
Area2 words [0x000~0x2000]	0x 0000	0x 40	40
Higher layer status	0x 0000		
Token surveillance timeout	255 [msec]		
Minimum frame interval [0~50*100usec]	0 [*100usec]	0 [*100usec]	
Vender name	HITACHI		
Maker form	LQE702		
Node name [10 characters]	Node1	NodeName0	
Protocol version	0x 80		
FA link status	0x 40		MW010
Self-node status	0x 20		MW000
Transparent reception task [0~255]	0		0
Transparent reception task's factor [0~32]	0		0
Transparent receiving flag area	000000		000000

(6) Change the rewrite value and each item of PCs allocation.

After completing the setup, click the Write button. To cancel the setup, click the Cancel button.

If the FL.NET system is running in online mode, clicking the Write button causes the FL.NET module to automatically isolate itself from the network and write the parameters to the actual target machine. Upon completion of the writing, the FL.NET module will automatically join the network again according to the written parameters.

(7) If the FL.NET system is running in online mode, you can also take any of the following actions:

To monitor the current value, click the Start monitoring button.

To display the contents of area 1 and area 2, click the Display node data button.

To join the FL.NET module in the network, click the Enter FL-net button.

To cause the FL.NET module to leave from the network, click the Leave FL-net button.

5.5.2 Other nodes display

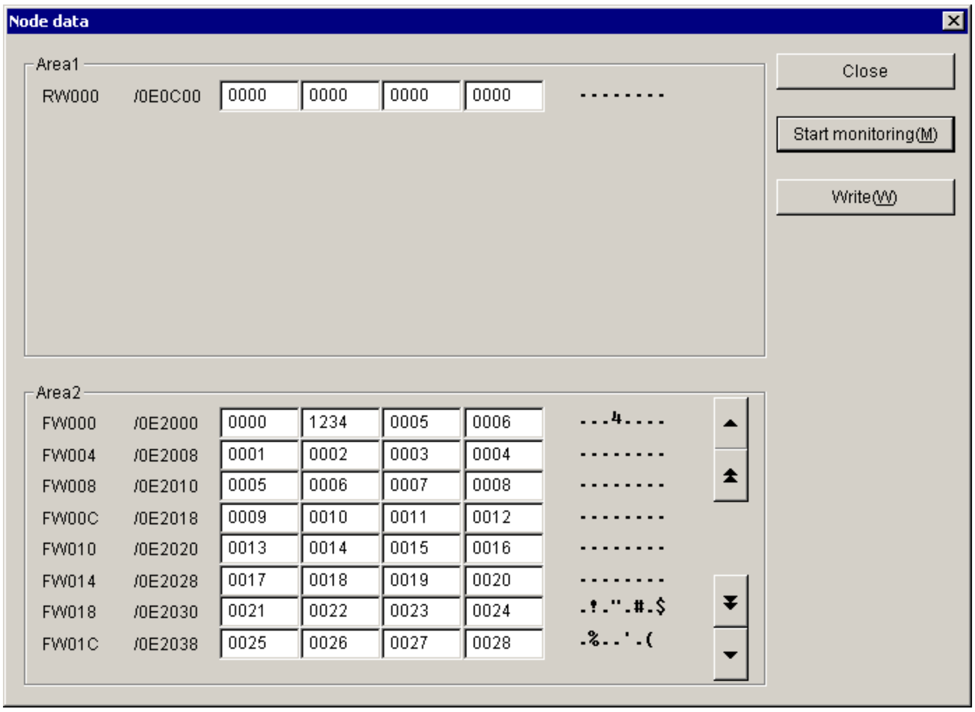
Function: Display the contents of areas 1 and 2 of either the local node (self-node) or another distant node, whichever has been selected in the [[Online] Self-node information] or [Other node information] window.

For using this function, it is a prerequisite that the PCs is connected with the PC by cable.

Operation: The procedure is shown below.

(1) When the FL.NET system is running in online mode, click the Display node data button in the [[Online] Self-node information] or [Other node information] window (see Subsection 5.5.4).

(2) The [Node data] window is displayed.



Area1

RW000	/0E0C00	0000	0000	0000	0000	....

Area2

FW000	/0E2000	0000	1234	0005	0006	...4....
FW004	/0E2008	0001	0002	0003	0004	....
FW008	/0E2010	0005	0006	0007	0008	....
FW00C	/0E2018	0009	0010	0011	0012	....
FW010	/0E2020	0013	0014	0015	0016	....
FW014	/0E2028	0017	0018	0019	0020	....
FW018	/0E2030	0021	0022	0023	0024	.!.".#.$
FW01C	/0E2038	0025	0026	0027	0028	.%..'.(

(3) If the data from area 1 or 2 cannot be displayed all at once because of limited screen size, the scroll buttons ▲ ⏫ and ▼ ⏬ are made available. Click any of these buttons to bring any unseen portion of the data into view.

(4) If you want to monitor the data in areas 1 and 2, click the Start monitoring button. If you want to add changes to the data, click the Write button. If nothing needs to be done on the data, click the Close button.
When using the Write button, note that no changes can be added to the data for any other node.

5.5.3 Other nodes reception setup

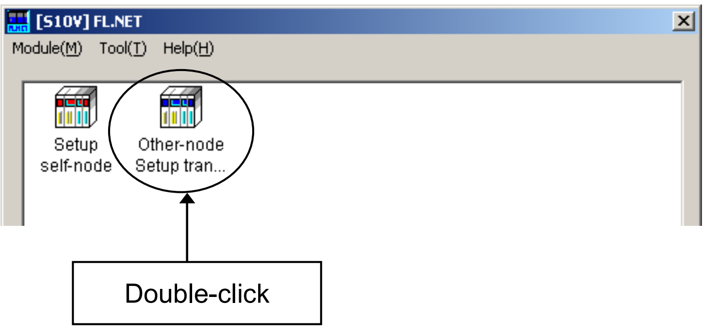
Function: Set other nodes PCs allocation.
Operation: The procedure is shown below.

(1) Select [Main] or [Sub] from the [Module] menu of the [[S10V] FL.NET] window.

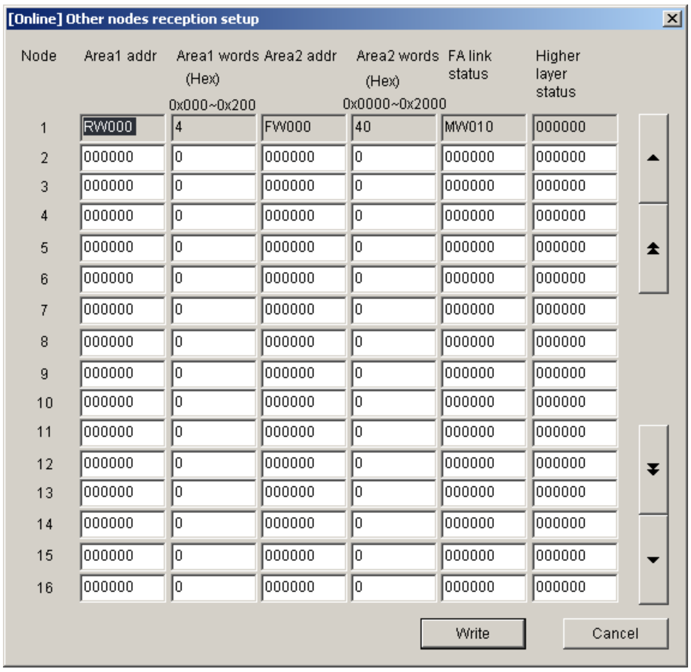
(2) Select the [Setup] menu.

(3) The [Setup self-node] icon (red) and [Other-node Setup tran…] icon (blue) are displayed.

(4) Double-click the [Other-node Setup tran…] icon (blue).

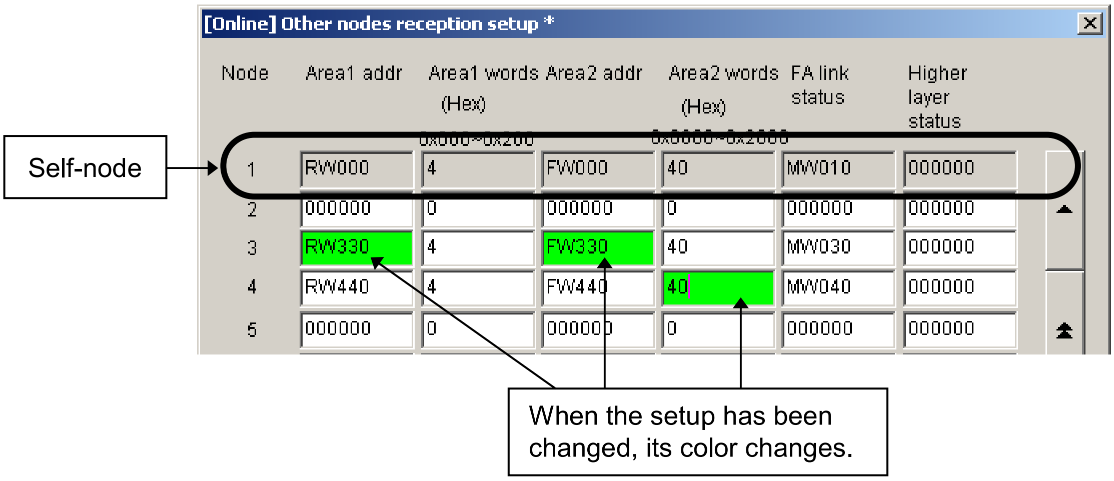


(5) The [[Online] Other nodes reception setup] window is displayed.



Node	Area1 addr	Area1 words (Hex) 0x000~0x200	Area2 addr	Area2 words (Hex) 0x0000~0x2000	FA link status	Higher layer status
1	RW000	4	FW000	40	MW010	000000
2	000000	0	000000	0	000000	000000
3	000000	0	000000	0	000000	000000
4	000000	0	000000	0	000000	000000
5	000000	0	000000	0	000000	000000
6	000000	0	000000	0	000000	000000
7	000000	0	000000	0	000000	000000
8	000000	0	000000	0	000000	000000
9	000000	0	000000	0	000000	000000
10	000000	0	000000	0	000000	000000
11	000000	0	000000	0	000000	000000
12	000000	0	000000	0	000000	000000
13	000000	0	000000	0	000000	000000
14	000000	0	000000	0	000000	000000
15	000000	0	000000	0	000000	000000
16	000000	0	000000	0	000000	000000

(6) The item whose setup has been changed is displayed in a different color.



[Online] Other nodes reception setup *

Node	Area1 addr	Area1 words (Hex) 0x000~0x200	Area2 addr	Area2 words (Hex) 0x0000~0x2000	FA link status	Higher layer status
1	RW000	4	FW000	40	MW010	000000
2	000000	0	000000	0	000000	000000
3	RW330	4	FW330	40	MW030	000000
4	RW440	4	FW440	40	MW040	000000
5	000000	0	000000	0	000000	000000

Self-node

When the setup has been changed, its color changes.

(7) If the node number to be displayed is not found on the window, click the ▲ ⏫ and ▼ ⏬ buttons to shift the display.

(8) After completing the setup, click the Write button. To cancel the setup, click the Cancel button.
The data of the self-node (Node No.1 in the above figure) is displayed in shadow form and the setup cannot be changed. Change it on the Self-node setup or [[Online] Self-node information] window (see "5.5.1 Self-node information").
If the FL.NET system is running in online mode, clicking the Write button causes the FL.NET module to automatically isolate itself from the network and have the parameters written to the actual target machine. Upon completion of the writing, the FL.NET module will automatically join the network again according to the written parameters.

5.5.4 Joining other nodes information

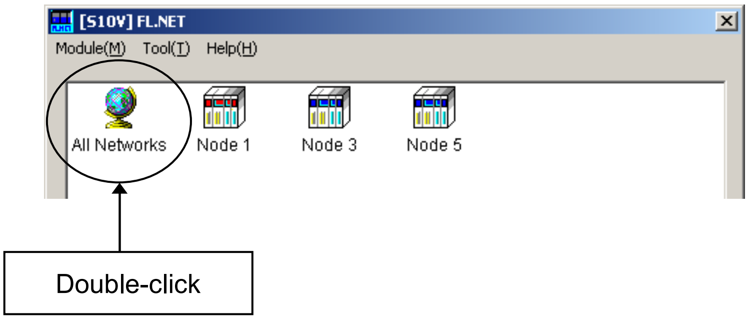
Function: Monitor the information on other nodes that join the network, and set PCs allocation.
For using this function, it is a prerequisite that the PCs is connected with the PC by cable.
Operation: The procedure is shown below.

(1) When the FL.NET system is running in online mode, choose [Main] or [Sub] from the [Module] menu in the [[S10V] FL.NET] window.

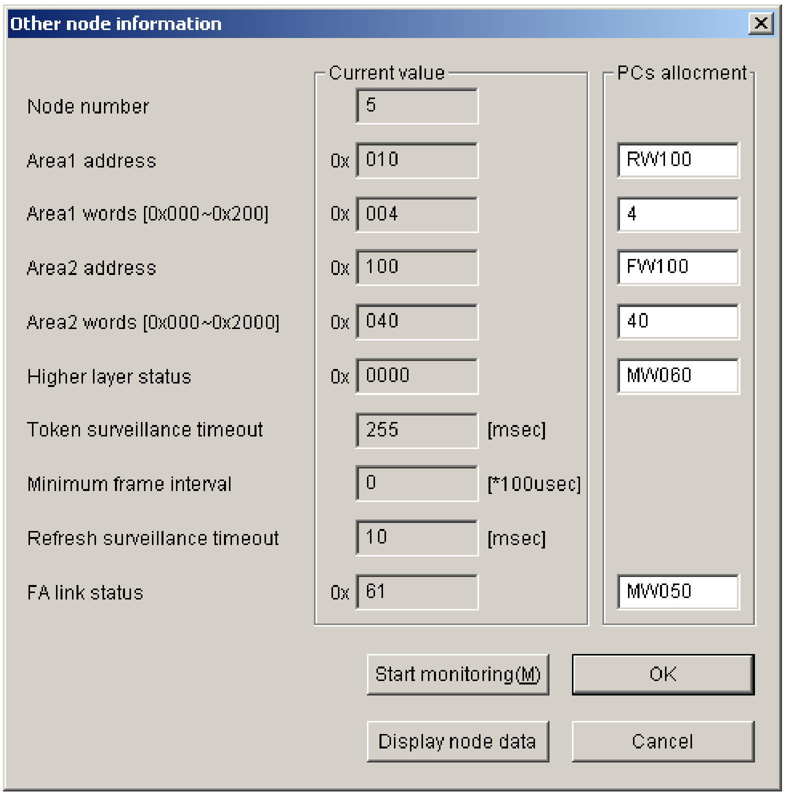
(2) Select the [Network] menu.

(3) The [All Networks] icon and each node No. icon (red or blue) connected to the network are displayed. The red node No. icon indicates the self-node. The blue node No. icon indicates other node connected to the network.

(4) Double-click the node No. icon (blue) to be set. (Double-clicking the red node No. icon displays the [[Online] Self-node information] window. (See “5.5.1 Self-node information.”)



(5) The [Other node information] window of the specified number is displayed.



	Current value		PCs allocment
Node number	5		
Area1 address	0x 010		RW100
Area1 words [0x000~0x200]	0x 004		4
Area2 address	0x 100		FW100
Area2 words [0x000~0x2000]	0x 040		40
Higher layer status	0x 0000		MW060
Token surveillance timeout	255	[msec]	
Minimum frame interval	0	[*100usec]	
Refresh surveillance timeout	10	[msec]	
FA link status	0x 61		MW050

(6) Change the each item setup of PCs allocation.
Click the Start monitoring button to monitor the current value.
Click the Display node data button to display the contents of area 1 and area 2.
After completing the setup, click the OK button. To cancel the setup, click the Cancel button.
Clicking the OK button causes the FL.NET module to automatically leave from the network to write the data. After writing the data, the FL.NET module joins the network automatically.

5.5.5 Network status

Function: Display the status when joining the network including the allowable refresh cycle time. For using this function, it is a prerequisite that the PCs is connected with the PC by cable.

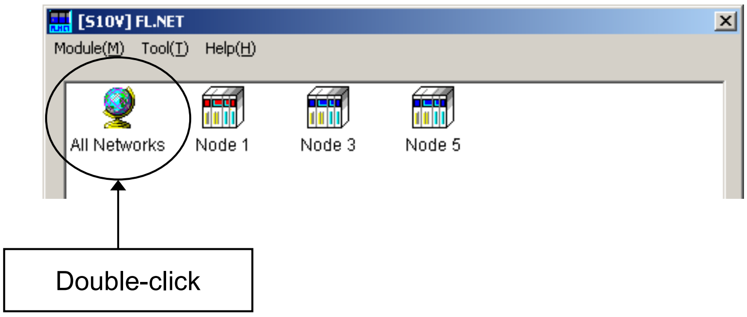
Operation: The procedure is shown below.

(1) Select [Main] or [Sub] from the [Module] menu of the [[S10V] FL.NET] window.

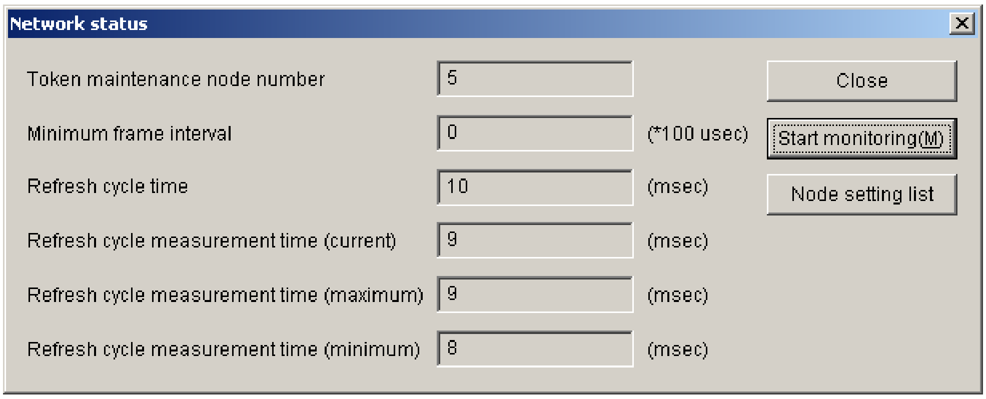
(2) Select the [Network] menu.

(3) The [All Networks] icon and each node No. icon connected to the network are displayed.

(4) Double-click the [All Networks] icon.



(5) The [Network status] window is displayed.



Token maintenance node number: 5

Minimum frame interval: 0 (*100 usec)

Refresh cycle time: 10 (msec)

Refresh cycle measurement time (current): 9 (msec)

Refresh cycle measurement time (maximum): 9 (msec)

Refresh cycle measurement time (minimum): 8 (msec)

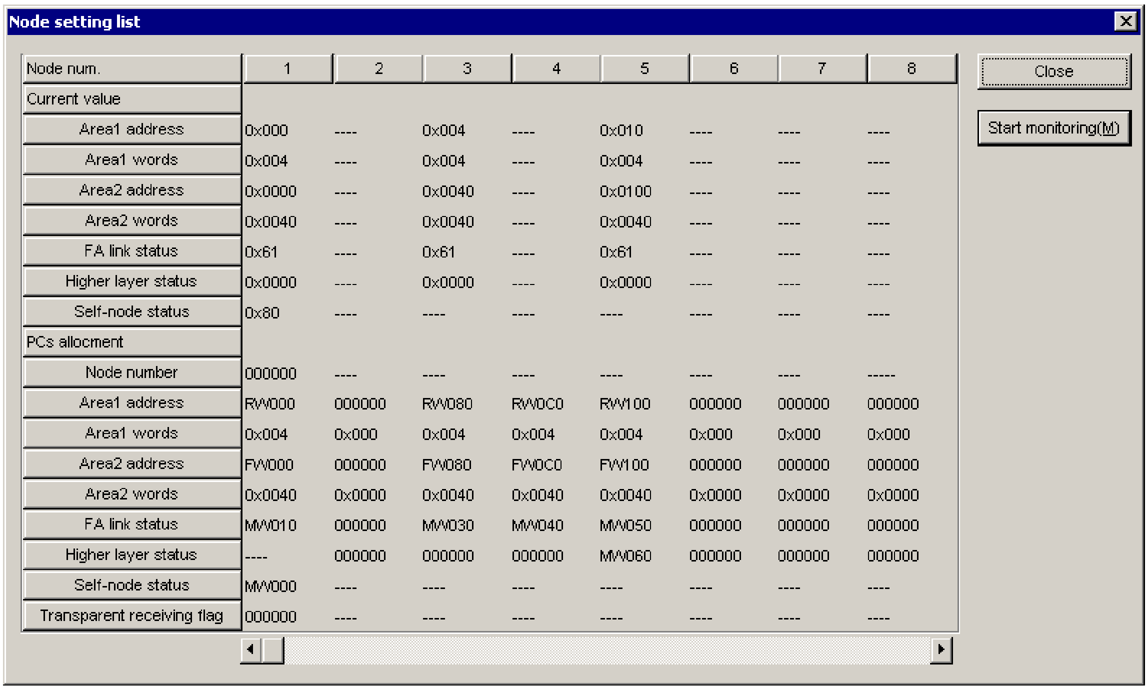
(6) Click the Start monitoring button to monitor the network status.
Click the Node setting list button to display the PCs allocation and current values of all nodes.
Click the Close button to exit the [Network status] window.

5.5.6 Node setting list

Function: Display the PCs allocation and current values of all nodes.
For using this function, it is a prerequisite that the PCs is connected with the PC by cable.
Operation: The procedure is shown below.

(1) Click the Node setting list button on the [Network status] window.

(2) The [Node setting list] window is displayed.



Node num.	1	2	3	4	5	6	7	8
Current value								
Area1 address	0x000	----	0x004	----	0x010	----	----	----
Area1 words	0x004	----	0x004	----	0x004	----	----	----
Area2 address	0x0000	----	0x0040	----	0x0100	----	----	----
Area2 words	0x0040	----	0x0040	----	0x0040	----	----	----
FA link status	0x61	----	0x61	----	0x61	----	----	----
Higher layer status	0x0000	----	0x0000	----	0x0000	----	----	----
Self-node status	0x80	----	----	----	----	----	----	----
PCs allocment								
Node number	000000	----	----	----	----	----	----	-----
Area1 address	RW000	000000	RW080	RW0C0	RW100	000000	000000	000000
Area1 words	0x004	0x000	0x004	0x004	0x004	0x000	0x000	0x000
Area2 address	FW000	000000	FW080	FW0C0	FW100	000000	000000	000000
Area2 words	0x0040	0x0000	0x0040	0x0040	0x0040	0x0000	0x0000	0x0000
FA link status	MW010	000000	MW030	MW040	MW050	000000	000000	000000
Higher layer status	----	000000	000000	000000	MW060	000000	000000	000000
Self-node status	MW000	----	----	----	----	----	----	----
Transparent receiving flag	000000	----	----	----	----	----	----	----

(3) Click the Start monitoring button to monitor the current values.
Click the Close button to exit the [Node setting list] window.

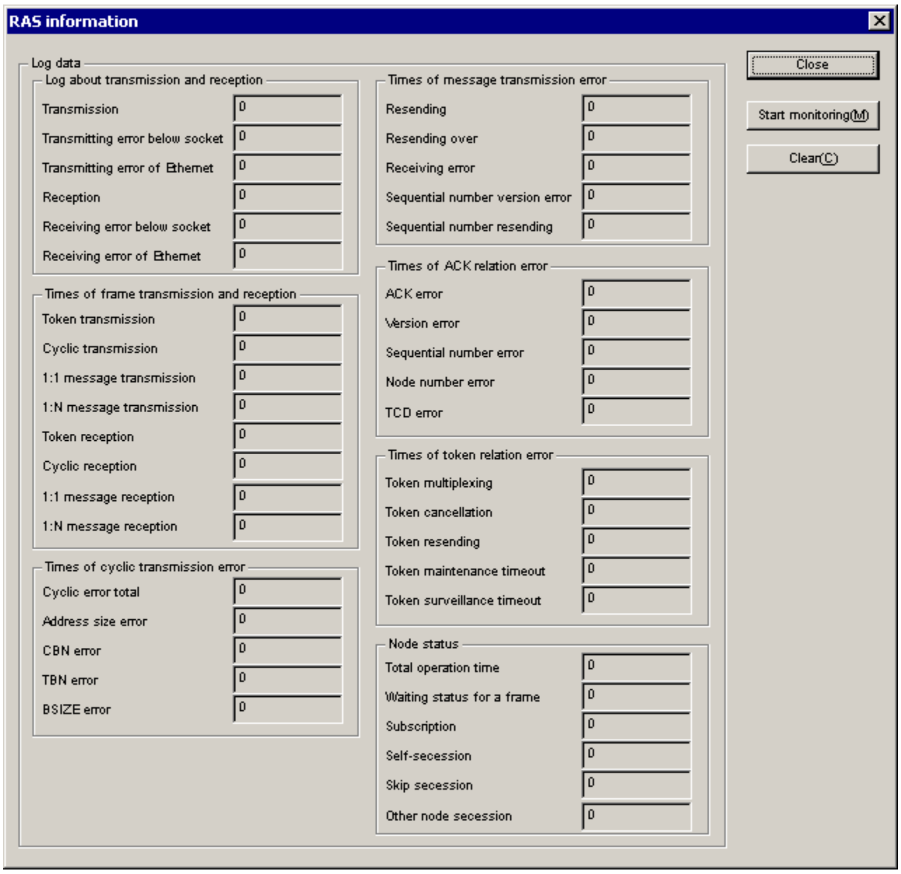
5.5.7 RAS information

Function: Display the RAS information.
For using this function, it is a prerequisite that the PCs is connected with the PC by cable.
Operation: The procedure is shown below.

(1) Select [Main] or [Sub] from the [Module] menu of the [[S10V] FL.NET] window.

(2) Select the [RAS information] menu.

(3) The [RAS information] window is displayed.



RAS information

Log data

Log about transmission and reception

Transmission	0
Transmitting error below socket	0
Transmitting error of Ethernet	0
Reception	0
Receiving error below socket	0
Receiving error of Ethernet	0

Times of frame transmission and reception

Token transmission	0
Cyclic transmission	0
1:1 message transmission	0
1:N message transmission	0
Token reception	0
Cyclic reception	0
1:1 message reception	0
1:N message reception	0

Times of cyclic transmission error

Cyclic error total	0
Address size error	0
CBN error	0
TBN error	0
BSIZE error	0

Times of message transmission error

Resending	0
Resending over	0
Receiving error	0
Sequential number version error	0
Sequential number resending	0

Times of ACK relation error

ACK error	0
Version error	0
Sequential number error	0
Node number error	0
TCD error	0

Times of token relation error

Token multiplexing	0
Token cancellation	0
Token resending	0
Token maintenance timeout	0
Token surveillance timeout	0

Node status

Total operation time	0
Waiting status for a frame	0
Subscription	0
Self-secession	0
Skip secession	0
Other node secession	0

(4) Click the Start monitoring button to monitor the RAS information.
Click the Clear button to reset all the RAS information to 0.
Click the Close button to exit the [RAS information] window.
The RAS information is all reset to 0 by resetting or turning off and on the power supply.

5.5.8 Saving setups

Function: Save the self-node setup of the PCs side, PCs allocation of all nodes, and IP address and subnet mask of the FL.NET module into the file on the PC.

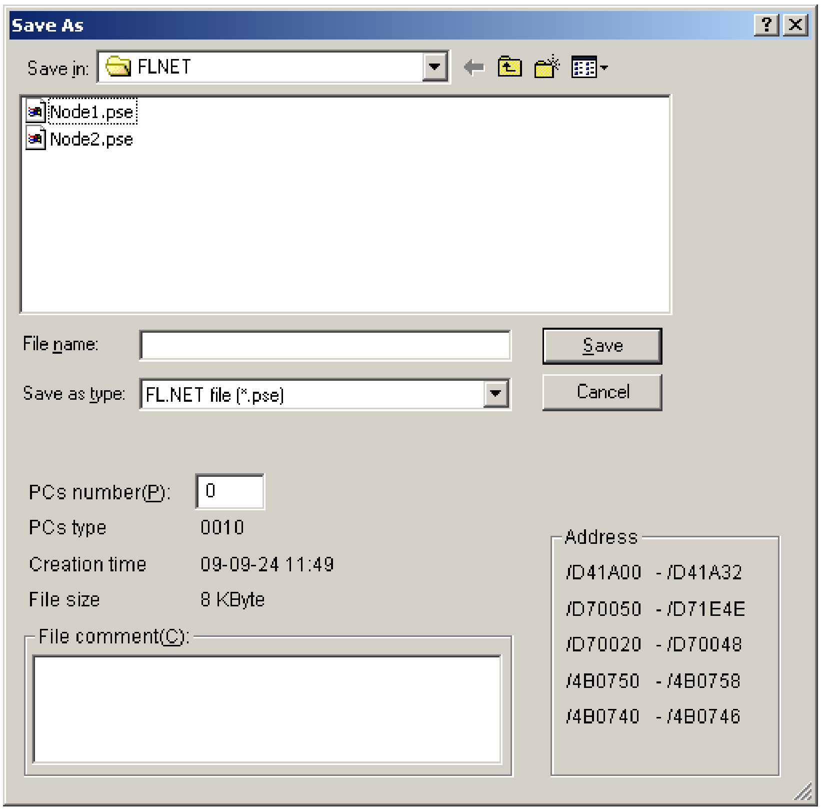
For using this function, it is a prerequisite that the PCs is connected with the PC by cable.

Operation: The procedure is shown below.

(1) Select [Main] or [Sub] from the [Module] menu of the [[S10V] FL.NET] window.

(2) Select the [Save setups] menu.

(3) The [Save As] window is displayed. Enter the file name.
If a comment is required, enter it in the file comment field.
(Maximum number of input characters: 128 characters)
To change the PCs number, enter it in the PCs number field. (Maximum number of input characters: 4-digit number)



Save in: FLNET

Node1.pse
Node2.pse

File name:

Save as type: FL.NET file (*.pse)

PCs number(P): 0
PCs type 0010
Creation time 09-09-24 11:49
File size 8 KByte
File comment(C):

Address
/D41A00 - /D41A32
/D70050 - /D71E4E
/D70020 - /D70048
/4B0750 - /4B0758
/4B0740 - /4B0746

(4) After completing the setup, click the Save button. To cancel the setup, click the Cancel button.
The areas to be saved are self-node setup, PCs allocation of all nodes, IP address, and subnet mask.

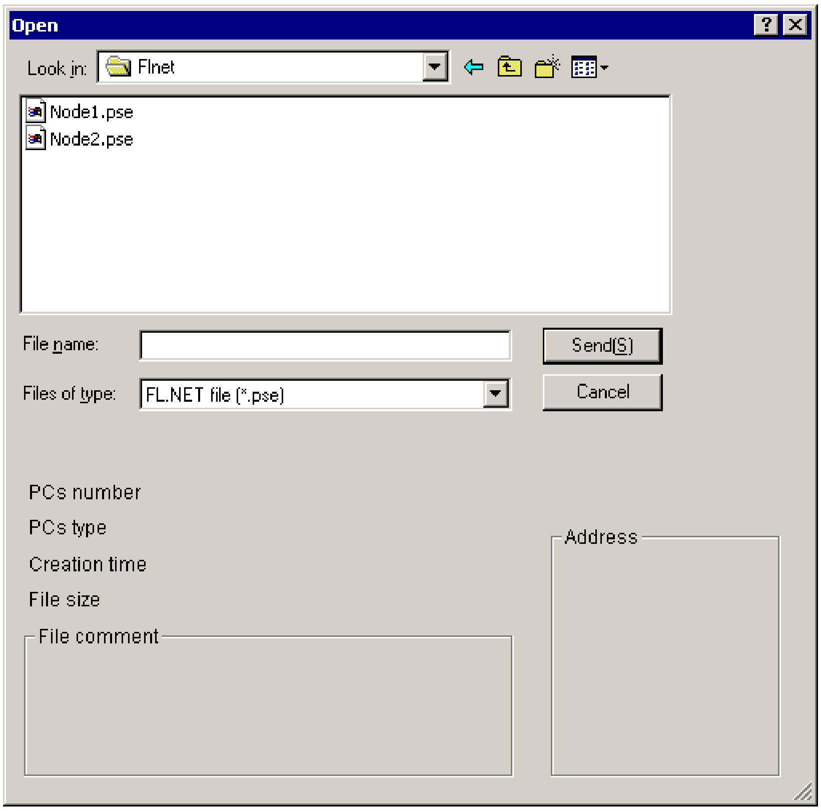
5.5.9 Sending setups

Function: Load the self-node setup, PCs allocation of all nodes, and IP address and subnet mask of the FL.NET module, which were saves on the PC by the saving function, onto the PCs.
For using this function, it is a prerequisite that the PCs is connected with the PC by cable.
Operation: The procedure is shown below.

(1) Select [Main] or [Sub] from the [Module] menu of the [[S10V] FL.NET] window.

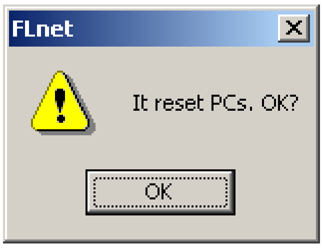
(2) Select the [Send setups] menu.

(3) The [Open] window is displayed. Enter the file name.



(4) When the file is specified, click the Send button to send it to the programmable controller. If you want to cancel this function, click the Cancel button.

(5) Upon completion of the sending, the message dialog box shown below appears. Contact the facility manager and check if the programmable controller is resettable now. If it is resettable, click the OK button. The programmable controller will then be automatically reset. Upon completion of the resetting, the "Sending terminated" message dialog box appears. Click the OK button.



It reset PCs. OK?

5.5.10 Setup IP address

● Online editing

Function: Set the IP address of the FL.NET module.

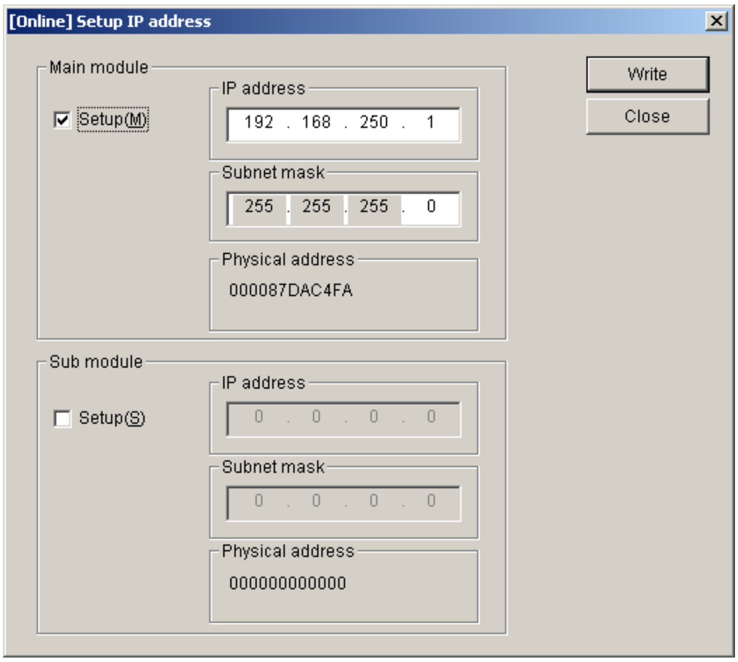
For using this function, it is a prerequisite that the PCs is connected with the PC by cable.

Note: Setting the IP address or subnet mask for the FL.NET module resets the programmable controller automatically.

Operation: The procedure is shown below.

(1) Select [IP address] from the [Module] menu of the [[S10V] FL.NET] window.

(2) The [[Online] Setup IP address] window is displayed. Enter the IP address and subnet mask.



[Online] Setup IP address

Main module

☑ Setup(M)

IP address: 192 . 168 . 250 . 1

Subnet mask: 255 . 255 . 255 . 0

Physical address: 000087DAC4FA

Sub module

□ Setup(S)

IP address: 0 . 0 . 0 . 0

Subnet mask: 0 . 0 . 0 . 0

Physical address: 000000000000

(3) After completing the setup, click the Write button. To cancel the setup, click the Close button.

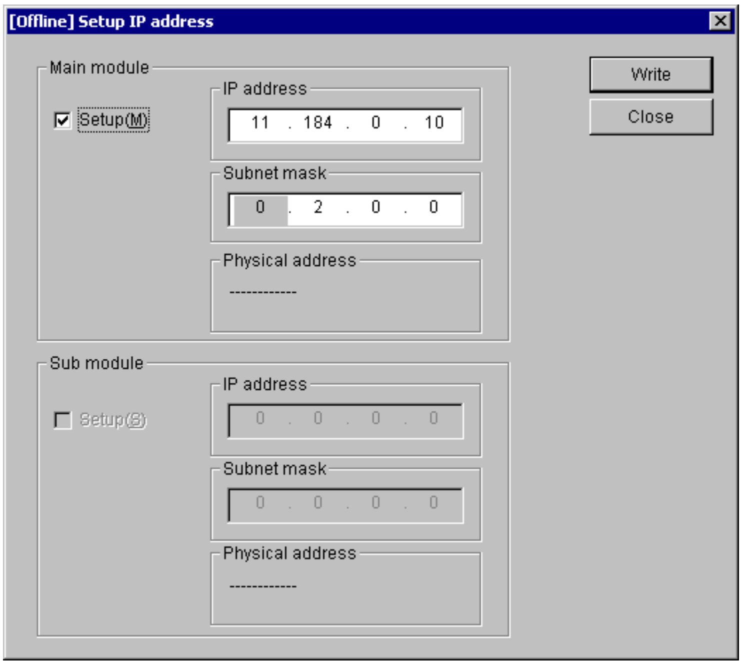
(4) Upon completion of the setup, the message dialog box shown below appears. Contact the facility manager and check if the programmable controller is resettable now. If it is resettable, click the OK button. The programmable controller will then be automatically reset. Upon completion of the resetting, the “Sending terminated” message dialog box appears. Click the OK button.

- Offline editing

Function: Set the IP address of the FL.NET module.
Operation: The procedure is shown below.

(1) Select [IP address] from the [Module] menu of the [[S10V] FL.NET] window.

(2) The [[Offline] Setup IP address] window is displayed. Enter the IP address and subnet mask.



[Offline] Setup IP address

Main module

☑ Setup(M)

IP address: 11 . 184 . 0 . 10

Subnet mask: 0 . 2 . 0 . 0

Physical address: ------------

Sub module

□ Setup(S)

IP address: 0 . 0 . 0 . 0

Subnet mask: 0 . 0 . 0 . 0

Physical address: ------------

(3) After completing the setup, click the Write button. To cancel the setup, click the Close button.

5.5.11 Printing

Function: The function of commands used for this purpose is to print on the printer one of the following two pieces of information: 1) the setup information for a selected module from the actual target machine if the FL.NET system is running in online mode; or 2) the content (setup information) of a selected file if it is running in offline mode.

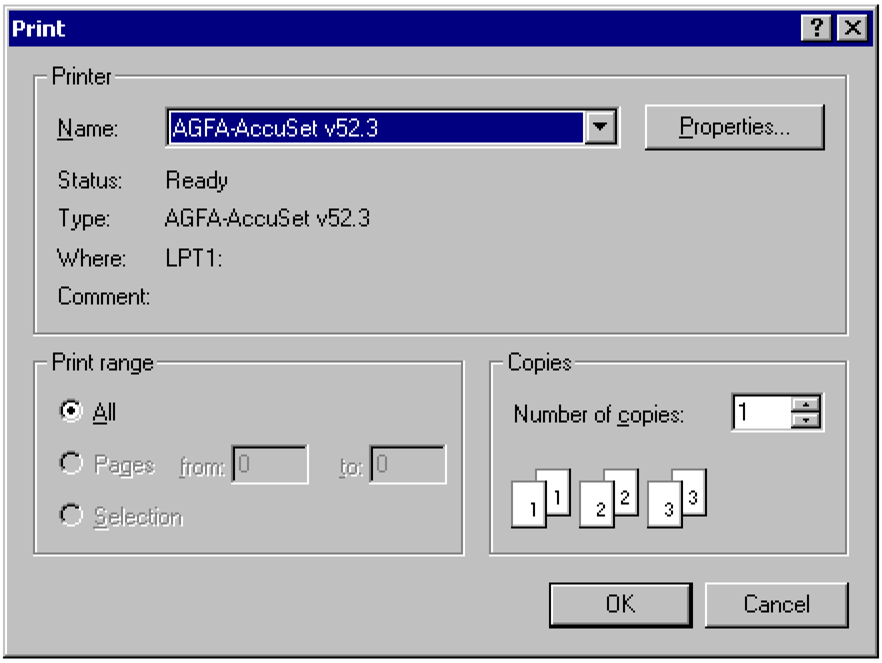
Operation: The procedure used is shown below.

(1) If the FL.NET system is running in online mode, establish a connection between the FL.NET system and the PCs (see "5.4.4 Changing connections"). If it is running in offline mode, choose the desired file for editing (see "5.4.5 Selecting an edition file").

(2) Choose [Main] – [Setup], or [Sub] – [Setup], from the [Module] menu. Then, the [Setup self-node] icon (red) and [Other-node setup transmission] icon (blue) appear.

(3) Choose [Main] – [Print], or [Sub] – [Print], from the [Module] menu.

(4) The [Print] dialog box appears. In this dialog box, specify the desired printer and its properties, and then click the OK button.


Print

Printer

Name: AGFA-AccuSet v52.3

Status: Ready

Type: AGFA-AccuSet v52.3

Where: LPT1:

Comment:

Print range

● All

○ Pages from: 0 to: 0

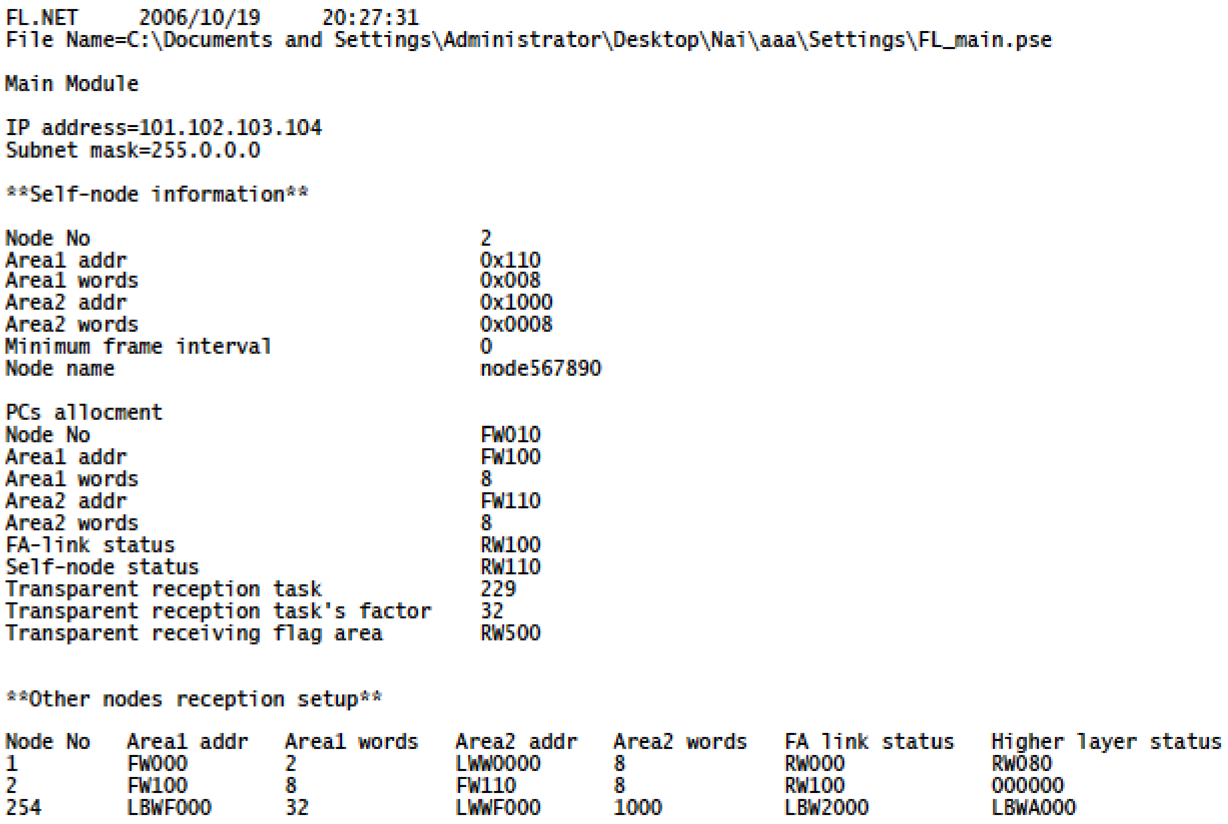
○ Selection

Copies

Number of copies: 1

Figure 5-52 The [Print] Dialog Box -- an Example

<Sample printout>



```
FL.NET     2006/10/19     20:27:31
File Name=C:\Documents and Settings\Administrator\Desktop\Nai\aaa\Settings\FL_main.pse

Main Module

IP address=101.102.103.104
Subnet mask=255.0.0.0

**Self-node information**

Node No                                 2
Area1 addr                              0x110
Area1 words                             0x008
Area2 addr                              0x1000
Area2 words                             0x0008
Minimum frame interval                  0
Node name                               node567890

PCs allocment
Node No                                 FW010
Area1 addr                              FW100
Area1 words                             8
Area2 addr                              FW110
Area2 words                             8
FA-link status                          RW100
Self-node status                        RW110
Transparent reception task              229
Transparent reception task's factor     32
Transparent receiving flag area         RW500


**Other nodes reception setup**

Node No   Area1 addr   Area1 words   Area2 addr   Area2 words   FA link status   Higher layer status
1         FW000        2             LWW0000      8             RW000            RW080
2         FW100        8             FW110        8             RW100            000000
254       LBWF000      32            LWWF000      1000          LBW2000          LBWA000
```

5.5.12 CSV output

Function: The function of this command is to output to a file in CSV format one of the following two pieces of information: 1) the setup information for a selected module from the actual target machine if the FL.NET system is running in online mode; or 2) the content (setup information) of a selected file if it is running in offline mode.

Operation: The procedure used is shown below.

(1) If the FL.NET system is running in online mode, establish a connection between the FL.NET system and the PCs (see “5.4.4 Changing connections”). If it is running in offline mode, choose the desired file for editing (see “5.4.5 Selecting an edition file”).

(2) Choose [Main] – [Setup], or [Sub] – [Setup], from the [Module] menu. Then, the [Setup self-node] icon (red) and [Other-node setup transmission] icon (blue) appear.

(3) Choose [Main] – [CSV output], or [Sub] – [CSV output], from the [Module] menu.

(4) The [Save As] dialog box as shown below appears. In this dialog box, specify the desired folder and file to which you want to output the setup information, and then click the Save button.

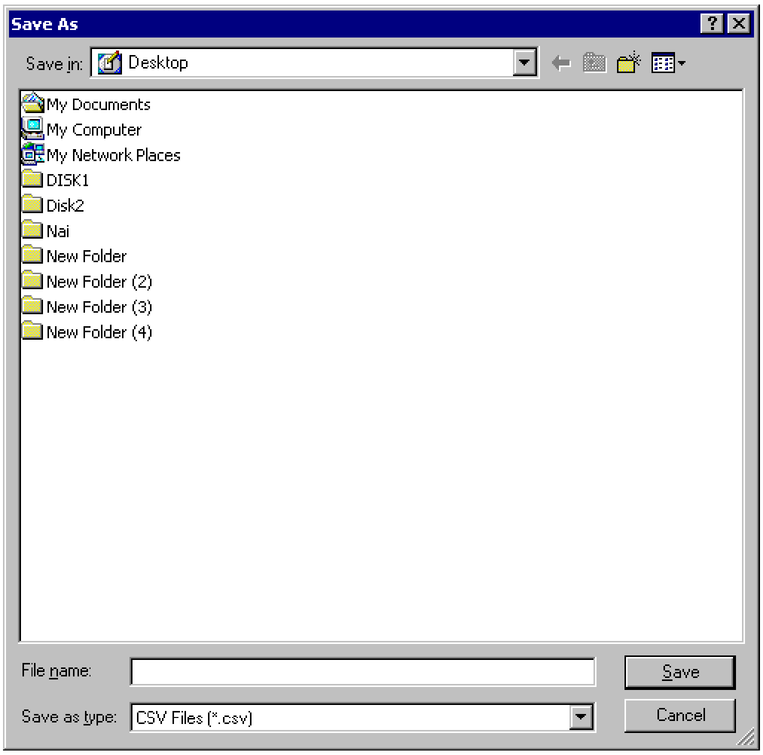


Figure 5-53 The [Save as] Dialog Box -- an Example

<Sample CSV file output>

```
FL.NET        2006/10/19        20:28:50
 File Name=C:\aaa\Settings\FL_main.pse

 Main Module

 IP address=101.102.103.104
 Subnet mask=255.0.0.0

 **Self-node information**

 Node No,2
 Area1 addr,0x110
 Area1 words,0x008
 Area2 addr,0x1000
 Area2 words,0x0008
 Minimum frame interval ,0
 Node name,node567890

 PCs allocment
 Node No,FW010
 Area1 addr,FW100
 Area1 words,8
 Area2 addr,FW110
 Area2 words,8
 FA-link status,RW100
 Self-node status,RW110
 Transparent reception task,229
 Transparent reception task’s factor,32
 Transparent receiving flag area,RW500


 **Other nodes reception setup**

 Node No,Area1 addr,Area1 words,Area2 addr,Area2 words,FA link status,Higher layer status,
 1,FW000,2,LWW0000,8,RW000,RW080,
 2,FW100,8,FW110,8,RW100,000000,
 254,LBWF000,32,LWWF000,1000,LBW2000,LBWA000,
```

This Page Intentionally Left Blank

6 MAINTENANCE

6.1 Maintenance and Inspection

To keep the module running in optimal condition, it requires checks. Make checks daily or periodically (twice a year or more often).

Table 6-1 Maintenance and Inspection Items

Item	Point to check
Module appearance	Check the module case for cracks, flaws and other defects. Such defects can be a sign of breakage in the internal circuitry, causing the system to malfunction.
LED	Check to see if the module ERR LED has not glowed.
Loose mounting screws	Check the module and communications cable mounting screws for tightness. Give additional tightening to screws found loose. Loose screws could cause the system to malfunction and lead to burnouts after heating.
Cable covering status	Check cable coverings for defects. A cable covering out of position could cause the system to malfunction, incur electrical shock hazards, or develop short circuits, resulting in burnouts.
Dust adhesion	Check to see if the module has not caught dust. If dust is noticed, remove it with a vacuum cleaner or other apparatus. Dust could cause short circuits in its internal circuitry, resulting in burnouts.
Module replacement	Replacing the module without switching it off could cause damage to its hardware and software. Before replacing the module, switch it off first.
Connector status	Connectors might have their characteristics degraded to cause failures if their contacts catch dust or foreign matter. Cover connectors out of use with the protective cap supplied.

⚠ **CAUTION**

Before replacing the module, switch it off to avoid electrical shock hazards and also to prevent it from being damaged or malfunctioning.

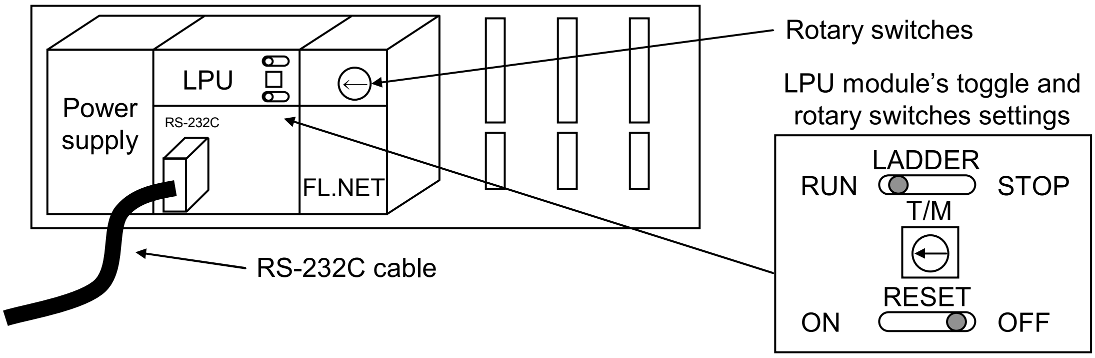
NOTICE

Static electricity could cause damage to the module. Before handling the module allow static charges on the human body to discharge.

6.1.1 Replacing or adding on the module

- What you should get in preparation
  ① Personal computer (with S10V FL.NET System installed in it)
  ② RS-232C cable (or 10BASE-T cable if the communication module used is an ET.NET module)
  ③ New or add-on FL.NET module (LQE702)
  ④ Copies of the parameter values for the module to be replaced. (These copies serve as a substitute for the original parameter values when the original become inaccessible due to, for example, a failure of the FL.NET module.)
  ⑤ Optional ET.NET module. Where an ET.NET module is installed, you can select Ethernet as the communication type. For more information, refer to Section 2.1, “Names and Functions of Each Part,” and Section 3.2, “Mounting the Module,” in the USER’S MANUAL OPTION ET.NET (LQE520) (manual number SVE-1-103).

- Replacement procedure
  ① Write down, on a piece of paper, the current settings of the rotary switches that are, as shown below, accessible at the front side of the FL.NET module to be replaced.
  ② Write down also the current settings of two switches, labeled LADDER (toggle switch) and T/M (rotary switch), respectively, that are, as shown below, accessible at the front side of the LPU module.



  ③ Connect the personal computer and the LPU module together with the RS-232C cable.
  ④ Start the S10V FL.NET SYSTEM and save the existing settings to a dedicated file, called an FL.NET setup information file (see “5.5.8 Saving setups” for details). If the existing settings are not accessible for some reason, skip this step and use item ④ you got in preparation.
  ⑤ Set the LPU module’s LADDER switch in STOP position and turn off the power supply of the controller unit.
  ⑥ Remove the connecting cables from the FL.NET module to be replaced.

⑦ Replace the existing FL.NET module with the new one and set the new FL.NET module's rotary switches in the same way as you wrote down in Step ①.

⑧ Turn on the power supply of the controller unit and send to the new FL.NET module the FL.NET setup information file that you have saved in Step ④ (see "5.5.9 Sending setups" for details).

⑨ Turn off the power supply of the controller unit.

⑩ Remove the RS-232C cable from both the personal computer and LPU module, which were connected together in Step ③.

⑪ Connect to the new FL.NET module the connecting cables that you removed in Step ⑥.

⑫ Set the LPU module's LADDER and T/M switches in the same way as you wrote down in Step ②.

⑬ Turn on the power supply of the controller unit and check that the new FL.NET module is running normally.

- Add-on procedure

① Write down, on a piece of paper, the current settings of two switches, labeled LADDER (toggle switch) and T/M (rotary switch), respectively, that are accessible at the front side of the LPU module, the one that is installed in the controller unit in which you are adding on a FL.NET module.

② Ensure that your application system has been shut down. Then, set the LPU module's LADDER switch in STOP position and turn off the power supply of the controller unit.

③ Mount the add-on FL.NET module in place according to the instructions given under "3.2 Mounting the Module."

④ Set the add-on FL.NET module's MAIN/SUB selector switches in such a way that a new module (1) setting, which must be a sub-module (1) setting, will not duplicate with the current rotary switch settings of the existing main FL.NET module.

⑤ Connect the personal computer and the LPU module together with the RS-232C cable. Then, turn on the power supply of the controller unit and set parameters for the add-on FL.NET module by using the S10V FL.NET System.

⑥ Turn off the power supply of the controller unit and connect the connecting cables to the add-on FL.NET module.

⑦ Set the LPU module's LADDER and T/M switches in the same way as you wrote down in Step ①.

⑧ Remove the RS-232C cable from both the personal computer and LPU module, which were connected together in Step ⑤.

⑨ Turn on the power supply of the controller unit and check that the add-on FL.NET module is running normally.

7 TROUBLESHOOTING

7.1 Troubleshooting

7.1.1 Procedure

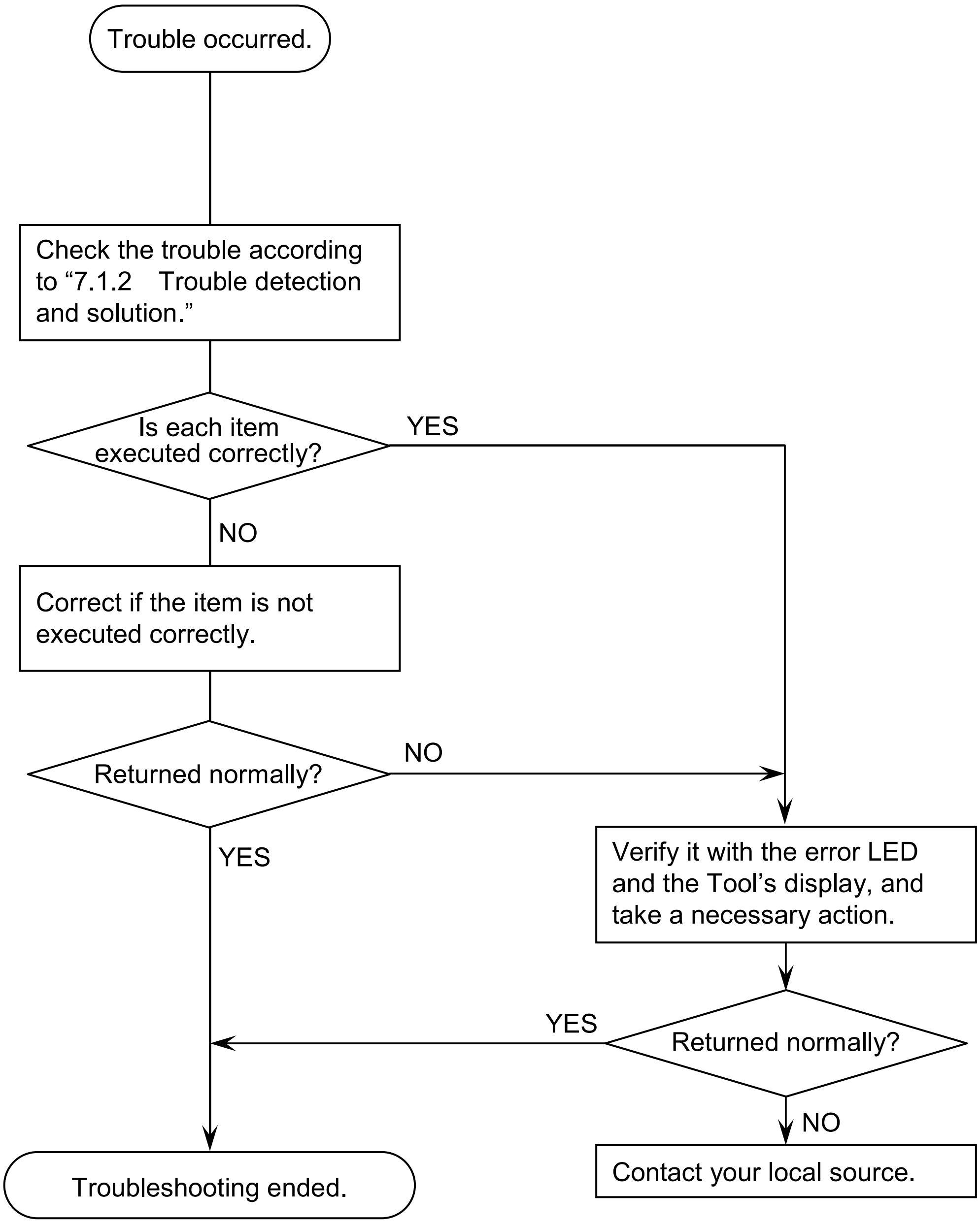
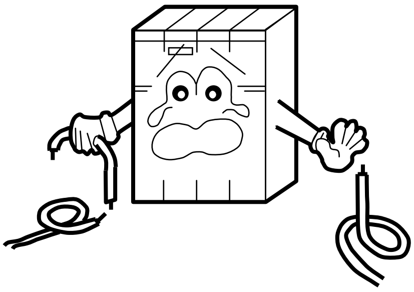

Trouble occurred.
Check the trouble according to "7.1.2 Trouble detection and solution."
Is each item executed correctly?
YES
NO
Correct if the item is not executed correctly.
Returned normally?
NO
YES
Verify it with the error LED and the Tool's display, and take a necessary action.
Returned normally?
YES
NO
Contact your local source.
Troubleshooting ended.

Figure 7-1 Troubleshooting Flow

7.1.2 Trouble detection and solution

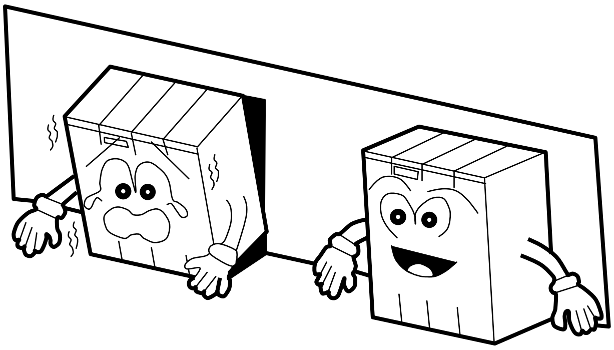
(1) Is the cabling correct?

- Check cables for disconnection or incorrect connection.



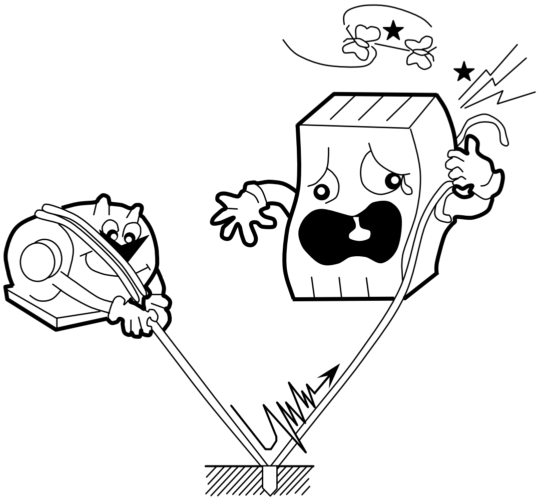
(2) Are the modules mounted correctly?

- Check that no set screws loosen.
- Check if a model LQE702 module is intermixed with a model LQE500/LQE502/LQE701 module among the same series of similar modules installed. The four different models may not be intermixed on the same mount base.



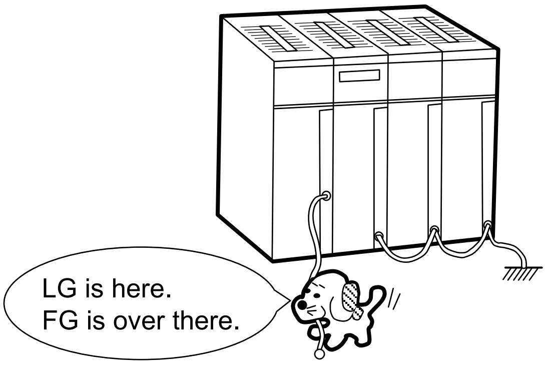
(3) Is grounding correct?

- Do not ground the ET.NET module in the same place where high-voltage equipment is grounded. They must be grounded in separate place.
- Perform grounding work conforming to Class D grounding or higher.



(4) Are LG and FG separated?

- Be sure to separate the LG from the FG or vice versa because power noise enters the FG via the LG. Failure to observe this rule may result in an equipment malfunction.
- Ground the LG at the power supply side.



7.2 Network Problems and Repairing

(1) Network problems and remedies (concerning unstable communications)

Table 7-1 Network Problems and Repairing (Concerning Unstable Communications)

Problem	Inspection item	Inspection procedure	Remedy
Communication cannot be established or is unstable.	Transmission path	Check that all the stations properly respond to the Ping command.	Check the power supplies and cables of stations that fail to respond.
		Check whether the collision lamp frequently glows.	Check that the cables and connectors are not in poor contact. With an analyzer, identify abnormalities.
		Check that the repeater power supply is ON.	Check for a faulty power supply, disconnected power cable, and improper voltage.
	Communicating-station device setup	Check that the network IP address is properly set.	With the support tool and analyzer, ensure that the IP address setting is correct.
		Check that the device station number is properly set.	With the support tool and analyzer, ensure that the station number setting is correct.
		Check that the device parameters are properly set.	With the support tool, ensure that the device parameter settings are correct.
		Check whether the CD (carrier detection) lamp glows steadily or intermittently.	Ensure that the communication cable, hub power supply, and other items are normal.
		Check whether the COM lamp glows steadily or intermittently.	Ensure that the device settings are correct.

(2) Checking the IP address with the personal computer's "ping feature"

The connection and IP address setting for the target FL-net device can be checked without using the FL-net network analyzer or any other special tool. Such a check can also be conducted with the "Ping feature" of a general-purpose personal computer or the like. The table below outlines the IP address checkout procedure that is to be performed with the "Ping feature."

The IP connection can be checked with the "Ping" command. The procedure is outlined below.

① Select Start button – [(All) Programs] – [Accessories] – [Command Prompt] to display [Command Prompt].

② Enter the "Ping" command to conduct a basic communication test for the purpose of checking the communication between the link unit and personal computer. To make a Ping command entry, type in "Ping [IP-address]" or "Ping [host-name] ."
Example (IP address): Ping 192.168.250. 13
When the target FL-net device is properly set, the following message is displayed.

```
Pinging 192.168.250. 13 with 32 bytes of data
Reply from 192.168.250. 13: bytes=32 time=2 ms TTL=32
Reply from 192.168.250. 13: bytes=32 time=1 ms TTL=32
Reply from 192.168.250. 13: bytes=32 time=1 ms TTL=32
Reply from JEMA 192.168.250. 13 : bytes=32 time=1 ms TTL=32
C: \WINDOWS>
```

③ If an NG result is obtained (the connection is not verified), the following timeout indication is displayed.

```
Pinging 192.168.250. 13 with 32 bytes of data:
Request timed out.
Request timed out.
Request timed out.
Request timed out.
C: \WINDOWS>
```

7.3 Precautions for FL-net Use

For the FL-net transmission path requirements, see the aforementioned section or IEEE802.3 standard. In addition to such requirements, you must observe the FL-net-specific precautions.

- Ensure that no other Ethernet communication data flows along the FL-net communication cable.
- Do not connect the FL-net to a router.
- The use of a switching hub for the FL-net does not produce any beneficial effect.
- The use of infrared, radio, or like medium may substantially decrease the real-time capability of communications.
- When you use a personal computer, the real-time capability of communications may substantially change depending on the personal computer capacity and the employed OS and applications.
- Use a specified IP address only. The same network address must be used (the standard network address is 192.168.250.). For the node number (station number) for the IP address, the following input range is recommended. When a node number is set, it is not checked for duplication until it is actually used for communication. If a duplicate node number is used for communication, a node number duplication error occurs. Exercise care not to set a duplicate node number.

Network address	Node number
192.168.250.	**1 to 249**

- Make a proper ground wire connection. Ensure that the employed ground cable has a sufficient thickness.
- Ensure that the FL-net is positioned at an adequate distance from a noise source. Also, avoid installing the FL-net together with a mains power line or the like.
- When cyclic data communication and message data communication are simultaneously effected, the real-time capability may deteriorate depending on the data amount and the like.
- Cyclic data communication areas (common memory areas) need not contiguously be allocated.
- The regular communication capability depends on the processing capacities of the connected devices. The entire communication is effected with the communication processing speeds of all the networked devices adjusted for the slowest device’s communication processing capability (minimum permissible frame interval). Therefore, the connection or addition of a single device may significantly deteriorate the real-time capability of the entire system.
- Although the header section of message data communications is in big-endian format, the data section is in little-endian format. Note, however, that the system parameters in the data section for a profile read are in big-endian format (the use of the big-endian format transmits the MSB first).

7.4 Error Indications and Countermeasures

(1) Tool’s error indications

For error indications given by the tool, see “5.5.7 RAS information.”

(2) Error log information to the LPU module

The collected error information in the LPU module can be referenced from [Error log] of [S10V BASE SYSTEM]. For details, refer to “S10V USER’S MANUAL BASIC MODULES (manual number SVE-1-100).”

[FA protocol errors]

When an error stated in the FA protocol is detected during an FL.NET module operation, the LPU unit module shows a error code as recorded in Table 7-2.
The FL.NET module may stop running in compliance with the FA protocol, depending on the contents of the displayed message.

[Hardware errors]

When the FL.NET module detects a hardware error, the LPU unit module shows an error code in record with Table 7-2. The FL.NET module also illuminates the error LED and operation comes to a stop.

If a checksum error occurs in the contents of parameter setup, this results in a parameter error. When this parameter error occurs, open the Parameter setup screen of the installed model and change the setup as required.

Table 7-2 Error Messages (1/2)

No.	Error code	Error message	Meaning	LER LED	ERR LED	Required recovery action
1	3621	Program error (Invalid Data Access)	Data access error	– (Turned off)	Turned on	Hardware item replacement
2	3661	Program error (Data Access Protection)	Data access protection error	–	Turned on	Hardware item replacement
3	3601	Program error (Data Page Fault)	Data access page fault	–	Turned on	Hardware item replacement
4	3421	Program error (Invalid Inst. Access)	Instruction access error	–	Turned on	Hardware item replacement
5	3461	Program error (Inst. Access Protection)	Instruction access protection error	–	Turned on	Hardware item replacement
6	3401	Program error (Instruction Page Fault)	Instruction access page fault	–	Turned on	Hardware item replacement
7	3031	Program error (Inst. Alignment Error)	Instruction alignment error	–	Turned on	Hardware item replacement
8	3081	Program error (Privileged Instruction)	Privileged-instruction error	–	Turned on	Hardware item replacement
9	3041	Program error (Illegal Instruction)	Illegal-instruction error	–	Turned on	Hardware item replacement
10	3391	Program error (FP Program Error)	Floating-point arithmetic error	–	Turned on	Hardware item replacement
11	3471	Program error (Data Alignment Error)	Data alignment error	–	Turned on	Hardware item replacement
12	5130	Undefined Macro	Undefined macro issued	–	Turned on	Hardware item replacement
13	5110	Macro parameter error	Incorrect macro parameter used	–	Turned on	Hardware item replacement
14	5C70	WDT timeout error	Watchdog timer timeout	–	Turned on	Hardware item replacement
15	3B70	Module error (Bus Target Abort)	Bus target aborted	–	Turned on	Hardware item replacement
16	5000	Module error (Invalid Interrupt)	Invalid interrupt detected	–	Turned on	Hardware item replacement
17	5001	Module error (Undefined Invalid Interrupt)	Undefined invalid interrupt detected	–	Turned on	Hardware item replacement
18	5002	Module error (INTEVT Invalid Interrupt)	INTEVT invalid interrupt detected	–	Turned on	Hardware item replacement
19	50F1	Module error (HERST Invalid Interrupt)	Serious-error invalid interrupt detected	–	Turned on	Hardware item replacement
20	50F2	Module error (HERST2 Invalid Interrupt)	Serious-error invalid interrupt of type 2 detected	–	Turned on	Hardware item replacement
21	50F3	Module error (BUERRSTAT Invalid Interrupt)	Bus serious-error interrupt status invalid	–	Turned on	Hardware item replacement
22	50F6	Module error (NHPMCLG Invalid Interrupt)	Memory serious-error interrupt status invalid	–	Turned on	Hardware item replacement
23	50F7	Module error (ECC 2bit Master Invalid Interrupt)	Memory ECC 2-bit serious-error status invalid	–	Turned on	Hardware item replacement
24	50F8	Module error (RERRMST Invalid Interrupt)	RERR interrupt status invalid	–	Turned on	Hardware item replacement
25	50C1	Module error (NINTR Invalid Interrupt)	NINT status invalid	–	Turned on	Hardware item replacement
26	50B1	Module error (PUINTR Invalid Interrupt)	PUINT status invalid	–	Turned on	Hardware item replacement
27	5051	Module error (RINTR Invalid Interrupt)	RINT status invalid	–	Turned on	Hardware item replacement
28	5031	Module error (LV3 INTST Invalid Interrupt)	Level-3 interrupt status invalid	–	Turned on	Hardware item replacement
29	5032	Module error (RQI6 INF Invalid Interrupt)	RQI6 status invalid	–	Turned on	Hardware item replacement
30	5011	Module error (RQI3 INT Invalid Interrupt)	RQI3 status invalid	–	Turned on	Hardware item replacement
31	5012	Module error (RQI3 Link Invalid Interrupt)	RQI3 link status invalid	–	Turned on	Hardware item replacement
32	5013	Module error (RQI3 Module Invalid Interrupt)	RQI3 module status invalid	–	Turned on	Hardware item replacement
33	7D01	Module error (INVALID EXCEPTION)	Invalid exception detected	–	Turned on	Hardware item replacement
34	7D10	Module error (INVALID MAIN/SUB SWITCH SETTING)	Main-/submodule setting switch operation error	–	Turned on	Set the switch correctly.
35	0114	Module error (INVALID MAC ADDRESS)	MAC address error	–	Turned on	Hardware item replacement
36	7D12	Module error (MAIN/SUB SWITCH SETTING DUPLICATION)	Duplicate setting of main-/submodule setting switch	–	Blinking	Set the switch correctly.
37	7D13	Module error (ETHERNET LSI CHECK ERROR)	LANCE diagnosis error	–	Turned on	Hardware item replacement
38	7D14	Module error (SDRAM CHECK ERROR)	SDRAM initialization error	–	Turned on	Hardware item replacement
39	7D15	Module error (OS-ROM CHECKSUM ERROR)	ROM checksum error (in CPMS)	–	Turned on	Hardware item replacement
40	7D16	Module error (LQE702 CANNOT BE MOUNTED ON THE SAME MOUNT BASE AS LQE500/LQE502/LQE701)	A model LQE702 module is mounted on the same mount base in conjunction with a model LQE500/LQE502/LQE701 module.	–	Turned on	The module cannot be mounted on the same mount base on which a model LQE500/502/701 module is mounted. Remove the model LQE500/502/701 module from the mount base.

Table 7-2 Error Messages (2/2)

No.	Error code	Error message	Meaning	LER LED	ERR LED	Required recovery action
41	7D18	Module error (TASK-ROM CHECKSUM ERROR)	ROM checksum error (communication task)	–	Turned on	Hardware item replacement
42	010B	Parameter type Mismatch (It is setted for S10mini)/ROM3 checksum error	ROM3 checksum error (communication parameter)	–	Turned on	Communication parameter re-setting
43	D010	Module error (Memory Alarm)	Memory single-bit error (solid)	–	Turned on	Hardware item replacement
44	D330	Module error (Hardware WDT timeout)	Hardware watchdog timer timeout	–	Turned on	Hardware item replacement
45	D340	Module error (Software WDT Timeout)	Software watchdog timer timeout	–	Turned on	Hardware item replacement
46	D810	System down (BPU Error)	BPU error	–	Turned on	Hardware item replacement
47	3389	System down (FP Unavailable)	Floating-point unavailability exception	–	Turned on	Hardware item replacement
48	30F9	System down (Illegal Exception)	Illegal-exception error	–	Turned on	Hardware item replacement
49	5700	System down (System Error)	System down (system error)	–	Turned on	Hardware item replacement
50	5800	System down (Kernel Trap)	System down (kernel trap)	–	Turned on	Hardware item replacement
51	7308	I/O error (SEND_TIMEOUT)	Transmission timeout error	–	Turned on	Hardware item replacement in case it is not recovered along with power recovery.
52	730A	I/O error (RESET_ERROR)	Hardware resetting error	–	Turned on	Hardware item replacement in case it is not recovered along with power recovery.
53	730E	I/O error (MEMORY)	Memory error	–	Turned on	Hardware item replacement in case it is not recovered along with power recovery.
54	7310	I/O error (LOSS)	Carrier loss error	–	–	Transmission path checkup (*1)
55	7311	I/O error (RETRY)	Retry error	–	–	Transmission path checkup (*3)
56	7312	I/O error (LATE)	Late-collision error	–	–	Transmission path checkup (*5)
57	7351	I/O error (TX_ABORT)	Abnormal end of transmission	–	–	Transmission path checkup
58	7353	I/O error (TX_DEFER)	Transmission error due to transmission delay	–	–	Transmission path checkup
59	7370	I/O error (EC_PCI_ERROR)	PCI error detected in communication LSI	–	Turned on	Hardware item replacement (*6)
60	7376	I/O error (TX_DATA_UNDER)	Communication data FIFO underrun	–	–	Transmission path checkup (*6)
61	7375	I/O error (RX_STAT_OVER)	Reception-status FIFO overrun	–	–	Check the load on the communication line. (*6)
62	7377	I/O error (RX_DATA_OVER)	Received-data FIFO overrun	–	–	Check the load on the communication line. (*7)
63	0113	I/O error (IP_ADDERSS_NOT_REGISTERED)	IP address not set yet	–	Blinking	Register the IP address.
64	7400	I/O error (PCI_BUS_ERR)	PCI bus error	–	Turned on	Hardware item replacement
65	7505	I/O error (INV_INTR)	Invalid interrupt generated from the communication line	–	Turned on	Hardware item replacement
66	7508	I/O error (BUF_OVF)	OS-managed send/receive buffer overflowed.	–	Turned on	Check the load on the communication line. (*2)
67	750F	I/O error (SOCKET_OVF)	Socket table is full.	–	–	Check the load on the communication line.
68	7510	I/O error (IFCONFIG_UP)	Network interface initialization error	–	Turned on	Review the settings made.
69	7512	I/O error (IPADDR_DUPL)	Duplicated-IP address error	–	Turned on	Review the settings made. (*4)
70	0200	Network participation not completed.	The FL.NET module currently not participating in the network	Turned on	–	Connect it to the network.
71	0201	Duplicate common memory settings	The local node's common memory setting overlaps with another distant node's common memory setting.	–	Blinking	The local node's common memory setting is zero (0). Disconnect the local node from the network and set common memory again.
72	0202	Duplicate node numbers	Another distant node uses the same node number as the local node in the same network.	Turned on	Blinking	Change either of the duplicated node numbers so that they will be unique in the network.
73	0203	FL.NET module setting error	The FL.NET module's setup is erroneous.	Turned on	Blinking	By using the setup tool, check the setup for error and correct the error. If this action does not solve the problem, replace the module with a new one.
74	0204	Token hold timeout	The defined token hold time was exceeded three times consecutively.	Turned on	–	Hardware item replacement

(*1) This message is displayed once when a total of 32 LSI carrier losses are detected. The LSI carrier loss occurs when an attempt is made to transmit data in the OFF state of the 100M LINK LED indicator, which indicates that a “link not established yet” condition exists. The LSI carrier loss also occurs when, after the startup of the LPU, a total of 32 send requests are issued by an FL.NET system program before the 100M LINK LED indicator comes on.

(*2) This type of error occurs due to an insufficient buffer space in a “high communication load” condition.

(*3) This message is displayed once when a total of 32 retries are made successively.

(*4) The IP address is duplicated with some other computer’s. If the local computer is connected to the network after the other computer, the ERR LED indicator will be turned on.

(*5) This message is displayed once when a total of eight such collisions are detected successively. If a total of 16 or more such collisions are detected successively, the message will be displayed once when a total of 256 such collisions are detected.

(*6) This message is displayed once when a total of five such errors or underruns/overruns are detected successively.

(*7) This message is displayed once when a total of 10 such overruns are detected successively.

[Code table of errors that can be detected by C mode handler and mathematical/logical function]

Table 7-3 shows error codes and remedies concerning the errors that may occur when the C mode handler or mathematical/logical function issues a request to the FL.NET module.

Table 7-3 List of Detectable Codes (1/2)

Error code	Description	Cause	Remedy
0x0000	Normal end of message handling	――――	――――
0x0001	Message response error	An abnormal response message is received from a specified node number.	The contents of the abnormal response message are stored in the error message storage table. Refer to the instruction manual for the specified node and check the status of the specified node.
0x0002	Message support not provided	The specified node does not support the user-requested message feature.	Do not issue any unsupported message to the node.
0xFE00	Parameter error	A user-specified parameter is abnormal. If a transparent message reception request has been issued, the associated transparent message is not received.	Check the parameters used when the request was issued. Do not initiate a transparent message reception until message receptions are verified.
0xFE01	Self-node not connected	The FL.NET module has not participated in the network.	Issue a request after the FL.NET module has participated in the network.
0xFE02	Specified node not connected	No node having a user-specified node number has participated in the network.	Specify the node number of a node that has participated in the network.
0xFE03	Message processing already in progress	The newly issued request cannot be accepted because the previously issued request is being processed.	Issue the new request again after the processing of the previously issued request terminates.
0xFE04	Message ACK response not received	An ACK response is not received from the node having a specified node number.	It is conceivable that the module may be defective. Replace the module.
0xFE06	No data received	No response to a message request was received within 30 seconds after message request issuance to a specified node number.	

Table 7-3 List of Detectable Codes (2/2)

Error code	Description	Cause	Remedy
0xFE08	ACK reception sequence number error	An ACK response returned from a specified node number reported a sequence number error.	It is conceivable that the module may be defective. Replace the module.
0xFE09	ACK reception sequence number version error	An ACK response returned from a specified node number reported a sequence number version error.	
0xFE12	Message queue full	The message queue for a specified node number is full. The specified node number cannot receive a request.	Reissue the request after a while or decrease the number of requests to the specified node number.
0xFE13	Initialization error	Message processing initialization is not completed for a specified node number.	Reissue the request after a while.
0xFE16	Message size error	A specified node number reported that an abnormal message size was requested by the self-node.	It is conceivable that the module may be defective. Replace the module.
0xF0XX or 0xFFXX	Driver abnormal	An abnormality was detected by a driver when a user-requested message was transmitted.	

(3) Module error indications

When an error occurs in the FL.NET module, the LER LED or ERR LED on the module turns ON or blinking.

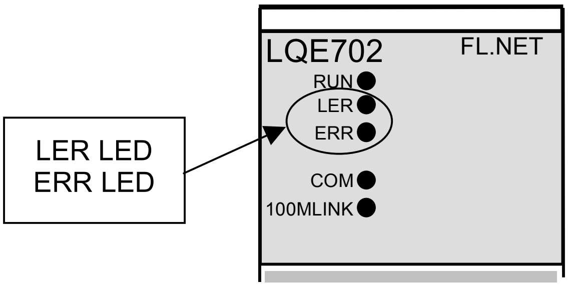


Figure 7-2 Module Error Lamp

Note that the LER LED turns ON even when the module has not participated in the network. Therefore, the LER LED indication is not adequate for judging whether the module is abnormal.

When the LER LED turns ON, error information is collected but no error is indicated to the LPU module. The collected error information can be referenced from [Error log] of [S10V BASE SYSTEM]. For confirming the collected error information, refer to “S10V USER’S MANUAL BASIC MODULES (manual number SVE-1-100).”

(4) Communication driver error alarm

The FL.NET module does not indicate an error alarm that is issued by a communication driver.

(5) Viewing the communication log data

You can view the communication log maintained in the module with the setup tool named [S10V FL.NET SYSTEM].

For detailed operating procedure, see “5.5.7 RAS information.”

This Page Intentionally Left Blank

8 APPENDIXES

8.1 Network System Definitions

8.1.1 Standard compliance of communication protocol

A communication protocol is a set of rules (communication regulations) to enable a system to exchange information with another system through a communications link. The communication protocols adopted by the FL-net comply with the following standards:

Table 8-1 FL-net Communication Protocols

FL-net communication protocol	Standard complied with
FL-net	FA Control Network [FL-net (OPCN-2)] – Protocol Specification (issued by Japan Electrical Manufacturers' Association (JEMA)
UDP	RFC768
IP ICMP, etc.	RFC791,792,919,922,950
ARP, etc.	RFC826,894
–	IEEE802.3

8.1.2 Communication protocol hierarchical structure

A communication protocol is modeled in hierarchical form to express and standardize a communication process by systematically dividing it into various levels. The FL-net consists of the following six protocol layers.

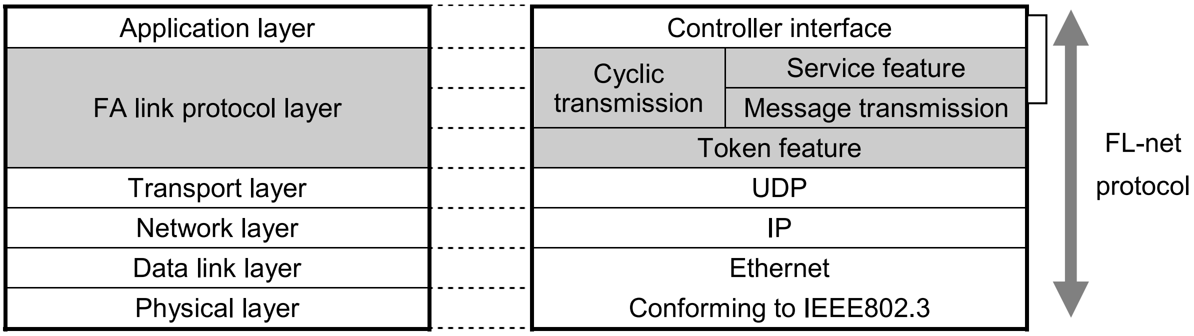


Figure 8-1 FA Link Protocol Definition

8.1.3 FL-net physical layer

For a transmission speed of 100 Mbps, the Ethernet physical layer offers five different transmission methods: 100BASE-TX, 100BASE-FX, 100BASE-T, 100BASE-T2, and 10BASE-T4.
The FL-net adopts 100BASE-TX.

8.1.4 IP address

An address named “IP address (INET address)” is used to differentiate a specified communication device from many other communication devices connected to Ethernet. Therefore, all the communication devices connected to Ethernet must have unique IP addresses.
The IP address is divided into two sections. One section shows a network address to which a communication device is connected. The other section indicates a host address for the communication device. Three different network classes (A, B, and C) are used depending on the network size (Classes D and E are additionally used for special purposes).

Table 8-2 IP Address Classification

	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 have the same network address section and a unique host address section that is not duplicated within the network.
The FL-net uses a Class C IP address. It is recommended that you set the network address to “192.168.250.N” (N is a node number between 1 and 254). It is also recommended that the low-order host address coincide with an FL-net protocol node number.

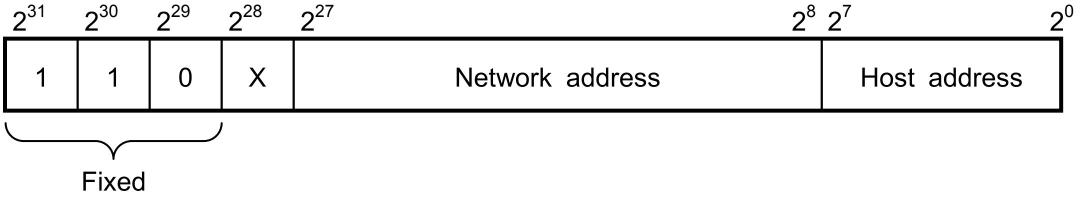


Figure 8-2 FL-net IP Address

8.1.5 Subnet mask

In compliance with the FL-net protocol, the subnet mask is fixed at 255.255.255.0. The FL-net user 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.

8.1.6 TCP/IP and UDP/IP communication protocols

TCP, UDP, and IP are important protocols that Ethernet uses.
IP corresponds to the communication protocol's network layer and controls the flow of communications data.
TCP and UDP correspond to the transport layer. They both use IP as a network layer but greatly differ in the contents of services.
TCP offers highly reliable services without causing a higher layer to be aware of data segments.
UDP, on the other hand, functions to transmit a datagram (one pack or unit of information) from IP to a higher layer, but does not guarantee the delivery of data to a transmission destination. It lets a higher layer perform data acknowledgment, retransmission, and other processes.
UDP is not as reliable as TCP. However, it offers communication services with a minimum of overhead.
The FL-net uses UDP. The reason is that TCP's complicated data verification/retransmission procedure is redundant for the FL.NET. The FL-net provides an increased data exchange speed by skipping such a complicated procedure and having the FL-net protocol layer on a higher level perform token-based transmission right management, multiple-frame division/synthesis, and other relevant processes.

8.1.7 Port numbers

For the FL-net, the following port numbers are predetermined 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 with parameters or the like.

Table 8-3 FL-net Port Numbers

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

8.1.8 FL-net data format

(1) FL-net data format overview

The data transmitted/received by the FL-net is capsuled in each communication protocol layer as indicated below:

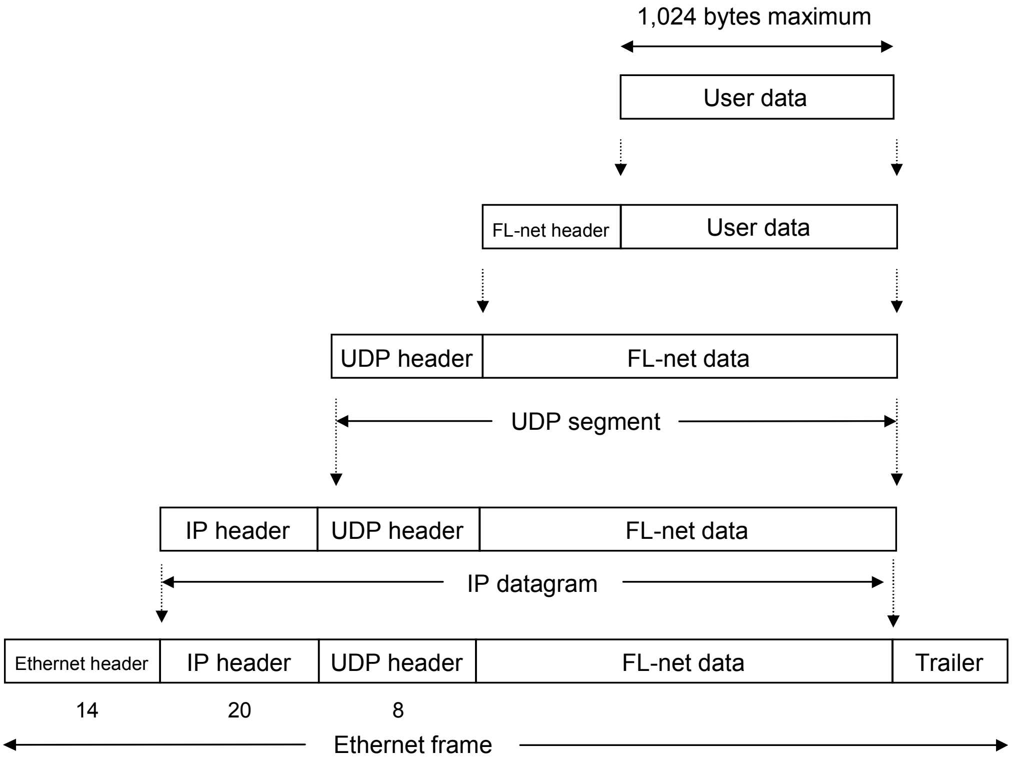
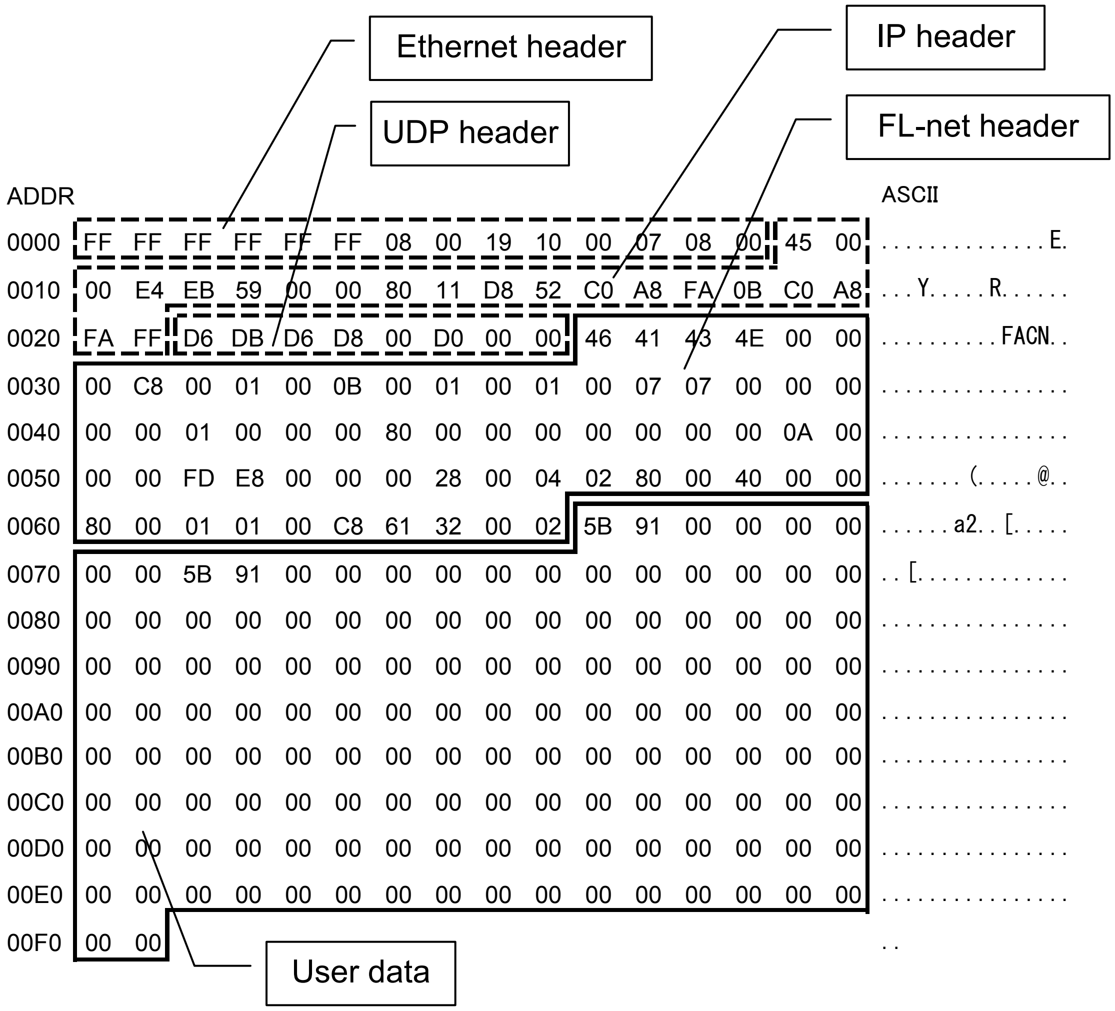


Figure 8-3 FL-net Data Format Overview

FL-net data (one-frame) observable over a communications line is indicated in the example below. In this example, 128-byte cyclic data is transferred.



```
ADDR                                                              ASCII
0000  FF FF FF FF FF FF 08 00 19 10 00 07 08 00 45 00   ..............E.
0010  00 E4 EB 59 00 00 80 11 D8 52 C0 A8 FA 0B C0 A8   ...Y.....R......
0020  FA FF D6 DB D6 D8 00 D0 00 00 46 41 43 4E 00 00   ..........FACN..
0030  00 C8 00 01 00 0B 00 01 00 01 00 07 07 00 00 00   ................
0040  00 00 01 00 00 00 80 00 00 00 00 00 00 00 0A 00   ................
0050  00 00 FD E8 00 00 00 28 00 04 02 80 00 40 00 00   .......(.....@..
0060  80 00 01 01 00 C8 61 32 00 02 5B 91 00 00 00 00   ......a2..[.....
0070  00 00 5B 91 00 00 00 00 00 00 00 00 00 00 00 00   ..[.............
0080  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
0090  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
00A0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
00B0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
00C0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
00D0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
00E0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
00F0  00 00                                             ..
```

Figure 8-4 FL-net Data (One-frame) Example

(2) FL-net header format

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

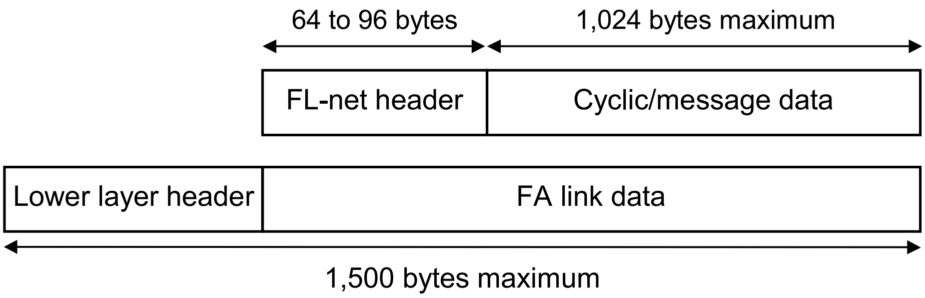


Figure 8-5 FL-net Header

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

8.1.9 FL-net transaction codes

The FL-net offers the following message transmission services:

- Word block data read
- Word block data write
- Network parameter read
- Network parameter write
- Stop directive (support for request only)
- Run directive (support for request only)
- Profile read (support for response only)
- Log data read
- Log data clear
- Message return
- Transparent type

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

Table 8-4 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	Enter 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

8.1.10 Transaction code receive operation at the UDP port

The following table shows the operations matched to the transaction codes upon receipt of frames at the UDP port defined in FL-net.

Table 8-5 Transaction Code Receives Processing at the UDP Port

Transaction code	For token frame or cyclic frame (UDP port = 55000)	For message frame (UDP port = 55001)	For trigger frame or participation request frame (UDP port = 55002)
00000 to 09999	Abandonment	Processing or abandonment	Abandonment
10000 to 59999	Abandonment	Processing (For transparent type message)	Abandonment
60000 to 64999	Abandonment	Abandonment	Abandonment
65000 to 65001	Processing	Abandonment	Abandonment
65002	Abandonment	Abandonment	Processing
65003 to 65011	Abandonment	Processing	Abandonment
65012	Abandonment	Abandonment	Processing
65013 to 65016	Abandonment	Processing	Abandonment
65017 to 65199	Abandonment	Unsupported processing	Abandonment
65200 to 65202	Abandonment	Abandonment	Abandonment
65203 to 65211	Abandonment	Processing	Abandonment
65212	Abandonment	Abandonment	Abandonment
65213 to 65216	Abandonment	Processing	Abandonment
65217 to 65399	Abandonment	Processing or abandonment	Abandonment
65400 to 65535	Abandonment	Abandonment	Abandonment

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

8.2 FL-net Network Management

8.2.1 FL-net token management

(1) Token

Under normal conditions, a node can transmit data while it holds 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 an enter request frame for situations where network enter is not completed.

- The FA net circulates one token among nodes.
- Upon receipt of the token, a node holds the right to transmit data to the network until it passes the token to the next node.
- The token circulates among all nodes that have entered in the FL-net.
- The token is monitored by a timer in such a manner that it is automatically reissued if it stops circulating through the network for a predetermined period of time.
- If two or more tokens should exist within the network, a unification feature works so that only one token prevails.

(2) Flow of token

Basically, only one token exists within the network.

If two or more tokens should exist within the network, the one having the lowest destination node number takes precedence with the others discarded.

A frame having a token (token frame) has a token's destination node number and a token transmission node number.

When a node matches the token's destination node number contained in the received token frame, it becomes a token holder.

The order of token rotation is determined by node numbers.

Token rotation for nodes occurs in the ascending order of node registrations in the entering node management table.

A node having the highest node number passes the token to a node having the lowest node number.

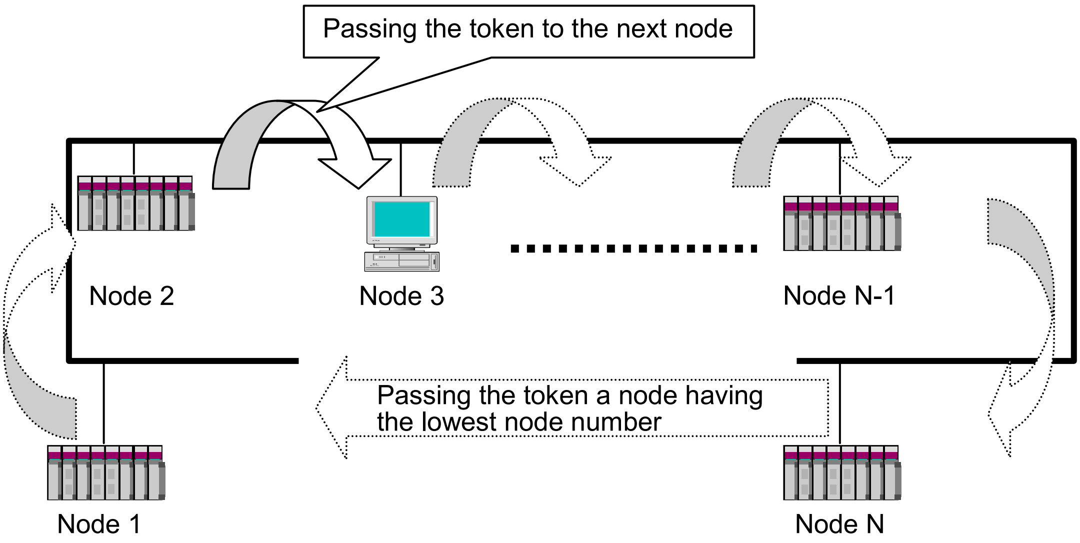
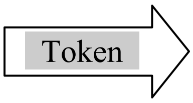
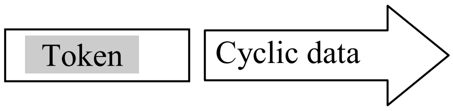
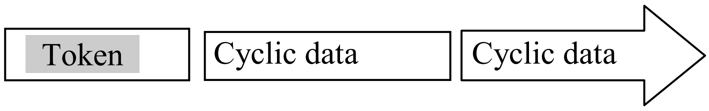
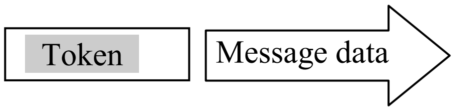
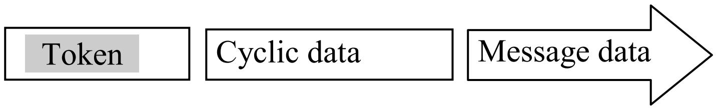
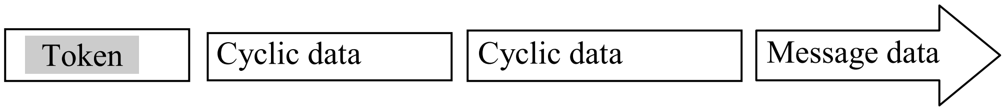

Passing the token to the next node
Node 2
Node 3
Node N-1
Passing the token a node having the lowest node number
Node 1
Node N

Figure 8-6 Token Flow

(3) Token and data

The following six different data patterns are used for token transmission:

Table 8-6 Token and Data

Pattern	Description
When no data is involved	Only the token is transmitted.
	
When only cyclic data is transmitted	A token is transmitted after cyclic data is transmitted.
	
When only cyclic data is transmitted after being divided into frames	Cyclic data is transmitted and, after the last frame, a token is transmitted.
	
When only message data is transmitted	The token is transmitted subsequently to message data transmission.
	
When cyclic data and message data is transmitted	Message data and cyclic data are transmitted, and then a token is transmitted.
	
When cyclic data and message data are transmitted with the cyclic data divided into frames	Message data and cyclic data are transmitted and, after the last frame, a token is transmitted.
	

(4) Frame interval (minimum permissible 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 a frame transmission.

The minimum permissible frame interval is the minimum period of time for which a node must wait to initiate a frame transmission.

In FL-net, this minimum permissible frame interval is commonly used within the whole network.

The maximum value of the minimum permissible frame intervals set by entering nodes in the network is calculated and updated for all nodes upon each node enter/leave.

8.2.2 Entering in/leaving from FL-net

(1) Entering in FL-net

When a node starts up, it monitors the line until an enter token detection timeout occurs. If it does not receive the token in this instance, it concludes that the network startup process is in progress, and then newly participates in the network. If it receives the token, on the other hand, it concludes that it is in a midway enter state, and then participates in the network on the spot.

(a) New enter

If a node does not receive the token within the enter token detection timeout period, the node prepares for trigger transmission and then start a transmission after an elapse of predetermined time which is determined by multiplying the remainder of node number/8 by 4 ms. If trigger reception occurs before trigger transmission, the node does not transmit a trigger. For a period of enter request frame reception wait time (1,200 ms) after trigger reception, all nodes wait to transmit an enter request frame while checking for a duplicate node number or address and updating the entering node management table. When the enter request frame transmission wait time (node number × 4 ms) elapses after trigger reception, the nodes transmit an enter request frame. If a node recognizes a duplicate address in accordance with an enter request frame of another node, the former node sets the common memory starting address and common memory size for area 1 and 2 to 0 and does not transmit cyclic data. When a node recognizes a duplicate address, it sets an address duplication flag and resets a common memory data validity notification flag. When an enter request frame reception wait timeout occurs, a entering node having the lowest node number transmits the token first in accordance with the entering node management table. When a node recognizes a duplicate node number, it starts no transmission/reception.

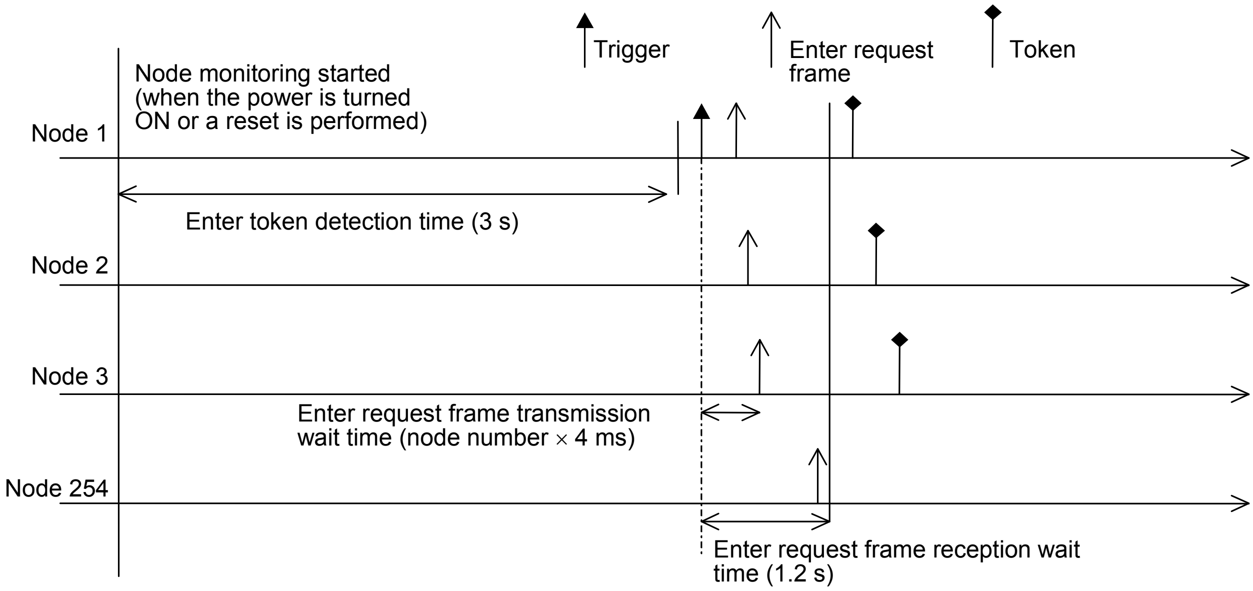

Trigger
Enter request frame
Token
Node monitoring started (when the power is turned ON or a reset is performed)
Node 1
Enter token detection time (3 s)
Node 2
Node 3
Enter request frame transmission wait time (node number × 4 ms)
Node 254
Enter request frame reception wait time (1.2 s)

Figure 8-7 Startup Timing Diagram 1

(b) Midway enter

If a node receives the token within the enter token detection time, it concludes that a link is already established, and does not transmit an enter request frame until the token makes three rounds. A frame received during such an interval is used to check for node number, address, and other duplications and update the entering node management table. If an address duplication is detected in such an instance, the common memory starting address and common memory size of area 1 and 2 are set to 0 and cyclic data transmission does not take place. When a node recognizes an address duplication, it sets an address multiplexing flag and resets the common memory data validity notification flag. If the node number is normal, the node transmits an enter request frame after an elapse of the enter request frame transmission wait time. The enter request frame is sent no matter whether the token is held. When a node recognizes a node number duplication, it does not transmit an enter request frame and refrains from entering in the network.

<Remarks>

Enter token detection time:

Period of time for checking whether the network is operating.

Round:

Rounds are determined with respect to the time at which the token addressed to the lowest node number is received.

Enter request frame transmission wait time:

A enter request frame transmission takes place after an elapse of [self-node number × 4 ms] in order to avoid a conflict with any other newly entering node.

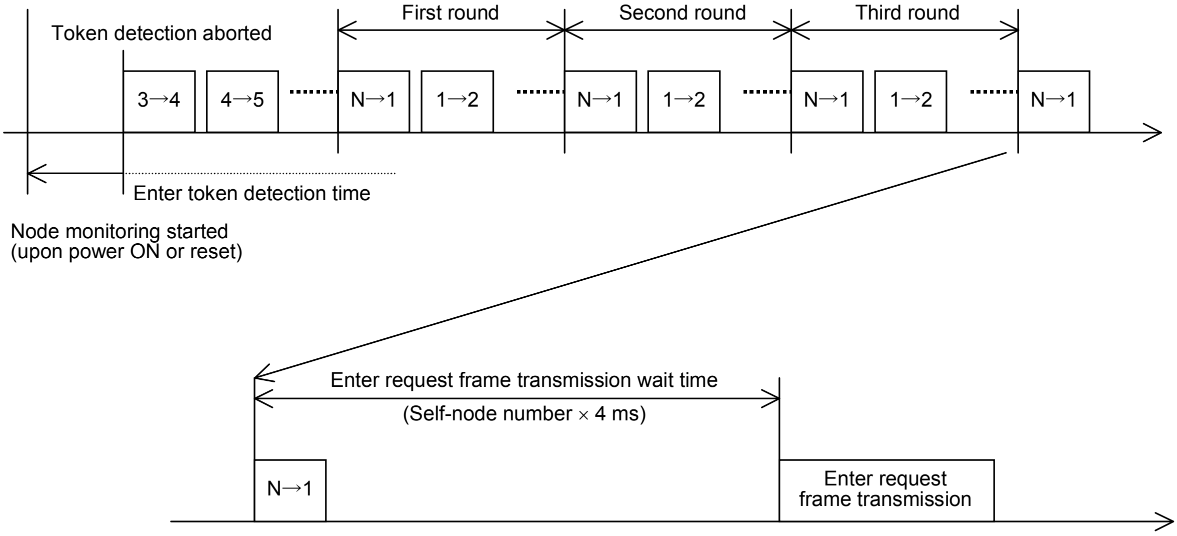


Figure 8-8 Startup Timing Diagram 2

(2) Leaving from FL-net

A node number check is conducted upon each token frame reception. If the token frame is not received from a certain node three successive times, that node is considered to have left from the FL-net (including a case where a token holder node does not transmit the token even after an elapse of the token monitoring time).

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

8.2.3 Node status management

As outlined in Table 8-7, node status management is exercised with the self-node management table, entering node management table, and network management table.

Table 8-7 Node Status Management Table Overview

Name	Description
Self-node management table	Manages the self-node settings.
Entering node management table	Manages the information about nodes that have entered in the network.
Network management table	Manages the common information about the entire network.

8.2.4 FL-net self-node management table

(1) Basic features

This table manages the data about self-node settings as summarized below:

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

(2) Management data

Table 8-8 Self-node Management Table

Item	Length	Description
Node number	1 byte	1 to 254
Area 1 data starting address for common memory	2 bytes	Word address (0 to 0x1FF)
Area 1 data size for common memory	2 bytes	Size (0 to 0x1FF)
Area 2 data starting address for common memory	2 bytes	Word address (0 to 0x1FFF)
Area 2 data size for common memory	2 bytes	Size (0 to 0x1FFF)
Higher layer status	2 bytes	RUN/STOP/ALARM/WARNING/NORMAL
Token surveillance timeout	1 byte	Variable in 1 ms units
Minimum frame interval	1 byte	Variable in 100 μs units
Vendor name	10 bytes	Name of the vendor
Maker form	10 bytes	Maker form/device name
Node name (equipment name)	10 bytes	User-defined node name
Protocol version	1 byte	Fixed at 0x80
FA link status	1 byte	Entering/leaving, etc.
Self-node status	1 byte	Node number duplication detection, etc.

8.2.5 FL-net entering node management table

(1) Basic features

The status of nodes that have entered in the network is monitored with the management table that each node has. This table handles on an individual node basis the data about nodes that have entered in the network. The basic functions of this table are summarized below:

- A token frame is received at startup to update the entering node management table and network management table.
- Upon each token frame reception, each node updates the entering node management table.
- Upon receipt of a new node’s enter request frame, the entering node management table is updated.
- When a node’s token frame no-reception condition or timeout is detected three successive times, the node is deleted from the table.

(2) Management data

The token of each node is constantly monitored to create and manage the entering node management table.

Table 8-9 Entering Node Management Table

Item	Length	Description
Node number	1 byte	1 to 254
Higher layer status	2 bytes	RUN/STOP/ALARM/WARNING/NORMAL
Area 1 data starting address for common memory	2 bytes	Word address (0 to 0x1FF)
Area 1 data size for common memory	2 bytes	Size (0 to 0x1FF)
Area 2 data starting address for common memory	2 bytes	Word address (0 to 0x1FFF)
Area 2 data size for common memory	2 bytes	Size (0 to 0x1FFF)
Refresh cycle time	2 bytes	Variable in 1 ms units
Token surveillance timeout	1 byte	Variable in 1 ms units
Minimum frame interval	1 byte	Variable in 100 µs units
FA link status	1 byte	Entering/leaving information, etc.

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

<Remark> Contained in a received token frame.

8.2.6 FL-net status management

(1) Basic feature

Provides management of parameters related to network status.

(2) Management data

Table 8-10 Network Management Table

Item	Length	Description
Token maintenance node number	1 byte	Node that currently holds the token.
Minimum frame interval	1 byte	Variable in 100 μs units.
Refresh cycle time	2 bytes	Variable in 1 ms units.
Refresh cycle measurement time (current)	2 bytes	Variable in 1 ms units.
Refresh cycle measurement time (maximum)	2 bytes	Variable in 1 ms units.
Refresh cycle measurement time (minimum)	2 bytes	Variable in 1 ms units.

8.2.7 FL-net message sequence number management

(1) Basic feature

Manages the message transmission sequence number and sequence number/version number.

(2) Transmission management data

Table 8-11 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-N transmission)	4 bytes	0x1 to 0xFFFFFFFF
Sequence number (One-to-one transmission)	4 bytes	0x1 to 0xFFFFFFFF

“0xFFFFFFFF” is a hexadecimal number (FFFFFFFF hex).

(3) Reception management data

Table 8-12 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-N reception)	4 bytes	0x1 to 0xFFFFFFFF

“0xFFFFFFFF” is a hexadecimal number (FFFFFFFF hex).

8.3 FL-net Profile

8.3.1 Device communication information classification

In the FL-net, the communications-related information about networked devices is classified into three types as shown in Figure 8-9.

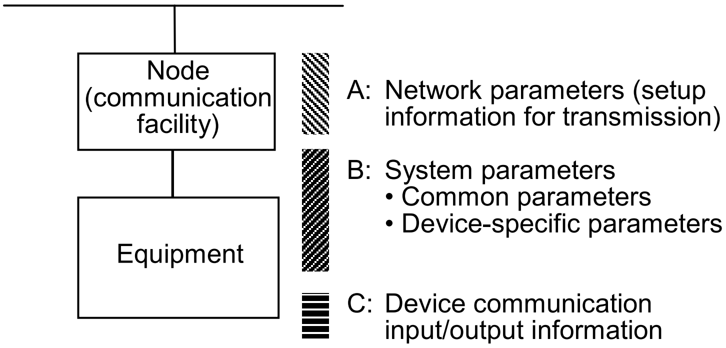

Node (communication facility)
Equipment
A: Network parameters (setup information for transmission)
B: System parameters
• Common parameters
• Device-specific parameters
C: Device communication input/output information

Figure 8-9 Device Communication Information Classification

- The network parameters (A) are essential to the information setup for transmission.
- The system parameters (B) are static parameters, which furnish management information for identifying the networked devices. These parameters are divided into common parameters and device-specific parameters.
- The device communication input/output information can be accessed from remote networked devices whenever it is needed for applications. This information includes dynamic information, which varies with application operations and device conditions.

8.3.2 Details of common parameters

Table 8-13 shows the details of common parameters.

Table 8-13 Details of Common Parameters

Parameter name	Name character 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, (0001-9999), [INTEGER], 1, (01-12), [INTEGER], 1, (01-31)	Example: 2, 1998 1, 9 1, 30
Device category	10, “DVCATEGORY”	PrintableString	Example: 3, “PLC” (see Note)
Vendor name	6, “VENDOR”	PrintableString	Example: 4, “MSTC”
Product model	7, “DVMODEL”	PrintableString	Example: 3, “JOP”

Note: The contents of the device category parameter are indicated below:

“PC” or “PLC”..........	Programmable controller
“NC” or “CNC”	Computer numerical control device
“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 syntactical rule for transfer, the entire system parameter set or common parameter set, the system parameter revision date, or the entire device-specific parameter set (optional) must be of a SEQUENCE structure type (device-specific parameter inner structuring is optional).

8.3.3 Details of device-specific parameters (when used)

Table 8-14 Details of Device-Specific Parameters

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

8.3.4 System parameter examples (PLC examples)

(1) PLC example of system parameter tabular document notation

Table 8-15 System Parameter Tabular Document Notation (PLC Examples)

Parameter name	Name character 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, (0001-9999), [INTEGER], 1, (01-12), [INTEGER], 1, (01-31)	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 character 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"
                        }
            plcmpara    {
                        nameID                          "ID",
                        paraID                          "DEVPARA",
                        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) Data arrangement on circuit

The sequence of data transmitted over a circuit is shown below.

Data transmission begins with address (0) in the relative address 00 column shown below. Data is transmitted, byte by byte, in the order indicated by the horizontal arrow mark. Upon completion of the transmission of the data in the relative address 00 column, the data in the relative address 10 column begins to be transmitted. Transmission continues in the same manner in the order of relative address.

Relative address → Data (in 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”									

8.3.5 System parameter examples (CNC examples)

Table 8-16 shows CNC system parameter examples.

Table 8-16 System Parameter Tabular Document Notation (CNC Examples)

Parameter name	Name character 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, (0001-9999), [INTEGER], 1, (01-12), [INTEGER], 1, (01-31)	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 character 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 data listed below.

System			
Option configuration flag	5, “SFLAG”	BIT STRING	8, “00100101”
Control axis count	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
                       }
```

<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  }
                  }
    }
```

8.4 Trouble Report

Fill out this form and submit it to local source.

Your company name			Person in charge	
Data and time of occurrence				(year / month / day / hour / minute)
Where to make contact	Address			
	Telephone			
	FAX			
	E-mail			
Model of defective module			LPU model	
OS Ver. Rev.		Program name:	Ver.	Rev.
Support program		Program name:	Ver.	Rev.
Symptom of defect				
Connection load	Type			
	Model			
	Wiring state			
System configuration and switch setting				
Space for correspondence				