e-Front runners

MICREX-SX series

SPH

USER'S MANUAL

Ethernet Interface Module

Type: NP1L-ET1 (10BASE-T, 100BASE-TX)
NP1L-ET2 (10BASE5)

FEH259a

Preface

This user's manual describes the specifications of the MICREX-SX series Ethernet interface module, NP1L-ET1 or NP1L-ET2, for connecting to Ethernet. Read this manual carefully to ensure correct operation. When using modules or peripheral devices, be sure to read the corresponding user's manuals listed below:

<Related to IEC conformed loader D300win>

Title	Manual No.	Contents
User's Manual Instruction, MICREX-SX series SPH	FEH200	Explains the memory, language and system definitions of the MICREX-SX series.
User's Manual Hardware, MICREX-SX series SPH	FEH201	Explains the system configurations and the hardware specifications of modules in the MICREX-SX series.
User's Manual D300win <Reference>, MICREX-SX series	FEH257	Explains the installation procedure, functions and operating method of D300winV3.
User's Manual LD/FBD Editor Operations, MICREX-SX series Guide, MICREX-SX series	FEH257-1	Explains the operating instructions of the LD/FBD editor which is added to D300winV3 as new function.

<Related to the Standard loader>

Title	Manual No.	Contents
User's Manual Instruction, MICREX-SX series	FEH588	Explains the memory, language and system definitions of the MICREX-SX series, viewed from the Standard loader.
User's Manual Hardware, MICREX-SX series SPH	FEH586	Explains the system configuration, the hardware specifications and operations of modules in the MICREX-SX series.
User's Manual Standard loader <Reference>, MICREX-SX series	FEH590	Explains the functions and operating method of the Standard loader.

* This manual is structured such that you can consult it, whether you use SX-Programmer Expert (hereinafter merely called "300win") or SX-Programmer Standard (hereinafter merely called "Standard loader"). For the Standard loader, word address is expressed by decimal number (the same as for MICREX-F series). When described merely as "loader" in this manual, it applies to both types of loader.

* In addition to the above manuals, the following Fuji Electric FA Components & Systems Co., Ltd. site offers various manuals and technical documents associated with MICREX-SX.
URL http://www.fujielectric.co.jp/fcs/eng/index.html

Notes

1. This manual may not be reproduced in whole or part in any form without prior written approval by the manufacturer.
2. The contents of this manual (including specifications) are subject to change without prior notice.
3. If you find any ambiguous or incorrect descriptions in this manual, please write them down (along with the manual No. shown on the cover) and contact FUJI.

Safety Precautions

Be sure to read the "Safety Precautions" thoroughly before using the module.
Here, the safety precautions items are classified into "Warning" and "Caution".

Warning	: Incorrect handling of the device may result in death or serious injury.
⚠ Caution	: Incorrect handling of the device may result in minor injury or physical damage.

Even some items indicated by "Caution" may result in a serious accident.
Both safety instruction categories provide important information. Be sure to strictly observe these instructions.

Warning
◊ Do not touch charged parts such as terminals while the power is ON. Touching such parts may result in electric shocks. ◊ Be sure to turn the power OFF before mounting or removing parts or components, wiring operations, maintenance, or inspection. Working with the power ON may cause electric shocks, malfunctions, or failures. ◊ Configure the emergency stop, interlocking, and other circuits outside the PLC. Otherwise, errors, if any, in the PLC may result in mechanical damages or accidents. ◊ The liquid (liquid crystal) of the liquid crystal display is harmful. If the display panel is damaged, the liquid flowing out of it must not get into your mouth. If your skin or clothes are stained with the liquid, rinse them with soap and water.

Safety Precautions

⚠Caution

◊ Do not use the product or parts found damaged or deformed when unpacking. If done, fire, malfunctions, or failures may occur.
◊ Do not give shock to the product by dropping or turning it over. If done, the product may be damaged or failures may occur.
◊ When mounting the product, follow the directions given in the User’s Manual and other manuals. The product incorrectly mounted may be dropped or cause malfunctions or failures.
◊ Use the rated voltage and current specified in the User’s Manual and other manuals. Otherwise, fire, malfunctions, or failures may occur.
◊ Use or keep the product under the environmental conditions described in the User’s Manual and other manuals. Use or storage of the product under high temperature, high humidity, dew condensation, dust, corrosive gas, oil, organic solvent, or excessive vibration or shock may result in electric shock, fire, malfunctions, or failures during operation.
◊ Select wires with the size appropriate to the applied voltage and current and tighten the wire terminals with the specified torque. Inappropriate wiring or tightening may cause fire, dropping of the product, malfunctions, or failures.
◊ Mount the product so that no foreign matters such as dust, wire chips, or iron powder do not enter it. Otherwise, fire, accidents, malfunctions, or failures may occur.
◊ After wiring, be sure to remove the dust cover from the module and unit. Using the product without removing the dust cover may cause fire, accidents, malfunctions, or failures.
◊ Periodically check that the terminal and mounting screws are securely tightened. Using the product with the screws not securely tightened may cause fire or malfunctions.
◊ Be sure to place the attached connector cover on a connector unused. Otherwise, malfunctions or failures may occur.
◊ Completely check safety when transferring screen data during system operation. Otherwise, incorrect operation might cause a machine damage or accident. Operation errors, if any, may suddenly start the machine resulting in accidents or damages to the machine.
◊ Insert the loader connectors in the correct direction. Wrong connections may cause malfunctions.
◊ Before operating the PLC, touch a grounded metallic object, etc. to discharge static electricity from your body. Excessive static electricity may cause malfunctions or failures.
◊ Carry out wirings securely and correctly, observing the instructions described in the User’s Manual and other manuals. Incorrect wirings may cause fire, malfunctions, or failures.
◊ Do not hold the cord when disconnecting the plug from the receptacle. Otherwise, the cable may be broken resulting in fire or failures.
◊ Do not press the key switches with tweezers, a tool or other thing having a sharp edge, nor press them too strongly. Otherwise, they might be flawed or damaged.
◊ If the product has a trouble, contact us for repair; never attempt to repair it in the field. When replacing the battery, be careful of the connecting directions. Wrong connections may cause fire, accidents or failures.
◊ Do not remodel or reassemble the product. Doing so may cause failures.
◊ When discarding the product, handle it as industrial waste observing the related regulations.
◊ The product described in this User’s Manual has not been designed or manufactured for purposes of use in equipment or systems which involve human lives.
◊ If you intend to use the product described in this User’s Manual for nuclear energy control, aerospace, medical treatment, transportation, or passenger vehicles or special purposes of such systems, please consult your nearest Fuji Electric agent.
◊ When using the product described in this User’s Manual for equipment or systems which, in case of failures, may lead to loss of human lives or serious damages, be sure to provide protective measures.

Revisions

*Manual No. is shown at lower right on the cover.

Printed on	*Manual No.	Revision contents
Sep. 2002	FEH259	First edition
Mar. 2004	FEH259a	· Errors in writing in "Appendix 1-2 Transmission Data Format of Commands" were corrected. · Specifications for NP1L-ET2 (of 10BASE5 interface specification) were added. · Specification for response speed was added. · Running status information RAS was added. · The operating method when the Standard program loader is used was added.

Contents

Preface

Safety Precautions

Revisions

Contents

Page

Contents

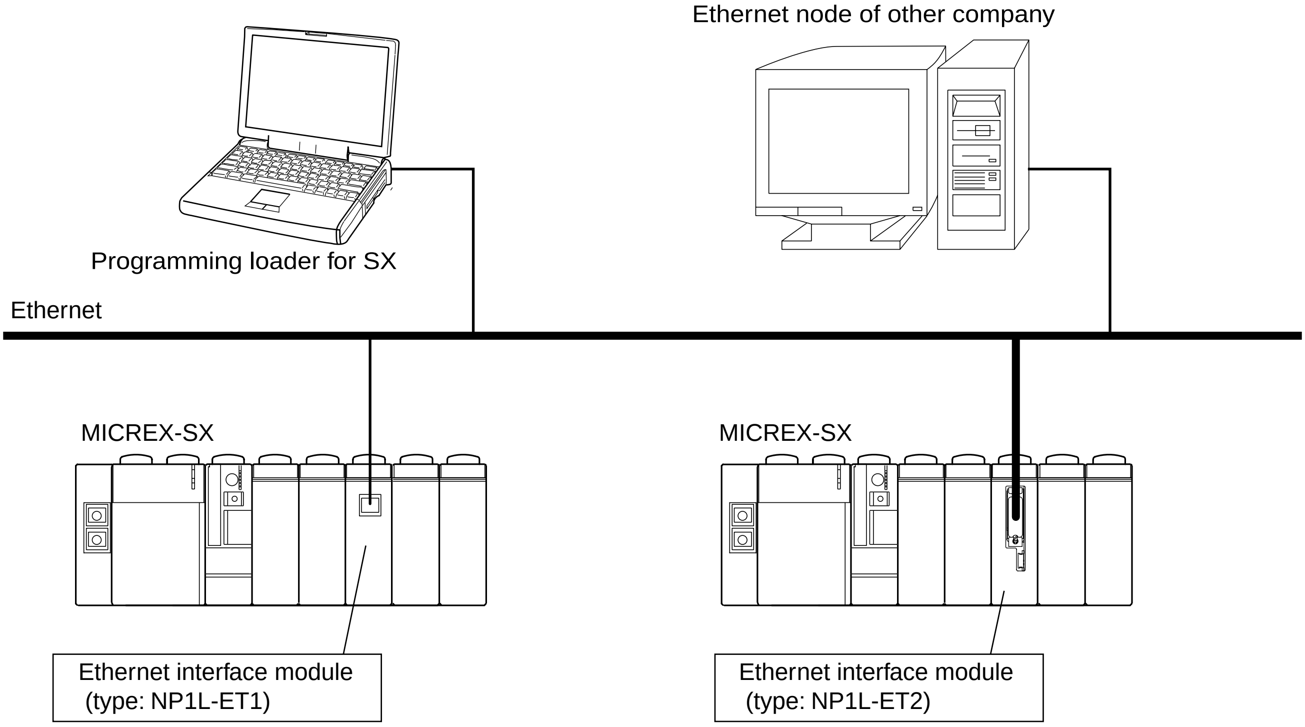
Page

Section 1 General

Section 1 General

1-1 Overview of Ethernet Interface Module

The Ethernet interface module (type: NP1L-ET1/ET2) is connected to the base board of MICREX-SX series SPH (connected to the SX bus) in order to connect the SPH to Ethernet.


Ethernet node of other company
Programming loader for SX
Ethernet
MICREX-SX
MICREX-SX
Ethernet interface module
(type: NP1L-ET1)
Ethernet interface module
(type: NP1L-ET2)

<Features>

- Because an Ethernet dedicated interface is build in the module, LAN card for Ethernet is unnecessary. (PC card interface module, or NP1F-PC2, requires such card.) As a result, no software needs to be downloaded to the module.
- NP1L-ET1 supports 10BASE-T as well as 100BASE-TX. The Auto Negotiation function automatically changes over between 10BASE-T and 100BASE-TX.
- NP1L-ET2 supports 10BASE5.

<Supported versions of CPU and Programming loader>

When you want to use the Ethernet interface module (NP1L-ET1/ET2) for communication via Ethernet, be sure to use the following version of CPU and loader.

- High-performance CPU software version: V25 or later (V58 or later when the redundancy function is used)
- Standard CPU software version: V30 or later (V42 or later when R_READ/R_WRITE FB is used)
- D300win version: V2.2.3.1 or later (However, be sure to select NP1L-ET1 when NP1L-ET2 is registered to system configuration with V2.2.3.1 of D300win.)
- Standard loader: V2.0.0.0 or later

Notes:

1) Sufficient safety measures need to be taken when installing Ethernet.
   It is recommended to entrust the installation work to an expert company.
   For the installation environment, refer to JIS X5252 (ISO 8802-3).
2) For the cable and other peripheral devices, use commercially available ones that conform to IEEE802.3.

* Ethernet is a registered trademark of U.S. based Xerox Corp.

Section 2 Specifications

Page

Section 2 Specifications

2-1 General Specifications

Item		Specification
Physical environmental conditions	Operating ambient temperature	0 to +55° C
	Storage temperature	-25 to +70° C
	Relative humidity	20 to 95%RH, no condensation
	Pollution degree	2 (no condensation)
	Corrosion immunity	Free from corrosive gases. Not stained with organic solvents
	Operating altitude	2000 m or less above sea level Transport condition: 70 kPa (equivalent to 3000 m above sea level) or more
Mechanical service conditions	Vibration	Half amplitude: 0.15 mm, Constant acceleration: 19.6 m/s^2
	Shock	Peak acceleration: 147 m/s^2 (3 cycles in each direction)
Electrical service conditions	Noise immunity	Rise time 1 ns, pulse width 1µs, 1.5 kV (noise simulator)
	Electrostatic discharge	Contact discharge: ± 6 kV, Aerial discharge: ± 8 kV (Class 3 of IEC 61000-4-2)
	Radioelectromagnetic field	80 MHz to 1000 MHz: 10 V/m (Class 3 of IEC 61000-4-3)
Isolation method		Pulse-transformer
Dielectric strength		500 V AC, 1 minute (between communication wires connected together and external power supply terminals connected together, between communication wires connected together and FG for external power supply)
Insulation resistance		10 MΩ or more with 500 V DC megger (between communication wires connected together and external power supply terminals connected together, between communication wires connected together and FG for external power supply)
Number of occupied slots		1
Internal current consumption		24 V DC, 140 mA or less (supplied from the power module via the base board)
External power supply		12 V DC, 500 mA or less (guideline: 14.28 to 15.75 V DC (NP1L-ET2 only)) (see note 2)
Installation conditions	Structure	Panel built in type, IP30
	Cooling method	Natural air cooled
	Mass	Approx. 140 g
	Outside dimensions	W35 x H105 x D87 mm (excl. protrusions)

Notes:

1) Influenced by the noise immunity of connected transceiver. Test and evaluation by Fuji Electric are performed using the wiring and installation environment described in Appendix 2-2.
2) For external power supply, use the one that meets the specifications for transceiver and transceiver cable (AUI cable).

[Reference]

IEEE 802.3 specifies as follows:

- Transceiver input terminal voltage: 12 V -6% to 15 V +15%
- DC resistance of transceiver cable: 40 Ω/km or less, max. 50 m
- Maximum current consumption of transceiver: 500 mA

2-2 Transmission Specifications

2-2-1 Transmission specifications

Item		Specification		
Type		NP1L-ET1		NP1L-ET2
Interface		10BASE-T	100BASE-TX	10BASE5
Transmission speed		10 Mbps	100 Mbps	10 Mbps
Medium (physical specification)		IEEE802.3	IEEE802.3u	IEEE802.3
Interface changeover system		Auto negotiation (automatic changeover)		–
Transmission medium		Twisted pair cable		Coaxial cable
Max. segment length		100 m		500 m
Max. number of nodes		1 piece/segment		100 pieces/segment
Communication function	General Purpose Communication mode	Performs communication between SX_CPU and other node, using TCP/IP or UDP/IP protocol. Because of free data format, it is possible to communicate with all kinds of node.		
	Fixed Buffer Communication mode	Communication function using the fixed buffer that Mitsuibishi PLC provides.		
	Loader Command Communication mode	This function can communicate dedicated loader commands between MICREX-SX series devices.		
Access method		CSMA/CD		
Transmission protocol		TCP/IP, UDP/IP, ICMP, ARP		
Max. number of mountable units		4 per configuration (* For details, refer to Section 3.)		
Max. number of simultaneous communication nodes (number of communication ports)		16 stations (ports)/module Note: The number of communication ports that can simultaneously be opened for one resource is 56. When multiple units of this module are installed or when other communication module exists, be careful that the number of opened ports per resource doesn't exceed 56.		
Transmission code		Binary/ASCII code		

Ethernet transmission frame

In Ethernet transmission frame (see the figure below), “type” or “length” is entered in the field following the source MAC address, but this module supports only “type” (DIX Ethernet specification).

Destination MAC address	Source MAC address	Type/Length	Destination IP address	• • •
6 bytes	6 bytes	2 bytes	4 bytes	• • •

2-2-2 Communication ports of the module

(1) Number of communication ports (client ports)
NP1L-ET1/ET2 has 16 ports for communication with other nodes. These 16 ports are shared by the following communication functions.

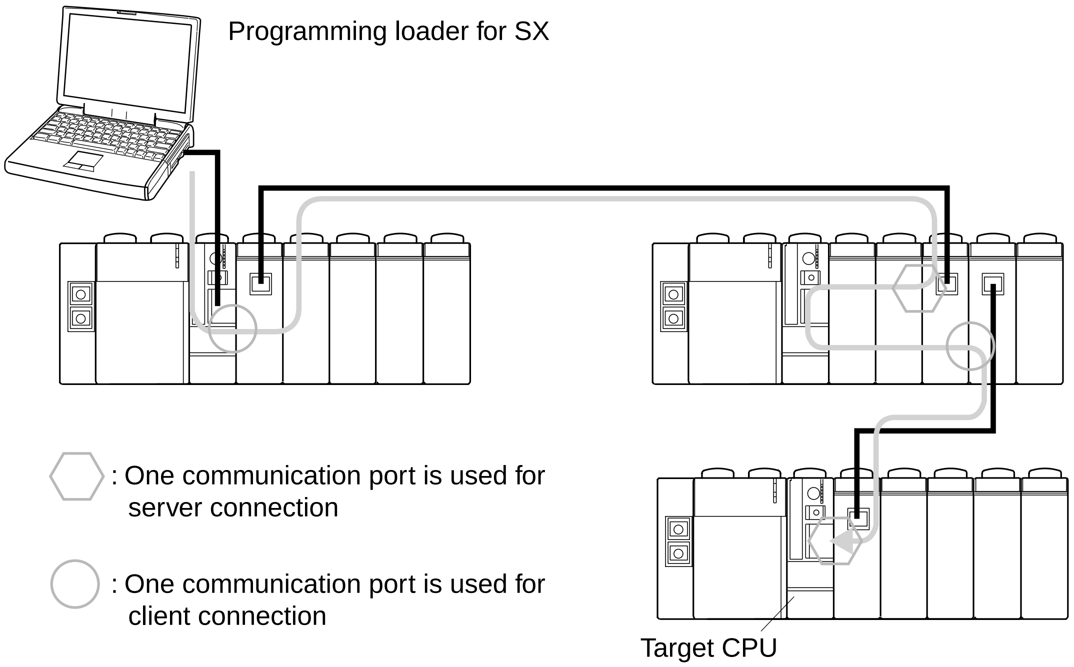
Communication function	Number of connections
1) Number of General Purpose Communication mode connections	Max. 16
2) Number of Fixed Buffer Communication mode connections	Max. 16
3) Number of Loader Command Communication (send) mode connections (see note)	Max. 4
4) Number of Loader Command Communication (receive) mode connections (see note)	Max. 8
5) Number of connections to loader	Max. 8

Note: Here "send" means the operation on the node side that sends the request of R_WRITE or R_READ; "receive", the operation on the node side that receives the request of R_WRITE or R_READ sent from a remote node.

(2) Control of communication ports
NP1L-ET1/ET2 controls communication ports according to the following rules:

- When the target is client connections (1), 2) and 3) in above table)
  Ports can be opened till there remains no free communication port on the NP1L-ET1/ET2.
- When the target is server connections (4) and 5) in above table)
  Ports can be opened till the total number of server connections becomes 8 or till there remains no free communication port on NP1L-ET1/ET2.

<Using client ports in loader network system>
When the programming loader is connected to a network, individual unit of this module supports one communication port for relaying loader commands.

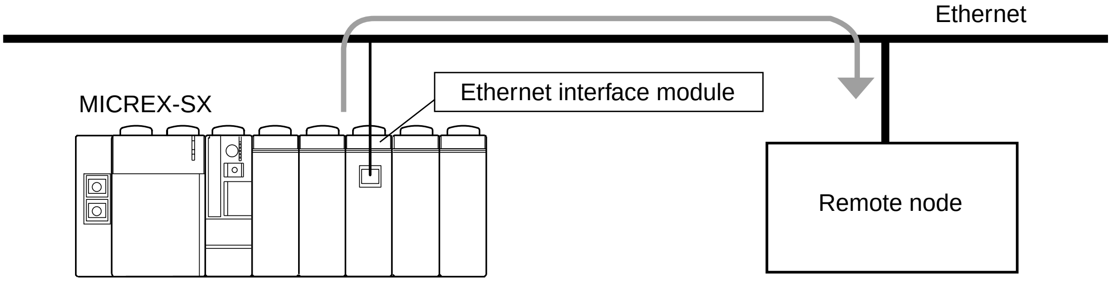

Programming loader for SX
⬡ : One communication port is used for server connection
○ : One communication port is used for client connection
Target CPU

2-2-3 Communication speed

The communication speed when message send/receive (M_SEND/M_RECEIVE) command or remote data read/write (R_READ/R_WRITE) command is used is calculated in the following manner:

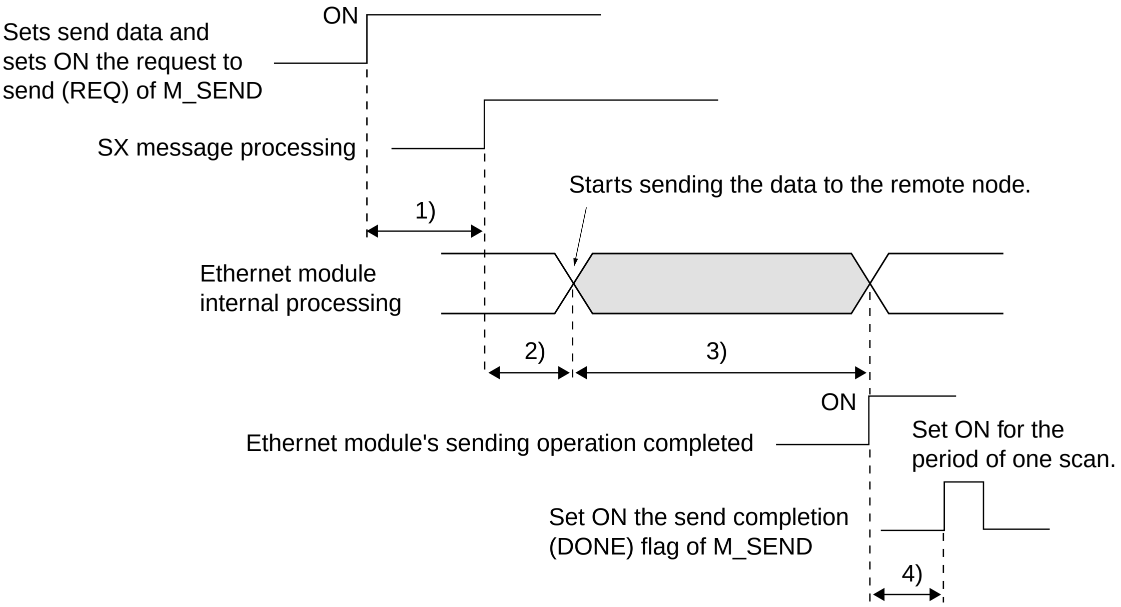
(1) When message send (M_SEND) command is used

<Example of system configuration>



<Elements for communication speed calculation>

For communication speed, the period of time till the send completion flag (DONE) of M_SEND FB is set ON since the request to send (REQ) of M_SEND FB is set ON is calculated. The elements used for this calculation are as follows:


ON
Sets send data and sets ON the request to send (REQ) of M_SEND
SX message processing
1)
Starts sending the data to the remote node.
Ethernet module internal processing
2)
3)
ON
Ethernet module's sending operation completed
Set ON for the period of one scan.
Set ON the send completion (DONE) flag of M_SEND
4)

1) SX message processing time

The time required to pass the send data from the CPU internal memory to the Ethernet module internal buffer. This processing time differs between high-performance CPU and standard CPU.

- High-performance CPU: Executes at 5-ms intervals (The volume of message data processed at a time: maximum 256 words)
- Standard CPU: Executes at the interval of SX bus tact time (The volume of message data processed at a time: maximum 256 words)

* When the volume of data exceeds 256 words, the remaining data is processed in the next cycle of SX message processing.

2) Ethernet module internal processing time

The time required to send the send data, which has been passed from the CPU to this module, on to the Ethernet transmission path.

Ethernet module internal processing time = 4 to 15 ms + (0.008 ms × Number of words to send)

3) Transmission time

The time required to pass the send data from Ethernet module to transmission path. This depends on transmission rate and the volume of data.

4) Send completion transmission time

The time required to transmit the completion of sending from Ethernet module to M_SEND FB. This transmission synchronizes with the processing of message by the CPU and therefore is performed at 5-ms intervals.

<Example of communication speed calculation>

* The conditions for this calculation are “high-performance CPU to use” and “receipt confirmation to disable”.
* For the volume of data, calculation is made in two cases: 100 words and 1017 words. (“1017 words” is an example that the operation to pass data from CPU to Ethernet is divided.)
* Transmission time is supposed to be 1 ms (approximate value when transmission speed = 10 Mbps).

Communication speed (maximum)	(Communication speed for 100 words)	=	1)×1	+2)	+3)	+4)
		=	5	+(15+0.008×100)	+1	+5
		=	26.8 ms			
	(Communication speed for 1017 words)	=	1)×1	+2)	+3)	+4)
		=	5	+(15+0.008×1017+(5+2.5)×3)	+1	+5
		=	5606 ms			

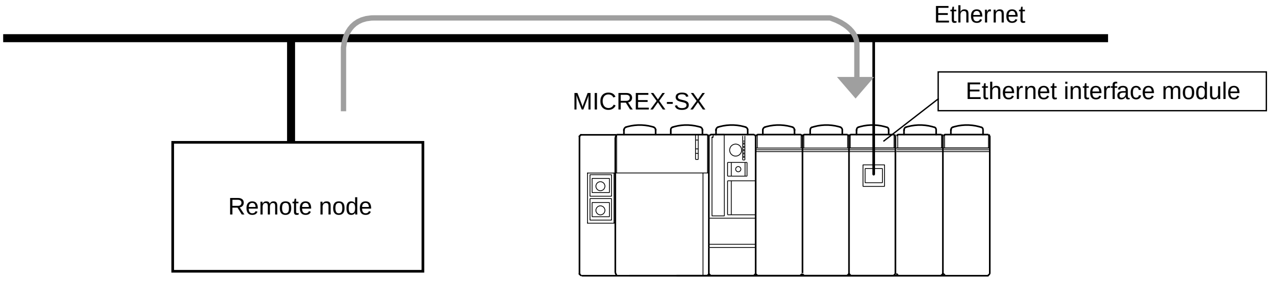
Communication speed (minimum)	(Communication speed for 100 words)	=	1)×0	+2)	+3)	+4)
		=	0	+(4+0.008×100)	+1	+0
		=	5.8 ms			
	(Communication speed for 1017 words)	=	1)×0	+2)	+3)	+4)
		=	0	+(4+0.008×1017+(5+2.5)×3)	+1	+0
		=	35.6 ms			

Notes:

1) When the volume of send data exceeds 256 words, 5 ms is added to message processing time 1), and 2.5 ms is added to Ethernet module internal processing time 2).
2) If the volume of send data is 730 words or less when receipt confirmation is enabled, 205 ms is added to the expression shown above.
If the volume of send data is 731 words or more when receipt confirmation is enabled, 5 ms is added to the expression shown above.

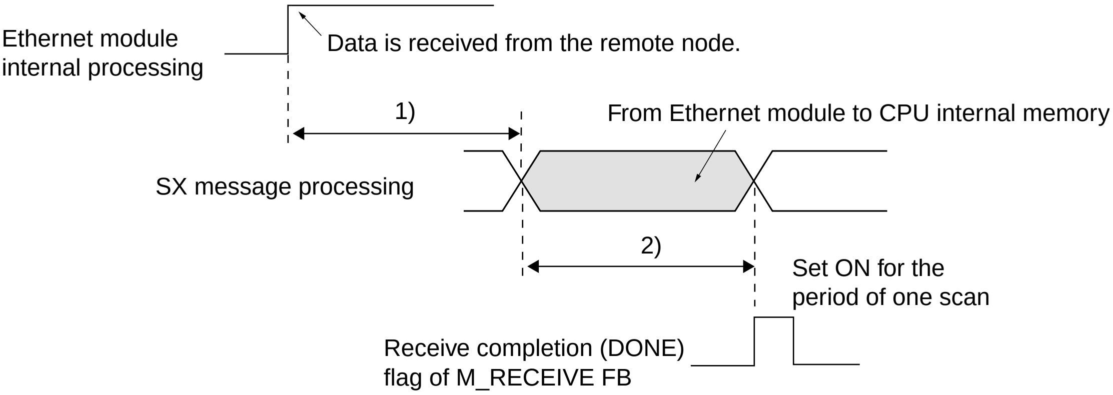
(2) When message receive (M_RECEIVE) command is used

<Example of system configuration>



<Elements for communication speed calculation>

For communication speed, the period of time till the receive completion flag (DONE) of M_RECEIVE FB is set ON since the data has been received from the remote node is calculated. The elements used for this calculation are as follows:


Ethernet module internal processing
Data is received from the remote node.
1)
From Ethernet module to CPU internal memory
SX message processing
2)
Set ON for the period of one scan
Receive completion (DONE) flag of M_RECEIVE FB

1) Ethernet module internal processing time

The time required to send the data, which has been received from the remote node and passed to the CPU module, on to the Ethernet transmission path.

Ethernet module internal processing time = 4 to 15 ms + (0.008 ms × Number of words)

2) SX message processing time

The time required to pass the receive data from Ethernet module to CPU internal memory.

This processing time differs between high-performance CPU and standard CPU.

- High-performance CPU: Executes at 5-ms intervals (The volume of message data processed at a time: maximum 256 words)
- Standard CPU: Executes at the interval of SX bus tact time (The volume of message data processed at a time: maximum 256 words)

* When the volume of data exceeds 256 words, the remaining data is processed in the next cycle of SX message processing.

<Example of communication speed calculation>

* The condition for this calculation is “high-performance CPU to use”.

* For the volume of data, calculation is made in two cases: 100 words and 1017 words. (“1017 words” is an example that the operation to pass data from CPU to Ethernet is divided.)

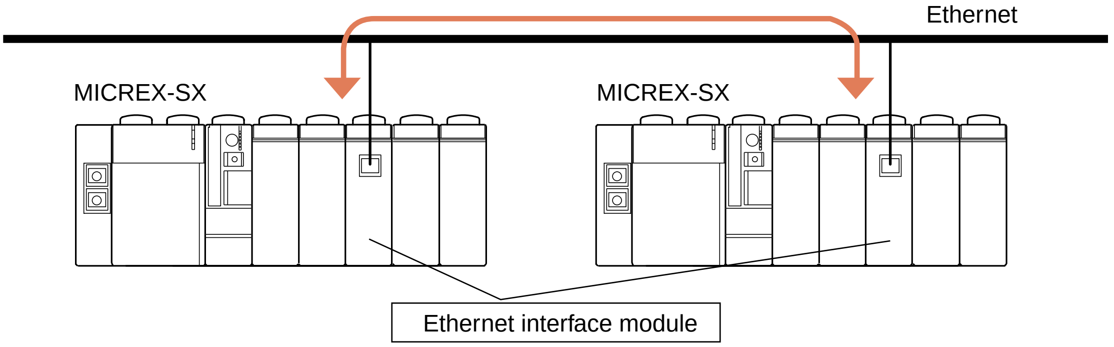
Communication speed (maximum)	(Communication speed for 100 words)	=	1)	+2)
		=	(15+0.008×100)	+5
		=	20.8 ms	
	(Communication speed for 1017 words)	=	1)	+2)
		=	(15+0.008×1017+(5+2.5)×3)	+5
		=	50.6 ms	

Communication speed (minimum)	(Communication speed for 100 words)	=	1)	+2)
		=	(4+0.008×100)	+0
		=	4.8 ms	
	(Communication speed for 1017 words)	=	1)	+2)
		=	(4+0.008×1017+(5+2.5)×3)	+0
		=	34.6 ms	

Note 1): When the volume of send data exceeds 256 words, 5 ms is added to SX message processing time 2), and 2.5 ms is added to Ethernet module internal processing time 1).

(3) When remote read/remote write (R_READ/M_WRITE) command is used

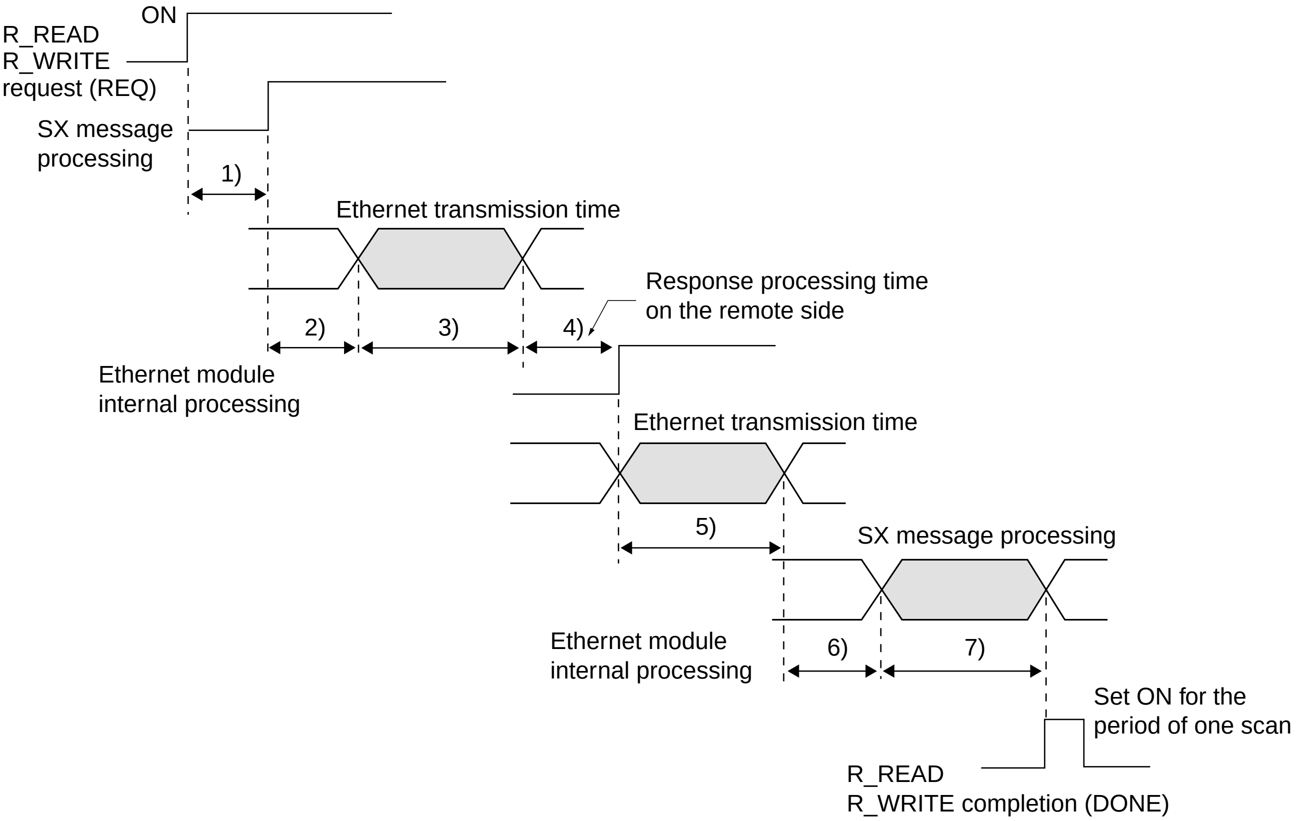
<Example of system configuration>



<Elements for communication speed calculation>

For communication speed, the period of time till the completion flag (DONE) is set ON since the request to communicate (REQ) flag of R_READ/R_WRITE FB has been set ON is calculated.

The elements used for this calculation are as follows:


ON
R_READ
R_WRITE
request (REQ)
SX message processing
1)
Ethernet transmission time
Response processing time on the remote side
2)
3)
4)
Ethernet module internal processing
Ethernet transmission time
5)
SX message processing
Ethernet module internal processing
6)
7)
Set ON for the period of one scan
R_READ
R_WRITE completion (DONE)

1), 7) SX message processing time

The time required to pass data between CPU internal memory and Ethernet module.

This processing time differs between high-performance CPU and standard CPU.

- High-performance CPU: Executes at 5-ms intervals (The volume of message data processed at a time: maximum 256 words)
- Standard CPU: Executes at the interval of SX bus tact time (The volume of message data processed at a time: maximum 256 words)

* When the volume of data exceeds 256 words, the remaining data is processed in the next cycle of SX message processing.

2) Ethernet module internal processing time

The time required to open/close communication port or send/receive data.

Maximum value for R_READ = 15 + 30 ms / Minimum value for R_READ = 7.5 + 5 ms

Maximum value for R_WRITE = 15 + (0.008 × Number of words to send) + 30 ms

Minimum value for R_WRITE = 7.5 + (0.008 × Number of words to send) + 5 ms

3), 5) Transmission time

The time required to pass the send data from Ethernet module to transmission path. This time depends on transmission rate and the volume of data.

4) Response processing time on the remote side

The time required till the remote side SX system passes response data to transmission path since it has received request data.
Maximum value for R_READ = 7.5 × 2 + (0.008 × Number of words) ms
Minimum value for R_READ = 7.5 × 2 + (0.008 × Number of words) ms
Maximum value for R_WRITE = 7.5 × 2 + (0.008 × Number of words) ms
Minimum value for R_WRITE = 7.5 × 2 + (0.008 × Number of words) ms

6) Ethernet module internal processing time

Maximum value for R_READ = 15 + (0.008 × Number of words) ms
Minimum value for R_READ = 7.5 + (0.008 × Number of words) ms
Maximum value for R_WRITE = 15 ms
Minimum value for R_WRITE = 7.5 ms

<Example of communication speed calculation for R_WRITE>

* The time required till the completion flag (DONE) is set ON since the request to write (REQ) has been set ON when high-performance CPU is used is calculated.
* For the volume of data, calculation is made in two cases: 100 words and 1017 words.
* Transmission time is supposed to be 1 ms (the approximate value when transmission rate is 10 Mbps).

Communication speed (maximum)

									Note)
(100 words)	=	1)×1	+2)	+3)	+4)	+5)	+6)	+7)	+32.5
	=	5	+15+(0.008×100)+30	+1	+7.5×2+(0.008×100)	+1	+15	+5	+32.5
	=	121.1 ms							
									Note)
(1017 words)	=	1)×1	+2)	+3)	+4)	+5)	+6)	+7)	+32.5×11
	=	5	+15+(0.008×1017)+30	+1	+7.5×2+(0.008×1017)	+1	+15	+5	+32.5×11
	=	460.1 ms							

Communication speed (minimum)

									Note)
(100 words)	=	1)×0	+2)	+3)	+4)	+5)	+6)	+7)	+15
	=	0	+7.5+(0.008×100)+5	+1	+7.5×2+(0.008×100)	+1	+7.5	+5	+15
	=	46.1 ms							
									Note)
(1017 words)	=	1)×0	+2)	+3)	+4)	+5)	+6)	+7)	+15×11
	=	0	+7.5+(0.008×1017)+5	+1	+7.5×2+(0.008×1017)	+1	+7.5	+5	+15×11
	=	263.3 ms							

Note: When the volume of data exceeds 90 words, +15 to 32.5 ms is added for every additional 90 words (every additional 97 words in the case of standard CPU).

2-3 Communication Functions

It is possible to communicate with maximum 16 stations (such as personal computer and PLC). NP1L-ET1/ET2 has the following 3 communication modes:

- General Purpose Communication mode
- Fixed Buffer Communication mode
- Loader Command Communication mode

Note: As for the number of stations to communicate with, the number of remote nodes as the total of these three communication modes must be 16 (stations) or less.

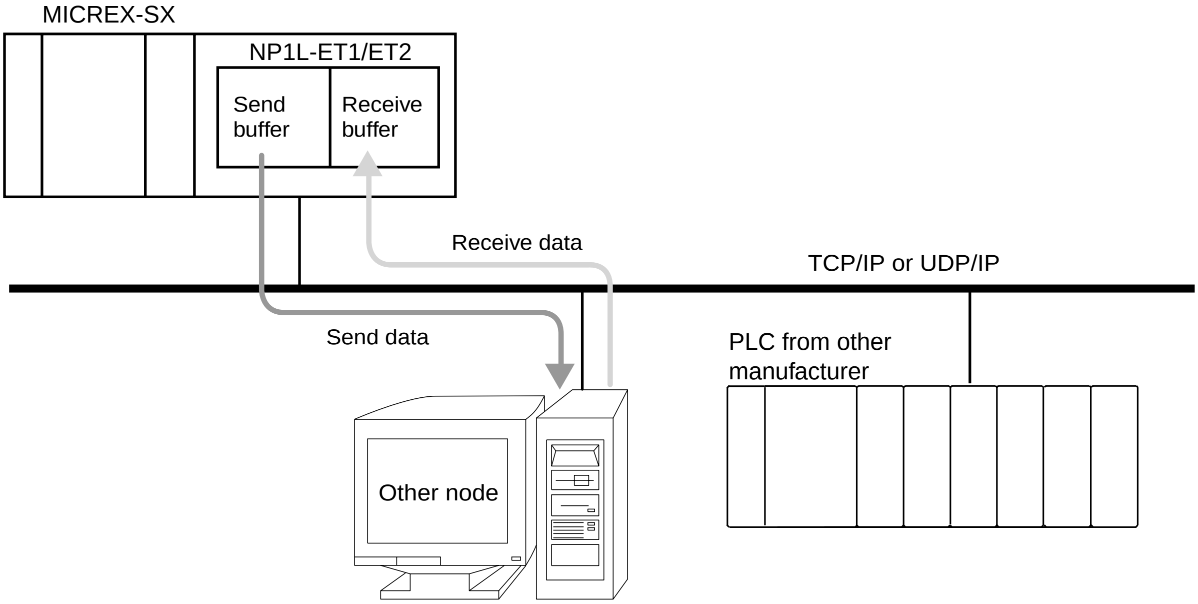
2-3-1 General Purpose Communication mode

This mode is used to communicate mainly between SX_CPU and other nodes, such as PLC systems of other manufacturers, using TCP/IP or UDP/IP protocol. To send data from SX_CPU to other nodes, M_OPEN (channel open) FB and M_SEND (message send) FB are used. To receive data from other nodes, M_OPEN (channel open) FB and M_RECEIVE (message receive) FB are used.

* For more information of concrete communication procedure, refer to "4-3-1 Communication in General Purpose Communication mode".

<Features of General Purpose Communication mode>

- Because of free data format, it is possible to communicate with all kinds of node.
- It is possible to communicate with maximum 16 stations.
- When remote node uses TPC division for data communication, maximum 730 words can be sent or received at a time. (MICREX-F, FLEX-PC, personal computer, etc.)
  When remote node uses IP division for data communication, maximum 1017 words can be sent or received at a time.
- Broadcast of UDP/IP is available.



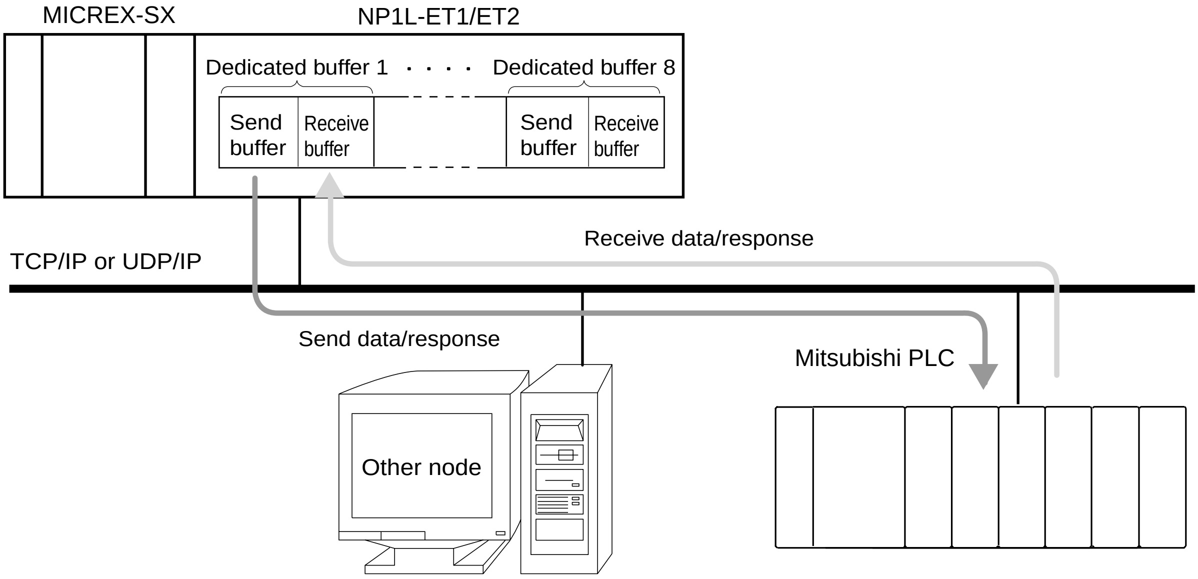
2-3-2 Fixed Buffer Communication mode

This mode is used to communicate data using the fixed buffer that type AJ71E71 Ethernet interface unit from Mitsubishi Electric Corp. provides.
This mode sets a subcommand in the data part of a TCP/IP or UDP/IP packet for General Purpose Communication mode to enable the communication of command type data, where handshaking is made for the communication between PLC and specific node.

* For more information of concrete communication procedure, refer to "4-3-2 Communication in Fixed Buffer Communication mode".

<Features of Fixed Buffer Communication mode>

- It is possible to communicate with maximum 16 stations (PLC system from Mitsubishi Electric Corp.)
- Maximum 1017 words can be sent or received at a time.



2-3-3 Loader Command Communication mode

This mode communicates data using the loader commands that are dedicated to MICREX-SX series. This mode is used to communicate data between nodes of an SX system or between an application program of a personal computer and SX_CPU. For communication between SX_CPU's, R_READ (remote data read) FB and R_WRITE (remote data write) FB are used on the request sending side. No program is necessary on the request receiving side.
For communication between personal computer and SX_CPU, the application program for sending the request to read or write needs to be created on the personal computer side. No application program is necessary on the SX_CPU side.

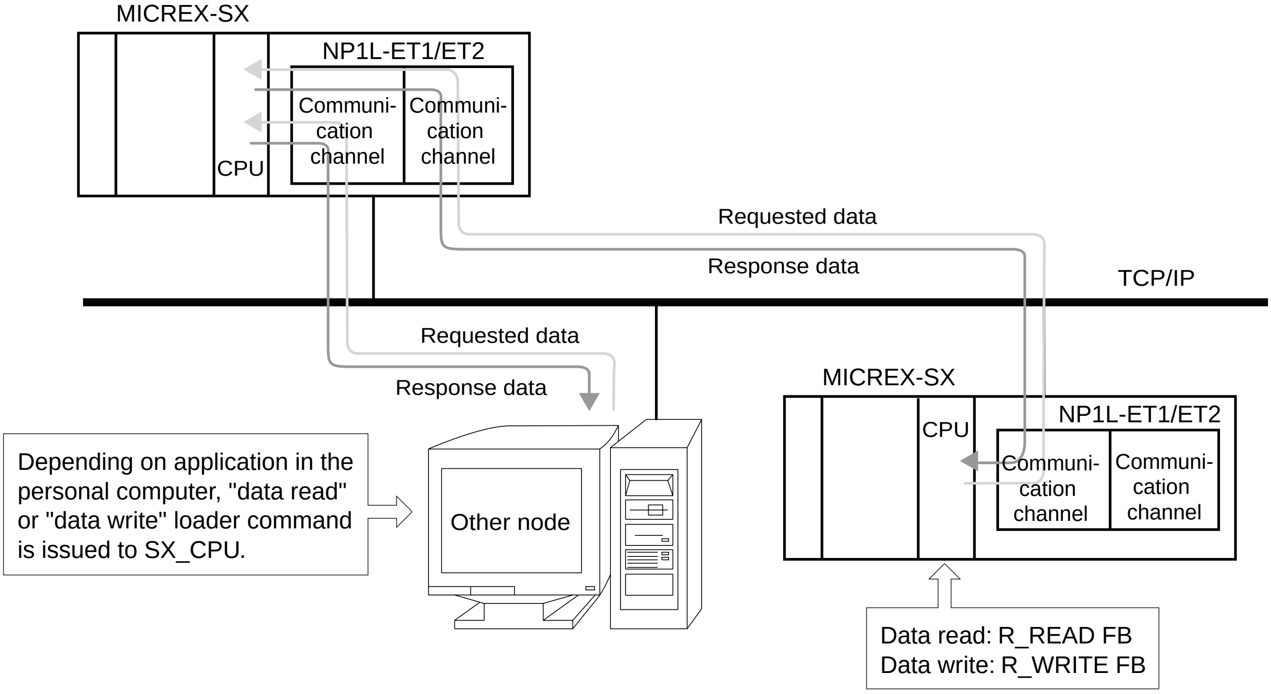
* For more information of concrete communication procedure, refer to "4-3-3 Communication in Loader Command Communication mode".

Communication conditions

Communication mode	General purpose
Communications protocol	TCP/IP
Open system	Unpassive
Local port No.	251 (loader command server), 253 (loader interface server) See note.

Note: There are the following restrictions on the use of 253 (loader interface server).
- Not adapted to the changeover to redundant CPU. Because communication is performed via Ethernet, use 251 (loader command server) when it is necessary to change over to redundant CPU.
- Detail RAS of modules other than CPU0 cannot be read via Ethernet.

<Features of Loader Command Communication mode>
- It is possible to communicate with maximum 8 stations.
- This mode is used mainly for the communication with other SX system that supports Ethernet or for the communication with personal computer.
  No application program is necessary for the communication with local node (the SX_CPU that receive loader command).



Depending on application in the personal computer, "data read" or "data write" loader command is issued to SX_CPU.

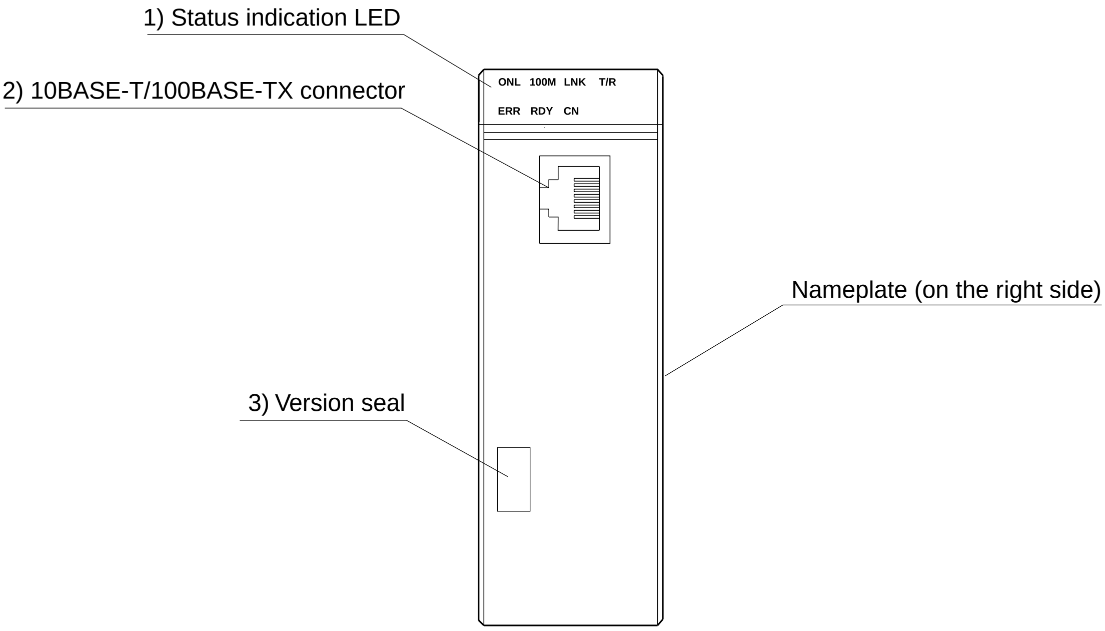
Data read: R_READ FB
Data write: R_WRITE FB

* Loader command communication port is local reference port number + 251 (or 253).
Default value is 256 + 251 = 507.
For example, when local reference port number is 1000, 1000 + 251 = 1251 becomes the port number for communicating loader commands.

2-4 Names and Functions

2-4-1 NP1L-ET1

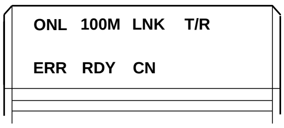
(1) Names



(2) Functions

1) Status indication LED

This LED indicates the current status of NP1L-ET1.



Symbol	Color	Description
ONL	Green	Lights when the module is operating normally; Blinks when SX bus is being connected; Unlit when SX bus is abnormal.
ERR	Red	Lights when the module is abnormal (module internal hardware error) or a cause of fatal fault is detected by software.
100M	Green	Lights when communication is performed using 100BASE-TX; unlit when communication is performed using 10BASE-T.
LNK	Green	Indicates the state of Ethernet connection. Lights when connecting to Ethernet.
T/R	Green	Indicates the state of data sending. Lights when a send/receive packet is generated.
RDY	Green	Lights when Initialization of the module is completed successfully.
CN	Green	Lights when there is at least one connection that is opened by SX user application.

2) 10BASE-T/100BASE-TX connector

10BASE-T or 100BASE-TX communication cable is connected here.

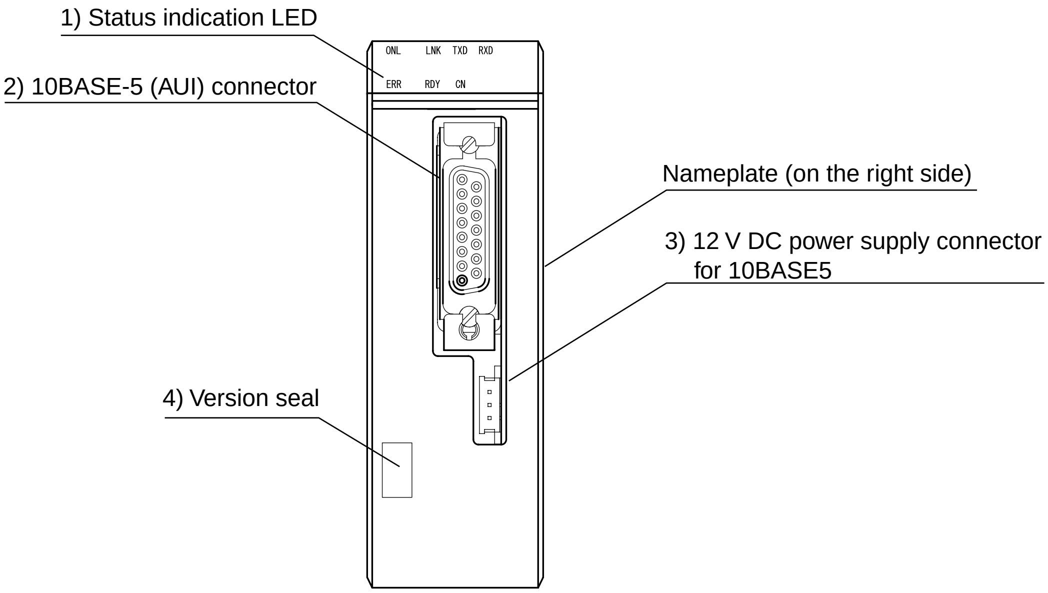
3) Version seal

Versions of the hardware and firmware of NP1L-ET1 are indicated on this seal.

20	← Hardware version
30	← Firmware version

2-4-2 NP1L-ET2

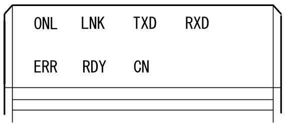
(1) Names


1) Status indication LED
2) 10BASE-5 (AUI) connector
Nameplate (on the right side)
3) 12 V DC power supply connector for 10BASE5
4) Version seal
ONL LNK TXD RXD
ERR RDY CN

(2) Functions

1) Status indication LED

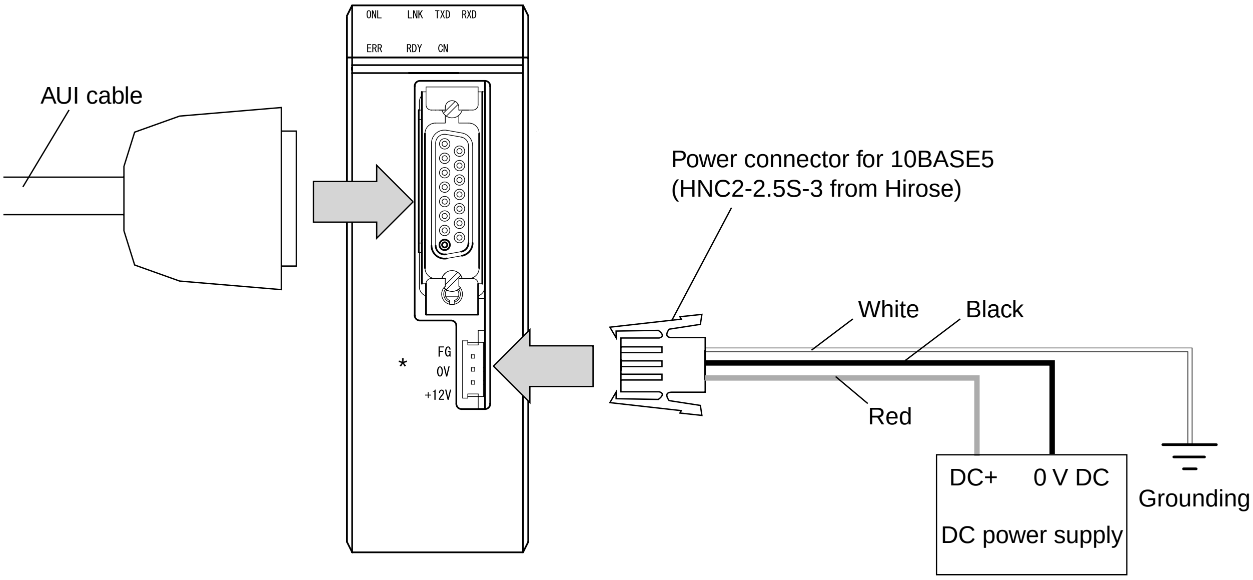
This LED indicates the current status of NP1L-ET2.



Symbol	Color	Description
ONL	Green	Indicates SX bus online condition. Lights when the module is operating normally; Blinks when SX bus is being connected; Unlit when SX bus is abnormal.
ERR	Red	Indicates module fault condition. Lights when the module is abnormal (module internal hardware error) or a cause of fatal fault is detected by software.
LNK	Green	Reserved
TXD	Green	Indicates the state of data sending. Lights when a send/receive packet is generated.
RXD	Green	Indicates the state of data receiving. Lights when a receive packet is generated.
RDY	Green	Lights when Initialization of the module is completed successfully.
CN	Green	Indicates the state of condition. Lights when there is at least one connection that is opened by SX user application.

2) 10BASE5 (AUI) connector
10BASE5 communication cable is connected here.

3) 12 V DC power supply connector for 10BASE5
12 V DC power (12 V DC, max. 500 mA) is supplied through this connector. For connection, be sure to use the connector (HNC2-2.5S-3 from Hirose) supplied with the product. For power specification, refer to "2-1 General Specifications".



* DC power supply will be prepared by customer.

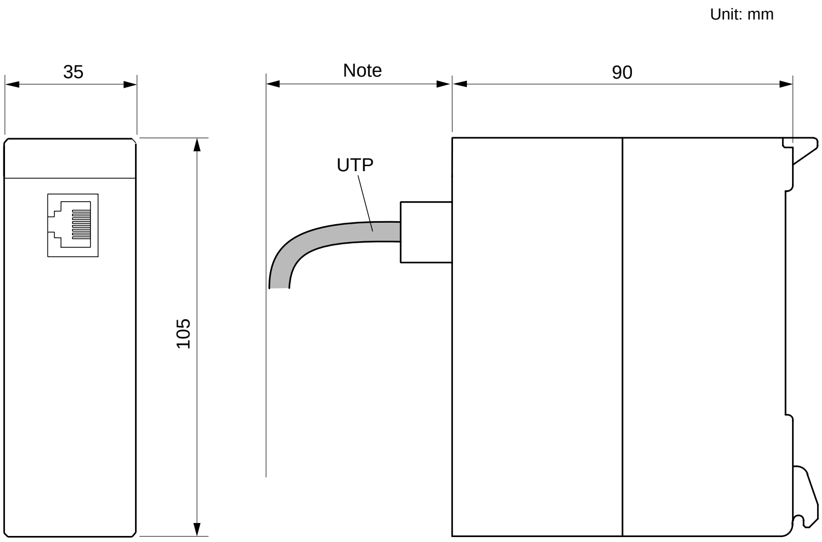
* Though NP1L-ET1/ET2 is equipped with the reverse-current preventive circuit for 12 V DC power supply for 10BASE5, be careful when wiring the DC power supply.

4) Version seal
Versions of the hardware and firmware of NP1L-ET1/ET2 are indicated on this seal.

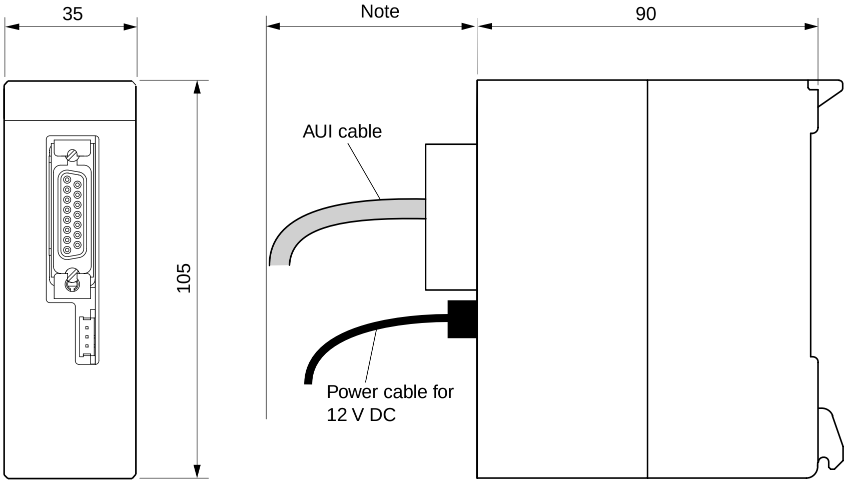
20 ← Hardware version
30 ← Firmware version

2-5 Dimensions

NP1L-ET1



NP1L-ET2



Note: In installing and wiring the NP1L-ET1/ET2 module, the bending radius of the cable needs to be taken into consideration.

Section 3 Installing the Module

Section 3 Installing the Module

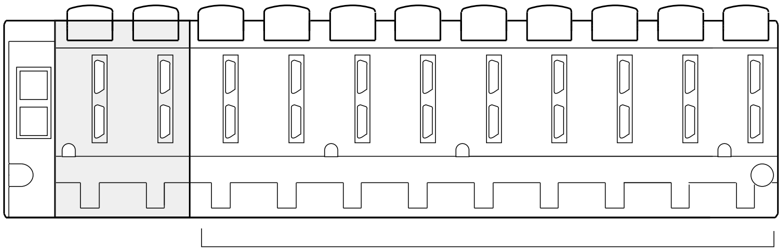
3-1 Mounting on Base Board

3-1-1 Mounting position

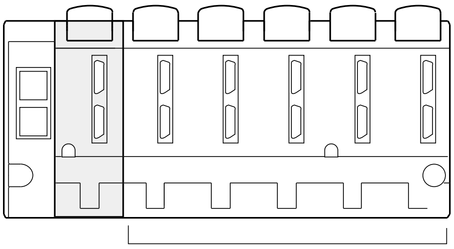
To use the NP1L-ET1/ET2 module, the module needs to be connected to the SX bus of MICREX-SX series SPH. It may be mounted at any location on a base board that is directly connected to the SX bus, but cannot be mounted on the OPCN-1, DeviceNet, T-link or other remote I/O base board.

The NP1L-ET1/ET2 module can be mounted in any slot except those for the power supply module (the two slots from the left on the base board).

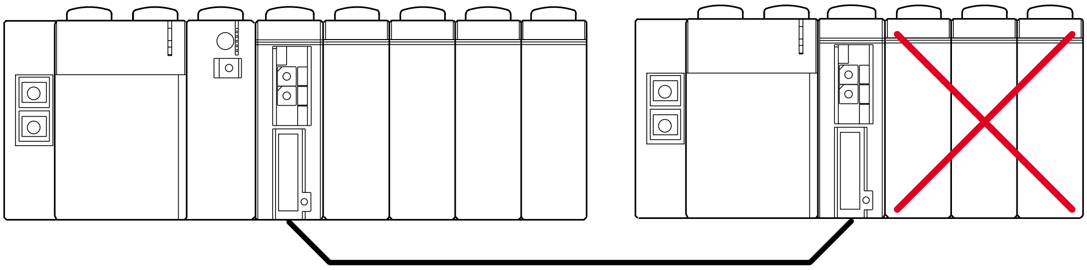
<8/11/13-slots base board>



<3/6-slots base board>



Note: NP1L-ET1/ET2 cannot be mounted on T-link, OPCN-1 or other slave station base board.

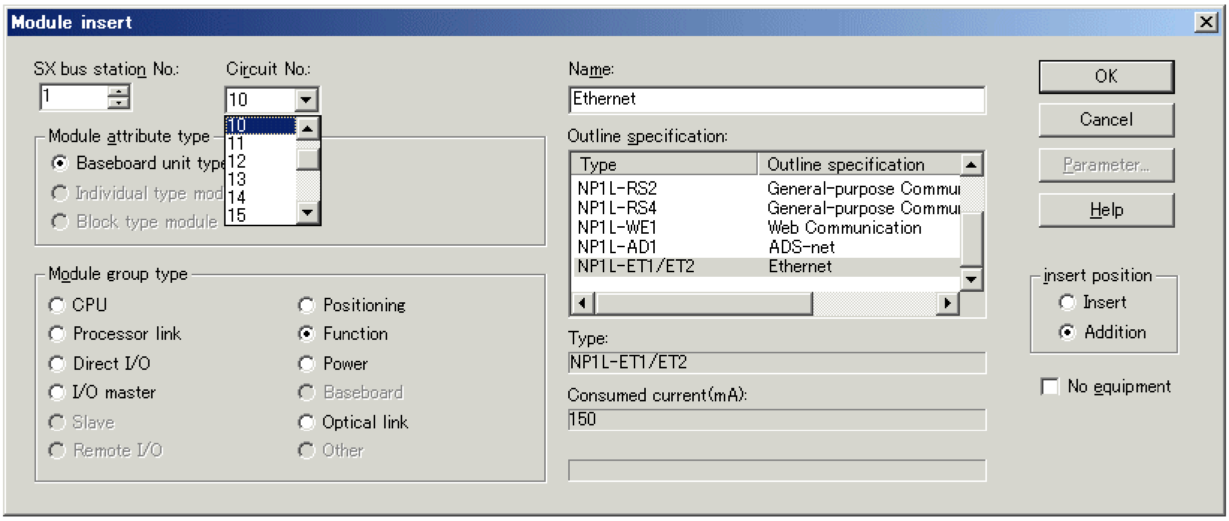


3-1-2 Number of mountable modules

Maximum four NP1L-ET1/ET2 modules can be connected for one configuration (including the case where PC card interface module is used for Ethernet communication). However, when the following modules are used in a same configuration, the system must be configured such that the total number of such modules and NP1L-ET1/ET2 modules becomes 16 or less.

Name	Type
P-link module	NP1L-PL1
PE-link module	NP1L-PE1
FL-net module	NP1L-FL1
FL-net2 module	NP1L-FL2
LE-net loop module	NP1L-LL1
General purpose communication module	NP1L-RS1/RS2/RS4
PC card interface module	NP1F-PC2
Memory card interface module	NP1F-MM1
ADS-net module	NP1L-AD1
WEB module	NP1L-WE1
LONWORKS interface module	NP1L-LW1
POD directly connected to SX bus	-

The modules in above table are controlled by “Circuit No.” that is set from the [Module insert] or [Module properties] dialog box. When the modules in above table are registered, circuit numbers from 8 to 23 are automatically assigned to them so that 17 or more modules won’t be registered to one system.



SX bus station No.: 1

Circuit No.: 10

10 11 12 13 14 15

Module attribute type

● Baseboard unit type
○ Individual type mod
○ Block type module

Module group type

○ CPU	○ Positioning
○ Processor link	● Function
○ Direct I/O	○ Power
○ I/O master	○ Baseboard
○ Slave	○ Optical link
○ Remote I/O	○ Other

Name: Ethernet

Outline specification:

Type	Outline specification
NP1L-RS2	General-purpose Commu
NP1L-RS4	General-purpose Commu
NP1L-WE1	Web Communication
NP1L-AD1	ADS-net
NP1L-ET1/ET2	Ethernet

Type: NP1L-ET1/ET2

Consumed current(mA): 150

insert position

○ Insert
● Addition

□ No equipment

Section 4 Communication Procedures

Page

Section 4 Communication Procedures

4-1 Overview of Communication Procedures

Communication with other node is performed basically in the following manner.

4-1-1 General purpose communication mode / Fixed buffer communication mode

1) Initialize

When the system is powered up, NP1L-ET1/ET2 module is initialized according to the setting of NP1L-ET1/ET2 module properties in the system configuration.

* For the content of the setting of NP1L-ET1/ET2, refer to "4-2 Definition of Module Operations".

2) Open communication channel

Using M_OPEN FB, make the system ready for communicating with remote node (open the port to establish connection).

* For more information of M_OPEN FB, refer to "4-4-2 Channel open (M_OPEN)".

3) Communicate with remote node

Send or receive data using M_SEND FB or M_RECEIVE FB.

* For mor* For more information of communication FBs, refer to "4-4-3 Message send (M_SEND)" and "4-4-4 Message receive (M_RECEIVE)".

4) Close communication channel

Set OFF the open request flag (a variable connected to the EN_C terminal) of M_OPEN FB to close the port and thus cut out the communication line with other node.

4-1-2 Loader command communication mode

1) Initialize and open communication channel

When the system is powered up, two ports are automatically opened to enable the communication using loader commands with maximum 8 stations.

2) Communicate with remote node

For communication between nodes of SX series devices, R_READ FB and/or R_WRITE FB are used. When the remote node of communication is a personal computer, loader commands (data read request and data write request) are issued from application program of the personal computer.

* For more information of communication FBs, refer to "4-4-5 Remote data write (R_WRITE)" and "4-4-6 Remote data read (R_READ)"
For more information of the loader commands that are used to issue command from personal computer, refer to "Appendix 1 Loader Commands".

3) Close communication channel

No application program is necessary for this. Automatically closed.

4-2 Definition of Module Operations

To communicate with other node, the definition of module operations (by setting module parameters) and communication program are necessary.

It is necessary to specify NP1L-ET1/ET2 operations. The setting for this is made in the system configuration definition of D300win (by setting NP1L-ET1/ET2 module parameters).

Item	Description		Initial value
IP address	IP address		None
	Subnet mask		255.255.255.0
Gateway	Default gateway		None
	New gateway	IP address	None
		Network address	None
	Installed gateway		None
Detail setting	TCP send timeout value (in 0.1-seconds steps)		300
	Maximum TCP end timer value (in 0.1-second steps)		0
	Response monitoring timer value (in 0.1-second steps)		300
	Server FTP command monitoring timer value (in 0.1-second steps)		3000
	Client FTP data connection monitoring timer value (in 0.1-second steps)		30
	Ethernet communication trace (disabled)		Not trace
	Processing for closing at TCP send timeout		Channel not closed
	Processing for closing at response receive timeout		Channel not closed
	Reference number of local port		256
	Reference number of remote port for communication		256

1) IP address

Sets IP address for NP1L-ET1/ET2 module (which is determined by network administrator). IP address is the address for distinguishing nodes on Ethernet. It consists of 32-bit binary data and is defined as a combination of IP network address and IP host address. IP network address is the address for distinguishing networks; IP host address is the ID for identifying the host (node) in a network.

2) Subnet mask

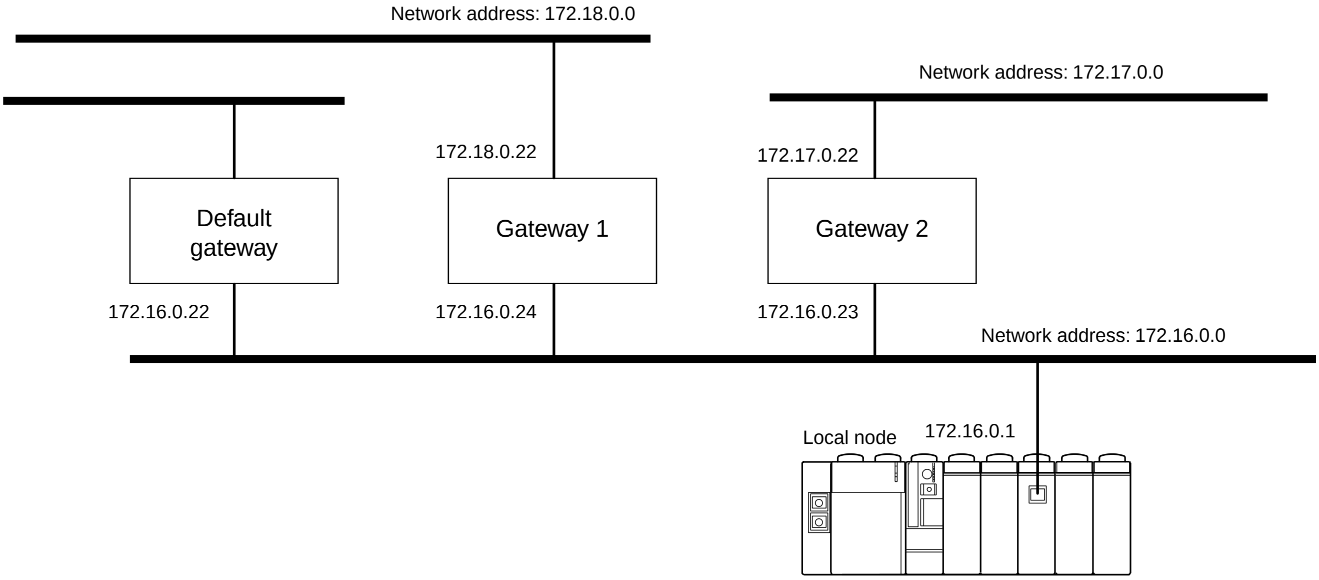
When many nodes are connected to one network, it is hard to control these nodes. When one network is divided into multiple subnets (groups), a value is set for subnet mask. Subnet uses part of host address, where bits are masked such that IP network address part and subnet part become "1" while IP host address part becomes "0". When there is no subnet, “0.0.0.0” is set.

Example 1) When a class B network is divided into 16 subnets, set as follows:
"255.255.240.0" (FFFFF000)

Example 2) When a class B network is divided into 256 subnets, set as follows:
"255.255.255.0" (FFFFFF00)

3) Gateway
When IP router (gateway) is used, multiple IP network segments can be connected. When networks are connected via IP router, it is possible to communicate with other network via the router. In this case, communication with other network becomes possible when gateway is set. The items to be set for this are the IP address of gateway and the IP address of network. Maximum 4 gateways can be set. When there is no gateway, set "0.0.0.0". When "0.0.0.0" is set for the IP address of network, the setting of the corresponding gateway is ignored.

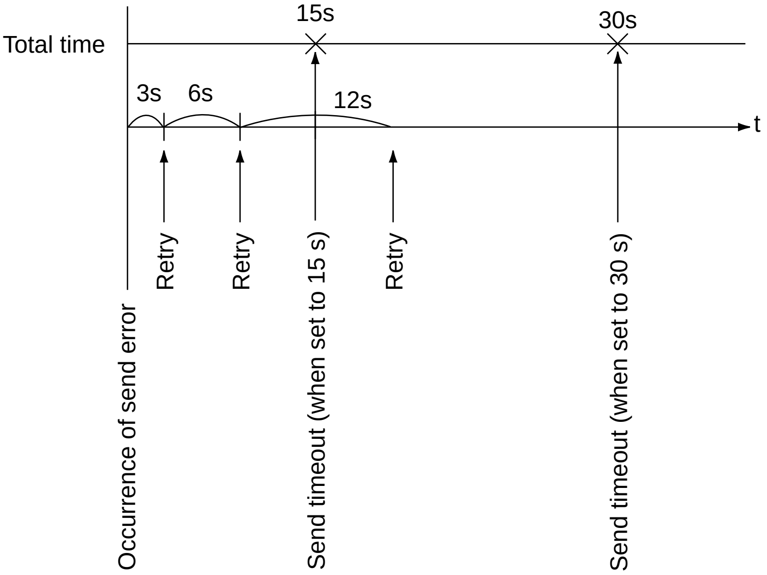
Example) When there are 3 gateways, including default gateway, set as follows:



4) TCP send timeout value (setting range: 0 to 300, in 0.1-second steps)
Communication error monitoring timer value when TCP channel is open and data is being sent.
If failed in opening a connection or sending data for TCP/IP communication, Ethernet automatically retries opening or sending. The interval of retry increases exponentially, like 3 seconds, 6 seconds, 12 seconds, …..

1) Abnormal end after the set timeout time elapses if repeating of retry cannot normally end the operation after opening is started
2) Abnormal end when the timing of retry is reached after the set timeout time elapses if repeating of retry cannot normally end the operation after sending is started

For example, when TCP send timeout value is set to 15 seconds or 30 seconds, the internal operation till error is detected becomes as follows:



5) Maximum TCP end timeout value (setting range: 0 to 600, in 0.1-second steps)
This item sets the waiting time to keep the socket for a given time after the closing of a TCP connection was attempted from local station and ended successfully. The maximum waiting time till the processing for closing the remote station is completed successfully when the closing of a TCP connection is attempted from local station is twice the set TCP send timeout value.
If the processing for closing the remote station cannot successfully be completed (when no FIN is received), RST is sent to the remote node to forcibly close the connection.

6) Response monitoring timer value (setting range: 0 to 300, in 0.1-second steps)
This item sets the waiting time till a response is returned after a command is sent.

7) Server FTP command monitoring timer value (setting range: 0 to 3000, in 0.1-second steps)
This item sets the time to automatically cut out the connection when a port is opened from other node in loader command mode but no command is sent from said node for a given period of time.
Set this item to “0” if you don’t want to cut out the connection even when no command is sent from other node, for example, when the loader is connected via NP1L-ET1/ET2.

8) Client FTP data connection monitoring timer value (setting range: 0 to 30, in 0.1-second steps)
Presently this item is not used.

9) Processing for closing at TCP send timeout
TCP send timeout occurs if no ACK is returned even when sending is retried specified times during communication under TCP/IP protocol. This item selects how to treat the connection.
Note: This item is disabled. If TCP send timeout occurs, the connection is forcibly closed.

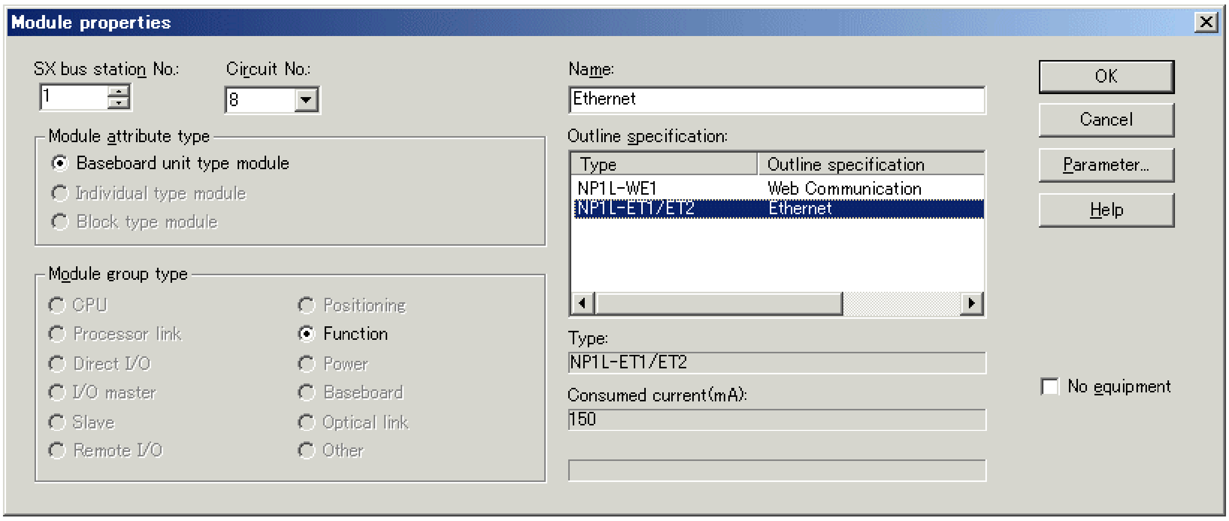
10) Processing for closing at response receive timeout
This item selects the treatment of connection in case of response receive timeout.

11) Reference number of local port (setting range: 0 to 65280)
Reference number of local port for TCP/IP or UDP/IP communication. TCP/IP or UDP/IP communication is performed using the port number that is determined by adding the reference number to the local port number that is specified by application FB (user interface). For example, when the values that are set from the parameter setting window for PC card interface module are
Reference number of local port = 256
Reference number of the remote port for communication = 256
and the setting of M_OPEN FB is
RPORT_NO =1
SPORT_NO = 2, then
PC card interface module side port number: Reference number of local port + SPORT_NO = 258, and
Port number of the remote node (personal computer side): Reference number of remote port + RPORT_NO = 257
Note: Set "256" for communication between MICREX-SX systems using R_READ and/or R_WRITE.

12) Reference number of remote port for communication (setting range: 0 to 65280)
Reference number of remote for TCP/IP or UDP/IP communication. TCP/IP or UDP/IP communication is performed using the port number that is determined by adding the reference number to the remote port number that is specified by M_OPEN FB (user application).
Note: Set "256" for communication between MICREX-SX systems using R_READ and/or R_WRITE.

<Operation>

1) Open the [Module properties] dialog box for the Ethernet module and click the [Parameter…] button, and the [Ethernet parameter setting] dialog box will be displayed.


Module properties

SX bus station No.: 1

Circuit No.: 8

Module attribute type: ● Baseboard unit type module ○ Individual type module ○ Block type module

Module group type: ○ CPU ○ Processor link ○ Direct I/O ○ I/O master ○ Slave ○ Remote I/O ○ Positioning ● Function ○ Power ○ Baseboard ○ Optical link ○ Other

Name: Ethernet

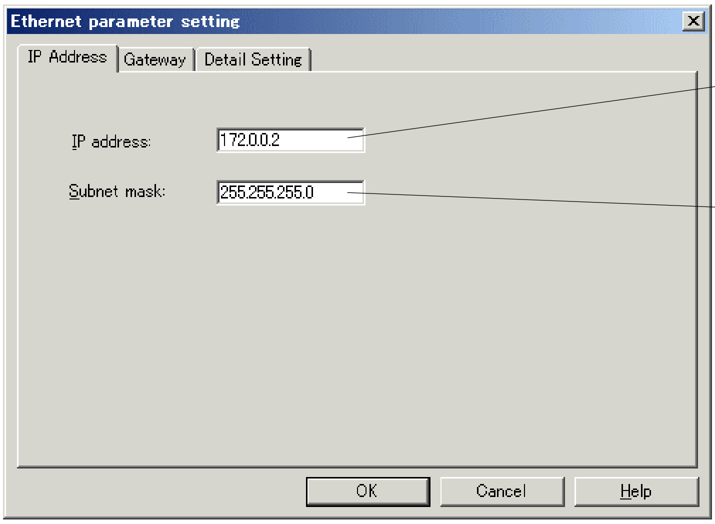
Outline specification:

Type	Outline specification
NP1L-WE1	Web Communication
NP1L-ET1/ET2	Ethernet

Type: NP1L-ET1/ET2

Consumed current(mA): 150

□ No equipment


Ethernet parameter setting

IP address: 172.0.0.2

Subnet mask: 255.255.255.0

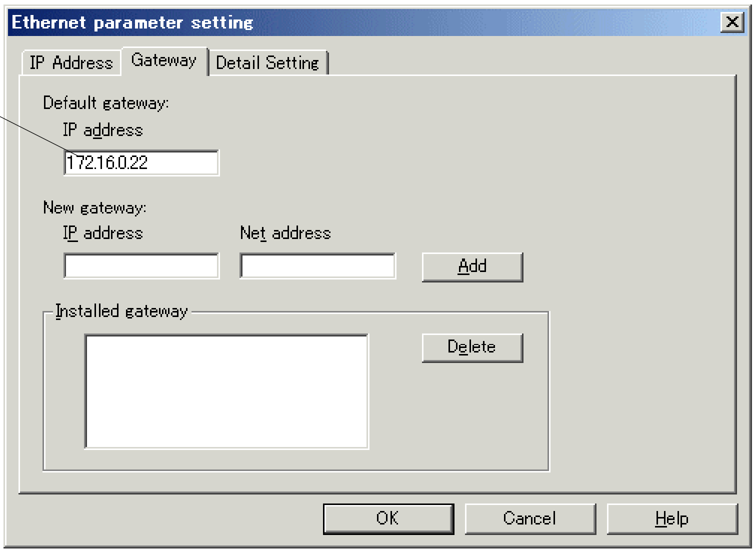
Enter IP address by a decimal number that is delimited by periods into 4 blocks.

Enter subnet mask by a decimal number that is delimited by periods into 4 blocks.

4-2 Definition of Module Operations

2) On the [IP Address] tab window, after setting IP address and subnet mask, click the [Gateway] tab to display the [Gateway] tab window.

Enter a decimal number that is delimited by periods into 4 blocks.



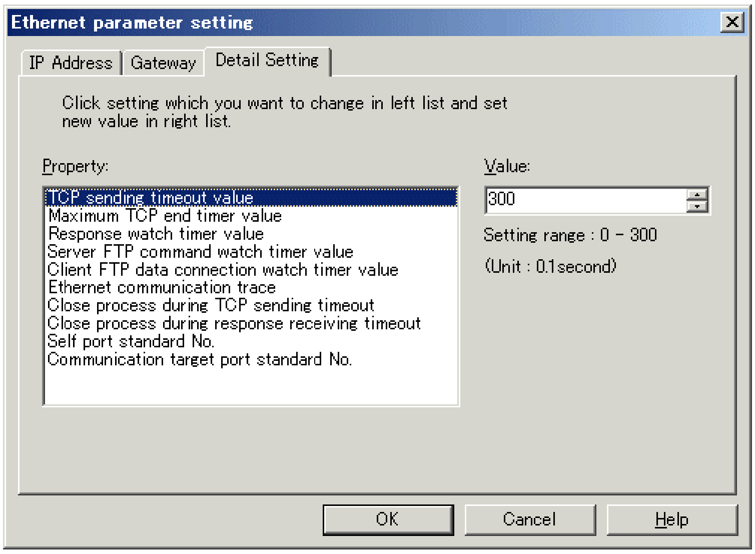
Ethernet parameter setting

Default gateway:
IP address: 172.16.0.22

New gateway:
IP address:
Net address:

Installed gateway

3) On the [Gateway] tab window, after setting IP address and subnet mask, click the [Detail Setting] tab to display the [Detail Setting] tab window.



Ethernet parameter setting

Click setting which you want to change in left list and set new value in right list.

Property:
- TCP sending timeout value
- Maximum TCP end timer value
- Response watch timer value
- Server FTP command watch timer value
- Client FTP data connection watch timer value
- Ethernet communication trace
- Close process during TCP sending timeout
- Close process during response receiving timeout
- Self port standard No.
- Communication target port standard No.

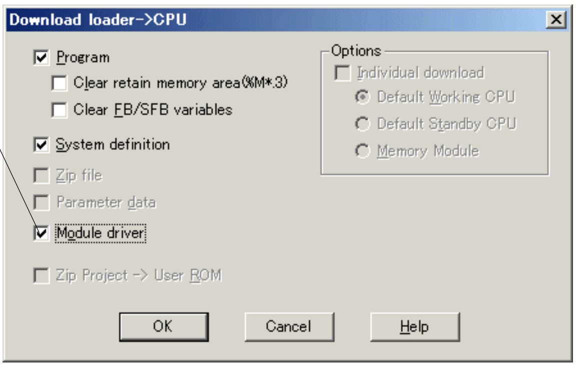
Value: 300

Setting range : 0 – 300

(Unit : 0.1second)

4) After setting all necessary items on this window, click the [OK] button to complete the setting.

Note: When a new system is configured or when an Ethernet parameter is changed, be sure to check the [Module Driver] box before downloading.



Download loader->CPU

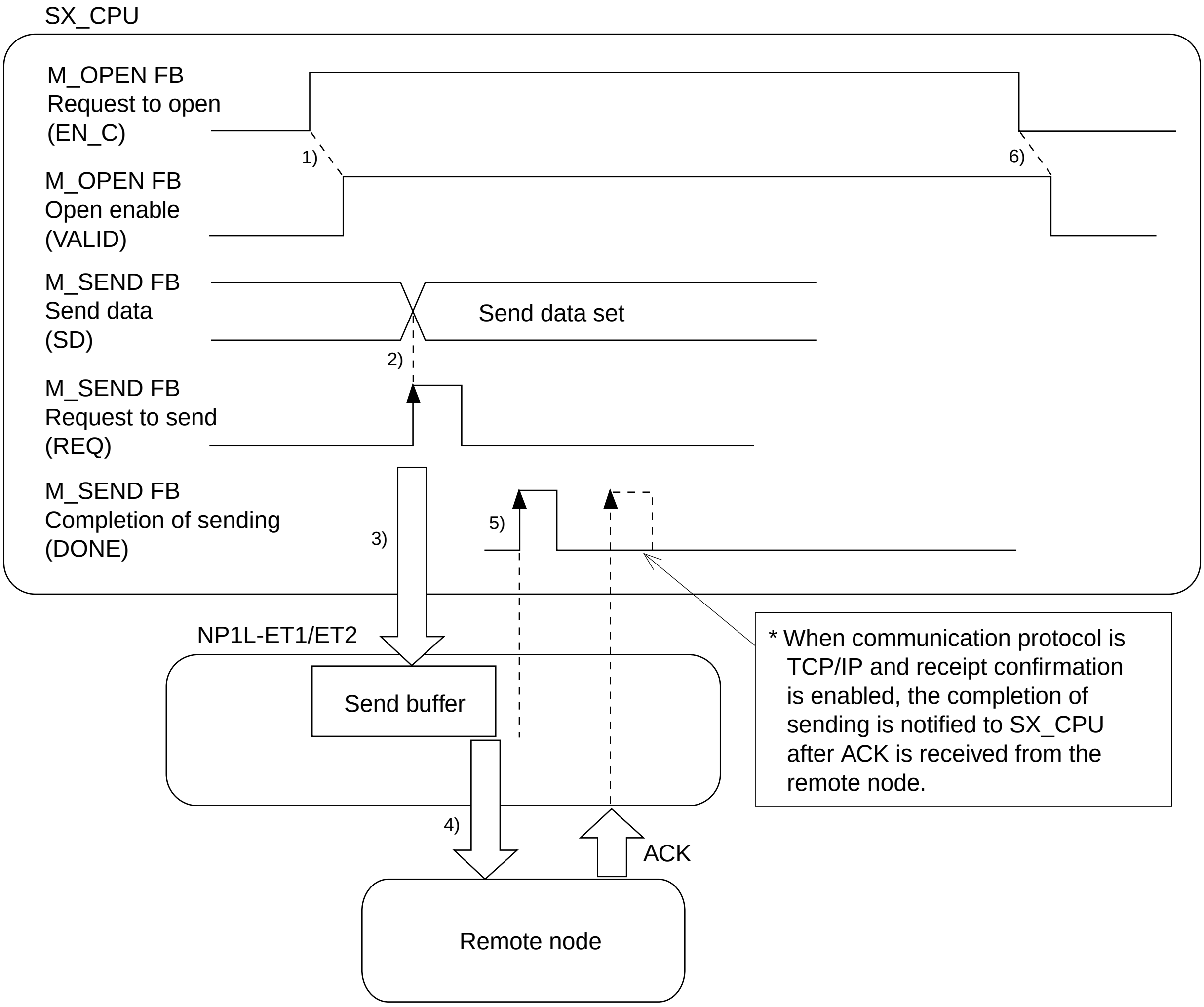
☑ Program
□ Clear retain memory area(%M*.3)
□ Clear FB/SFB variables
☑ System definition
□ Zip file
□ Parameter data
☑ Module driver
□ Zip Project -> User ROM

Options
□ Individual download
● Default Working CPU
○ Default Standby CPU
○ Memory Module

4-3-1 Communication in general purpose communication mode

(1) Data sending

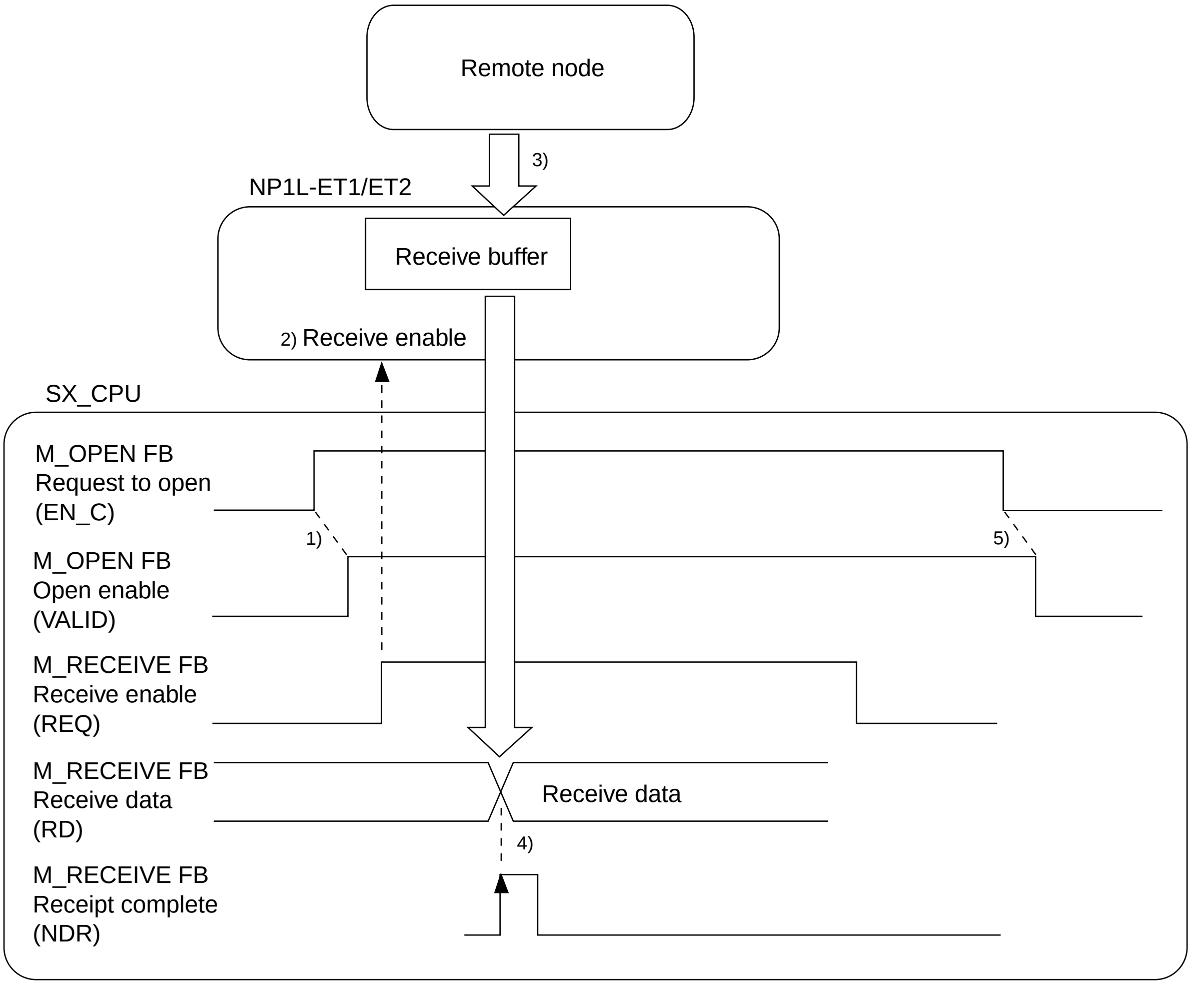
To send data to other node, M_OPEN FB is used to establish the connection, and M_SEND FB is used to send the data.



1) M_OPEN FB opens the communication channel. When the request to open (EN_C) is set ON, the connection with the remote node is established according to the content of the set M_OPEN FB, and the open enable flag (VALID) is set ON.
2) The target data is set in the send data (SD) of M_SEND FB, and the request to send (REQ) is set ON.
3) At the rising edge of the request to send (REQ), the data is sent to the send buffer of NP1L-ET1.
4) NP1L-ET1/ET2 adds the header part necessary for communication to the data and then sends the data to the remote node.
5) When the sending of data from NP1L-ET1/ET2 is completed, the sending completion flag (DONE) is set ON for the period of one scan. When communication protocol is TCP/IP and receipt confirmation is enabled, the sending completion flag (DONE) is set ON for the period of one scan after receipt complete signal (ACK) is sent from the remote node.
   Note: If the sending of data cannot successfully be ended, the sending error flag (ERROR) is set ON for the period of one scan. In such case, set the request to send OFF and re-execute the operation for sending (from step 3)).
6) To cut out the communication line after the end of communication, set OFF the request to open (EN_C) of M_OPEN FB. Then VALID will be set OFF when the processing for closing is completed.

(2) Data receiving

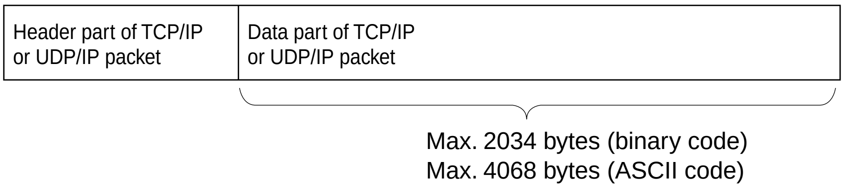
To receive data from other node, M_OPEN FB is used to establish the connection, and M_RECEIVE FB is used to receive the data.



1) M_OPEN FB opens the communication channel. When the request to open (EN_C) is set ON, the connection with the remote node is established according to the content of the set M_OPEN FB, and the open enable flag (VALID) is set ON.
2) The receive enable flag (EN_R) of M_RECEIVE FB is set ON to make the module become ready for receiving data.
3) The data sent from the remote node is received in the receive buffer.
4) The receive data that is stored in the receive buffer of NP1L-ET1 is transferred to the receive data (RD) of M_RECEIVE FB, and the receipt completion flag (NDR) is set ON for the period of one scan.
   Note: If abnormal data is received, the receipt completion flag (NDR) is not set ON, nor the receive data is stored in the receive buffer.
5) To cut out the communication line after the end of communication, set OFF the request to open (EN_C) of M_OPEN FB. Then VALID will be set OFF when the processing for closing is completed.

(3) Data format

Data received from or sent to other node


Header part of TCP/IP or UDP/IP packet
Data part of TCP/IP or UDP/IP packet
Max. 2034 bytes (binary code)
Max. 4068 bytes (ASCII code)

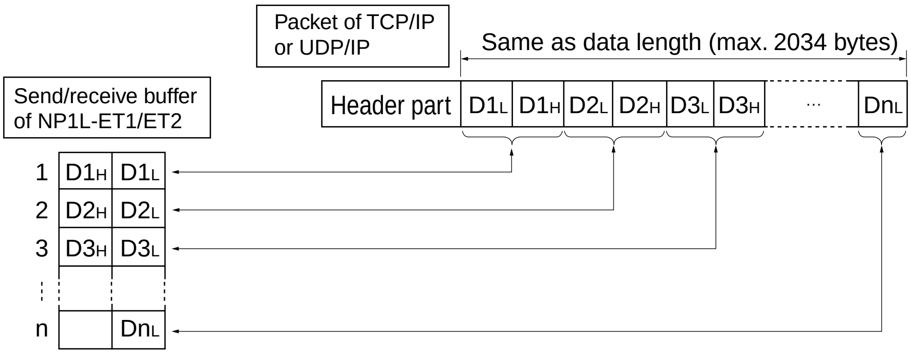
1) Header part

Header part is the header of Ethernet TCP/IP or UDP/IP packet. When an application program sends data to other node, NP1L-ET1/ET2 automatically adds the header and sends the data with the header to the remote node. When data is received from the remote node, NP1L-ET1/ET2 removes the header part from the received data, and only data part is sent to the FB.

2) Data part

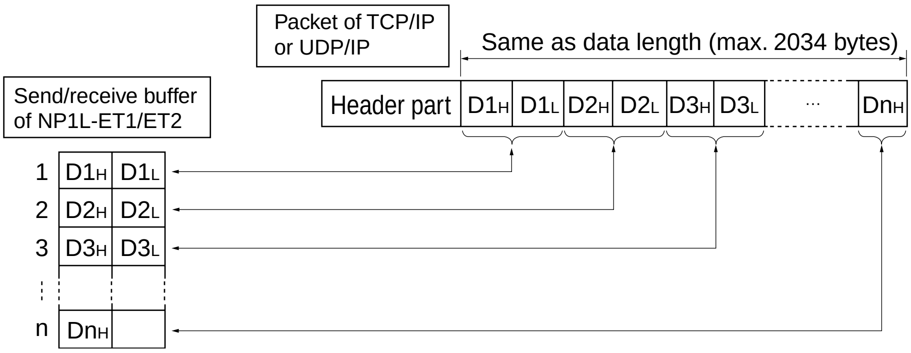
Data part is the data of Ethernet TCP/IP or UDP/IP packet. When data is sent to other node by M_SEND FB, the data transferred in the send buffer of NP1L-ET1/ET2 is set in the data part of TCP/IP or UDP/IP packet, and this data is sent. In the case of binary code, the data in the send buffer is sent as it is; in the case of ASCII code, the data in the send buffer is converted into ASCII code, and the converted data is sent.
When data is received from other node, the data in the data part of TCP/IP or UDP/IP packet is stored in the receive buffer of NP1L-ET1/ET2. In the case of binary code, received data is stored in the buffer as it is; in the case of ASCII code, received data is converted into ASCII code, and the converted data is stored in the buffer.

<Binary code (without data swap)>


Packet of TCP/IP or UDP/IP
Same as data length (max. 2034 bytes)
Send/receive buffer of NP1L-ET1/ET2
Header part
$D1_L$ $D1_H$ $D2_L$ $D2_H$ $D3_L$ $D3_H$ ··· Dn_L
1 $D1_H$ $D1_L$
2 $D2_H$ $D2_L$
3 $D3_H$ $D3_L$
n Dn_L

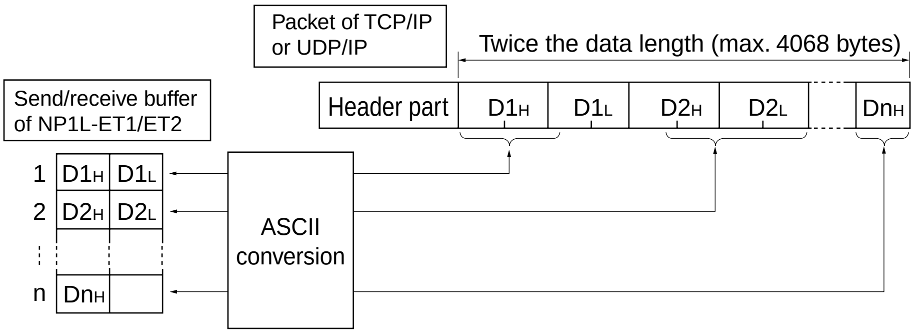
Data length is the number of bytes of the data that is stored in the send or receive buffer, expressed by decimal number (maximum 2034 bytes).
When data length is an odd number, the data up to the low-order byte of the last word is effective.

<Binary code (with data swap)>


Packet of TCP/IP or UDP/IP
Same as data length (max. 2034 bytes)
Send/receive buffer of NP1L-ET1/ET2
Header part
$D1_H$ $D1_L$ $D2_H$ $D2_L$ $D3_H$ $D3_L$ ··· Dn_H
1 $D1_H$ $D1_L$
2 $D2_H$ $D2_L$
3 $D3_H$ $D3_L$
n Dn_H

Data length is the number of bytes of the data that is stored in the send or receive buffer, expressed by decimal number (maximum 2034 bytes).
When data length is an odd number, the data up to the high-order byte of the last word is effective.

<ASCII code>


Packet of TCP/IP or UDP/IP
Twice the data length (max. 4068 bytes)
Send/receive buffer of NP1L-ET1/ET2
Header part
D1H
D1L
D2H
D2L
DnH
1
2
n
ASCII conversion

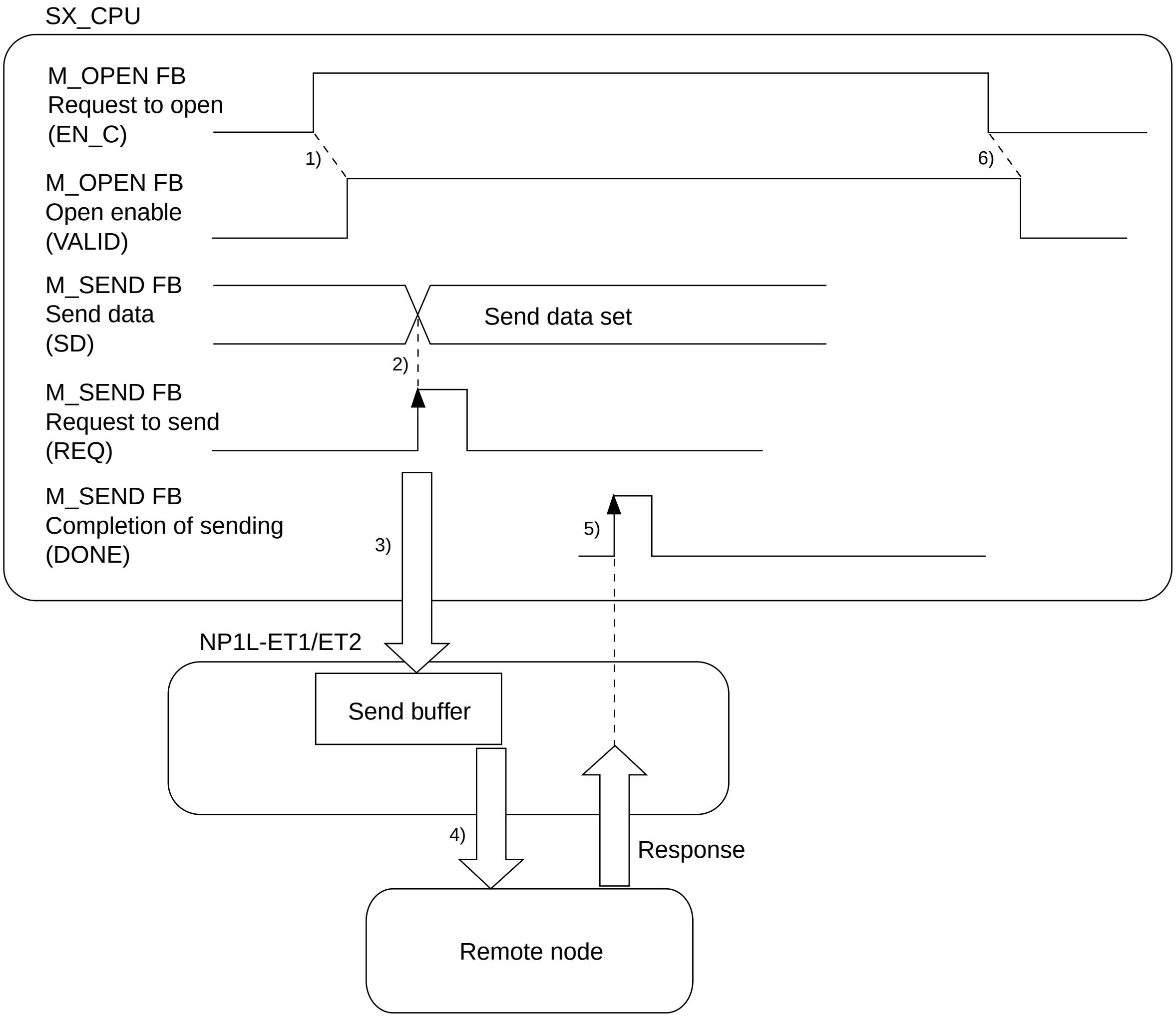
Data length is the number of bytes of the data that is stored in the send or receive buffer, expressed by decimal number (maximum 2034 bytes).

When data length is an odd number, the data up to the high-order byte of the last word is effective.

4-3-2 Communication in fixed buffer communication mode

(1) Data sending

To send data to other node, M_OPEN FB is used to establish the connection, and M_SEND FB is used to send the data.

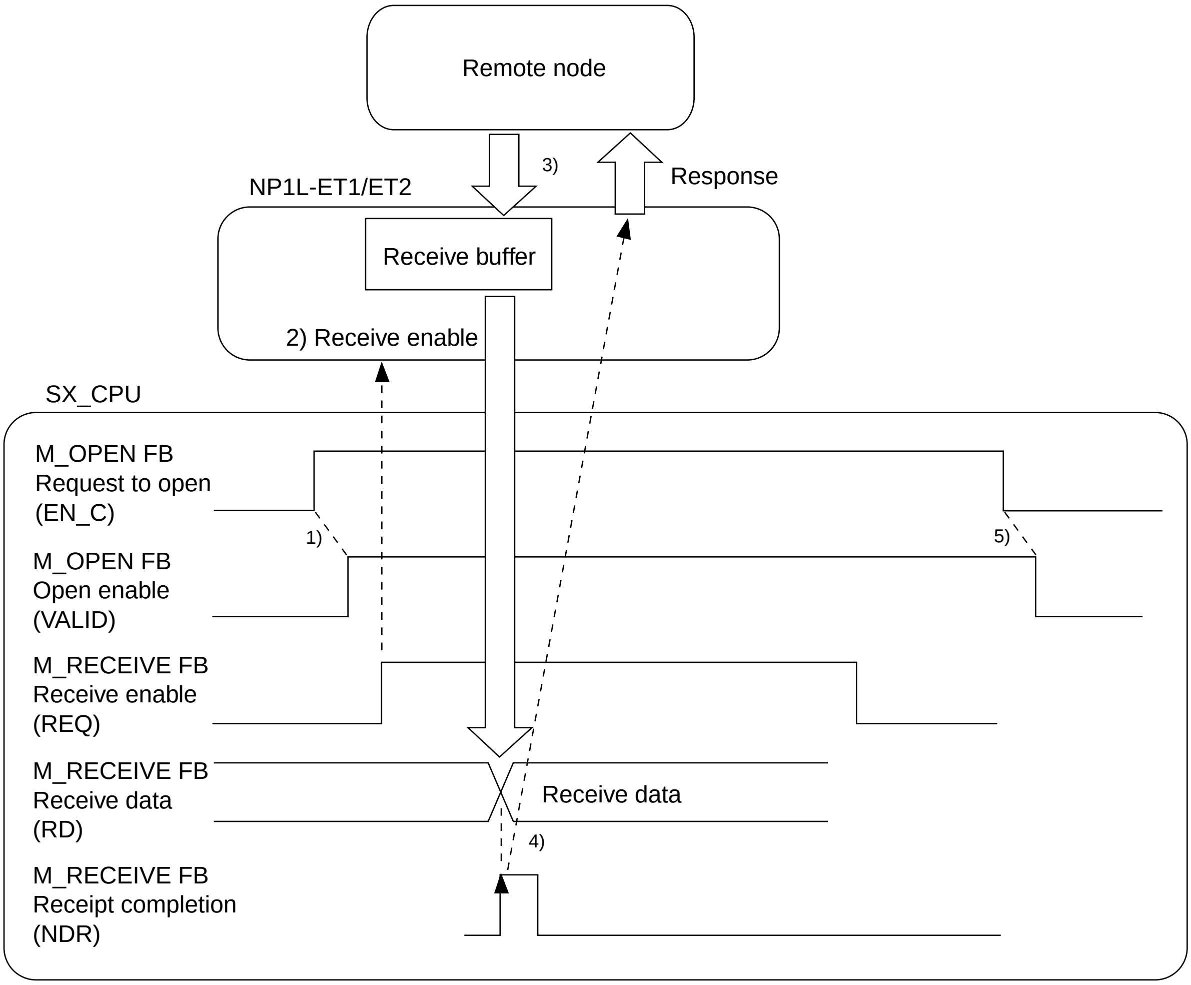


1) M_OPEN FB opens the communication channel. When the request to open (EN_C) is set ON, the connection with the remote node is established according to the content of the set M_OPEN FB, and the open enable flag (VALID) is set ON.
2) The target data is set in the send data (SD) of M_SEND FB, and the request to send (REQ) is set ON.
3) At the rising edge of the request to send (REQ), the data is sent to the send buffer of NP1L-ET1/ET2.
4) NP1L-ET1/ET2 adds the header part necessary for communication to the data and then sends the data to the remote node.
5) When the sending of data from NP1L-ET1/ET2 is completed and a response is received from the remote mode, the sending completion flag (DONE) of M_SEND FB is set ON for the period of one scan.
   Note: If the sending of data cannot successfully be completed (if no response is sent from the remote node or if the status of the end code returned as the response is not "00"), the sending error flag (ERROR) is set ON for the period of one scan. In such case, set the request to send once OFF and re-execute the operation for sending (from step 3)).
6) To cut out the communication line after the end of communication, set OFF the request to open (EN_C) of M_OPEN FB. Then VALID will be set OFF when the processing for closing is completed.

* When the Fixed Buffer Communication mode is used, set separate ports for sending and receiving.

(2) Data receiving

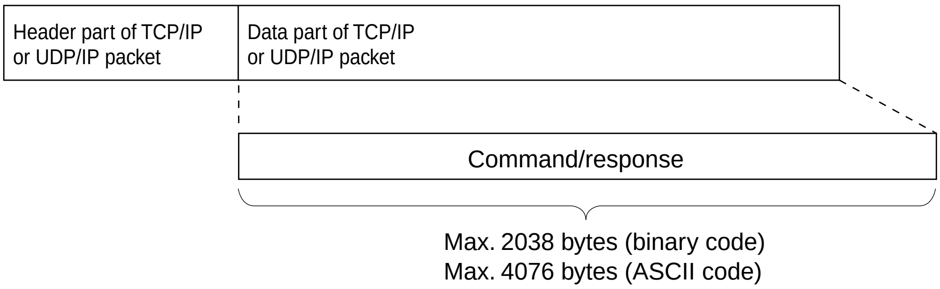
To receive data from other node, M_OPEN FB is used to establish the connection, and M_RECEIVE FB is used to receive the data.



1) M_OPEN FB opens the communication channel. When the request to open (EN_C) is set ON, the connection with the remote node is established according to the content of the set M_OPEN FB, and the open enable flag (VALID) is set ON.
2) The receive enable flag (EN_R) of M_RECEIVE FB is set ON to make the module become ready for receiving data.
3) The data sent from the remote node is received in the receive buffer.
4) The receive data that is stored in the receive buffer of NP1L-ET1/ET2 is transferred to the receive data (RD) of M_RECEIVE FB, and the receipt completion flag (NDR) is set ON for the period of one scan. When the receipt completion flag (NDR) is set ON, NP1L-ET1/ET2 returns a response to the other node.
   In processing for receiving, the data length to receive and the size of received data part are checked. Therefore, till all data is received, the system waits for the remaining data being received. Receiving completes when all data is received. If the size of received data is greater than the data length to receive, “E052(h)” is returned as response (E052: abnormal number of data words). At that time, the abnormal end (ERROR) flag is not set ON in the case of R_RECEIVE, and the received data is abandoned.
   Note: If abnormal data is received, the receipt completion flag (NDR) is not set ON, nor the receive data is stored in the receive buffer.
5) To cut out the communication line after the end of communication, set OFF the request to open (EN_C) of M_OPEN FB. Then VALID will be set OFF when the processing for closing is completed.

* When the Fixed Buffer Communication mode is used, set separate ports for sending and receiving.

(3) Data format


Header part of TCP/IP or UDP/IP packet
Data part of TCP/IP or UDP/IP packet
Command/response
Max. 2038 bytes (binary code)
Max. 4076 bytes (ASCII code)

1) Header part

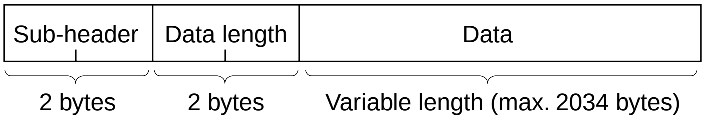
Header part is the header of Ethernet TCP/IP or UDP/IP packet. When an application program sends data to a remote node, NP1F-ET1/ET2 automatically adds the header and sends the data with the header to the remote node. When data is received from a remote node, NP1F-ET1/ET2 removes the header part from the received data, and only data part is sent to the FB.

2) Data part

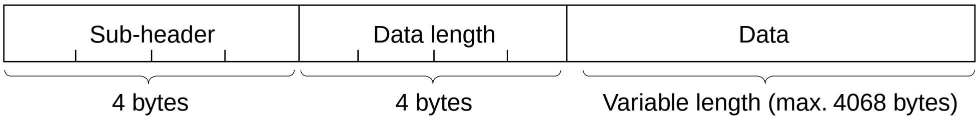
Data part is the data of Ethernet TCP/IP or UDP/IP packet. When data is sent to a remote node by M_SEND FB, sub-header is added to the data transferred in the send buffer of NP1F-ET1/ET2, and the sub-header added data is set in the data part of TCP/IP or UDP/IP packet, and this data is sent. In the case of binary code, the data in the send buffer is sent as it is; in the case of ASCII code, the data in the send buffer is converted into ASCII code, and the converted data is sent. When data is received from a remote node, the data that is removed the sub-header from that in the data part of TCP/IP or UDP/IP packet is stored in the receive buffer of NP1L-ET1/ET2. In the case of binary code, received data is stored in the buffer as it is; in the case of ASCII code, received data is converted into ASCII code, and the converted data is stored in the buffer.

<Command format>

Communication by binary code

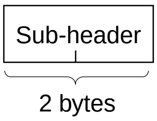


Communication by ASCII code

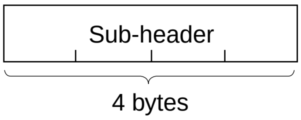


<Sub-header>

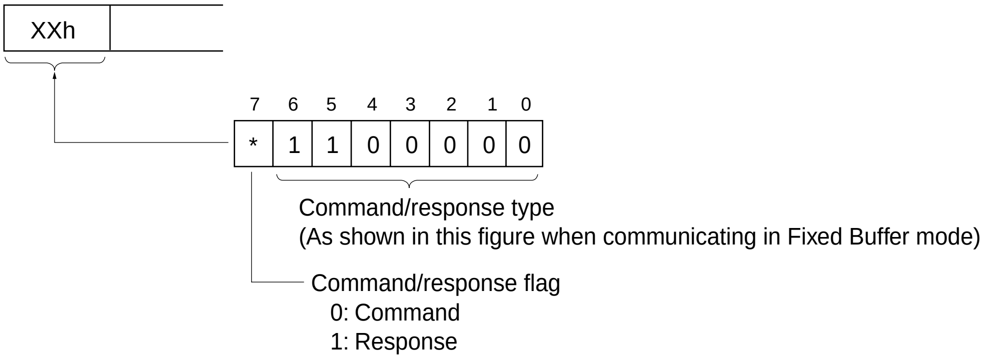
Communication by binary code



Communication by ASCII code



<Sub-header>

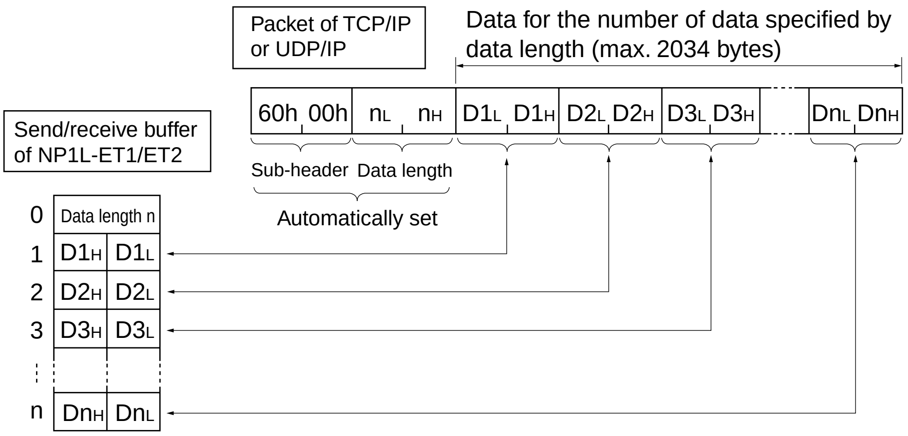


	Command	Response
Binary code	60h 00h	E0h End code
ASCII code	36h 30h 36h 30h "6" "0" "0" "0"	45h 30h End code "E" "0"

End code

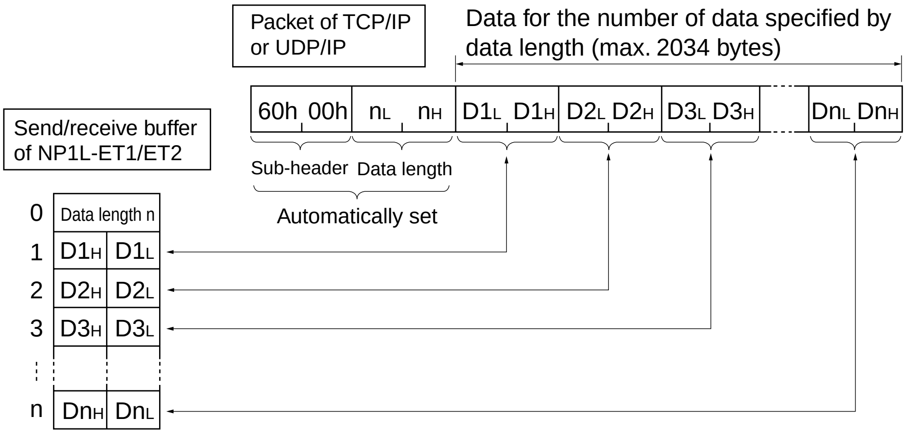
Binary code	ASCII code	Meaning	Description
00h	30h 30h	Normal end	—
50h	35h 30h	Command/response type definition error	Unspecified code is set for the type of command/response in sub-header.
52h	35h 32h	Abnormal number of data words	Data length exceeds specified value.
—	35h 34h	ASCII conversion error	During communication using ASCII code, an ASCII code that cannot be converted into binary code was sent from the remote node.

<Data, binary code (without data swap)>

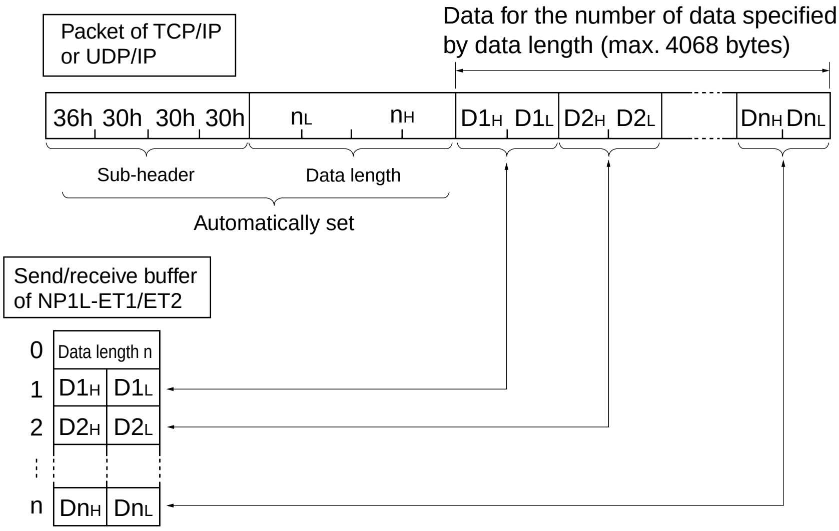

Packet of TCP/IP or UDP/IP
Data for the number of data specified by data length (max. 2034 bytes)
60h 00h
nL nH
D1L D1H
D2L D2H
D3L D3H
DnL DnH
Sub-header
Data length
Automatically set
Send/receive buffer of NP1L-ET1/ET2
0
Data length n
1
D1H D1L
2
D2H D2L
3
D3H D3L
n
DnH DnL

* Data length is the number of bytes of the data that is stored in the send or receive buffer (maximum 1017 words).

<Data, binary code (with data swap)>


Packet of TCP/IP or UDP/IP
Data for the number of data specified by data length (max. 2034 bytes)
60h 00h
n_L n_H
$D1_L$ $D1_H$
$D2_L$ $D2_H$
$D3_L$ $D3_H$
Dn_L Dn_H
Sub-header
Data length
Automatically set
Send/receive buffer of NP1L-ET1/ET2
0 Data length n
1 $D1_H$ $D1_L$
2 $D2_H$ $D2_L$
3 $D3_H$ $D3_L$
n Dn_H Dn_L

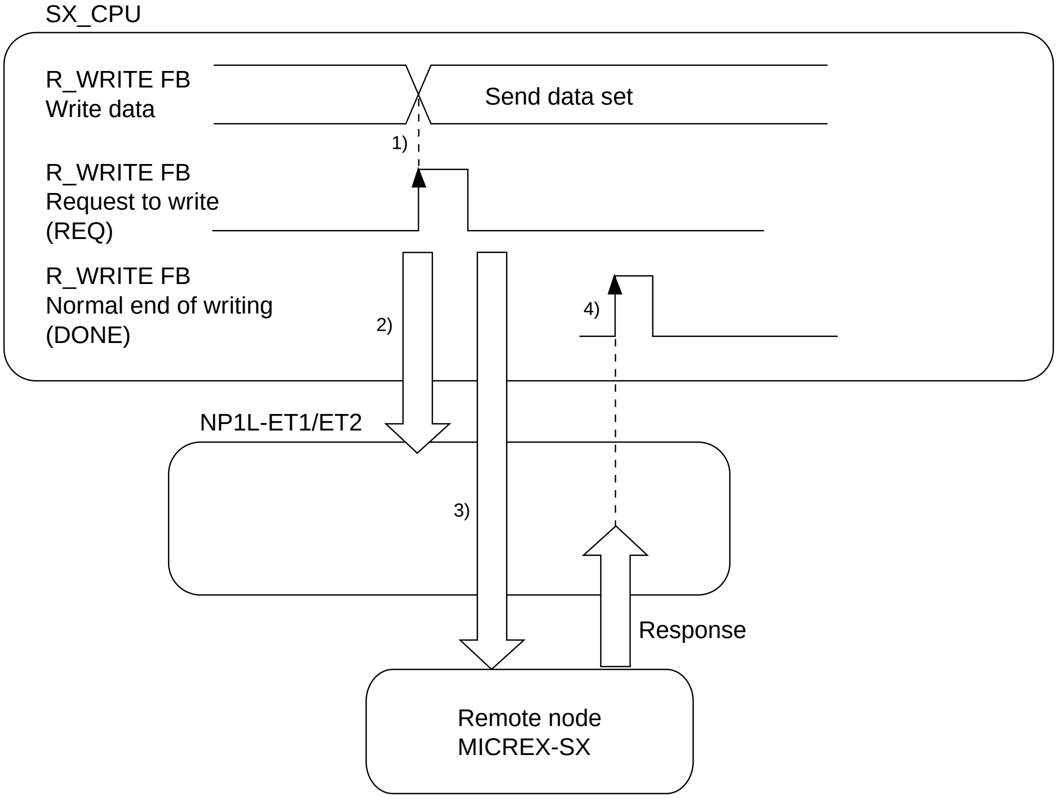
<Data, ASCII code>


Packet of TCP/IP or UDP/IP
Data for the number of data specified by data length (max. 4068 bytes)
36h 30h 30h 30h
n_L n_H
$D1_H$ $D1_L$
$D2_H$ $D2_L$
Dn_H Dn_L
Sub-header
Data length
Automatically set
Send/receive buffer of NP1L-ET1/ET2
0 Data length n
1 $D1_H$ $D1_L$
2 $D2_H$ $D2_L$
n Dn_H Dn_L

4-3-3 Communication in loader command communication mode

(1) Data writing

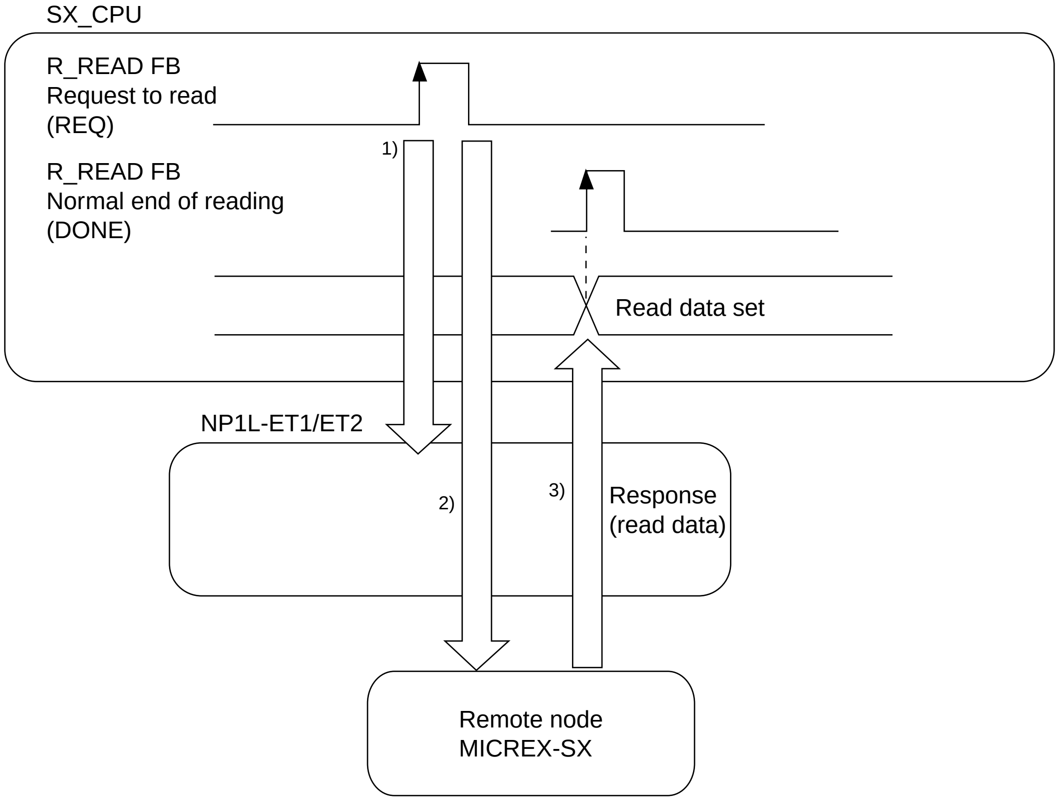
R_WRITE FB is used to write data in a remote SX node (SX_CPU internal memory). R_WRITE FB automatically opens the communication channel and writes data in the remote node (SX system).



1) Target data is set in the send data (SD) of R_WRITE FB, and the request to write (REQ) is set ON.
2) When the request to write (REQ) of R_WRITE FB is set ON, the communication channel to NP1L-ET1/ET2 is automatically opened.
3) When the connection is established, loader command is sent to the remote SX node via Ethernet to execute writing.
4) The remote SX node (SX_CPU) executes the write command that is issued by R_WRITE FB and sends the result of operation (response) to NP1L-ET1/ET2. When normal end signal is received, the flag for the normal end of writing (DONE) is set ON for the period of one scan.
Note: If the writing of data cannot successfully be ended (if no response is sent from the remote node or if the status of the end code returned as the response is not "00"), the receiving error flag (ERROR) is set ON for the period of one scan. In such case, set the request to write once OFF and re-issue the request to write.

(2) Data reading

R_READ FB is used to read data from a remote SX node (SX_CPU internal memory). R_READ FB automatically opens the communication channel and reads data from the remote node (SX system).



1) When the request to read (REQ) of R_WRITE FB is set ON, the communication channel to NP1L-ET1/ET2 is automatically opened.
2) When the connection is established, loader command is sent to the remote SX node via Ethernet to request reading data from the internal memory of SX_CPU.
3) The remote SX node (SX_CPU) executes the read command that is issued by R_READ FB and returns the result of operation (read data and status). When normal end signal and read data are received, the flag for the normal end of reading (DONE) is set ON for the period of one scan.
   Note: If the reading of data cannot successfully be ended (if no response is sent from the remote node or if the status of the end code returned as the response is not "00"), the receiving error flag (ERROR) is set ON for the period of one scan. In such case, set the request to read once OFF and re-issue the request to read.

4-4-1 FBs used in communication program

Ethernet communication programs use the following FBs.

Communication mode	FB used	Purpose and use
General Purpose Communication mode Fixed Buffer Communication mode	M_OPEN	Used in order for the CPU to request the Ethernet interface module to open local ports for it and thus establishing connection.
	M_SEND	Used to send data from SX_CPU (SX system) to other node
	M_RECEIVE	Used in order for SX_CPU to receive data sent from other node.
Loader Command Communication mode	R_WRITE	Used to write data from SX_CPU (SX system) in the Ethernet node of other SX system. No application program is necessary on the written side.
	R_READ	Used in order for SX_CPU (SX system) to read data from the Ethernet node of other SX system. No application program is necessary on the side that receives the request to read.

Note: Loader Command Communication mode requires no program to open the communication port because communication port is automatically opened at the rising edge of the Ethernet interface module.

4-4-2 Channel open (M_OPEN)

M_OPEN (channel open) FB is used when the CPU requests the NP1L-ET1/ET2 module to open a local port for it and thus to establish the connection for it to communicate with other node that is connected to the module. (Maximum 16 ports can be opened at a time.)

When the processing for opening is completed successfully, a connection number is output as the proof that the connection is established. After this, M_SEND and M_RECEIVE FBs communicate data using this connection number.

(1) Operation of the command

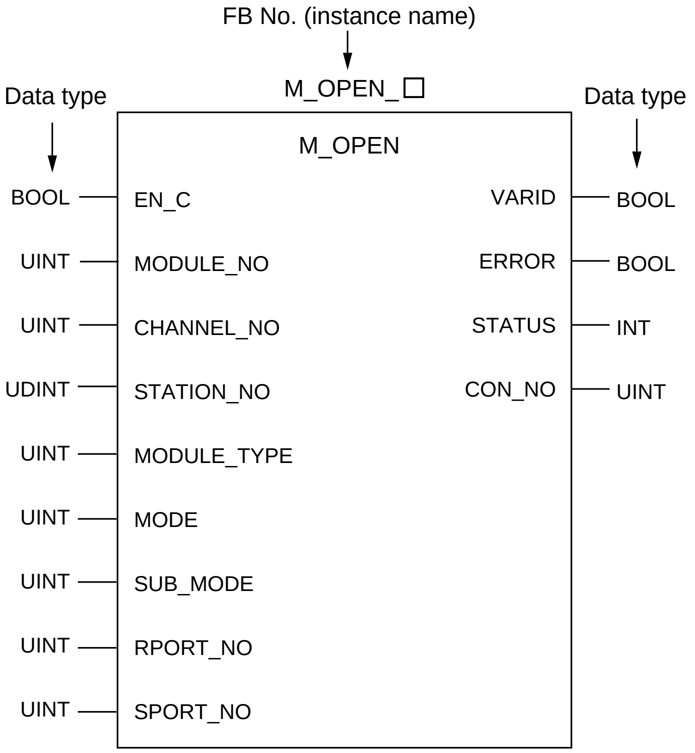
1) At the rising edge of "EN_C", the processing for opening is performed for the NP1L-ET1/ET2 that is specified by "MODULE_NO". (The processing for opening does not complete within the period of one scan.)
2) When the processing for opening is completed successfully, the "VALID" flag is set ON, and a connection number is output to "CON_NO". Then it becomes possible to use M_SEND and M_RECEIVE FBs.
3) If the processing for opening cannot successfully be completed, the "ERROR" flag is set ON for the period of one scan, and the corresponding error code is output to "STATUS".
4) When "EN_C" is set to 0 (zero), the processing for closing is star5ed. (The processing for closing does not end within the period of one scan.)
5) When the processing for closing is completed successfully, the "VALID" flat is set to (zero). (Processing for closing does not ends abnormally.)

Notes:

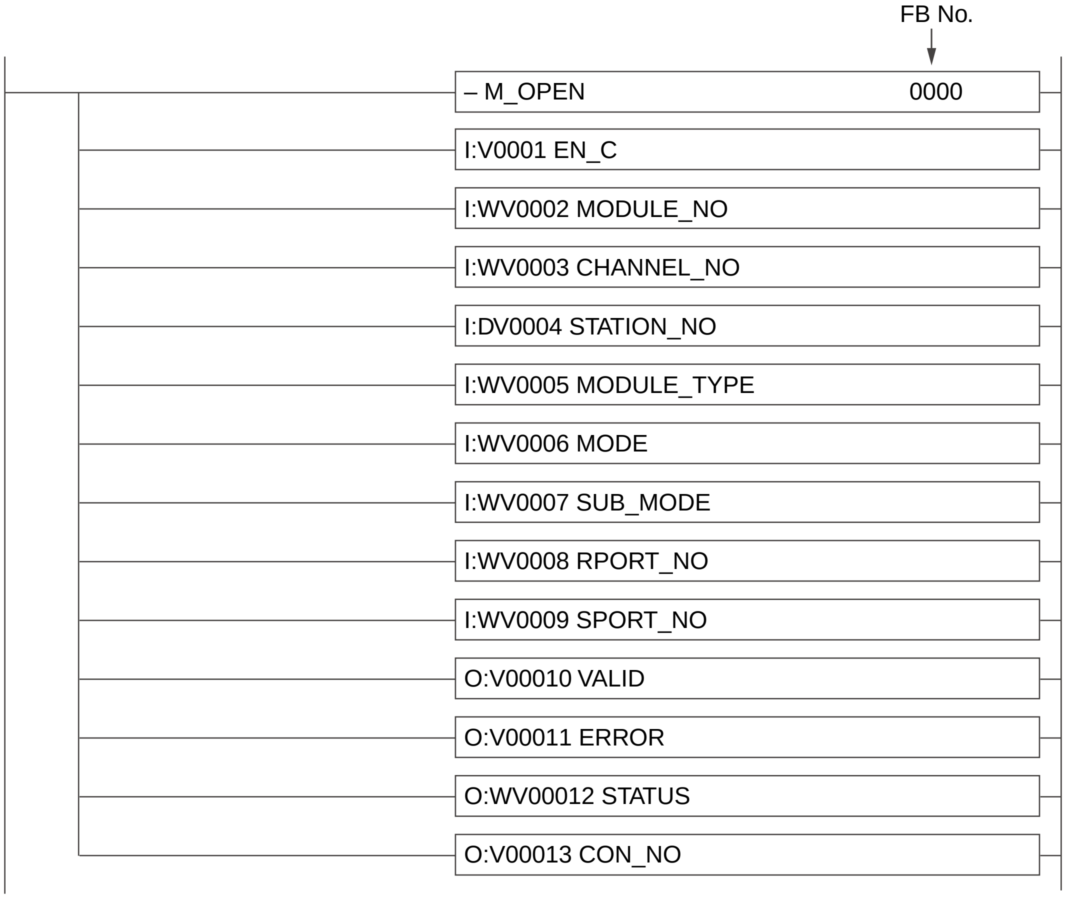
1) There are two modes for opening the ports: "passive mode" for receiving and "active mode" for sending. To send data, the remote node must be ready for receiving data. Therefore, the processing for opening ports in "passive mode" for receiving data needs to be completed in advance.
2) If reopening is attempted on a same port (when local port and the remote port for communication are the same), the connection will be established during the retry that is performed 3 seconds later because it takes 1 second to open the socket.

(2) FB format

1) When D300win loader is used



2) When the Standard loader is used



(3) Explanation of each FB terminal

Terminal function name	Terminal name	Data type	I/O	Description
Request to open	EN_C	BOOL	IN	ON : The processing for opening ports is started at the rising edge of this signal OFF: The processing for opening ports is started at the falling edge of this signal.
Communication module No.	MODULE_NO	UINT	IN	Specify the target of communication by SX bus station number of the Ethernet interface module.
Communication channel No.	CHANNEL_NO	UINT	IN	For this module, this item is fixed to 0 (zero)
Remote station No. *1	STATION_NO	UDINT	IN	Specifies IP address for the remote node of communication.
Module type	MODULE_TYPE	UINT	IN	Sets the type of network for this module. Fixed to 1 (one).
Communication mode *2	MODE	UINT	IN	Sets communication conditions for the port to be opened.
Communication submode *3	SUB_MODE	UINT	IN	Sets enable/disable of receipt confirmation. 0: Disable, 1: Enable
Remote port No.	RPORT_NO	UINT	IN	Specifies port No. for the remote node of communication.
Local port No.	SPORT_NO	UINT	IN	Specifies port No. for the NP1L-ET1 module (setting range: 1 to 127).
Open enable	VALID	BOOL	OUT	Turns ON when the processing for opening is completed successfully and the modules becomes ready for communication. Turns OFF when the processing for closing is performed (when EN_C is set OFF).
Error	ERROR	BOOL	OUT	Turns ON for the period of one scan if the processing for opening is ended abnormally.
Status *4	STATUS	INT	OUT	The result of the processing for opening is output by a status code.
Connection No.	CON_NO	UINT	OUT	When the processing for opening is completed, a connection number is assigned. This number is used for communication by M_SEND and M_RECEIVE FBs.

* Data type is used to create a program with D300win.

***1 Remote station No.**

IP address is set in the following manner. Example: When IP address is set to "172.16.0.1"

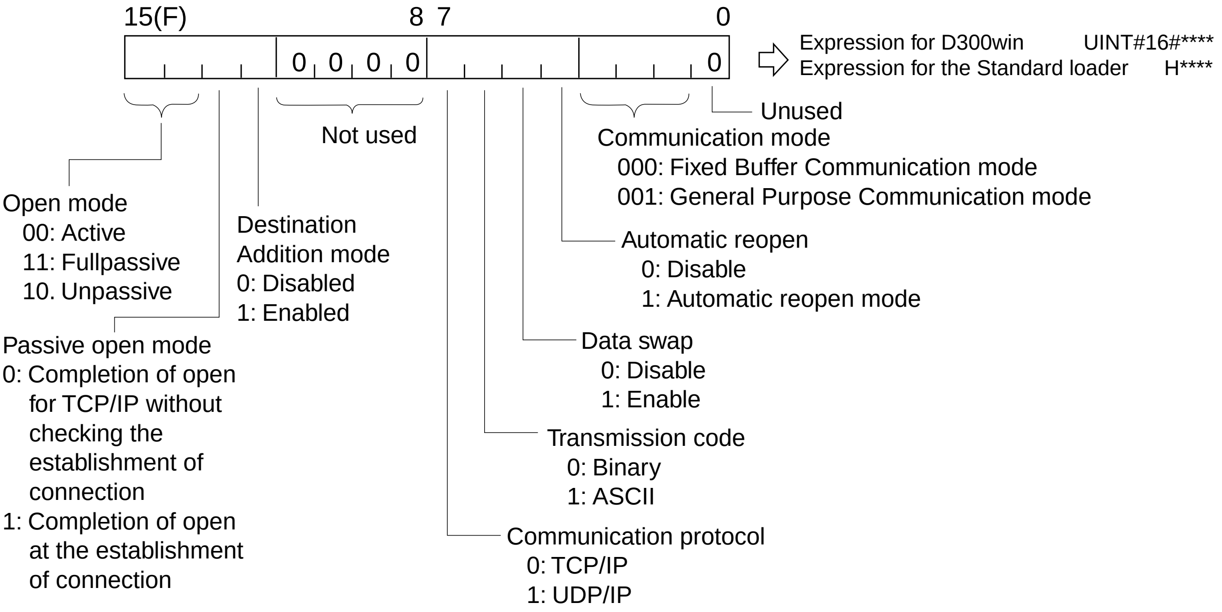
172	16	0	1
ACh	10h	00h	01h

⇨ Expression for D300win UDINT#16#AC100001
Expression for the Standard loader HAC100001

* When UDP/IP protocol is used, broadcast is possible by setting "255.255.255.255", and partial broad cast is possible by setting "***.***.***.255".

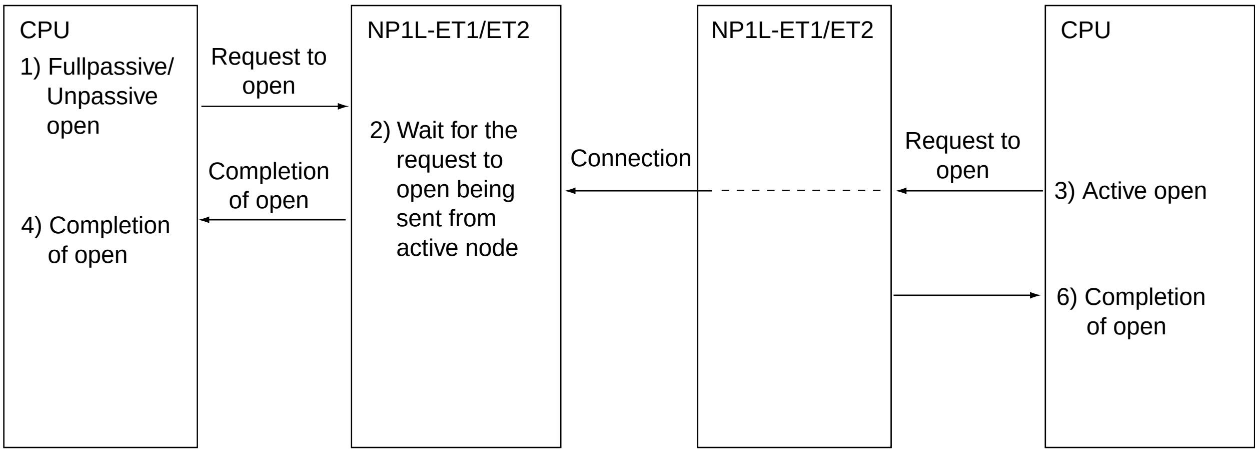
***2 Communication mode**

Sets communication conditions for the connection at which a channel is to be opened, in the following format.


15(F)
8 7
0
0 0 0 0
0
Expression for D300win UINT#16#****
Expression for the Standard loader H****
Unused
Not used
Communication mode
000: Fixed Buffer Communication mode
001: General Purpose Communication mode
Open mode
00: Active
11: Fullpassive
10. Unpassive
Destination Addition mode
0: Disabled
1: Enabled
Automatic reopen
0: Disable
1: Automatic reopen mode
Passive open mode
0: Completion of open for TCP/IP without checking the establishment of connection
1: Completion of open at the establishment of connection
Data swap
0: Disable
1: Enable
Transmission code
0: Binary
1: ASCII
Communication protocol
0: TCP/IP
1: UDP/IP

<Open mode>

To open ports for TCP/IP communication, the processing for Fullpassive open as well as that for Unpassive open must be finished before the processing for active open is performed.



1) Active open mode

The processing for active open is performed for other nodes that are ready for receiving TCP connection open command.

2) Fullpassive open mode

The processing for passively opening TCP connection is performed only for the nodes that are set in the communication address setting area. Then the module waits for a request of active open being sent from other node that is set in the communication address setting area.

3) Unpassive open mode

The processing for passively opening TCP connection is performed for all other nodes that are connected to the network. Then the module waits for a request of active open being sent from other node connected to the network.

<Passive open mode>

If the completion of open at the establishment of connection is selected, VALID of M_OPEN is set ON when Ethernet connection is established. If connection is not established for 30 seconds after M_OPEN request is issued, open error (status: 193) occurs.

<Destination Addition mode>

When Destination Addition mode is enabled, the first 4 words of data are used as destination information area.

- For TCP/IP

Send		Receive	
Reserved	Max. 1021 words	Receive data length (number of bytes)	Max. 1021 words
Reserved		Remote IP address for communication (L)	
Reserved		Remote IP address for communication (H)	
Reserved		Remote port number for communication	
Send data Max. 1017 words		Receive data Max. 1017 words	

- For UDP/IP

Send		Receive	
Reserved	Max. 1021 words	Receive data length (number of bytes)	Max. 1021 words
Remote IP address for communication (L)		Remote IP address for communication (L)	
Remote IP address for communication (H)		Remote IP address for communication (H)	
Remote port number for communication		Remote port number for communication	
Send data Max. 1017 words		Receive data Max. 1017 words	

- IP address and port number are stored by decimal number.
- For UDP/IP sending, data is sent to the remote station set above. It is set by user application.

<Data swap>

When binary is specified as transmission code, the treatment of high-order and low-order bytes of transmission data are reversed. When ASCII is specified as transmission code, this setting does not take effect.

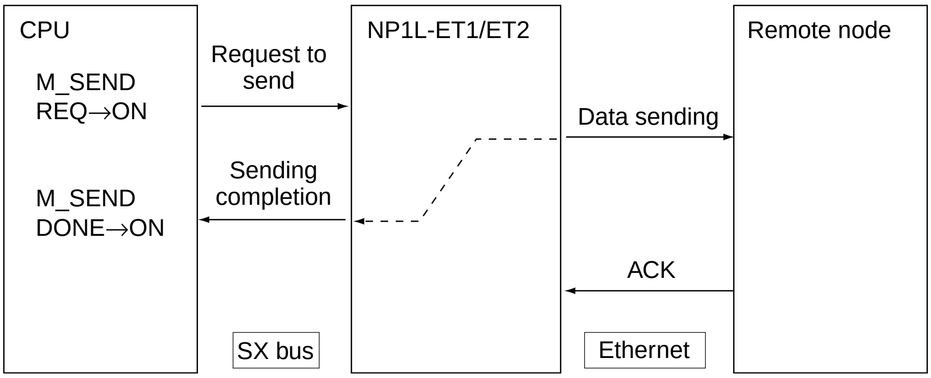
<Automatic reopen>

If Automatic Reopen mode is enabled when "Fullpassive" or "Unpassive" is specified as open mode, the NP1L-ET1 module automatically starts the processing for closing the moment the request to close is received from the remote mode after the connection has been established, and then automatically comes in Request-to-Open waiting mode.

***3 Communication submode**

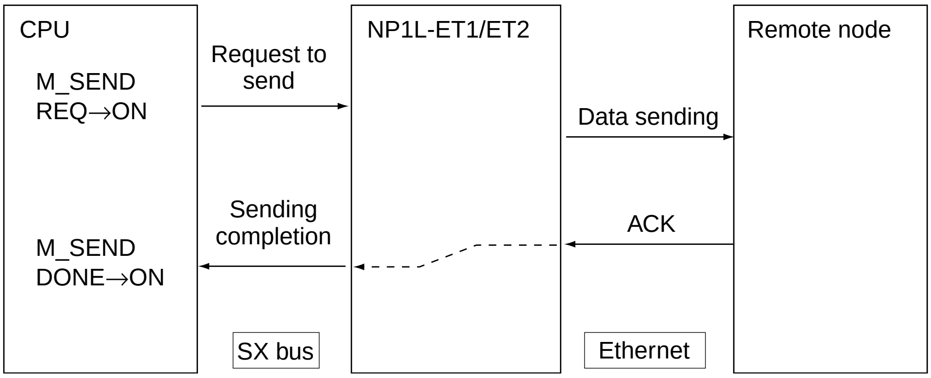
When data is sent using TCP/IP protocol, this submode specifies whether or not to enable receipt confirmation.

1) Operation when receipt confirmation is disabled (SUB_MODE=UINT#0)



* Sending is completed when data is sent onto Ethernet without waiting for ACK from the remote node.

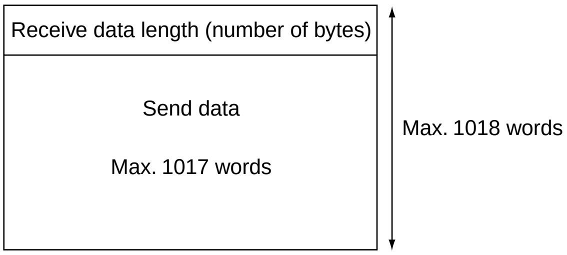
2) Operation when receipt confirmation is enabled (SUB_MODE=UINT#1)



* Sending is completed after waiting for ACK being sent from the remote node.

<Store receive data length>

When “Store receive data length” is enabled, receive data length is stored in the first word of the receive buffer, as shown in the figure below.



***4 Status**
The table below shows the probable errors that may occur while processing to open ports.

Error code	Cause
160	Message destination setting error • No module exists at the SX bus station No. that is specified by "MODULE_NO".
162 (see note)	No response for command • No response from remote node due to defective wiring or condition of remote node • No response from remote node due to other causes
177	Parameter error • An out-of-range value was input. (The set value of MODULE_NO exceeds the range of 1 to 256, or an out-of-spec value is input for MODULE_TYPE.)
193	Channel open error • An abnormal value is set for station No. • An abnormal value is set for communication mode • "Active" is set for open mode (on the sending side), but the station No. (IP address) of the remote node or RPORT_NO cannot be found on the network. * If UDP/IP is selected for communication protocol, because no communication is performed in opening a channel, no error occurs even when the station No. (IP address) of the remote node cannot be found on the network. In addition, no connection is established in opening a port during UDP/IP communication. NP1L-ET1/ET2 modules regard a connection as being established when data is received from the remote node. • Connection is not established
200	Port setting error • The code specified by "SPORT_NO" is out of the range of 1 to 127. • Same value is already set as "SPORT_NO" in the resource (CPU). • The same combination of "SPORT_NO" and "RPORT_NO" is already registered in the same communication module.
201	Connection numbers and client numbers are full • It was attempted to open 57 or more ports in the resource at a time. • It was attempted to open ports exceeding the limit on the number of ports that can be opened for one communication module. (The number of ports that can be opened simultaneously changes with modules: maximum 16 ports for NP1L-ET1/ET2 module.)
202	Mismatching of network type • The code specified for module type does not match the network of said communication module

Note: According to the “running status information” of detail RAS, check the following item:

- SMC send 16 collision (offset address: 00000020)
- SMC send memory allocation incomplete (TASK) (offset address: 00000028)
- SMC send memory allocation (INT) (offset address: 00000030)
- SMC receive overrun (offset address: 00000090)
- SMC receive USNET buffer full (offset address: 000000A0)

If one of these counters has counted up, it seems that the load of the network line is too heavy. Adjust to reduce the load of the network line.

4-4-3 Message send (M_SEND)

M_SEND (message send) FB is used to send data in General Communication mode or Fixed Buffer Communication mode to a remote node, the communication channel to which is opened by SX_CPU and thus the communication with which becomes possible.

(1) Operation of the command

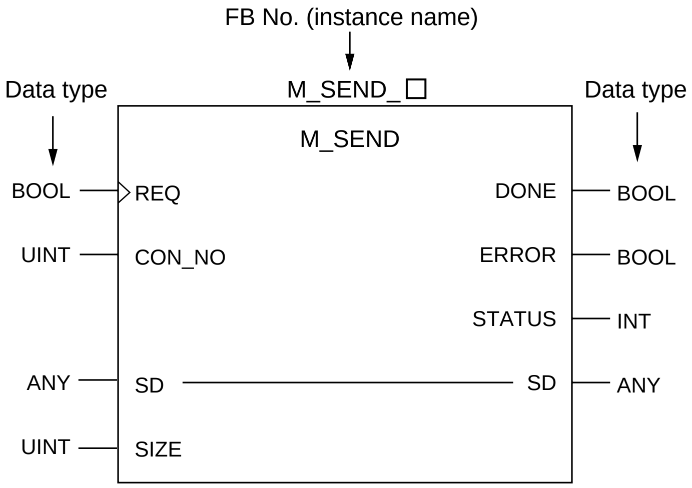
1) At the rising edge of "REQ", message is sent to the station having the connection number that is set by "CON_NO". (The processing for sending does not complete within the period of one scan.)
2) When the sending of message is completed successfully, the "DONE" flag is set ON for the period of one scan.
3) If the sending of message cannot successfully be completed, the "ERROR" flag is set ON for the period of one scan, and the corresponding error code is output to "STATUS".

Notes:

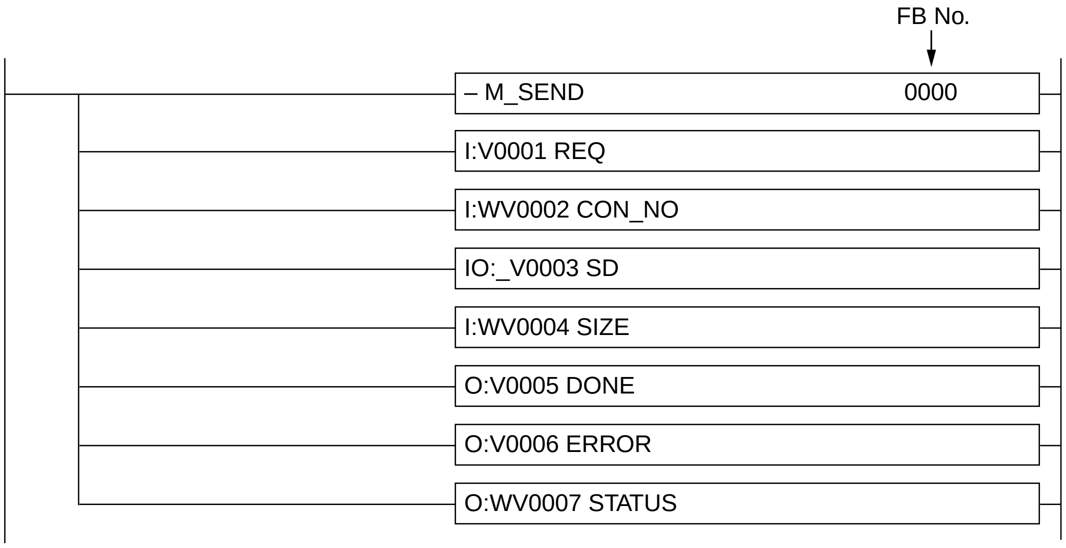
1) Maximum 1017 words of data can be sent at a time.
2) While a message is being sent (since the rising edge of "REQ" till the "DONE" or "ERROR" flag is set ON), additional "REQ" command (OFF→ON) does not take effect.
3) Do not change "SD" while a message is being sent. Otherwise, the send data cannot be guaranteed.
4) If the number of data specified by "SIZE" exceeds the size of variable specified by "SD", the data for the excess part becomes undefined. Be sure to enter specified size of variable for "SIZE".
5) Programming should be made such that request to send, or "REQ", is turned ON after the "VALID" flag turns ON.

(2) FB format

1) When D300win loader is used



2) When the Standard loader is used



(3) Explanation of each FB terminal

Terminal function name	Terminal name	Data type	I/O	Description
Request to send	REQ	BOOL	IN	At the rising edge of this signal, the data prepared in "SD" is sent.
Connection No.	CON_NO	UINT	IN	Sets a number for the connection that is opened by M_OPEN. The "CON_NO" terminal of M_OPEN FB is connected to this terminal.
Send data size	SIZE	UINT	IN	Specify the size of send data in units of word.
Send data	SD	ANY	IN-OUT	Connect a variable in which the data to be sent is stored.
Normal end of sending	DONE	BOOL	OUT	Turn ON for the period of one scan when the sending of data is completed successfully.
Abnormal end of sending	ERROR	BOOL	OUT	Turn ON for the period of one scan if the sending of data ended abnormally.
Status *1	STATUS	INT	OUT	The result of sending a message is output by a status code.

***1 Status**

Error code	Cause
164	Message sending error • The remote node of communication is disconnected.
165 (see note)	Message receiving busy • No message can be sent because the message transmission line of the SX bus is busy. (After waiting for a while, re-execute the FB. It the error recurs, the message transmission load of the remote node may be too high.)
177	Parameter error • 0 (zero) is input to the "SIZE" terminal. • The variable that stores send data exceeds specified memory area.
195	Message sending error (communication error between Ethernet module and other node) • No data can be sent to the communication module at the remote node. • No response is returned from the communication module at the remote node (sending is completed but no ACK is returned). • Other transmission error on the network
199	Channel forcible close (communication error between Ethernet module and other node) • Request to close is received from the remote node. Note: If this code is received, close once the corresponding port and then re-execute the processing for opening the port.
206	Buffer over • The number of send data exceeds 1017 words. • In sending data via a communication module, the limit on the number of data that the communication module can send is exceeded.
207	Connection No. error • A connection number that is not yet opened is used. • A currently used connection number is used. (When two M_SEND FBs are used in parallel for one connection)

Note: According to the “running status information” of detail RAS, check the following item:

- SMC send 16 collision (offset address: 00000020)
- SMC send memory allocation incomplete (TASK) (offset address: 00000028)
- SMC send memory allocation (INT) (offset address: 00000030)
- SMC receive overrun (offset address: 00000090)
- SMC receive USNET buffer full (offset address: 000000A0)

If one of these counters has counted up, it seems that the load of the network line is too heavy. Adjust to reduce the load of the network line.

4-4-4 Message receive (M_RECEIVE)

M_RECEIVE (message receive) FB is used to receive data in General Communication mode or Fixed Buffer Communication mode from a remote node, the communication channel to which is opened by SX_CPU and thus the communication with which is possible.

(1) Operation of the command

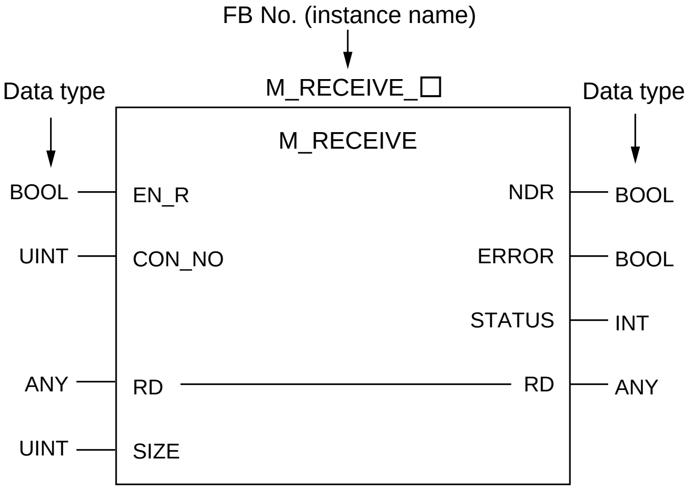
1) At the rising edge of "EN_R", message is received from the station having the connection number that is set by "CON_NO". (The processing for receiving does not complete within the period of one scan.)
2) When the receiving of message is completed successfully, the "NDR" flag is set ON for the period of one scan.
3) If the receiving of message cannot successfully be completed, the "ERROR" flag is set ON for the period of one scan, and the corresponding error code is output to "STATUS".

Notes:

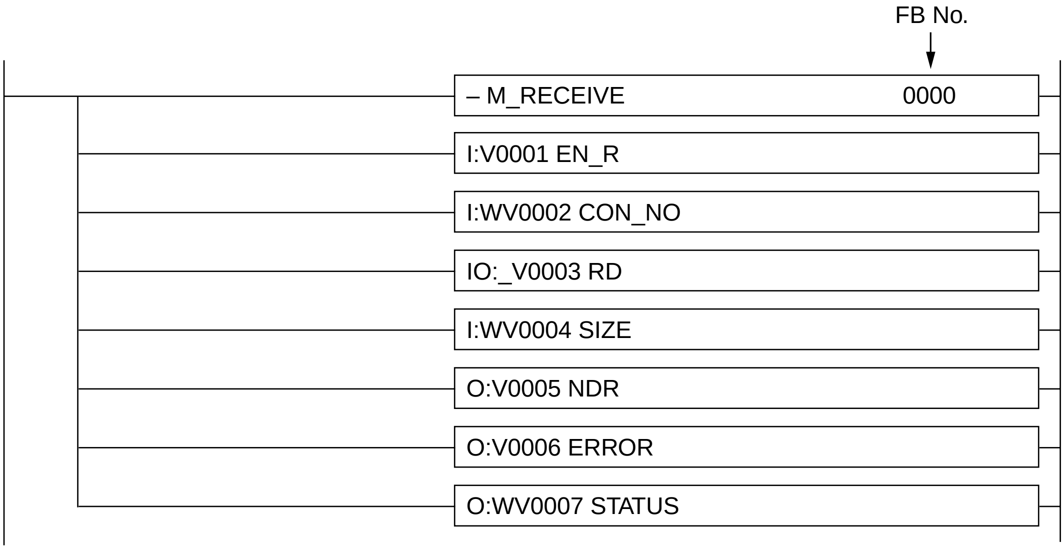
1) Maximum 1017 words of data can be received at a time.
2) While a message is being received (since the rising edge of "EN_R" till the "NDR" or "ERROR" flag is set ON), "EN_R" must be kept ON. Receiving is halted if "EN_R" is set OFF.
3) Receiving is resumed when "EN_R" is set ON after receiving has been halted. Receiving is resumed using the input values that were effective when operation was halted even if "CON_NO", "RD" or "SIZE" is changed. These changes have no influence on the processing for receiving a message.
4) If "EN_R" is set ON during the next scan after the processing to receive message has been completed, the processing to receive a message is restarted.
5) "RD" must be kept unchanged while a message is being sent. Otherwise, the receive data cannot be guaranteed.
6) If the number specified by "SIZE" exceeds the size of variable specified by "RD", the data of other variable area is rewritten. Be sure to enter specified size of variable for "SIZE".
7) Programming should be done such that 1 (one) is input to "EN_R" after "VALID" of M_OPEN FB is set ON.

(2) FB format

1) When D300win loader is use



2) When the Standard loader is used



(3) Explanation of each FB terminal

Terminal function name	Terminal name	Data type	I/O	Description
Receive enable	ENR	BOOL	IN	While this signal is turned ON, receiving is enabled and the module can receive messages.
Connection No.	CON_NO	UINT	IN	Set a number for the connection that is opened by M_OPEN. The "CON_NO" terminal of M_OPEN FB is connected to this terminal.
Receive data size	SIZE	UINT	IN	Specify the size of receive data (the size of the variable that is connected to RD) in units of word.
Receive data	RD	ANY	IN-OUT	Connect a variable in which received data is stored.
Normal end of receiving	NDR	BOOL	OUT	Turn ON for the period of one scan when the receiving of data is completed successfully.
Abnormal end of receiving	ERROR	BOOL	OUT	Turn ON for the period of one scan if the receiving of data ended abnormally.
Status *1	STATUS	INT	OUT	A status code for the result of receiving a message is output.

***1 Status**

Error code	Cause
165 (see note)	Message receiving busy • When no message can be received during message communication via SX bus (After waiting for a while, re-execute the FB. If the error recurs, the message transmission load of the remote node may be too heavy.)
177	Parameter error • 0 (zero) is input to the "SIZE" terminal. • The variable that stores receive data exceeds specified memory area.
196	Message receive error (communication error between Ethernet module and other node)
199	Channel forcible close (communication error between Ethernet module and other node) • Request to close is received from the remote node. Note: If this code is received, close once the corresponding port and then re-execute the processing to open the port.
206	Buffer over • Data was received exceeding specified size of receive data
207	Connection number error • A connection number that is not yet opened is used. • "EN_R" (receive enable) is set ON when "EN_C" (request to open) of M_OPEN is set OFF. * Application must be made such that "receive enable" is also set OFF when the request to open of M_OPEN is set OFF. • A currently used connection number is used. (When two M_SEND FBs are used in parallel for one connection)

Note: According to the “running status information” of detail RAS, check the following item:

- SMC send 16 collision (offset address: 00000020)
- SMC send memory allocation incomplete (TASK) (offset address: 00000028)
- SMC send memory allocation (INT) (offset address: 00000030)
- SMC receive overrun (offset address: 00000090)
- SMC receive USNET buffer full (offset address: 000000A0)

If one of these counters has counted up, it seems that the load of the network line is too heavy. Adjust to reduce the load of the network line.

<Precautions for using M_SEND and M_RECEIVE FBs>

1) When communicating in General Purpose Communication mode using UDP/IP protocol, no receipt confirmation or flow control is performed. If the processing on the receiver side is too slow, the receive buffer becomes full and the data sent after that will be discarded. As a result, mismatch occurs between the number of data successfully sent from the sender side and the number of data successfully received on the receiver side.
If the receive buffer becomes full, it takes approximately 10 seconds to release the buffer, during which the operation to receive data may be stopped.
2) If a request to open is received in Full Passive open mode from a remote node whose IP address and port number do not match, the connection is established first, and then a request to close is sent from the Full Passive side to the Active side. As a result, on the Active side, forcible close (error code 199) occurs after the opening of port is successfully completed and data is sent.
3) If the port number on the sender side does not coincide with that on the receiver side, data send error occurs, and the receiver side forcibly closes the port. As a result, "forcible close" error (error code 199) occurs.
4) During communication between MICREX-SX devices, if one-word sending operation is performed consecutively, depending on the timing of M_RECEIVE, the receiver side may link the one word that is received first with the one ward that is received next and return the response to the effect that it received 2 words to the CPU. To avoid this, when one-word data is sent, the buffer area for 2 words should be secured on the receiver side. When 2 or more words are sent, the buffer area for the same number of words as that of send data should be secured on the receiver side.
5) When sending data after converting it into ASCII code in General Purpose Communication mode using UDP/IP protocol, if the number of bytes of send data exceeds 1017, the sender side divides the data into two pieces and performs twice the processing to send data. Therefore, the request to receive must be sent twice to the receiver side. In addition, the buffer area on the receiver side must be greater than the send data.

4-4-5 Remote data write (R_WRITE)

This function is used in Loader Command Communication mode to write data directly in the internal memory of the CPU of MICREX-SX system via Ethernet from SX_CPU. It can be used only when the remote node is an SX system.

(1) Operation of the command

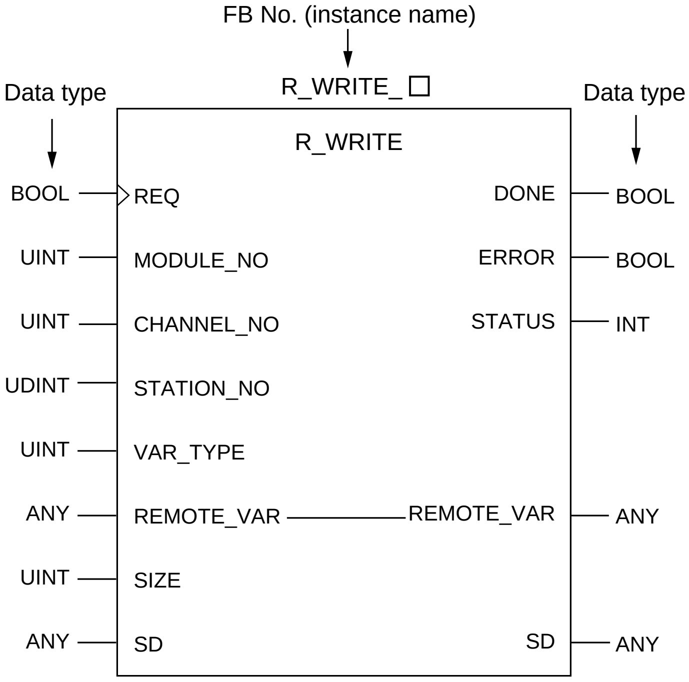
1) At the rising edge of "REQ", the data specified by "SD" is written in the area that is specified by "MODULE_NO", "CHANNEL_NO", "STATION_NO" and "REMOTE_VAR". (This operation does not complete within the period of one scan.)
2) When the writing of data is completed successfully, the "DONE" flag is set ON for the period of one scan.
3) If the writing of data cannot successfully be completed, the "ERROR" flag is set ON for the period of one scan, and the corresponding error code is output to "STATUS".

Notes:

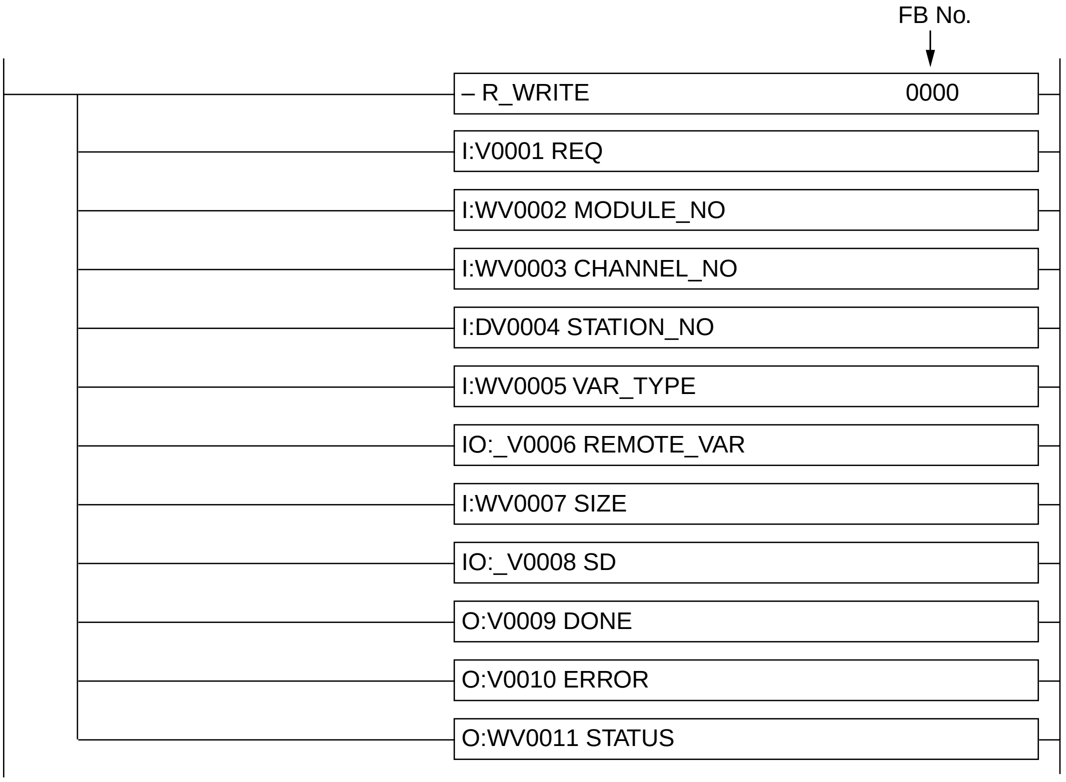
1) "REQ" command takes effect at its rising edge and is disabled during writing operation (since the rising edge of "REQ" till "DONE" or "ERROR" is set ON).
2) During writing operation "SD" must not be changed. If changed, the write data cannot be guaranteed.
3) There is no limit on "SIZE" when 0 (zero) is set for "VAR_TYPE". When other value is set, "SIZE" is limited by the communication module through which data is sent.
4) If the number of data specified by "SIZE" exceeds the size of variable specified by "SD", the data for the excess part becomes undefined. Be sure to enter specified size of variable for "SIZE".
5) When "R_WRITE" command is executed successively, the initially set maximum TCP end timer value must be reset to 0 (zero).
6) When "R_READ" and/or "R_WRITE" commands are executed, the reference number of local port and that of remote port must be set to default value (256).

(2) FB format

1) When D300win loader is used



2) When the Standard loader is used



(3) Explanation of each FB terminal

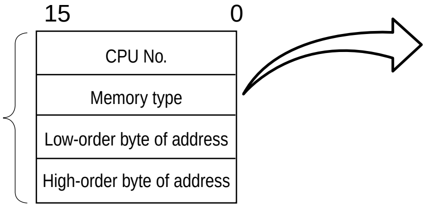
Terminal function name	Terminal name	Data type	I/O	Description
Request to write	REQ	BOOL	IN	Command is executed at the rising edge of this signal.
Communication module No.	MODULE_NO	UINT	IN	Specify the target of communication by SX bus station number of the Ethernet interface module (local module).
Communication channel No.	CHANNEL_NO	UINT	IN	For this module, this item is fixed to 0 (zero)
Remote station No.	STATION_NO	UDINT	IN	Specifies IP address for the remote node of communication.
Variable specification method	VAR_TYPE	UINT	IN	Set the method to specify variables. For this module, this item is fixed to 0 (zero).
Target variable to access *1	REMOTE_VAR	ANY	IN-OUT	Specify the target to write (top address) of remote SX_CPU.
Write data size	SIZE	UINT	IN	Specify the size of write data in units of word.
Write data	SD	ANY	IN-OUT	Connect the variable in which write data is stored
Normal end of writing	DONE	BOOL	OUT	Turn ON for the period of one scan when writing is completed successfully
Abnormal end of writing	ERROR	BOOL	OUT	Turn ON for the period of one scan when writing ended abnormally
Status *2	STATUS	INT	OUT	The result of writing operation is output by a status code.

***1 Target variable to access (REMOTE_VAR)**

The target SX_CPU internal memory in which data is to be written is specified in the following format:

<For D300win>

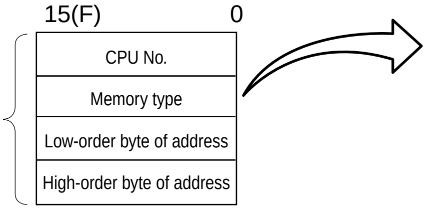
UINT type array data of 4 elements



Standard memory: UINT#1
Retain memory: UINT#3
System memory: UINT#10
Note: Values other than these must not be set for memory type.

<For the Standard loader>

Consecutive data areas



Standard memory: 1 (H1)
Retain memory: 3 (H3)
System memory: 10 (HA)
Note: Values other than these must not be set for memory type.

***2 Status**

Error code	Cause
35	Transmission interlock error • Remote CPU is running, such as downloading.
68	Memory address setting error • A wrong address is specified by REMOTE_VAR.
69	Memory size over • The address specified by REMOTE_VAR + SIZE exceed the effective range for the address of remote CPU.
160	Message send destination setting error • No module exists at the SX bus station No. specified by MODULE_NO. • Inexistent CPU No. is specified as the other party of communication by REMOTE_VAR.
171	Internal resources exhausted • If multiple R_READ and R_WRITE commands are executed at a time, internal resources may be exhausted. In such case, restart after waiting a while.
177	Parameter error • 0 (zero) was input to the SIZE terminal. • An out-of-spec value was input to VAR_TYPE. • No module exists at the station No. specified by MODULE_NO.
193	Channel open error • An abnormal value is set for station No.
201	Client port full • It was attempted to open ports exceeding the number specified for one communication module. (For this module, maximum 16 ports as the total of all communication modes can be opened.)

4-4-6 Remote data read (R_READ)

This function is used to directly read from SX_CPU the data stored in the CPU internal memory of MICREX-SX system via Ethernet.

(1) Operation of the command

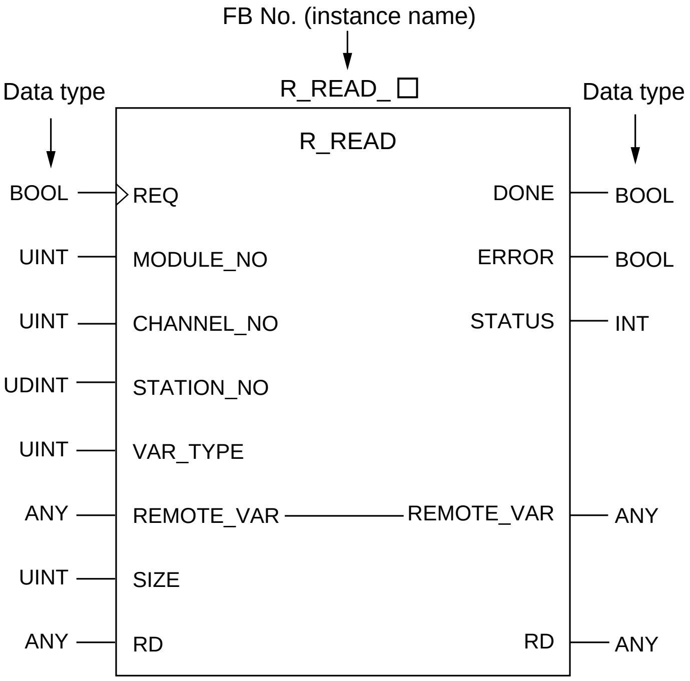
1) At the rising edge of "REQ", the data specified by "MODULE_NO", "CHANNEL_NO", "STATION_NO" and "REMOTE_VAR" is read out into the area that is specified by "RD". (This operation does not complete within the period of one scan.)
2) When the reading of data is completed successfully, the "DONE" flag is set ON for the period of one scan.
3) If the reading of data cannot successfully be completed, the "ERROR" flag is set ON for the period of one scan, and the corresponding error code is output to "STATUS".

Notes:

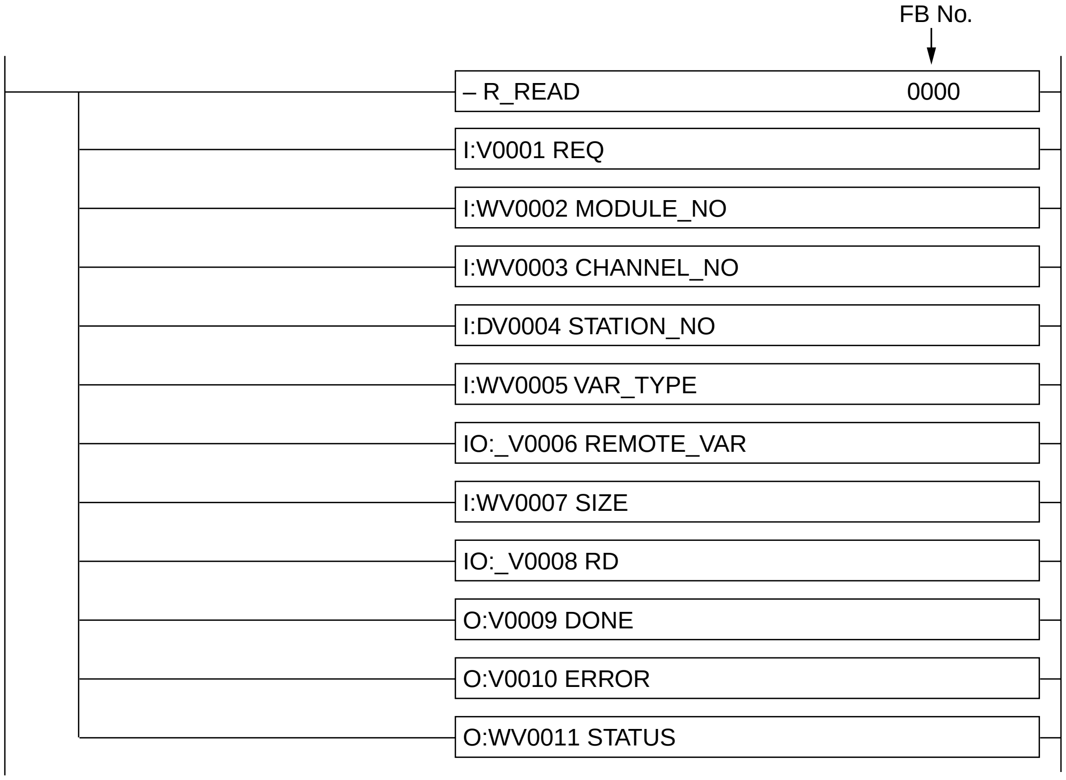
1) "REQ" command is disabled during reading operation (since the rising edge of "REQ" till "DONE" or "ERROR" is set ON).
2) During reading operation "RD" must not be changed. If changed, the read data cannot be guaranteed.
3) There is no limit on "SIZE" when 0 (zero) is set for "VAR_TYPE". When other value is set, "SIZE" is limited by the communication module through which data is sent.
4) If the number of data specified by "SIZE" exceeds the size of variable specified by "RD", other variable area is rewritten. Be sure to enter specified size of variable for "SIZE".
5) When "R_READ" command is executed successively, the initially set maximum TCP end timer value must be reset to 0 (zero).

(2) FB format

1) When D300win loader is used



2) When the Standard loader is used



(3) Explanation of each FB terminal

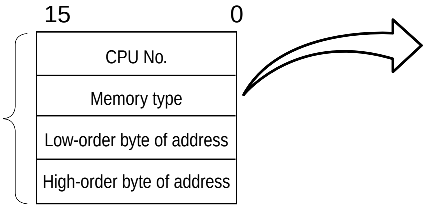
Terminal function name	Terminal name	Data type	I/O	Description
Request to write	REQ	BOOL	IN	Command is executed at the rising edge of this signal.
Communication module No.	MODULE_NO	UINT	IN	Specify the target of communication by SX bus station number of the Ethernet interface module (local module).
Communication channel No.	CHANNEL_NO	UINT	IN	For this module, this item is fixed to 0 (zero)
Remote station No.	STATION_NO	UDINT	IN	Specifies IP address for the remote node of communication.
Variable specification method	VAR_TYPE	UINT	IN	Set the method to specify variables. For this module, this item is fixed to 0 (zero).
Target variable to access *1	REMOTE_VAR	ANY	IN-OUT	Specify the target to write (top address) of remote SX_CPU.
Write data size	SIZE	UINT	IN	Specify the size of write data in units of word.
Write data	SD	ANY	IN-OUT	Connect the variable in which write data is stored
Normal end of writing	DONE	BOOL	OUT	Turn ON for the period of one scan when writing is completed successfully
Abnormal end of writing	ERROR	BOOL	OUT	Turn ON for the period of one scan when writing ended abnormally
Status *2	STATUS	INT	OUT	The result of writing operation is output by a status code.

***1 Target variable to access (REMOTE_VAR)**

The target SX_CPU internal memory in which data is to be written is specified in the following format:

<For D300win>

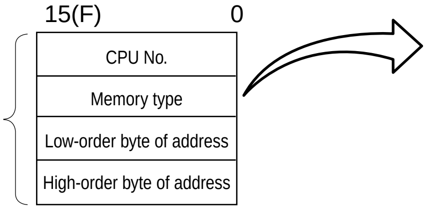
UINT type array data of 4 elements



Standard memory: UINT#1
Retain memory: UINT#3
System memory: UINT#10
Note: Values other than these must not be set for memory type.

<For the Standard loader>

Consecutive data areas



Standard memory: 1 (H1)
Retain memory: 3 (H3)
System memory: 10 (HA)
Note: Values other than these must not be set for memory type.

***2 Status**

Error code	Cause
35	Transmission interlock error • Remote CPU is running, such as downloading.
68	Memory address setting error • A wrong address is specified by REMOTE_VAR.
69	Memory size over • The address specified by REMOTE_VAR + SIZE exceed the effective range for the address of remote CPU.
160	Message send destination setting error • No module exists at the SX bus station No. specified by MODULE_NO. • Inexistent CPU No. is specified as the other party of communication by REMOTE_VAR.
171	Internal resources exhausted • If multiple R_READ and R_WRITE commands are executed at a time, internal resources may be exhausted. In such case, restart after waiting a while.
177	Parameter error • 0 (zero) was input to the SIZE terminal. • An out-of-spec value was input to VAR_TYPE. • No module exists at the station No. specified by MODULE_NO.
193	Channel open error • An abnormal value is set for station No.
201	Client port full • It was attempted to open ports exceeding the number specified for one communication module. (For this module, maximum 16 ports as the total of all communication modes can be opened.)

Section 5 Sample Programs

Page

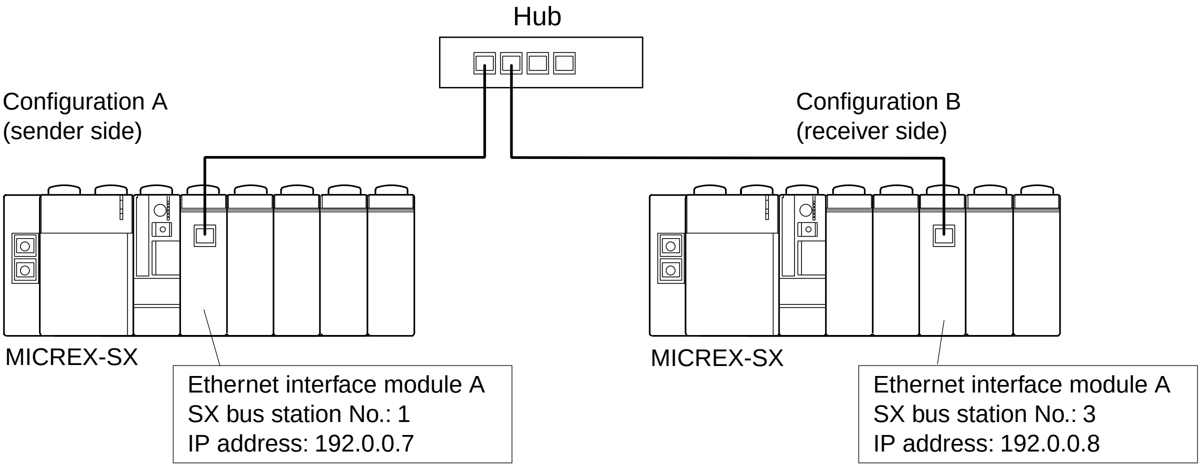
Section 5 Sample Programs

This section introduces sample programs that use the D300win loader. When the Standard loader is used, create a program referring to these examples.

5-1 General Purpose Communication Mode

A sample program is shown below, which sends data from configuration A to configuration B in General Purpose Communication mode under the following system configuration.

<System configuration>



Ethernet interface module A
SX bus station No.: 1
IP address: 192.0.0.7

Ethernet interface module A
SX bus station No.: 3
IP address: 192.0.0.8

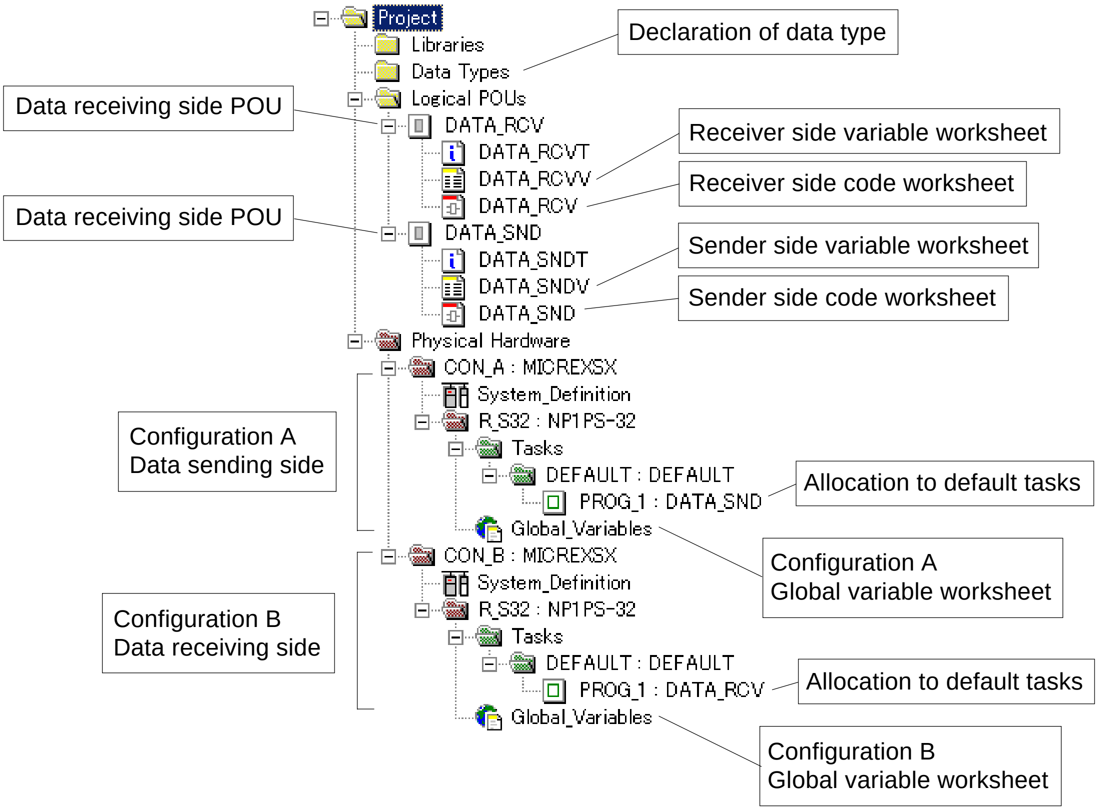
<Operation of the sample program>
The sample program sends 5 words of word type data from the internal memory (variable) of the CPU of configuration A to the internal memory (variable) of the CPU of configuration B.

<Execution sequence of the communication program>

1) In configuration B (receiver side), M_OPEN FB is executed to open a port on the receiver side.
2) In configuration B (receiver side), M_RECEIVE FB is executed to make the module ready for receiving data.
3) In configuration A (sender side), M_OPEN FB is executed to open a port on the sender side.
4) In configuration A (sender side), M_SEND FB is executed to send data.

(1) Project tree

The sample program shown above was created based on the following project tree.



```
Project
  Libraries
  Data Types
  Logical POUs
    DATA_RCV
      DATA_RCVT
      DATA_RCVV
      DATA_RCV
    DATA_SND
      DATA_SNDT
      DATA_SNDV
      DATA_SND
  Physical Hardware
    CON_A : MICREXSX
      System_Definition
      R_S32 : NP1PS-32
        Tasks
          DEFAULT : DEFAULT
            PROG_1 : DATA_SND
        Global_Variables
    CON_B : MICREXSX
      System_Definition
      R_S32 : NP1PS-32
        Tasks
          DEFAULT : DEFAULT
            PROG_1 : DATA_RCV
        Global_Variables
```

Declaration of data type

Data receiving side POU

Receiver side variable worksheet

Receiver side code worksheet

Data receiving side POU

Sender side variable worksheet

Sender side code worksheet

Configuration A
Data sending side

Allocation to default tasks

Configuration A
Global variable worksheet

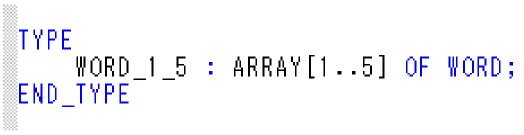
Configuration B
Data receiving side

Allocation to default tasks

Configuration B
Global variable worksheet

(2) Declaration of data type

In order to communicate 5 words of data, it is necessary to declare the data type of word type array of 5 words.

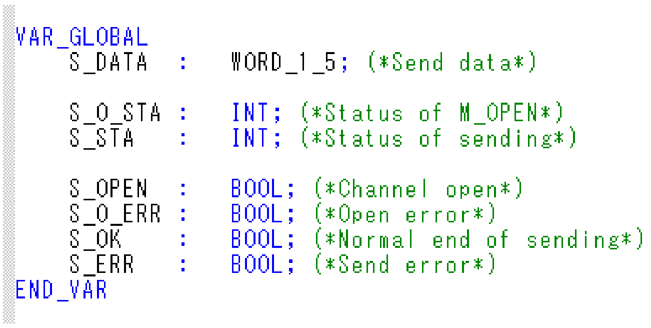


```
TYPE
    WORD_1_5 : ARRAY[1..5] OF WORD;
END_TYPE
```

* In this example, the name "WORD_1_5" is given so that the starting and end element numbers of array and the data type of elements can be known.

(3) Data send program (configuration A side)

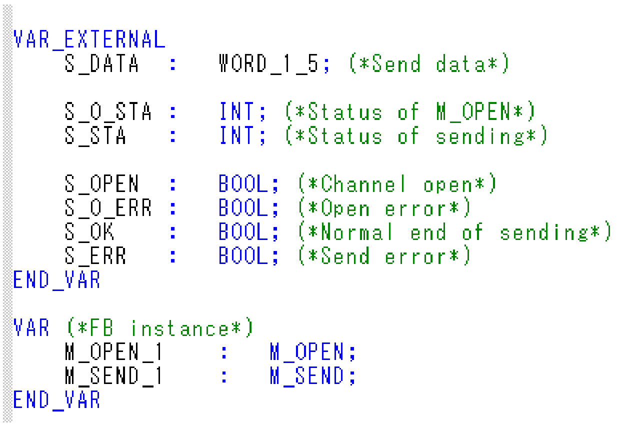
1) Global variable worksheet



```
VAR_GLOBAL
    S_DATA  :   WORD_1_5; (*Send data*)

    S_O_STA :   INT; (*Status of M_OPEN*)
    S_STA   :   INT; (*Status of sending*)

    S_OPEN  :   BOOL; (*Channel open*)
    S_O_ERR :   BOOL; (*Open error*)
    S_OK    :   BOOL; (*Normal end of sending*)
    S_ERR   :   BOOL; (*Send error*)
END_VAR
```

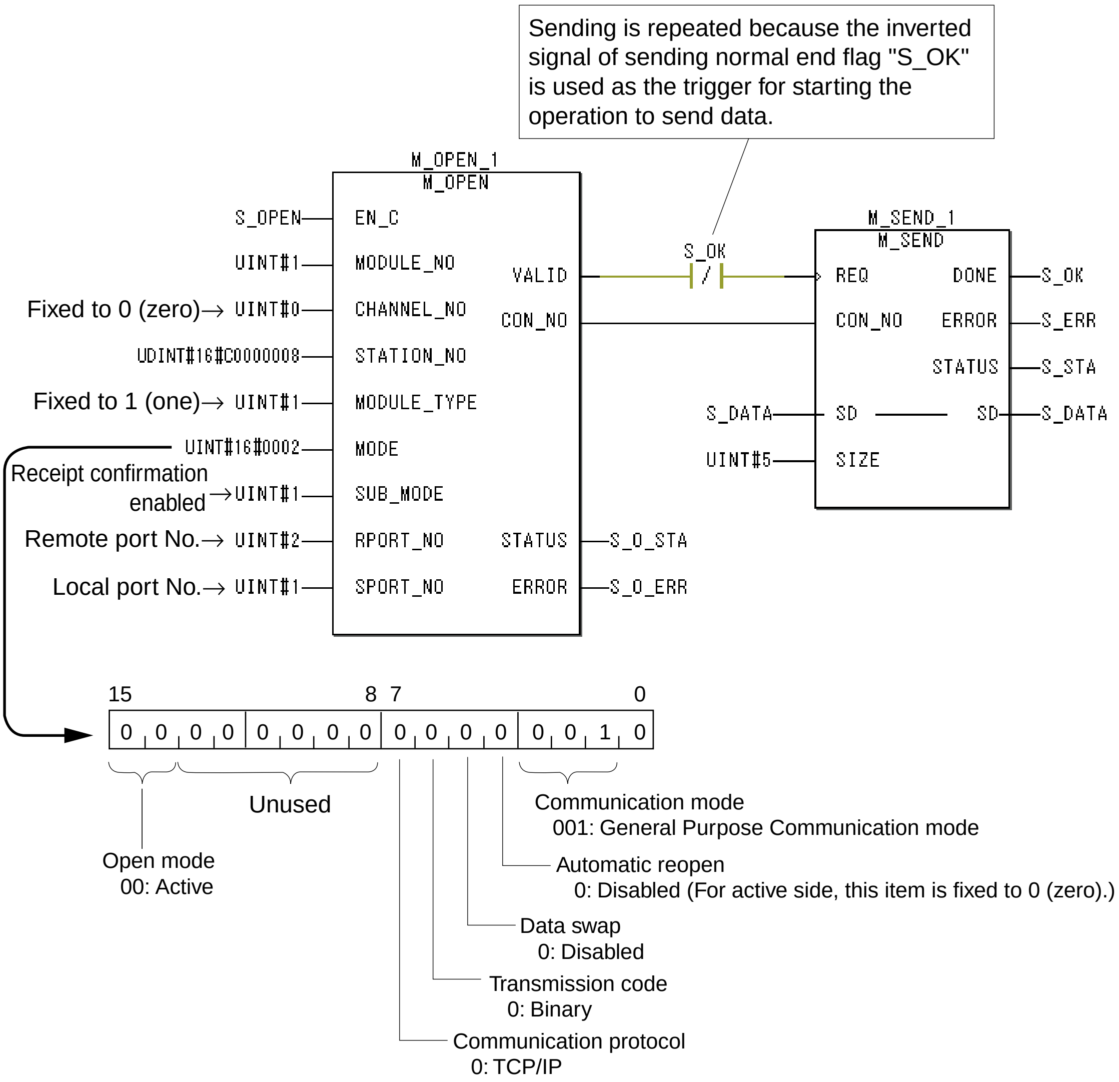
5-1 General Purpose Communication Mode

2) Local variable worksheet



```
VAR_EXTERNAL
    S_DATA   :   WORD_1_5; (*Send data*)

    S_O_STA :   INT; (*Status of M_OPEN*)
    S_STA   :   INT; (*Status of sending*)

    S_OPEN   :   BOOL; (*Channel open*)
    S_O_ERR :   BOOL; (*Open error*)
    S_OK     :   BOOL; (*Normal end of sending*)
    S_ERR    :   BOOL; (*Send error*)
END_VAR

VAR (*FB instance*)
    M_OPEN_1     :   M_OPEN;
    M_SEND_1     :   M_SEND;
END_VAR
```

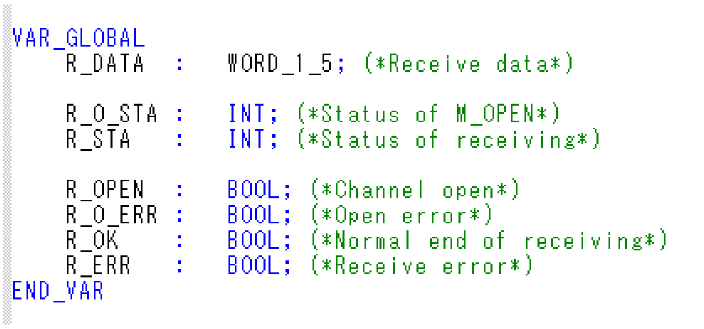
3) Code worksheet



Sending is repeated because the inverted signal of sending normal end flag "S_OK" is used as the trigger for starting the operation to send data.

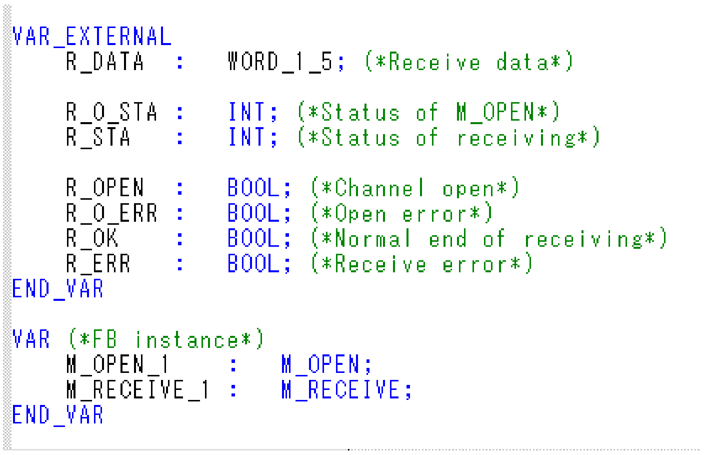
(4) Data receive program (configuration B side)

1) Global variable worksheet



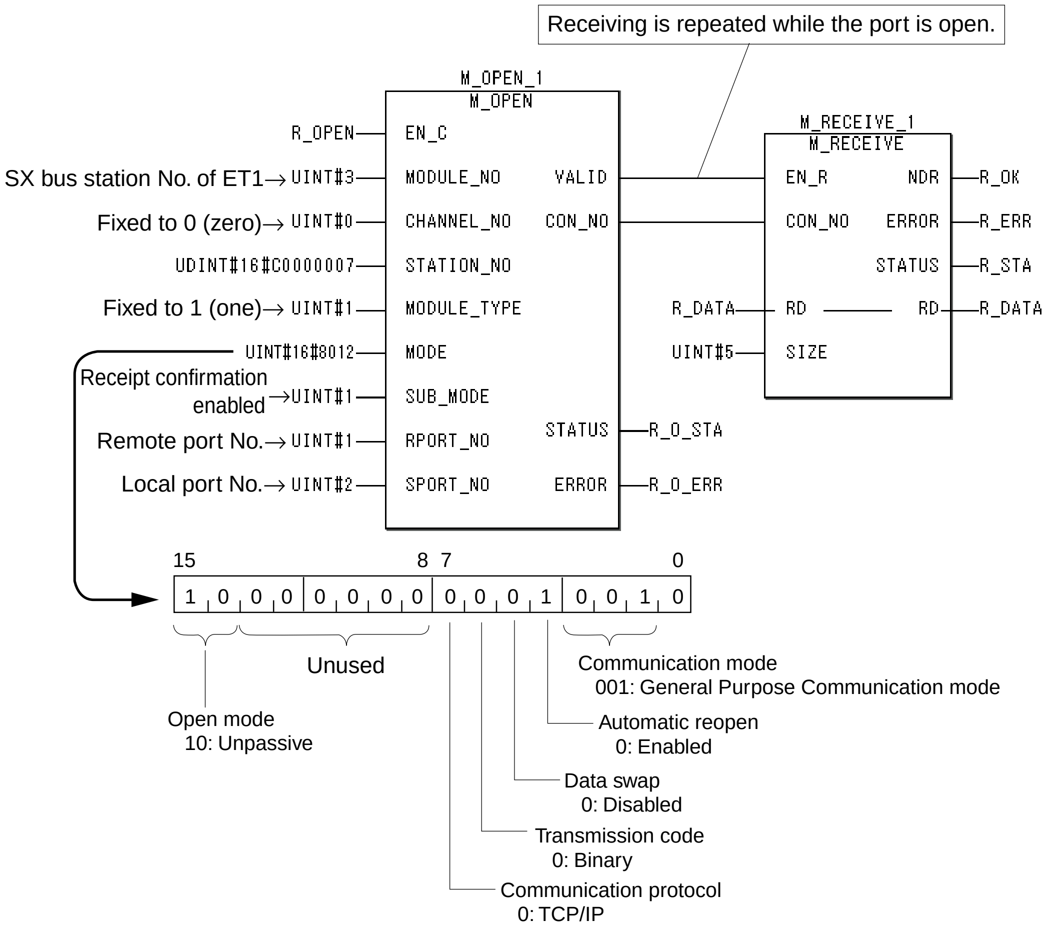
```
VAR_GLOBAL
    R_DATA  :   WORD_1_5; (*Receive data*)

    R_O_STA :   INT; (*Status of M_OPEN*)
    R_STA   :   INT; (*Status of receiving*)

    R_OPEN  :   BOOL; (*Channel open*)
    R_O_ERR :   BOOL; (*Open error*)
    R_OK    :   BOOL; (*Normal end of receiving*)
    R_ERR   :   BOOL; (*Receive error*)
END_VAR
```

2) Local variable worksheet



```
VAR_EXTERNAL
    R_DATA  :   WORD_1_5; (*Receive data*)

    R_O_STA :   INT; (*Status of M_OPEN*)
    R_STA   :   INT; (*Status of receiving*)

    R_OPEN  :   BOOL; (*Channel open*)
    R_O_ERR :   BOOL; (*Open error*)
    R_OK    :   BOOL; (*Normal end of receiving*)
    R_ERR   :   BOOL; (*Receive error*)
END_VAR

VAR (*FB instance*)
    M_OPEN_1    :   M_OPEN;
    M_RECEIVE_1 :   M_RECEIVE;
END_VAR
```

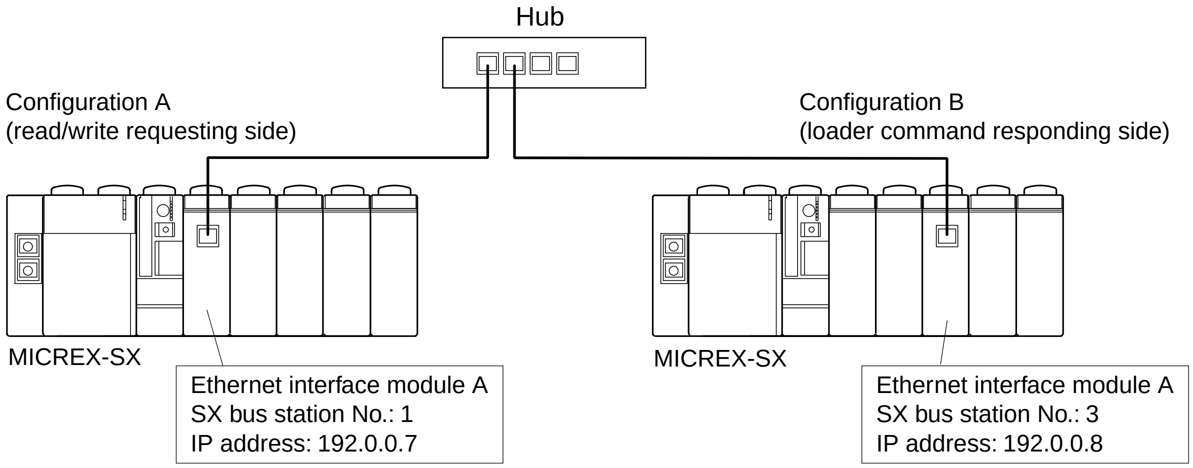
3) Code worksheet



5-2 Loader Command Communication Mode

Sample programs are shown below, which writes data in the internal memory of the CPU of configuration B from configuration A or reads data from the internal memory of the CPU of configuration B in Loader Command Communication mode under the following system configuration.

<System configuration>



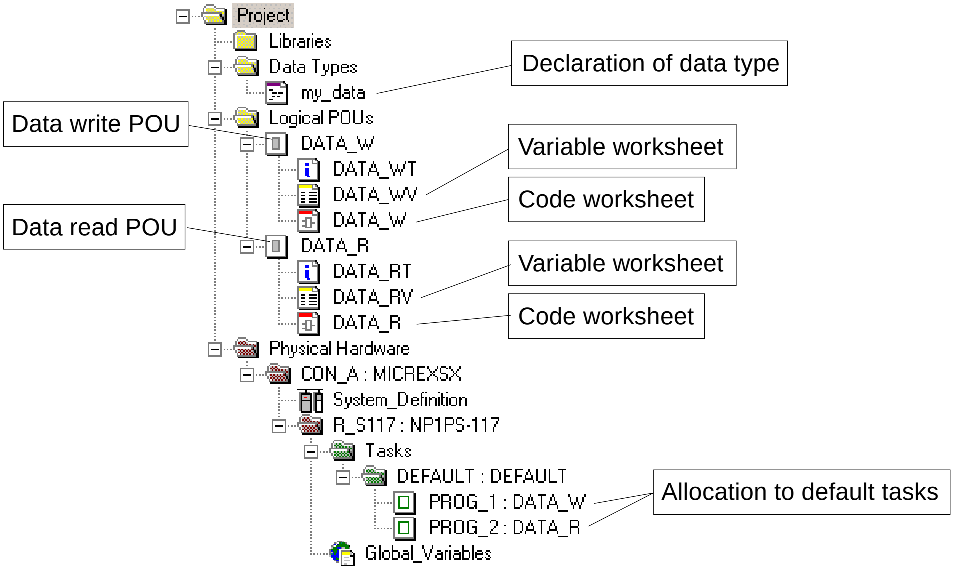
<Operation of the sample program>
The sample program writes data from the internal memory of the CPU of configuration A into the internal memory of the CPU of configuration B (data write program) or reads data from the internal memory of the CPU of configuration B into the internal memory of the CPU of configuration A (data read program).
  * For these sample programs, no communication program is necessary on the side of configuration B.

<Execution sequence of communication program, data writing>
1) In configuration A (data write requesting side), R_WRITE FB is executed.

<Execution sequence of communication program, data reading>
1) In configuration A (data read requesting side), R_READ FB is executed.

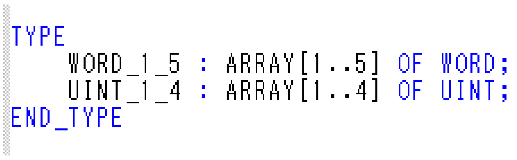
(1) Project tree

The sample program shown above was created based on the following project tree.



(2) Declaration of data type

In order to write or read 5 words of data, it is necessary to declare the data type of word type array of 5 words. It is also necessary to declare the data type of UINT type array of 4 words for the purpose of specifying internal memories of the CPU of the other party of communication (configuration B).

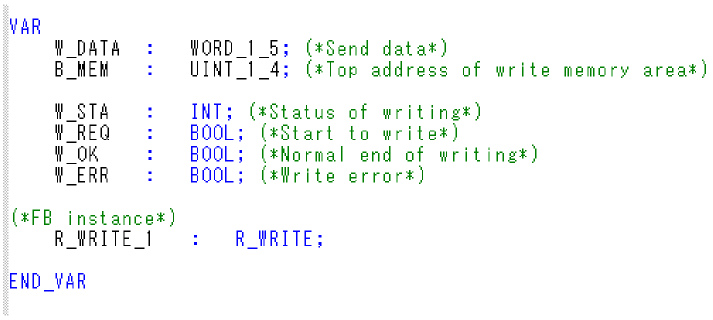


```
TYPE
    WORD_1_5 : ARRAY[1..5] OF WORD;
    UINT_1_4 : ARRAY[1..4] OF UINT;
END_TYPE
```

* In this example, the names "WORD_1_5" and "UINT_1_4" are given so that the starting and end element numbers of array and the data type of elements can be known.

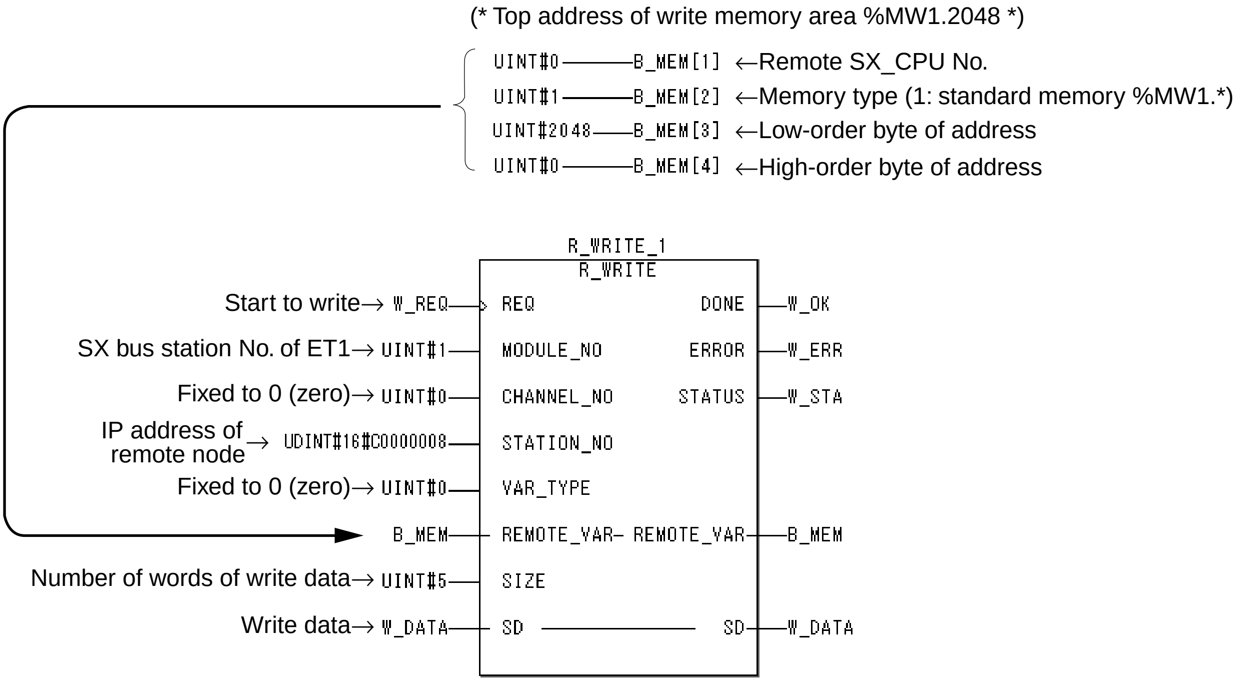
(3) Data write program

1) Local variable worksheet



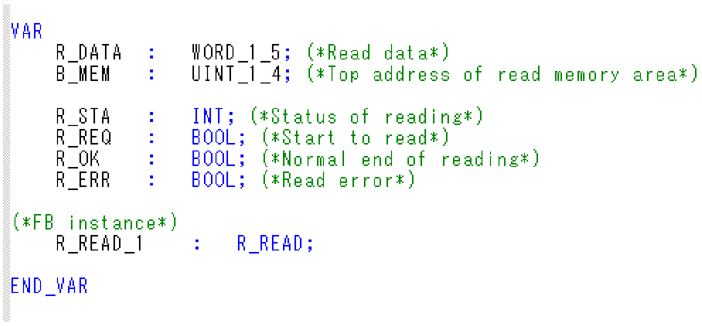
```
VAR
    W_DATA   :   WORD_1_5; (*Send data*)
    B_MEM    :   UINT_1_4; (*Top address of write memory area*)

    W_STA    :   INT; (*Status of writing*)
    W_REQ    :   BOOL; (*Start to write*)
    W_OK     :   BOOL; (*Normal end of writing*)
    W_ERR    :   BOOL; (*Write error*)

(*FB instance*)
    R_WRITE_1    :   R_WRITE;

END_VAR
```

2) Code worksheet


(* Top address of write memory area %MW1.2048 *)
UINT#0 — B_MEM[1] ←Remote SX_CPU No.
UINT#1 — B_MEM[2] ←Memory type (1: standard memory %MW1.*)
UINT#2048 — B_MEM[3] ←Low-order byte of address
UINT#0 — B_MEM[4] ←High-order byte of address
R_WRITE_1
R_WRITE
Start to write→ W_REQ
REQ
DONE
W_OK
SX bus station No. of ET1→ UINT#1
MODULE_NO
ERROR
W_ERR
Fixed to 0 (zero)→ UINT#0
CHANNEL_NO
STATUS
W_STA
IP address of remote node→ UDINT#16#C0000008
STATION_NO
Fixed to 0 (zero)→ UINT#0
VAR_TYPE
B_MEM
REMOTE_VAR
REMOTE_VAR
B_MEM
Number of words of write data→ UINT#5
SIZE
Write data→ W_DATA
SD
SD
W_DATA

(4) Data read program

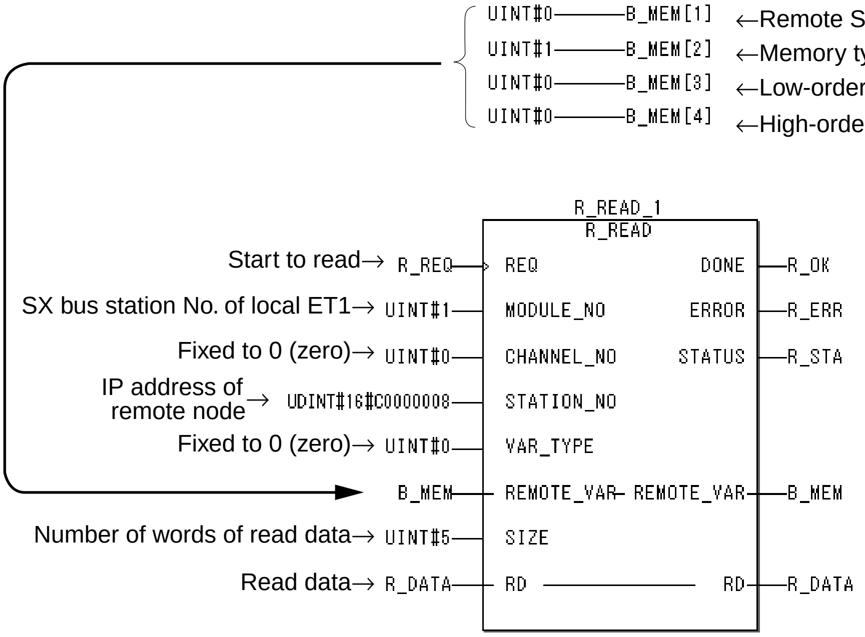
1) Local variable worksheet



```
VAR
    R_DATA  :   WORD_1_5; (*Read data*)
    B_MEM   :   UINT_1_4; (*Top address of read memory area*)

    R_STA   :   INT; (*Status of reading*)
    R_REQ   :   BOOL; (*Start to read*)
    R_OK    :   BOOL; (*Normal end of reading*)
    R_ERR   :   BOOL; (*Read error*)

(*FB instance*)
    R_READ_1    :   R_READ;

END_VAR
```

2) Code worksheet

(* Top address of read memory area %MW1.0 *)



←Remote SX_CPU No.
←Memory type (1: standard memory %MW1.*)
←Low-order byte of address
←High-order byte of address

Start to read→
SX bus station No. of local ET1→
Fixed to 0 (zero)→
IP address of remote node→
Fixed to 0 (zero)→
Number of words of read data→
Read data→

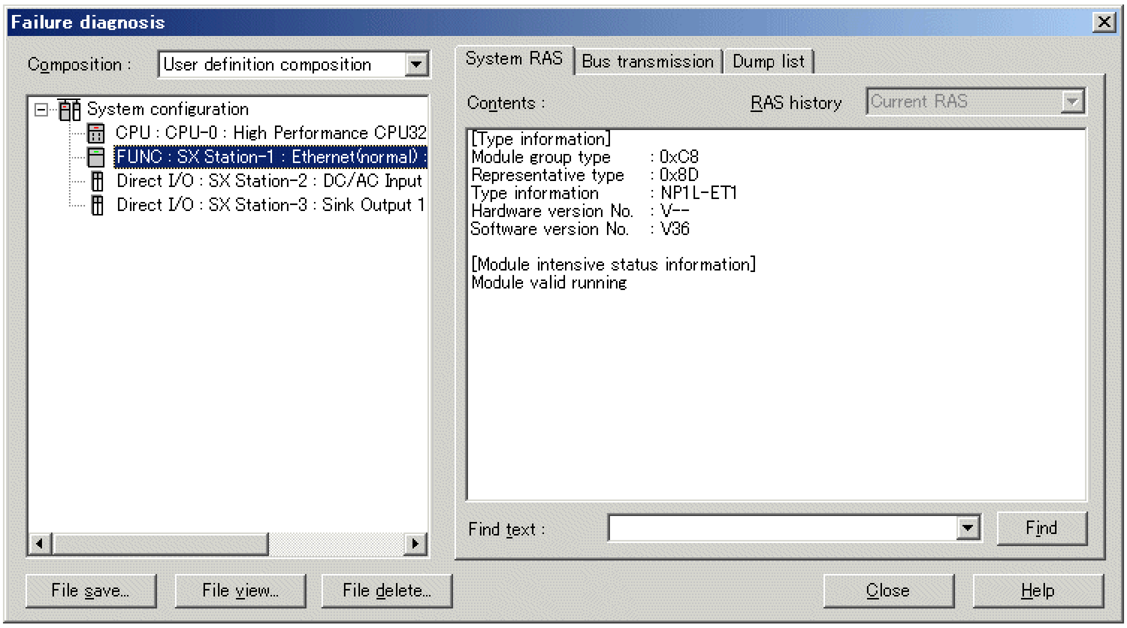
Section 6 Fault Diagnosis

Page

To diagnose NP1L-ET1/ET2, the diagnostic function of the loader is used. When this module is selected from the “Fault diagnosis” dialog box, various information related to the module is displayed, as described below. (The layout of the fault diagnosis window depends on the version of loader. In this section, explanation is made based on the fault diagnosis window for Version 3 of D300win and the Standard loader.)

[Content of system RAS]

When the [System RAS] tab is clicked, the following information is displayed.


Failure diagnosis
Composition :
User definition composition
System configuration
CPU : CPU-0 : High Performance CPU32
FUNC : SX Station-1 : Ethernet(normal) :
Direct I/O : SX Station-2 : DC/AC Input
Direct I/O : SX Station-3 : Sink Output 1
System RAS
Bus transmission
Dump list
Contents :
RAS history
Current RAS
Find text :
Find
File save...
File view...
File delete...
Close
Help

```
[Type information]
Module group type       : 0xC8
Representative type     : 0x8D
Type information        : NP1L-ET1
Hardware version No.    : V--
Software version No.    : V36

[Module intensive status information]
Module valid running
```

(1a) Type information

Model, type and software version of this module are displayed.

(1b) Module intensive status information

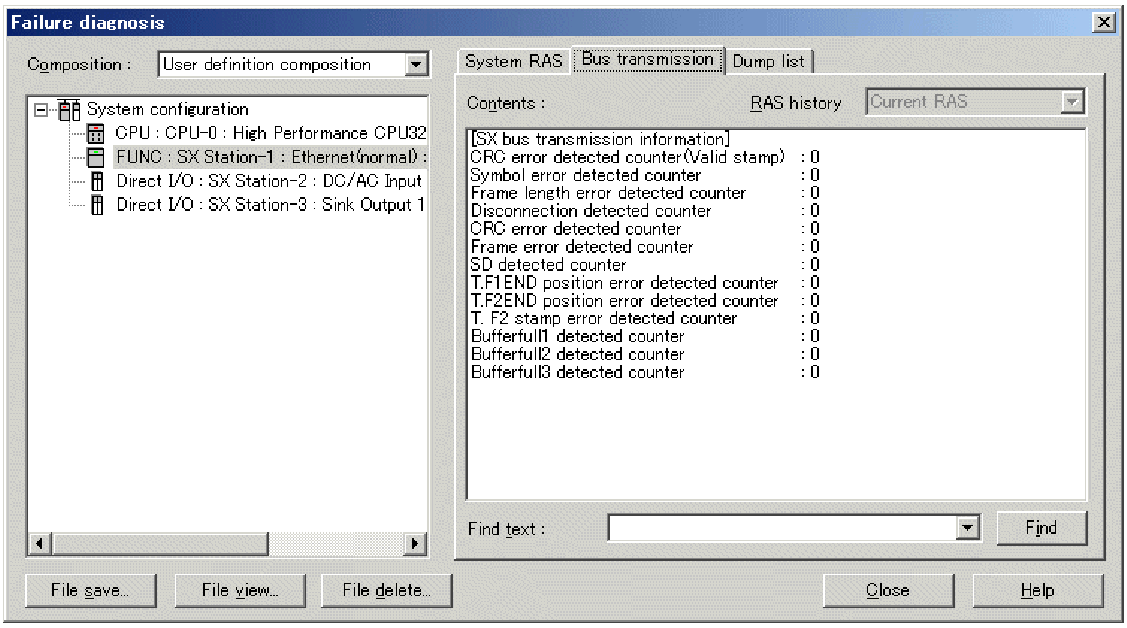
Intensive status information for this module is displayed.

<List of displayed messages>

Classification	Displayed message	Description
Normal	Module valid running	Initialization is completed and module is running normally.
Fatal fault	Module fatal fault: WDT Up error	Hardware fault. The module requires being replaced.
	Module fatal fault: CPU error	Hardware fault. The module requires being replaced.
	Module fatal fault: ROM error	Hardware fault. The module requires being replaced.
	Module fatal fault: RAM error	Hardware fault. The module requires being replaced.
	Module fatal fault: SX bus error	Hardware fault. The module requires being replaced.
	Module fatal fault: ROM write error	Hardware fault. The module requires being replaced.
Nonfatal fault	Module nonfatal fault: Setting error	Communication initialization error

[Content of bus transmission]

When the [Bus transmission] tab is clicked, the following information is displayed.


Failure diagnosis
Composition :
User definition composition
System configuration
CPU : CPU-0 : High Performance CPU32
FUNC : SX Station-1 : Ethernet(normal) :
Direct I/O : SX Station-2 : DC/AC Input
Direct I/O : SX Station-3 : Sink Output 1
System RAS
Bus transmission
Dump list
Contents :
RAS history
Current RAS
Find text :
Find
File save...
File view...
File delete...
Close
Help

```
[SX bus transmission information]
CRC error detected counter(Valid stamp)   : 0
Symbol error detected counter             : 0
Frame length error detected counter       : 0
Disconnection detected counter            : 0
CRC error detected counter                : 0
Frame error detected counter              : 0
SD detected counter                       : 0
T.F1END position error detected counter   : 0
T.F2END position error detected counter   : 0
T. F2 stamp error detected counter        : 0
Bufferfull1 detected counter              : 0
Bufferfull2 detected counter              : 0
Bufferfull3 detected counter              : 0
```

(2) SX bus transmission information

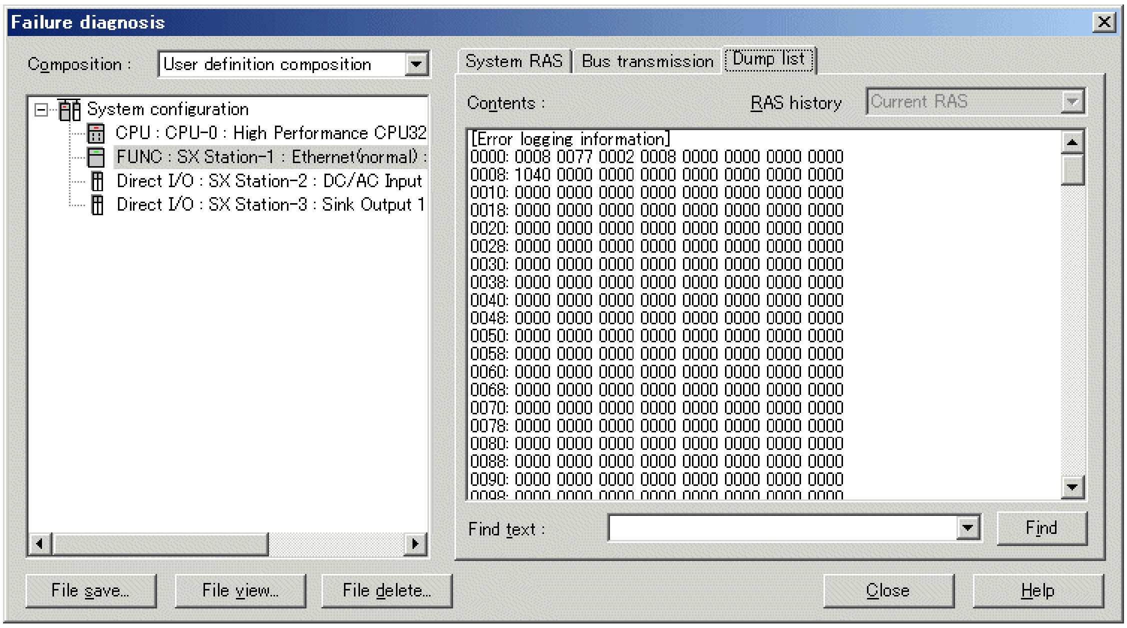
The information of the errors related to SX bus transmission that have been detected by this module is displayed.
This information is counted up when the transmission waveform of SX bus is disturbed by noise or other causes.

<List of SX bus transmission statuses>

SX bus transmission status	Description
CRC error detected counter (Valid stamp)	Counted up if CRC error is detected but stamp is valid.
Symbol error detected counter	Counted up if symbol error is detected.
Frame length error detected counter	Counted up if frame length error is detected.
Disconnection detected counter	Counted up if disconnection is detected (if transmission is stopped for 130 ms or longer time.)
CRC error detected counter	Counted up if CRC error is detected.
Frame error detected counter	Counted up if frame receiving is stopped in the middle of frame receiving operation.
SD detected counter	Counted up if SD (the code indicating the top of a frame) is received in the middle of frame receiving operation.
TF1 END position detected counter	Counted up if END is not at specified position during the 1st cycle of total frame receiving.
TF2 END position detected counter	Counted up if END is not at specified position during the 2nd cycle of total frame receiving.
TF2 stamp error detected counter	Counted up if error is found in a received frame during the 2nd cycle of total frame receiving.
Buffer full 1 detected counter	Counted up if buffer full is detected in the middle of total frame receiving or multicast frame receiving.
Buffer full 2 detected counter	Counted up if buffer full is detected in the middle of message, ACK or NAK receiving.
Buffer full 3 detected counter	Counted up if buffer full is detected in the middle of token frame receiving.

Content of Dump list

When the [Dump list] tab is clicked, the following information is displayed.



Composition : User definition composition

- System configuration
  - CPU : CPU-0 : High Performance CPU32
  - FUNC : SX Station-1 : Ethernet(normal) :
  - Direct I/O : SX Station-2 : DC/AC Input
  - Direct I/O : SX Station-3 : Sink Output 1

Contents :

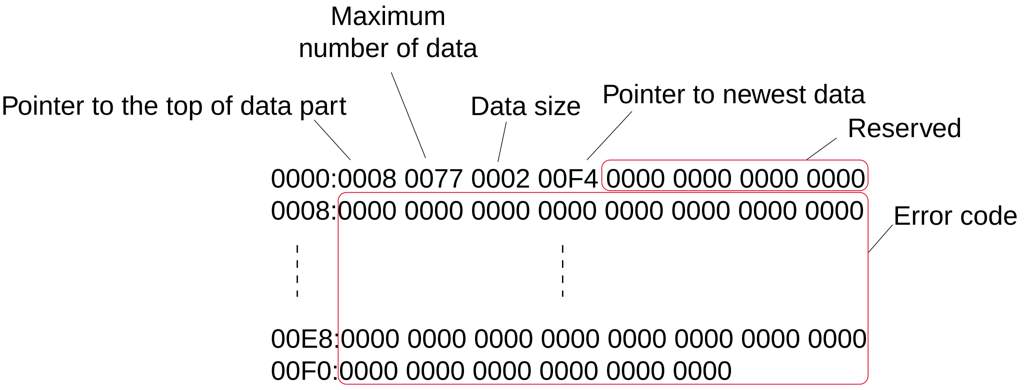
RAS history: Current RAS

```
[Error logging information]
0000: 0008 0077 0002 0008 0000 0000 0000 0000
0008: 1040 0000 0000 0000 0000 0000 0000 0000
0010: 0000 0000 0000 0000 0000 0000 0000 0000
0018: 0000 0000 0000 0000 0000 0000 0000 0000
0020: 0000 0000 0000 0000 0000 0000 0000 0000
0028: 0000 0000 0000 0000 0000 0000 0000 0000
0030: 0000 0000 0000 0000 0000 0000 0000 0000
0038: 0000 0000 0000 0000 0000 0000 0000 0000
0040: 0000 0000 0000 0000 0000 0000 0000 0000
0048: 0000 0000 0000 0000 0000 0000 0000 0000
0050: 0000 0000 0000 0000 0000 0000 0000 0000
0058: 0000 0000 0000 0000 0000 0000 0000 0000
0060: 0000 0000 0000 0000 0000 0000 0000 0000
0068: 0000 0000 0000 0000 0000 0000 0000 0000
0070: 0000 0000 0000 0000 0000 0000 0000 0000
0078: 0000 0000 0000 0000 0000 0000 0000 0000
0080: 0000 0000 0000 0000 0000 0000 0000 0000
0088: 0000 0000 0000 0000 0000 0000 0000 0000
0090: 0000 0000 0000 0000 0000 0000 0000 0000
0098: 0000 0000 0000 0000 0000 0000 0000 0000
```

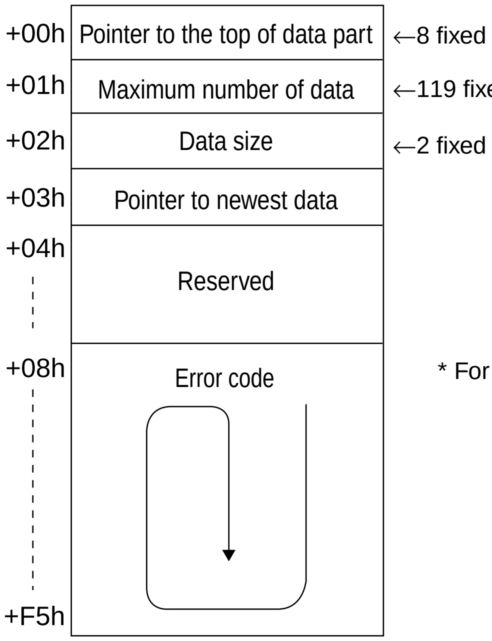
Find text :

(3) Error logging function

This function stores the error information that is detected during Ethernet communication. Error log is stored in ring files. Using 2 words for one information, maximum 119 error information can be stored.



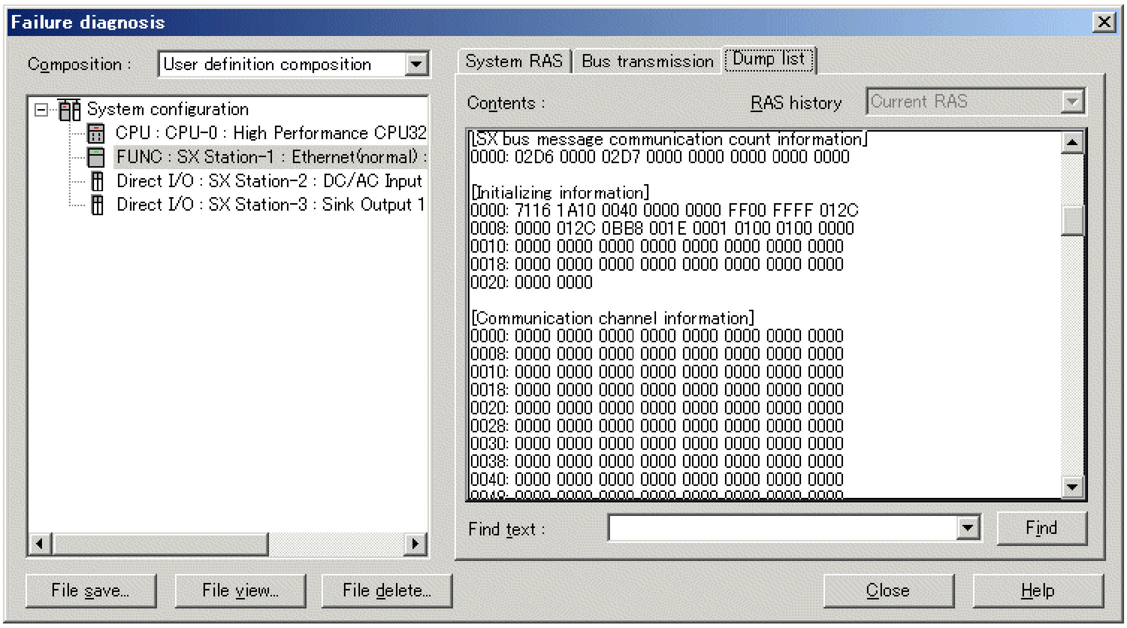
```
0000:0008 0077 0002 00F4 0000 0000 0000 0000
0008:0000 0000 0000 0000 0000 0000 0000 0000

00E8:0000 0000 0000 0000 0000 0000 0000 0000
00F0:0000 0000 0000 0000 0000 0000
```



* For error codes, refer to paragraph (9).

(4) SX bus message transmission count information

This function stores the total number of operations having done for message transmission via SX bus.


Failure diagnosis
System configuration
CPU : CPU-0 : High Performance CPU32
FUNC : SX Station-1 : Ethernet(normal) :
Direct I/O : SX Station-2 : DC/AC Input
Direct I/O : SX Station-3 : Sink Output 1
System RAS
Bus transmission
Dump list
Find
File save...
File view...
File delete...
Close
Help

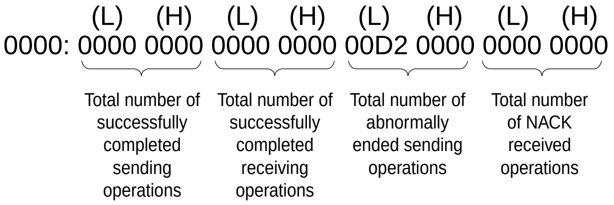
Composition : User definition composition

Contents :

RAS history: Current RAS

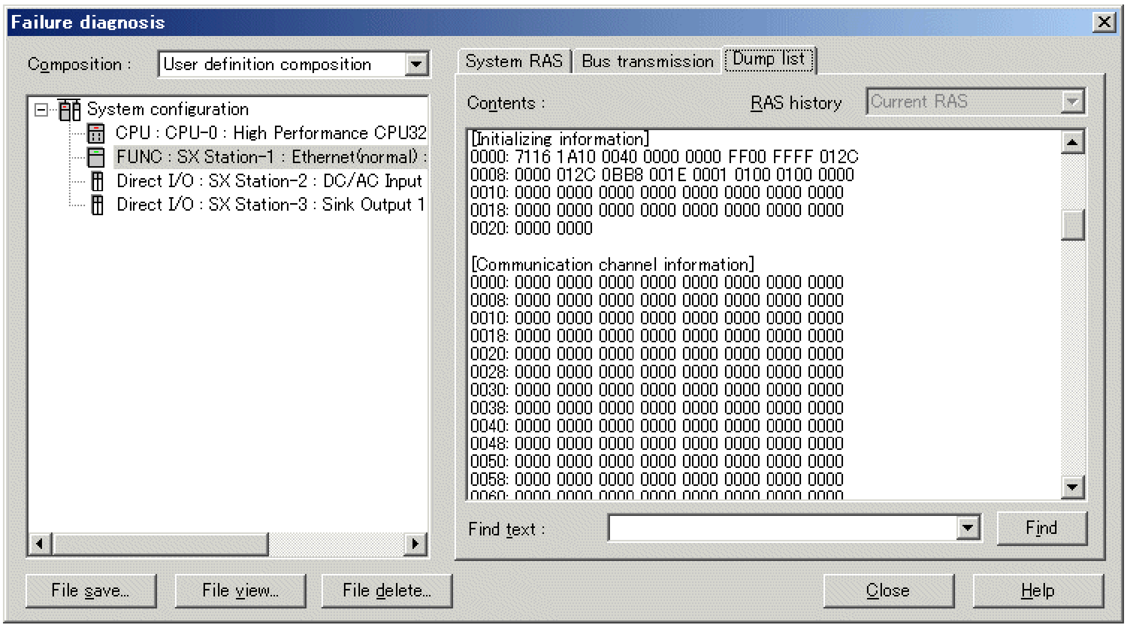
```
[SX bus message communication count information]
0000: 02D6 0000 02D7 0000 0000 0000 0000 0000

[Initializing information]
0000: 7116 1A10 0040 0000 0000 FF00 FFFF 012C
0008: 0000 012C 0BB8 001E 0001 0100 0100 0000
0010: 0000 0000 0000 0000 0000 0000 0000 0000
0018: 0000 0000 0000 0000 0000 0000 0000 0000
0020: 0000 0000

[Communication channel information]
0000: 0000 0000 0000 0000 0000 0000 0000 0000
0008: 0000 0000 0000 0000 0000 0000 0000 0000
0010: 0000 0000 0000 0000 0000 0000 0000 0000
0018: 0000 0000 0000 0000 0000 0000 0000 0000
0020: 0000 0000 0000 0000 0000 0000 0000 0000
0028: 0000 0000 0000 0000 0000 0000 0000 0000
0030: 0000 0000 0000 0000 0000 0000 0000 0000
0038: 0000 0000 0000 0000 0000 0000 0000 0000
0040: 0000 0000 0000 0000 0000 0000 0000 0000
0048: 0000 0000 0000 0000 0000 0000 0000 0000
```

Find text :


(L)
(H)
(L)
(H)
(L)
(H)
(L)
(H)
0000: 0000 0000 0000 0000 00D2 0000 0000 0000
Total number of successfully completed sending operations
Total number of successfully completed receiving operations
Total number of abnormally ended sending operations
Total number of NACK received operations

(5) Initialization information

The information of Ethernet address, IP address, gateway setting, etc. for this module is displayed.



Composition : User definition composition

System configuration
CPU : CPU-0 : High Performance CPU32
FUNC : SX Station-1 : Ethernet(normal) :
Direct I/O : SX Station-2 : DC/AC Input
Direct I/O : SX Station-3 : Sink Output 1

Contents :

RAS history: Current RAS

```
[Initializing information]
0000: 7116 1A10 0040 0000 0000 FF00 FFFF 012C
0008: 0000 012C 0BB8 001E 0001 0100 0100 0000
0010: 0000 0000 0000 0000 0000 0000 0000 0000
0018: 0000 0000 0000 0000 0000 0000 0000 0000
0020: 0000 0000

[Communication channel information]
0000: 0000 0000 0000 0000 0000 0000 0000 0000
0008: 0000 0000 0000 0000 0000 0000 0000 0000
0010: 0000 0000 0000 0000 0000 0000 0000 0000
0018: 0000 0000 0000 0000 0000 0000 0000 0000
0020: 0000 0000 0000 0000 0000 0000 0000 0000
0028: 0000 0000 0000 0000 0000 0000 0000 0000
0030: 0000 0000 0000 0000 0000 0000 0000 0000
0038: 0000 0000 0000 0000 0000 0000 0000 0000
0040: 0000 0000 0000 0000 0000 0000 0000 0000
0048: 0000 0000 0000 0000 0000 0000 0000 0000
0050: 0000 0000 0000 0000 0000 0000 0000 0000
0058: 0000 0000 0000 0000 0000 0000 0000 0000
0060: 0000 0000 0000 0000 0000 0000 0000 0000
```

Find text :

Initializing information consists of top address and data part, as shown below.

Top address	Data							
0000:	Item 0	Item 1	Item 2	Item 3	Item 4	Item 5	Item 6	Item 7
0008:	Item 8	Item 9	Item A	Item B	Item C	Item D	Item E	Item F
0010:	Item 10	Item 11	Item 12	Item 13	Item 14	Item 15	Item 16	Item 17
0018:	Item 18	Item 19	Item 1A	Item 1B	Item 1C	Item 1D	Item 1E	Item 1F
0020:	Item 20	Item 21						

The content of data part is as follows:
It is the information of network parameter setting, etc. that is downloaded from the loader and set in this module.
Though the setting on the loader window is made by decimal number, the initializing information is stored by hexadecimal value.

Note: Item No. 16 and following are displayed only when the version of loader and this module is as follows:

- Firmware version of this module: V36 or later
- D300win: V3.1.2.0 or later version, or
  Standard loader: V2.0.0.0 or later version

Item No.	Description
0	Ethernet address for local station (L)
1	Ethernet address for local station (M)
2	Ethernet address for local station (H)
3	IP address for local station (L)
4	IP address for local station (H)
5	Sub-net mask (L)
6	Sub-net mask (H)
7	TCP send time-out value
8	Maximum TCP end timer value
9	Response monitoring timer value
A	Server FTP command monitoring timer value
B	Client FTP connection monitoring timer value
C	Fixed to 0001
D	Reference number for local port
E	Reference number for remote port of communication
F	Fixed to 0000

Item No.	Description
10	IP address for default gateway (L)
11	IP address for default gateway (H)
12	Gateway 1 - Network address (L)
13	Gateway 1 - Network address (H)
14	Gateway 1 - IP address (L)
15	Gateway 1 - IP address (H)
16	Gateway 2 - Network address (L)
17	Gateway 2 - Network address (H)
18	Gateway 2 - IP address (L)
19	Gateway 2 - IP address (H)
1A	Gateway 3 - Network address (L)
1B	Gateway 3 - Network address (H)
1C	Gateway 3 - IP address (L)
1D	Gateway 3 - IP address (H)
1E	Gateway 4 - Network address (L)
1F	Gateway 4 - Network address (H)
20	Gateway 4 - IP address (L)
21	Gateway 4 - IP address (H)

An example of how to read the initializing information is shown below.

```
  Ethernet address       IP address

0000: 104A 1A12 0040  0007 C000  0000 0000 0000
0008: 0000 0000 0000  0000 0000  0000 0000 0000

                       Unused
```

104A→Ethernet address (L)
1A12→Ethernet address (M)
0040→Ethernet address (H)

⇨ Ethernet address is 00041A12104A

0007→IP address (L)
C000→IP address (H)

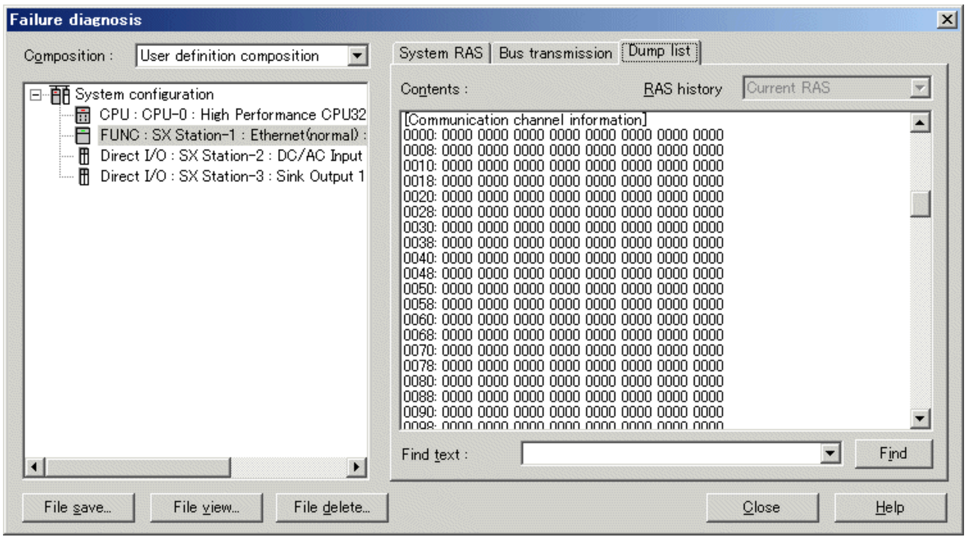
⇨

C0	00	00	07
192	0	0	7

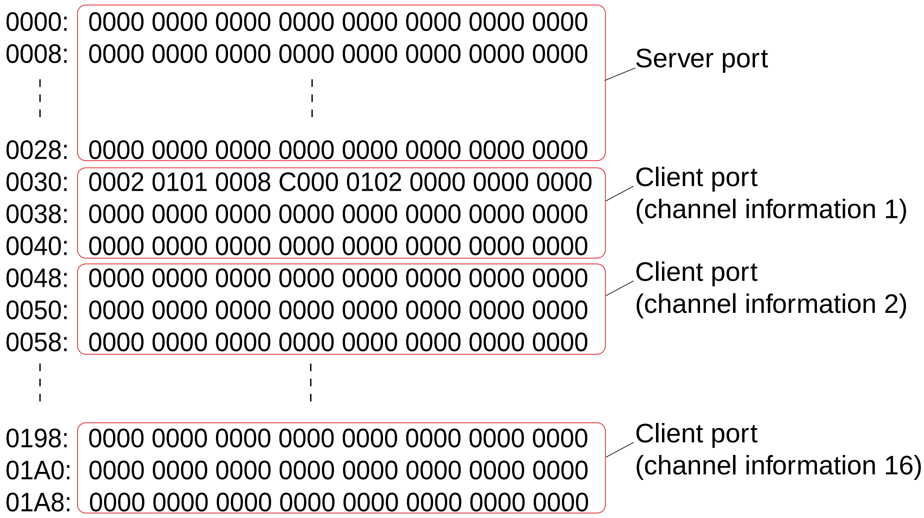
⇨ IP address is 192.0.0.7

(6) Communication channel information

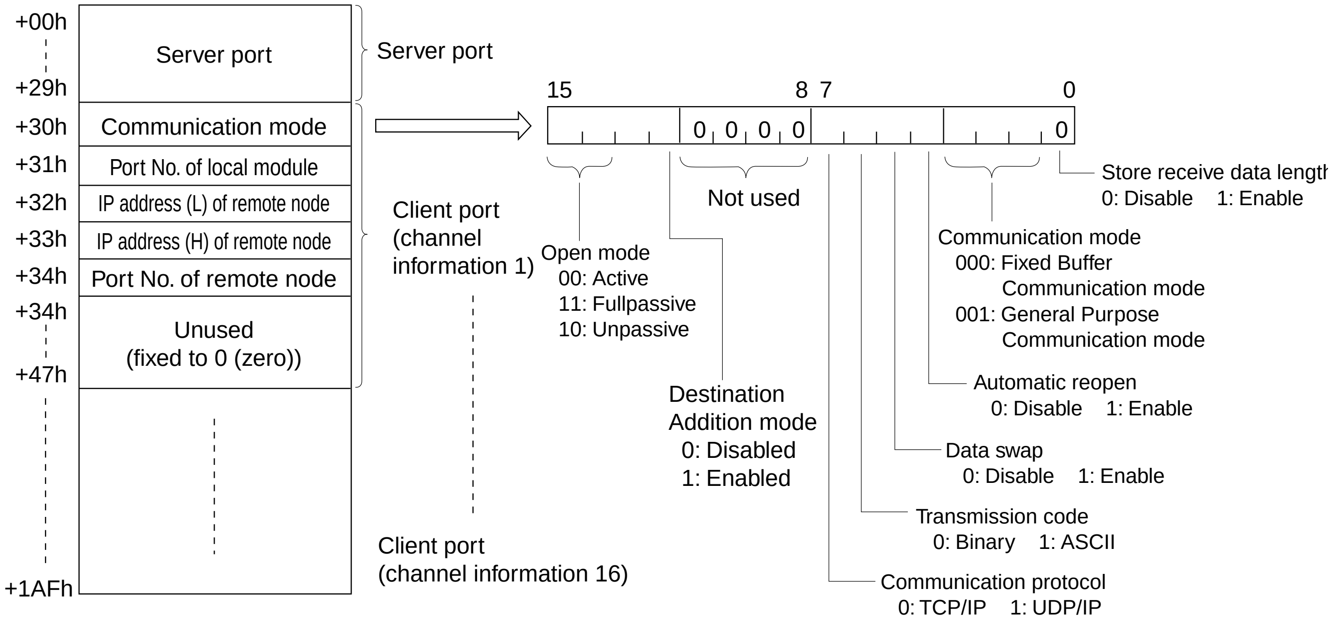
For this type of RAS group, the information of individual communication channel specified by client port number of this module when it is opened is displayed (the information for maximum 16 channels can be displayed). Communication mode, the port number of local module and the port number of remote node are displayed.


Failure diagnosis
Composition :
User definition composition
System configuration
CPU : CPU-0 : High Performance CPU32
FUNC : SX Station-1 : Ethernet(normal) :
Direct I/O : SX Station-2 : DC/AC Input
Direct I/O : SX Station-3 : Sink Output 1
System RAS
Bus transmission
Dump list
Contents :
RAS history
Current RAS
Find text :
Find
File save...
File view...
File delete...
Close
Help

```
[Communication channel information]
0000: 0000 0000 0000 0000 0000 0000 0000 0000
0008: 0000 0000 0000 0000 0000 0000 0000 0000
0010: 0000 0000 0000 0000 0000 0000 0000 0000
0018: 0000 0000 0000 0000 0000 0000 0000 0000
0020: 0000 0000 0000 0000 0000 0000 0000 0000
0028: 0000 0000 0000 0000 0000 0000 0000 0000
0030: 0000 0000 0000 0000 0000 0000 0000 0000
0038: 0000 0000 0000 0000 0000 0000 0000 0000
0040: 0000 0000 0000 0000 0000 0000 0000 0000
0048: 0000 0000 0000 0000 0000 0000 0000 0000
0050: 0000 0000 0000 0000 0000 0000 0000 0000
0058: 0000 0000 0000 0000 0000 0000 0000 0000
0060: 0000 0000 0000 0000 0000 0000 0000 0000
0068: 0000 0000 0000 0000 0000 0000 0000 0000
0070: 0000 0000 0000 0000 0000 0000 0000 0000
0078: 0000 0000 0000 0000 0000 0000 0000 0000
0080: 0000 0000 0000 0000 0000 0000 0000 0000
0088: 0000 0000 0000 0000 0000 0000 0000 0000
0090: 0000 0000 0000 0000 0000 0000 0000 0000
0098: 0000 0000 0000 0000 0000 0000 0000 0000
```

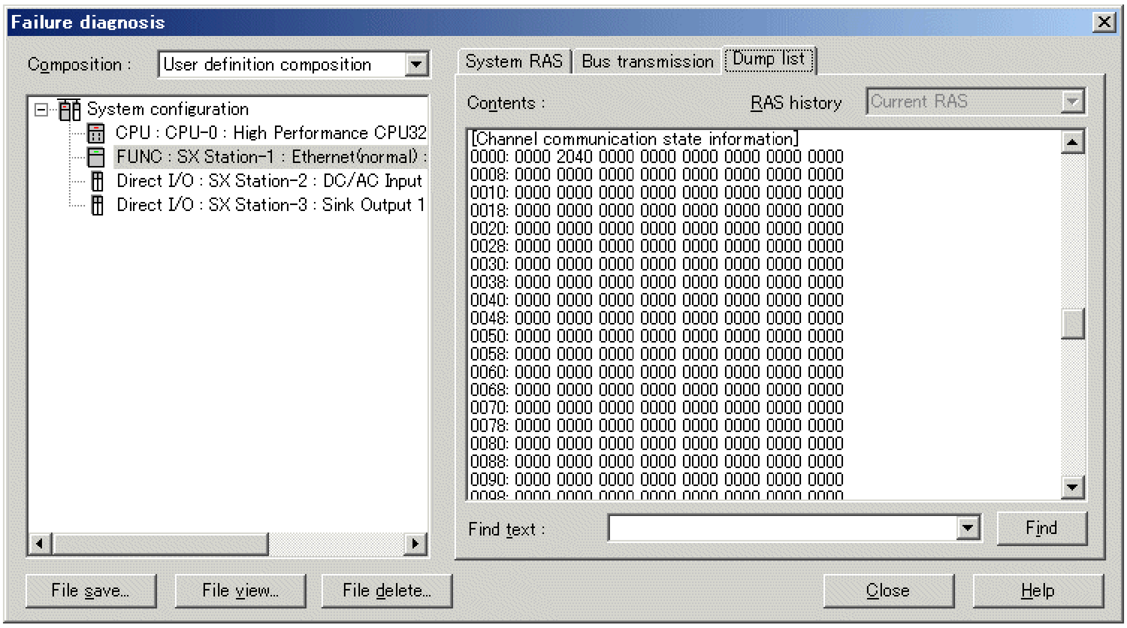


```
0000: 0000 0000 0000 0000 0000 0000 0000 0000
0008: 0000 0000 0000 0000 0000 0000 0000 0000

0028: 0000 0000 0000 0000 0000 0000 0000 0000
0030: 0002 0101 0008 C000 0102 0000 0000 0000
0038: 0000 0000 0000 0000 0000 0000 0000 0000
0040: 0000 0000 0000 0000 0000 0000 0000 0000
0048: 0000 0000 0000 0000 0000 0000 0000 0000
0050: 0000 0000 0000 0000 0000 0000 0000 0000
0058: 0000 0000 0000 0000 0000 0000 0000 0000

0198: 0000 0000 0000 0000 0000 0000 0000 0000
01A0: 0000 0000 0000 0000 0000 0000 0000 0000
01A8: 0000 0000 0000 0000 0000 0000 0000 0000
```



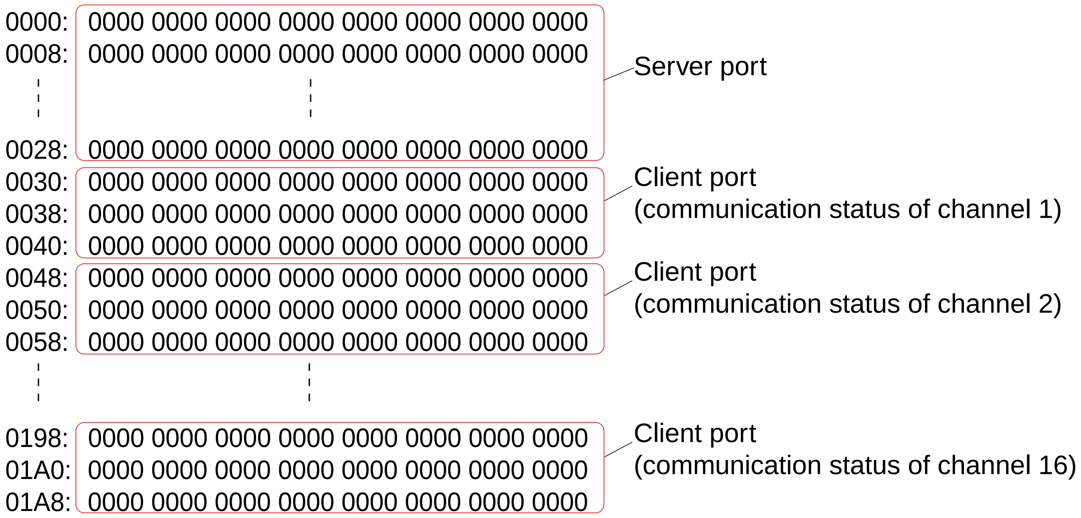
(7) Channel communication status information

For this type of RAS group, the information of individual communication channel specified by client port number of this module when it is opened is displayed (the information for maximum 18 channels can be displayed). 48 words from the top indicate the communication status of server port.


Failure diagnosis
Composition :
User definition composition
System configuration
CPU : CPU-0 : High Performance CPU32
FUNC : SX Station-1 : Ethernet(normal) :
Direct I/O : SX Station-2 : DC/AC Input
Direct I/O : SX Station-3 : Sink Output 1
System RAS
Bus transmission
Dump list
Contents :
RAS history
Current RAS
Find text :
Find
File save...
File view...
File delete...
Close
Help

```
[Channel communication state information]
0000: 0000 2040 0000 0000 0000 0000 0000 0000
0008: 0000 0000 0000 0000 0000 0000 0000 0000
0010: 0000 0000 0000 0000 0000 0000 0000 0000
0018: 0000 0000 0000 0000 0000 0000 0000 0000
0020: 0000 0000 0000 0000 0000 0000 0000 0000
0028: 0000 0000 0000 0000 0000 0000 0000 0000
0030: 0000 0000 0000 0000 0000 0000 0000 0000
0038: 0000 0000 0000 0000 0000 0000 0000 0000
0040: 0000 0000 0000 0000 0000 0000 0000 0000
0048: 0000 0000 0000 0000 0000 0000 0000 0000
0050: 0000 0000 0000 0000 0000 0000 0000 0000
0058: 0000 0000 0000 0000 0000 0000 0000 0000
0060: 0000 0000 0000 0000 0000 0000 0000 0000
0068: 0000 0000 0000 0000 0000 0000 0000 0000
0070: 0000 0000 0000 0000 0000 0000 0000 0000
0078: 0000 0000 0000 0000 0000 0000 0000 0000
0080: 0000 0000 0000 0000 0000 0000 0000 0000
0088: 0000 0000 0000 0000 0000 0000 0000 0000
0090: 0000 0000 0000 0000 0000 0000 0000 0000
0098: 0000 0000 0000 0000 0000 0000 0000 0000
```

The communication status of one channel is displayed using 24 words, as shown below.



```
0000: 0000 0000 0000 0000 0000 0000 0000 0000
0008: 0000 0000 0000 0000 0000 0000 0000 0000
  :                  :
0028: 0000 0000 0000 0000 0000 0000 0000 0000
0030: 0000 0000 0000 0000 0000 0000 0000 0000
0038: 0000 0000 0000 0000 0000 0000 0000 0000
0040: 0000 0000 0000 0000 0000 0000 0000 0000
0048: 0000 0000 0000 0000 0000 0000 0000 0000
0050: 0000 0000 0000 0000 0000 0000 0000 0000
0058: 0000 0000 0000 0000 0000 0000 0000 0000
  :                  :
0198: 0000 0000 0000 0000 0000 0000 0000 0000
01A0: 0000 0000 0000 0000 0000 0000 0000 0000
01A8: 0000 0000 0000 0000 0000 0000 0000 0000
```

Server port
Client port
(communication status of channel 1)
Client port
(communication status of channel 2)
Client port
(communication status of channel 16)

+00h	Open status code
	Open error code
	Send status code
	Send error code
	Fixed Buffer Communication mode response code
	Number of send data
	Receive status code
+07h	Receive error code

+08h	Status of fixed buffer overwriting
	Number of receive data
	Sending operation time (max.)
	Sending operation time (min.)
	Sending operation time (newest)
	Receiving operation time (max.)
	Receiving operation time (min.)
+0Fh	Receiving operation time (newest)

+10h	Send count
	Send error count
	Receive count
	Receive error count
	Unused
	Unused
	Unused
+17h	Unused

1) Open status code
The open status of a channel is stored in this area. 0: Not open, 1: Opening, 2: Already open, 3: Closing, 8xh: Forcible close

2) Open error code
Binary value for the code of an error that occurred while opening a channel is stored in this area. For more information of open error code, refer to "(9) Error code list".
* The error code is cleared when the next opening operation is completed successfully or communication is started.

3) Send status code
The status of data sending operation is stored in this area. 0: No request to send or sending completed, 1: Sending

4) Send error code
Binary value for the code of an error that occurred while sending data is stored in this area. For more information of send error code, refer to "(9) Error code list".
* The error code is cleared when the next opening operation is completed successfully, communication is started, or sending is completed successfully.

5) Fixed Buffer Communication mode response code
Binary value for the end code of the response that is returned when sending data in Fixed Buffer Communication mode is stored in this area. For more information of the end code of response, refer to "4-3-2 Fixed Buffer Communication mode".
* The error code is cleared when the next opening operation is completed successfully or communication is started.

6) Number of send data
Binary value for the number of send data that is specified when data sending is requested is stored in this area. 0 (zero) is stored when there is no request to send data.
* The error code is cleared when the next opening operation is completed successfully or communication is started.

7) Receive status code
The status of data receiving operation is stored in this area. 0: No request to receive or receiving completed, 1: Receiving, 2: Receive data exists (when there is no request to receive)

8) Receive error code
Binary value for the code of an error that occurred while receiving data is stored in this area. For more information of receive error code, refer to "(9) Error code list".
To clear the error code, execute one of the following operations:
* The error code is cleared when the next opening operation is completed successfully, communication is started, or receiving is completed successfully.

9) Status of fixed buffer overwriting
It is stored in this area whether or not data was received again (whether or not overwriting occurred) in the period after data had been received from other node till the response for it was sent in Fixed Buffer Communication mode.
0: Fixed buffer overwriting not occurred, 1: Fixed buffer overwriting occurred
* Fixed buffer overwriting status is cleared when the next opening operation is completed successfully or communication is started.

10) Number of receive data
Binary value for the number of bytes of receive data that is transferred to the receive data storage area during receipt confirmation after data receiving has been requested is stored in this area.

11) Sending operation time

Maximum, minimum and current values of the operation time for sending in each communication performed are stored in these areas.

(a) Operation time is stored by binary value in steps of 55 ms.
(b) The time required till local module finishes sending after sending is requested.

* This data is cleared when the next opening operation is completed or communication is started.

12) Receiving operation time

Maximum, minimum and current values of the operation time for receiving in each communication performed are stored in these areas.

(a) Operation time is stored by binary value in steps of 55 ms.
(b) For the operation time for receiving, the following values are stored:

<General Purpose Communication mode>

The time till receipt confirmation is called by a user program after local module finished receiving

<Fixed Buffer Communication mode>

The time till receipt confirmation is called by a user program and local module finishes the operation to return a response after local module finished receiving

<Client FTP/TFTP communication>

The time till local module finishes receiving after receiving is requested

* This data is cleared when the next opening operation is completed or communication is started.

13) Send count

The total number of successfully completed operations for sending data to other node is stored in this area.

(a) Counted when the communication of command and response is completed successfully.
(b) Counted when one of the following communications is completed successfully.

- Sending data to other node in Fixed Buffer Communication mode
- Reading the data of sheared buffer from other node in Shared Buffer Communication mode
- Sending data to other node in General Purpose Communication mode
- File transfer to other node during server FTP/TFTP communication
- File transfer to other node during client FTP/TFTP communication

* This data is cleared when the next opening operation is completed or communication is started.

14) Send error count

The total number of communication errors that occurred in sending data to other node (when the communication of command and response ended abnormally) is stored in this area.

* This data is cleared when the next opening operation is completed or communication is started.

15) Receive count

The total number of successfully completed operations for receiving data from other node is stored in this area.

(a) Counted when the communication of command and response is completed successfully.
(b) Counted when one of the following communications is completed successfully.

- Receiving data from other node in Fixed Buffer Communication mode
- Writing the data from other node in the sheared buffer in Shared Buffer Communication mode
- Receiving data from other node in General Purpose Communication mode
- File transfer from other node during server FTP/TFTP communication
- File transfer from other node during client FTP/TFTP communication

* This data is cleared when the next opening operation is completed or communication is started.

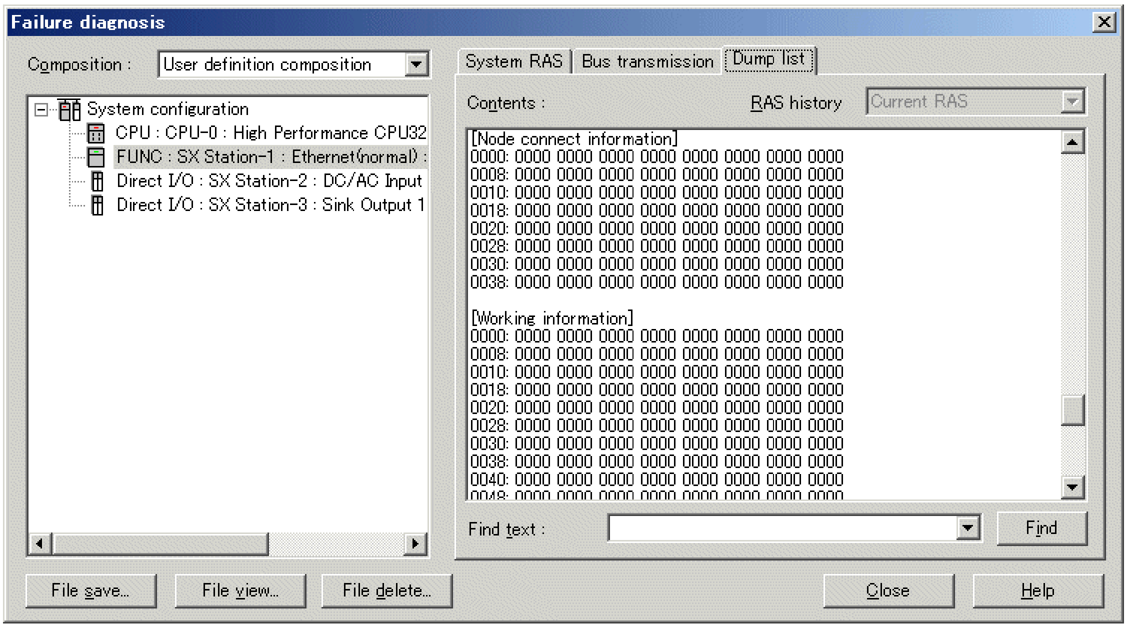
16) Receive error count

The total number of communication errors that occurred in receiving data from other node (when the communication of command and response ended abnormally) is stored in this area.

* This data is cleared when the next opening operation is completed or communication is started.

(8) Node connection information

For this type of RAS group, the intensive information of the status of individual communication channel that is specified by client port number of this module is displayed in the [Content] text box.


Failure diagnosis
Composition : User definition composition
System configuration
CPU : CPU-0 : High Performance CPU32
FUNC : SX Station-1 : Ethernet(normal) :
Direct I/O : SX Station-2 : DC/AC Input
Direct I/O : SX Station-3 : Sink Output 1
System RAS
Bus transmission
Dump list
Contents :
RAS history
Current RAS
Find text :
Find
File save...
File view...
File delete...
Close
Help

```
[Node connect information]
0000: 0000 0000 0000 0000 0000 0000 0000 0000
0008: 0000 0000 0000 0000 0000 0000 0000 0000
0010: 0000 0000 0000 0000 0000 0000 0000 0000
0018: 0000 0000 0000 0000 0000 0000 0000 0000
0020: 0000 0000 0000 0000 0000 0000 0000 0000
0028: 0000 0000 0000 0000 0000 0000 0000 0000
0030: 0000 0000 0000 0000 0000 0000 0000 0000
0038: 0000 0000 0000 0000 0000 0000 0000 0000

[Working information]
0000: 0000 0000 0000 0000 0000 0000 0000 0000
0008: 0000 0000 0000 0000 0000 0000 0000 0000
0010: 0000 0000 0000 0000 0000 0000 0000 0000
0018: 0000 0000 0000 0000 0000 0000 0000 0000
0020: 0000 0000 0000 0000 0000 0000 0000 0000
0028: 0000 0000 0000 0000 0000 0000 0000 0000
0030: 0000 0000 0000 0000 0000 0000 0000 0000
0038: 0000 0000 0000 0000 0000 0000 0000 0000
0040: 0000 0000 0000 0000 0000 0000 0000 0000
0048: 0000 0000 0000 0000 0000 0000 0000 0000
```

```
0000: 0001 0001 0000 0000 0000 0000 0000 0000
0008: 0000 0000 0000 0000 0000 0000 0000 0000   Status of connection
0010: 0000 0000 0000 0000 0000 0000 0000 0000
0018: 0000 0000 0000 0000 0000 0000 0000 0000
0020: 0000 0000 0000 0000 0000 0000 0000 0000
0028: 0000 0000 0000 0000 0000 0000 0000 0000   Error code
0030: 0000 0000 0000 0000 0000 0000 0000 0000
0038: 0000 0000 0000 0000 0000 0000 0000 0000
```

+00h	Status of connection (server port 1)
	Status of connection (server port 2)
	Status of connection (channel 1)
	Status of connection (channel 2)
	⋮
+11h	Status of connection (channel 16)
+12h ⋮ +1Fh	Unused
+20h	Error code (server port 1)
	Error code (server port 2)
	Error code (channel 1)
	Error code (channel 2)
	⋮
+31h	Error code (channel 16)
+32h ⋮ +3Fh	Unused

1) Status of connection

The open status of a channel is stored in this area.

0: Not open
1: Opening
2: Already open
3: Closing
8x: Forcible close

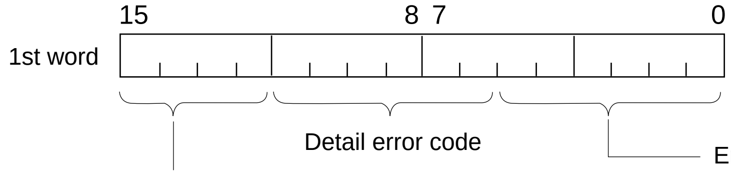
2) Error code

Binary value for the code of an error that occurred when opening a channel, sending data or receiving data is stored in this area.
For more information of the error codes, refer to "Error code list".

* The data is cleared when the next opening operation is completed successfully or communication is started.

(9) Error code

The error codes that are stored as "error logging information", "channel communication status information" or "node connection information" as well as their contents are shown below:



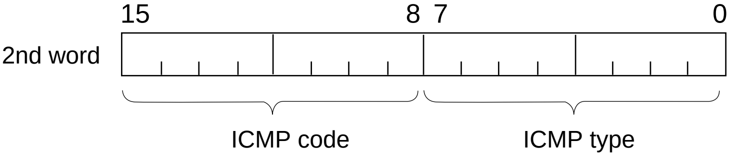
Error type

0001: Parameter error
0010: Sequence error
0011: Communication error in upper layer
0100: Communication error between physica
layer and transport layer

Error detected channel No.

00h: Cannot pinpoint channel No.
01h: Server port 1
02h: Server port 2
03h: Channel No. 1
⋮
12h: Channel No. 16

Detail error code (only for error logging information)



<Detail error codes for individual error type>

1) Parameter error

Code	Description	Timing of occurrence
01h	Local module does not yet complete the operation for starting communication.	During initialization
03h	The number of bits of subnet mask is smaller than the number of bits of netmask that is obtained from the IP address of this module.	During initialization
04h	IP address of default gateway or the specification of gateway has one of the following errors: • "255.255.255.255" is specified for the IP address of default gateway. • "255.255.255.255" is specified for the network address of gateway. • "0.0.0.0" or "255.255.255.255" is specified for the IP address of gateway.	During initialization
10h	The IP address of local module, "0.0.0.0" or "255.255.255.255" is specified for the IP address of other node.	When opening
13h	The same combination of "port No. of local module" and "port No. of other node" as that for already opened connection is set.	When opening
15h	The IP address of other network the communication with which is impossible is specified for the IP address of remote node.	When opening
16h	Wrong communication mode is specified.	When opening
20h	Send data length is out of specified range (1017 words).	When opening
22h	The storage area for receive data is smaller than receive data length.	When opening

2) Sequence error

Code	Description	Timing of occurrence
01h	Local module does not yet complete the operation for starting communication.	When opening, sending or receiving
02h	The operation to open the channel is not yet completed.	When sending or receiving
03h	Data was sent or received while the channel is being closed.	When sending or receiving

3) Communication error in upper layer

Code	Description	Timing of occurrence
01h	A code other than 00h (normal end) was returned from other node as the end code for response in Fixed Buffer Communication mode.	When sending or receiving
02h	No response was returned within the set response monitoring timer value.	When sending or receiving
03h	• The volume of data that is specified by data length could not be received within the set response monitoring timer value. • The volume of actual data is smaller than the value specified by data length. • The remaining part of the data divided at TCP/IP level could not be received within the set response monitoring timer value.	When sending or receiving

4) Communication error between physical layer and transport layer

Code	Description	Timing of occurrence
02h	Connection could not be established by the operation to open TCP connection.	When opening
03h	During communication using TCP protocol, checksum error was found in the receive data.	
04h	During communication using UDP protocol, checksum error was found in the receive data.	
05h	• During communication using TCP/IP protocol, TCP send timeout occurred. • During communication using TCP/IP protocol, no ACK was returned from other node.	When opening, sending or receiving
06h	• Invalid IP address (network No.) is used. (When the IP address of the IP packet sent to other node was different, ICMP error packet was received.) ICMP TYPE=3, CODE=0	
07h	Invalid IP address (host No.) is used. (When the IP address of the IP packet sent to other node was the same, ICMP error packet was received.) ICMP TYPE=3, CODE=1	
08h	Invalid port No. is used. (When the port number of the IP packet sent to a remote node was not registered to the remote node, ICMP error packet was received.) ICMP TYPE=3, CODE=2, 3	
09h	When IP packet was destroyed due to increased traffic, ICMP error packet was received.) ICMP TYPE=4	
0Ah	When assembly timeout occurred at remote node, ICMP error packet was received. ICMP TYPE=11	
0Bh	• An ICMP error packet that is not supported by the system was received. The type and code of received ICMP error packet are stored in the detail error information area.	
0Ch	• No Ethernet address exists that corresponds to the specified IP address. • Invalid IP address (network No., host No.) or invalid port No. is used.	When opening
0Eh	Checksum error of received IP packet	
0Fh	Cannot open the channel because the internal buffer for opening channels is full.	When opening
10h	Send error occurred.	When sending or receiving
11h	No send buffer could be secured.	When sending or receiving
12h	• During communication using TCP/IP protocol, TCP timeout occurred. • Cannot send data due to too small window size of the remote node.	When sending or receiving
13h	Other errors related to socket interface	When sending or receiving

(10) Running status information

Running status information consists of counter information and supplementary information. Counter information is the frequency of occurrences of an event, and supplementary information is set for each event. They are overwritten whenever updated to always provide the newest information.

This RAS information is supported by V3.1.0.0 or later or V2.2.4.0 or later version of D300win, and Ethernet module is supported by of V34 or later software version.

Running status information is classified into the following 5 groups:

The error codes that are stored as “error logging information”, “channel communication status information” or “node connection information” as well as their contents are shown below:

Offset address (hexadecimal)	Description
00000000 ~	Related to Ethernet communication SMC91C111 (LANC)
000000C0 ~	Related to Ethernet communication SMC91C111 (Physical)
00000100 ~	Related to USNET socket I/F
00000150 ~	Related to SBM communication
000001B0 ~	Related to APP communication

<Related to Ethernet communication SMC91C111 (LANC)>

Offset address (hexadecimal)	Description
00000000 00000004	Counter: SMC initialization error Supplementary information: None
00000008 0000000C	Counter: SMC send delay collision Supplementary information: None
00000010 00000014	Counter: SMC send carrier lost Supplementary information: None
00000018 0000001C	Counter: SMC send under run Supplementary information: None
00000020 00000024	Counter: SMC send 16 collision Supplementary information: None
00000028 0000002C	Counter: SMC send memory allocation incomplete (TASK) Supplementary information: None
00000030 00000034	Counter: SMC send memory allocation incomplete (INT) Supplementary information: None
00000038 0000003C	Counter: SMC send queuing Supplementary information: None
00000040 00000044	Counter: SMC send queue full Supplementary information: None
00000048 0000004C	Counter: SMC send buffer no request source Supplementary information: None
00000050 00000054	Counter: SMC send request (TASK) Supplementary information: None
00000058 0000005C	Counter: SMC send request (INT) Supplementary information: None
00000060 00000064	Counter: SMC send completion Supplementary information: None
00000068 0000006C	Counter: SMC receive alignment error Supplementary information: None
00000070 00000074	Counter: SMC receive CRC error Supplementary information: None
00000078 0000007C	Counter: SMC receive long packet Supplementary information: None

Offset address (hexadecimal)	Description
00000080 00000084	Counter: SMC receive short packet Supplementary information: None
00000088 0000008C	Counter: SMC receive data size over Supplementary information: None
00000090 00000094	Counter: SMC receive overrun Supplementary information: None
00000098 0000009C	Counter: SMC receive completion Supplementary information: None
000000A0 000000A4	Counter: SMC receive USNET queue full Supplementary information: None
000000A8 000000AC	Counter: SMC receive USNET buffer full Supplementary information: None
000000B0 000000B4	Counter: SMC memory reset Supplementary information: None
000000B8 000000BC	Counter: SMC restart Supplementary information: None

<Related to Ethernet communication SMC91C111 (Physical)>

Offset address (hexadecimal)	Description
000000C0 000000C4	Counter: Illegal 4B5B code detected in PHY receive data Supplementary information: None
000000C8 000000CC	Counter: No starting point of stream delimiter found in PHY receive data Supplementary information: None
000000D0 000000D4	Counter: No end point of stream delimiter found in PHY receive data Supplementary information: None
000000D8 000000DC	Counter: PHY Jabber detected Supplementary information: None
000000E0 000000E4	Counter: PHY changed over to full duplex Supplementary information: None
000000E8 000000EC	Counter: PHY changed over to half duplex Supplementary information: None
000000F0 000000F4	Counter: Reserved Supplementary information: None
000000F8 000000FC	Counter: Reserved Supplementary information:

<Related to USNET socket I/F>

Offset address (hexadecimal)	Description
00000100 00000104	Counter: USN socket error Supplementary information: Error code
00000108 0000010C	Counter: USN bind error Supplementary information: Error code
00000110 00000114	Counter: USN connect error Supplementary information: Error code
00000118 0000011C	Counter: USN listen error Supplementary information: Error code
00000120 00000124	Counter: USN accept error Supplementary information: Error code
00000128 0000012C	Counter: USN send error Supplementary information: Error code
00000130 00000134	Counter: USN sentto error Supplementary information: Error code
00000138 0000013C	Counter: USN recv error Supplementary information: Error code
00000140 00000144	Counter: USN recvfrom error Supplementary information: Error code
00000148 0000014C	Counter: USN closesocket error Supplementary information: Error code

<Related to SBM communication>

Offset address (hexadecimal)	Description
00000150 00000154	Counter: Abnormal end of sending NACK to SBM Supplementary information: None
00000158 0000015C	Counter: Abnormal end of other than sending NACK to SBM Supplementary information: None
00000160 00000164	Counter: Timeout of sending to SBM Supplementary information: Number of times of consecutively occurred PHY register data mismatch
00000168 0000016C	Counter: Retry over of sending to SBM Supplementary information: None
00000170 00000174	Counter: Buffer full for receiving from SBM (FULL) Supplementary information: None
00000178 0000017C	Counter: Buffer full for receiving from SBM (NEARFULL) Supplementary information: None
00000180 00000184	Counter: Send request to SBM Supplementary information: None
00000188 0000018C	Counter: Normal end of sending to SBM Supplementary information: Test data
00000190 00000194	Counter: Completion of receiving from SBM Supplementary information: None
00000198 0000019C	Counter: Reserved Supplementary information:
000001A0 000001A4	Counter: Reserved Supplementary information:
000001A8 000001AC	Counter: Reserved Supplementary information:

<Related to the processing of APP part>

Offset address (hexadecimal)	Description
000001B0 000001B4	Counter: Abnormal end of APP communication initialization Supplementary information: None
000001B8 000001BC	Counter: Abnormal end of APP channel open Supplementary information: None
000001C0 000001C4	Counter: Abnormal end of APP channel open port specification Supplementary information: None
000001C8 000001CC	Counter: APP channel open client port full Supplementary information: None
000001D0 000001D4	Counter: APP channel open type error (other than Ethernet) Supplementary information: None
000001D8 000001DC	Counter: APP channel open LDC server re-open Supplementary information: None
000001E0 000001E4	Counter: Abnormal end of APP channel close Supplementary information: None
000001E8 000001EC	Counter: APP channel close unopened port Supplementary information: None
000001F0 000001F4	Counter: APP channel close SA, SP mismatch Supplementary information: None
000001F8 000001FC	Counter: APP channel close specified data count over Supplementary information: None
00000200 00000204	Counter: APP connection server undefined command Supplementary information: None
00000208 0000020C	Counter: APP connection server split parameter error Supplementary information: None
00000210 00000214	Counter: APP connection server communication command header error Supplementary information: None
00000218 0000021C	Counter: APP connection server forcible close Supplementary information: None
00000220 00000224	Counter: Abnormal end of APP message send Supplementary information: None
00000228 0000022C	Counter: Abnormal end of APP message receive Supplementary information: None
00000230 00000234	Counter: APP auto communication Error other than channel close error Supplementary information: None
00000238 0000023C	Counter: APP auto communication Channel close error Supplementary information: None
00000240 ~ 00000274	Counter: Reserved Supplementary information:
00000278 0000027C	System time (H) System time (L)

[Running status information occurrence conditions]

<Related to Ethernet communication SMC91C111 (LANC)>

Offset address (hexadecimal)	Cause, conditions and timing of occurrence
00000000	SMC initialization error During the initialization of LAN controller
00000008	SMC send delay collision Collision that occurred out of collision detection time during half duplex communication at 10 Mbps (illegal operation)
00000010	SMC send carrier lost Occurs due to a disconnected cable or an unstable line during send operation.
00000018	SMC send under run Occurs when send FIFO of LANC is full.
00000020	SMC 16 collision Occurs when collision occurs 16 times during half duplex communication at 10 Mbps. (Line load)
00000028	SMC send memory allocation incomplete (TASK) LANC send buffer cannot be acquired. (Occurs under slightly heavy load.)
00000030	SMC send memory allocation incomplete (INT) LANC send buffer cannot be acquired. (Occurs under slightly high load.)
00000038	SMC send queuing Operation counter when allocation is incomplete
00000040	SMC send queue full When send queues (64 pieces) are all full (LANC cannot send or has failed.)
00000048	SMC send buffer no request source Occurs according to the comparison with USNET definition.
00000050	SMC send request (TASK) LANC send operation times (no send delay)
00000058	SMC send request (INT) Number of times to send the queued send buffer data
00000060	SMV send completion Number of times of send completion interruption
00000068	SMC receive alignment error Fraction bit received (LANC)
00000070	SMC receive CRC error Receive frame check error (LANC)
00000078	SMC receive long packet 1514-byte or longer packet was received (except FCS).
00000080	SMC receive short packet 60-byte or shorter packet was received (except FCS).
00000088	SMC receive data size over Long packer check error on LANC driver
00000090	SMC receive overrun Occurs when there is no LANC receive buffer remained. (Line load)
00000098	SMC receive completion Number of receive completion times
000000A0	SMC receive USNET queue full USNET receive queues (64 pieces) are full. (LANC received data exceeding the performance of this module.)
000000A8	SMC receive USNET buffer full USNET receive buffers (200 pieces) are full.
000000B0	SMC memory reset Occurs when send queues are full. Clear LANC receive buffer.
000000B8	SMC restart Occurs if time out occurs while waiting for the completion of memory reset operation. (Reset the chip.)

<Related to Ethernet communication SMC91C111 (Physical)>

Offset address (hexadecimal)	Description
000000C0	Illegal 4B5B code detected in PHY receive data Receive error on physical layer
000000C8	No starting point of stream delimiter found in PHY receive data Receive error on physical layer
000000D0	No end point of stream delimiter found in PHY receive data Receive error on physical layer
000000D8	PHY Jabber detected Receive error on physical layer
000000E0	PHY changed over to full duplex Number of times to change over from half duplex to full duplex
000000E8	PHY changed over to half duplex Number of times to change over from half duplex to full duplex
000000F0	Reserved
000000F8	Reserved

<Related to USNET socket I/F>

Offset address (hexadecimal)	Description
00000100	USN socket error Failed to acquire a socket, etc.
00000108	USN bind error Failed to set local port number, etc.
00000110	USN connect error Error during active open. This error also occurs due to timeout.
00000118	USN listen error Error during passive open
00000120	USN accept error Error during passive open. This error also occurs due to timeout.
00000128	Counter: USN send error TCP/IP send error. Connection not established. This error also occurs due to timeout, etc.
00000130	Counter: USN sentto error UDP/IP send error
00000138	Counter: USN recv error TCP/IP receive error. Connection not established. This error also occurs due to timeout, etc.
00000140	Counter: USN recvfrom error UDP/IP receive error
00000148	Counter: USN closesocket error Error while returning a socket

<Related to SBM communication>

Offset address (hexadecimal)	Description
00000150	Abnormal end of sending NACK to SBM
00000158	Abnormal end of other than sending NACK to SBM
00000160	Timeout of sending to SBM
00000168	Retry over of sending to SBM After retried 3 times.
00000170	Buffer full for receiving from SBM (FULL) No receive buffer remained
00000178	Buffer full for receiving from SBM (NEARFULL) Only one receive buffer is remained.
00000180	Send request to SBM
00000188	Normal end of sending to SBM
00000190	Completion of receiving from SBM
00000198	Reserved
000001A0	Reserved
000001A8	Reserved

<Related to the processing of APP part>

Offset address (hexadecimal)	Description
000001B0	Abnormal end of APP communication initialization Number of times to notify FB error response
000001B8	Abnormal end of APP channel open Number of times to notify FB error response (including the detection of duplicated open)
000001C0	Abnormal end of APP channel open port specification Number of times to notify FB error response
000001C8	APP channel open client port full Number of times to notify FB error response
000001D0	APP channel open type error (other than Ethernet) Number of times to notify FB error response
000001D8	APP channel open LDC server re-open Number of times to notify FB error response
000001E0	Abnormal end of APP channel close Number of times to notify FB error response
000001E8	APP channel close unopened port Number of times to notify FB error response
000001F0	APP channel close SA, SP mismatch Number of times to notify FB error response
000001F8	APP channel close specified data count over Number of times to notify FB error response
00000200	APP connection server undefined command Number of times to notify FB error response
00000208	APP connection server split parameter error Number of times to notify FB error response
00000210	APP connection server communication command header error Number of times to notify FB error response

Offset address (hexadecimal)	Description
00000218	APP connection server forcible close Number of times to notify FB error response
00000220	Abnormal end of APP message send Number of times to notify FB error response
00000228	Abnormal end of APP message receive Number of times to notify FB error response
00000230	APP auto communication Error other than channel close error Number of times to notify FB error response
00000238	APP auto communication Channel close error Number of times to notify FB error response
00000240 to 00000274	Reserved
00000278 0000027C	System time (H) System time (L)

Appendix 1 Loader Commands

Page

Appendix 1 Loader Commands

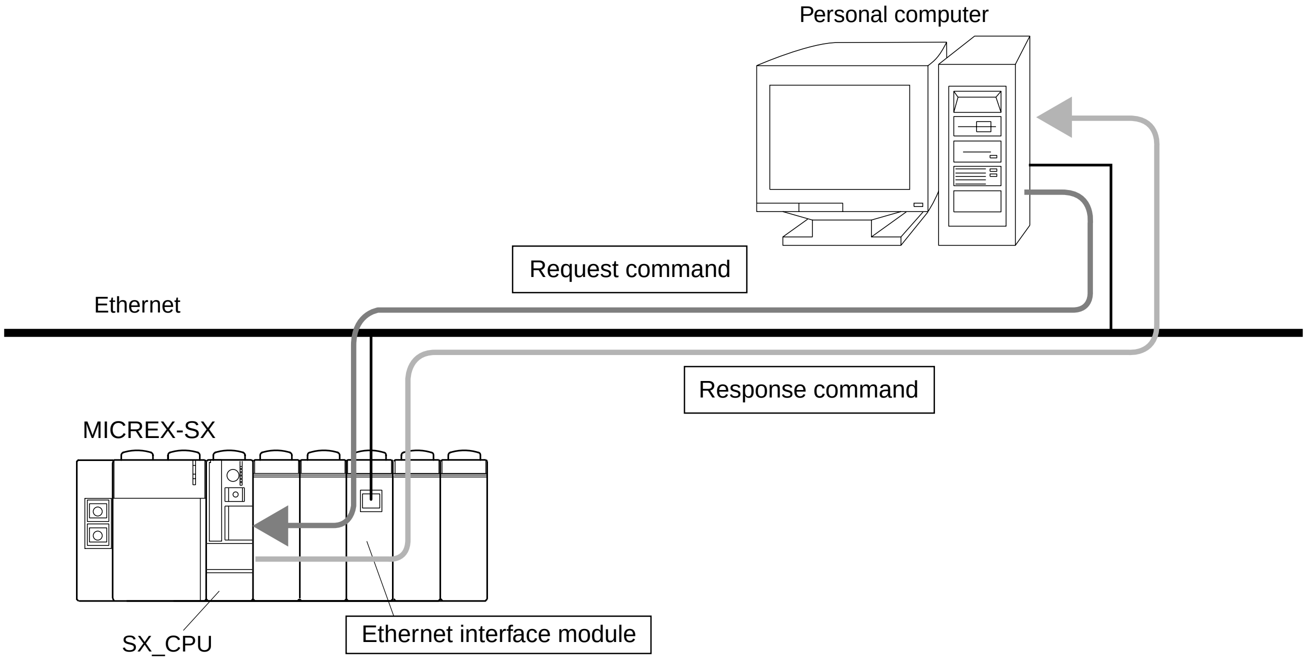
Appendix 1-1 Overview of Loader Commands

When accessing the data of the internal memory of MICREX-SX series CPU module from an intelligent device such as personal computer via NP1L-ET1/ET2, it is possible to read or write data only by issuing a request command from the personal computer, without installing any application program for data sending/receiving in the CPU module. For this purpose, the following 10 commands are prepared.

Classification	Command name	Command	Mode	Description
Read	Read data	00h	00h	Read data from specified data memory.
Write	Write data	01h	00h	Write data in specified data memory.
PLC control	CPU batch start	04h	00h	Start all the CPUs existing in a configuration as a batch.
	CPU batch initialize and start		01h	Initialize and start all the CPUs existing in a configuration as a batch.
	CPU batch stop		02h	Stop all the CPUs existing in a configuration as a batch.
	CPU batch reset		03h	Reset all the CPUs existing in a configuration as a batch.
	CPU individual start		04h	Start CPUs individually by specifying them.
	CPU individual initialize and start		05h	Initialize and start CPUs individually by specifying them
	CPU individual stop		06h	Stop CPUs individually by specifying them.
	CPU individual reset		07h	Reset CPUs individually by specifying them.

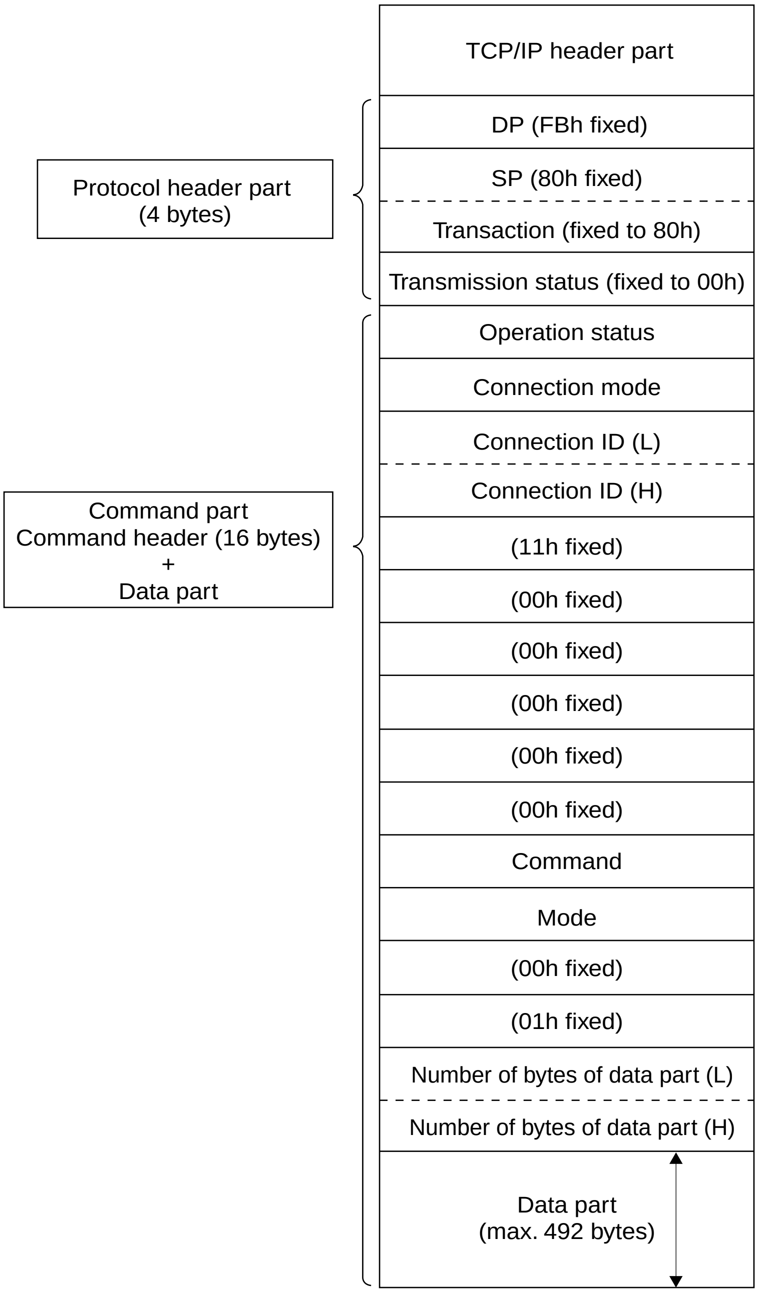
Note: When "CPU batch initialize and start" or "CPU individual initialize and start" is executed, retain memory is cleared before starting the CPUs.

<Image of data access>
A request command is created by an application program installed in the personal computer, and the command is issued to the CPU module via the Ethernet interface module. On receiving the request command, the CPU returns a response command.
No data communication program is necessary in the application program installed in the CPU module.



Appendix 1-2 Transmission Data Format of Commands

The transmission data consists of “TCP/IP header part”, “Protocol header part”, “Command part” (request command and response command), and "check part". The format is the same for both request command and response command.



TCP/IP header part

Protocol header part
(4 bytes)

DP (FBh fixed)

SP (80h fixed)

Transaction (fixed to 80h)

Transmission status (fixed to 00h)

Command part
Command header (16 bytes)
+
Data part

Operation status

Connection mode

Connection ID (L)

Connection ID (H)

(11h fixed)

(00h fixed)

(00h fixed)

(00h fixed)

(00h fixed)

(00h fixed)

Command

Mode

(00h fixed)

(01h fixed)

Number of bytes of data part (L)

Number of bytes of data part (H)

Data part
(max. 492 bytes)

<Detail explanation of transmission data>

1) TCP/IP header part

In the case of request command, this part is added at top of the transmission data part of loader command on the personal computer side before the command is sent.
In the case of response command, this part is added at top of the transmission data part by NP1L-ET1 before response data is returned.

2) DP (Destination Port)

Destination port number on the destination module (not TCP/IP port number). This is specified by the requestor.
  * In general 251 (FBh) is specified, which is the loader interface server port on CPU0.
  * When accessing CPU0, 253 (FDh) may be specified, which is the loader command server port.
  * DP and SP are changed for response data.

3) SP (Source Port)

Requestor port number on the source module. This number is specified by the requestor.

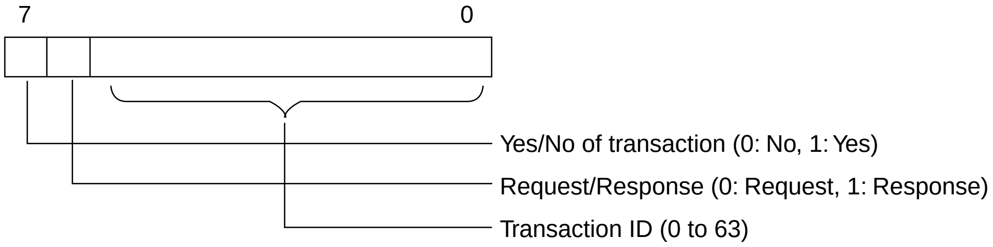
- * Fixed to 128 (80h).
- * DP and SP are changed for response data.

4) Transaction

Specifies Yes/No of transaction, Request/Response, and Transaction ID.

Yes/No of transaction is set by the requestor. Transaction ID is dynamically assigned when the request to send message is issued. Response bit is set by the destination server when response is returned.

- * Fixed to 80h when transaction needs not be controlled according to the request from this module.


7
0
Yes/No of transaction (0: No, 1: Yes)
Request/Response (0: Request, 1: Response)
Transaction ID (0 to 63)

5) Transmission status

Indicates the result of transmission via SX bus or network

- * Fixed to 00h when requesting
- * Means abnormal end if a value other than 00h is set for response command. Meaning of individual code is the same as “operation status”.

6) Operation status

Indicates the result of individual operation requested by command. It is checked by response command. Request command is issued when this item is set to FFh (fixed).

<Status list>

Code	Status	Description
00h	Ended normally	The processing of command is completed successfully.
10h	CPU error	Command cannot be executed because an abnormality occurred on the CPU.
11h	CPU running	Command cannot be executed because the CPU is running.
12h	Command unexecutable	Command cannot be executed due to the key switch condition of the CPU.
20h	Undefined command	CPU received undefined command or mode.
22h	Parameter error	Setting error was found in command header part.
23h	Transmission interlocked	Transmission is interlocked by a command from other device.
28h	Processing a command	Requested command cannot be executed because other command is now being executed.
2Bh	Remote loader now processing	Requested command cannot be executed because D300win loader is now processing.
2Fh	Initialization not competed	Requested command cannot be executed because the system is now being initialized.
40h	Data setting error	Invalid data type or number was specified.
41h	Inexistent data	Specified data cannot be found.
44h	Memory address setting error	Specified address exceeds the valid range.
45h	Memory size over	Address + the number of read/write words exceed the valid range.
A0h	Command send destination setting error	No module exists at specified destination station number.
A2h	No response to command	No response data is returned from the remote module.
A4h	SX bus send error	Command cannot be communicated because an abnormality occurred on the SX bus.
A5h	SX bus send NAK	Command cannot be communicated because NAK occurred while sending data via SX bus.
FFh	Specification at sending request command	"Operation status" must be set to FF when issuing a request command.

7) Specification of target to connect, connection ID

This item specifies a module as the target of data transmission (the destination for sending a command).

Target module of transmission	Connection mode	Connection ID (L)	Connection ID (H)
CPU0	7Ah	00h	00h
CPU1 to CPU7	7Bh	SX bus station No. of remote CPU	00h
P/PE-link, FL-net	7Bh	SX bus station No. of remote module	00h

8) Command, mode

Command: Major classification code of command
Mode: Detail code of command
For more information of commands and modes, refer to “Appendix 1-3 Detail Description of Loader Commands”.

9) No. of bytes of data part

Indicates the number of bytes of command data part. Neither request command for writing data nor response command for reading data may exceed 492 bytes.

Appendix 1-3 Detail Description of Loader Commands

Appendix 1-3-1 Read data

This command is used to read the data for specified number of words from specified address of specified data memory. If the requested number of words to read exceeds the range of specified memory, the data is read out up to the last address of specified memory.

1) Command/mode code

Command: 00h, Mode: 00h

2) Format of the data part of request command and response command

Request data

Memory type
Memory address (L)
Memory address (M)
Memory address (H)
Number of read words (L)
Number of read words (H)

Response data

Memory type
Memory address (L)
Memory address (M)
Memory address (H)
Number of read words (L)
Number of read words (H)
Read data • • •

3) Layout of read data

Read data is laid out as follows:

MICREX-SX module

15	0
12	34
56	78

↔

Personal computer

34
12
78
56

4) Example of data read command

[When the D300win is used]

Following shows the the command part for reading 2 words of data from standard memory %MW1.0 of CPU module (CPU station No. 0).
(The data is for the case of %MW1.1000=WORD#16#1234 and %MW1.1001=WORD#16#5678.)

[When the Standard loader is used]

When 2 words of data, beginning with WM1000 of the standard memory of CPU module (CPU station No. 0), are to be read, the command part becomes as follows. (The data is for the case of WM1000=H1234 and WM1001=H5678.)

	Request command	Response command
Operation status	FF	00
Connection mode	7A	7A
Connection ID (L)	00	00
Connection ID (H)	00	00
Fixed to 11h	11	11
Fixed to 00h	00	00
Fixed to 00h	00	00
Fixed to 00h	00	00
Fixed to 00h	00	00
Fixed to 00h	00	00
Command	00	00
Mode	00	00
Fixed to 00h	00	00
Fixed to 01h	01	01
No. of bytes of data part (L)	06	0A
No. of bytes of data part (H)	00	00
Memory type	02	02
Memory address (L)	E8	E8
Memory address (M)	03	03
Memory address (H)	00	00
No. of read words (L)	02	02
No. of read words (H)	00	00
Read data	-	34
	-	12
	-	78
	-	56

Appendix 1-3 Detail Description of Loader Commands

<Memory type code>

Memory type code for individual MICREX-SX internal memory is as follows: (The expression of address for D300win loader/Standard loader)

Target module	Memory type	Area name	Remarks
CPU module	02h	Standard memory (%MW1.*/WM*)	
	04h	Retain memory (%MW3.*/WL*)	
	08h	System memory (%MW10.*/WSM*)	
PE-link module	FFh	Broadcast area (%MW□.1.*/WM□.*)	See note 1
P-link module	FFh	Broadcast area (%MW□.1.*/WM□.*)	See note 2
FL-net module	FFh	Common memory, etc. (%MW□.1.*/WM□.*)	

Notes:

1) When the broadcast area of PE-link module is accessed by a command, the address does not begin with 0 (zero). Refer to the address correspondence table shown below.
2) When the broadcast area of P-link module is accessed by a command, high-speed bit area, high-speed word area, and low-speed word area are not consecutive. Refer to the address correspondence table shown below.

<Address correspondence table for PE-link>

	Access from application	Access from command		
		L	M	E
High-speed area 1 (high-speed bit area)	%MW□ .1.0/WM □.0 | %MW□.1.511/WM□.511	00 | 00	1E | 1F	00 | FF
High-speed area 2	%MW□.1.512/WM□.512 | %MW□.1.8703/WM□.8703	00 | 00	20 | 3F	00 | FF
Low-speed area 1	%MW□.1.8704/WM□.8704 | %MW□.1.12799/WM□.12799	00 | 00	40 | 4F	00 | FF
Low-speed area 2	%MW□.1.12800/WM□.12800 | %MW□.1.25087/WM□.25087	00 | 00	50 | 7F	00 | FF

<Address correspondence table for P-link>

	Access from application	Access from command		
		L	M	E
High-speed area 1 (high-speed bit area)	%MW□.1.0/WM□.0 | %MW□.1.511/WM□.511	00 | 00	00 | 01	00 | FF
High-speed area 2	%MW□.1.512/WM□.512 | %MW□.1.1663/WM□.1663	00 | 00	02 | 06	00 | 7F
Low-speed area 1	%MW□.1.1664/WM□.1664 | %MW□.1.4735/WM□.4735	00 | 00	08 | 13	00 | FF
Low-speed area 2	%MW□.1.4736/WM□.4736 | %MW□.1.7807/WM□.7807	00 | 00	14 | 1F	00 | FF

Appendix 1-3 Detail Description of Loader Commands

Appendix 1-3-2 Write data

This command is used to write the data for specified number of words in the area starting from specified address of specified data memory

1) Command/mode code

Command: 01h, Mode: 00h

2) Format of the data part of request command and response command

Request data

Memory type
Memory address (L)
Memory address (M)
Memory address (H)
Number of write words (L)
Number of write words (H)
Write data

Response data

Memory type
Memory address (L)
Memory address (M)
Memory address (H)
Number of write words (L)
Number of write words (H)

3) Layout of write data

Write data is laid out as follows:

MICREX-SX module

15	0
12	34
56	78

↔

Personal computer

34
12
78
56

4) Example of data write command

[When the D300win is used]

The command part for w1 word of data in the retain memory %MW3.1 of CPU module (CPU station No. 0) becomes as follows (see the table at right).
(Write data: WORD#16#1234)

[When the Standard loader is used]

The command part for w1 word of data in the retain memory WL1 of CPU module (CPU station No. 0) becomes as follows (see the table at right).
(Write data: H1234)

	Request command	Response command
Operation status	FF	00
Connection mode	7A	7A
Connection ID (L)	00	00
Connection ID (H)	00	00
Fixed to 11h	11	11
Fixed to 00h	00	00
Fixed to 00h	00	00
Fixed to 00h	00	00
Fixed to 00h	00	00
Fixed to 00h	00	00
Command	01	01
Mode	00	00
Fixed to 00h	00	00
Fixed to 01h	01	01
No. of bytes of data part (L)	08	06
No. of bytes of data part (H)	00	00
Memory type	04	04
Memory address (L)	01	01
Memory address (M)	00	00
Memory address (H)	00	00
No. of write words (L)	01	01
No. of write words (H)	00	00
Write data	34	-
	12	-

Appendix 1-3-3 CPU batch start

This command is used to start all the CPUs that exist in a configuration as a batch. Individual CPU is cold-started or warm-started, depending on its condition. If a CPU is already started up, or if the key switch is set at "RUN" position, the processing for startup is not executed for the CPU, which, however, does not result in an error, and response is returned normally.

1) Command/mode code

Command: 04h, Mode: 00h

2) Format of the data part of request command and response command

None

3) Batch start command

The command part for batch start is as follows:

	Request command	Response command
Operation status	FF	00
Connection mode	7A	7A
Connection ID (L)	00	00
Connection ID (H)	00	00
Fixed to 11h	11	11
Fixed to 00h	00	00
Fixed to 00h	00	00
Fixed to 00h	00	00
Fixed to 00h	00	00
Fixed to 00h	00	00
Command	04	04
Mode	00	00
Fixed to 00h	00	00
Fixed to 01h	01	01
No. of bytes of data part (L)	00	00
No. of bytes of data part (H)	00	00

Appendix 1-3-4 CPU batch initialize and start

This command is used to initialize and start all the CPUs that exist in a configuration as a batch. Individual CPU is cold-started. If a CPU is already started up, or if the key switch is set at "RUN" position, the processing for initialize and startup is not executed for the CPU, which, however, does not result in an error, and response is returned normally.

1) Command/mode code

Command: 04h, Mode: 01h

2) Format of the data part of request command and response command

None

3) Batch initialize and start command

The command part for batch initialize and start is as follows:

	Request command	Response command
Operation status	FF	00
Connection mode	7A	7A
Connection ID (L)	00	00
Connection ID (H)	00	00
Fixed to 11h	11	11
Fixed to 00h	00	00
Fixed to 00h	00	00
Fixed to 00h	00	00
Fixed to 00h	00	00
Fixed to 00h	00	00
Command	04	04
Mode	01	01
Fixed to 00h	00	00
Fixed to 01h	01	01
No. of bytes of data part (L)	00	00
No. of bytes of data part (H)	00	00

Appendix 1-3-5 CPU batch stop

This command is used to stop all the CPUs that exist in a configuration as a batch.
If a CPU is already stopped, or if the key switch is set at "RUN" position, the processing for stop is not executed for the CPU, which, however, does not result in an error, and response is returned normally.

1) Command/mode code

Command: 04h, Mode: 02h

2) Format of the data part of request command and response command

None

3) Batch stop command

The command part for batch stop is as follows:

	Request command	Response command
Operation status	FF	00
Connection mode	7A	7A
Connection ID (L)	00	00
Connection ID (H)	00	00
Fixed to 11h	11	11
Fixed to 00h	00	00
Fixed to 00h	00	00
Fixed to 00h	00	00
Fixed to 00h	00	00
Fixed to 00h	00	00
Command	04	04
Mode	02	02
Fixed to 00h	00	00
Fixed to 01h	01	01
No. of bytes of data part (L)	00	00
No. of bytes of data part (H)	00	00

Appendix 1-3-6 CPU batch reset

This command is used to reset all the CPUs that exist in a configuration as a batch.
This command is executed independent of the setting of the key switch of the CPU.

1) Command/mode code

Command: 04h, Mode: 03h

2) Format of the data part of request command and response command

None

3) Batch reset command

The command part for batch reset is as follows:

	Request command	Response command
Operation status	FF	00
Connection mode	7A	7A
Connection ID (L)	00	00
Connection ID (H)	00	00
Fixed to 11h	11	11
Fixed to 00h	00	00
Fixed to 00h	00	00
Fixed to 00h	00	00
Fixed to 00h	00	00
Fixed to 00h	00	00
Command	04	04
Mode	03	03
Fixed to 00h	00	00
Fixed to 01h	01	01
No. of bytes of data part (L)	00	00
No. of bytes of data part (H)	00	00

Appendix 1-3-7 CPU individual start

This command is used to start an arbitrary CPU existing in a configuration by specifying it. The CPU may be cold-started or warm-started, depending on its condition. Error occurs if the CPU is already started. Target CPU is specified by connection mode and connection ID.

1) Command/mode code

Command: 04h, Mode: 04h

2) Format of the data part of request command and response command

None

3) Example of individual start command

The command part for individual start is as follows (when CPU1 of a multi-CPU system is started):

	Request command	Response command
Operation status	FF	00
Connection mode	7B	7B
Connection ID (L)	FD	FD
Connection ID (H)	00	00
Fixed to 11h	11	11
Fixed to 00h	00	00
Fixed to 00h	00	00
Fixed to 00h	00	00
Fixed to 00h	00	00
Fixed to 00h	00	00
Command	04	04
Mode	04	04
Fixed to 00h	00	00
Fixed to 01h	01	01
No. of bytes of data part (L)	00	00
No. of bytes of data part (H)	00	00

Appendix 1-3-8 CPU individual initialize and start

This command is used to initialize and start an arbitrary CPU existing in a configuration by specifying it. The CPU is cold-started. Error occurs if the CPU is already started or if the key switch is set at "RUN" or "STOP" position. Target CPU is specified by connection mode and connection ID.

1) Command/mode code

Command: 04h, Mode: 05h

2) Format of the data part of request command and response command

None

3) Example of individual initialize and start command

The command part for individual initialize and start is as follows (when CPU1 of a multi-CPU system is initialized and started):

	Request command	Response command
Operation status	FF	00
Connection mode	7B	7B
Connection ID (L)	FD	FD
Connection ID (H)	00	00
Fixed to 11h	11	11
Fixed to 00h	00	00
Fixed to 00h	00	00
Fixed to 00h	00	00
Fixed to 00h	00	00
Fixed to 00h	00	00
Command	04	04
Mode	05	05
Fixed to 00h	00	00
Fixed to 01h	01	01
No. of bytes of data part (L)	00	00
No. of bytes of data part (H)	00	00

Appendix 1-3-9 CPU individual stop

This command is used to stop an arbitrary CPU existing in a configuration by specifying it. Error occurs if the CPU is already stopped or if the key switch is set at "RUN" or "STOP" position. Target CPU is specified by connection mode and connection ID.

1) Command/mode code

Command: 04h, Mode: 06h

2) Format of the data part of request command and response command

None

3) Example of individual stop command

The command part for individual stop is as follows (when CPU1 of a multi-CPU system is stopped):

	Request command	Response command
Operation status	FF	00
Connection mode	7B	7B
Connection ID (L)	FD	FD
Connection ID (H)	00	00
Fixed to 11h	11	11
Fixed to 00h	00	00
Fixed to 00h	00	00
Fixed to 00h	00	00
Fixed to 00h	00	00
Fixed to 00h	00	00
Command	04	04
Mode	06	06
Fixed to 00h	00	00
Fixed to 01h	01	01
No. of bytes of data part (L)	00	00
No. of bytes of data part (H)	00	00

Appendix 1-3-10 CPU individual reset

This command is used to reset an arbitrary CPU existing in a configuration by specifying it. Error occurs if the key switch is set at "RUN" or "STOP" position. Target CPU is specified by connection mode and connection ID.

1) Command/mode code

Command: 04h, Mode: 07h

2) Format of the data part of request command and response command

None

3) Example of individual reset command

The command part for individual reset is as follows (when CPU1 of a multi-CPU system is reset):

	Request command	Response command
Operation status	FF	00
Connection mode	7B	7B
Connection ID (L)	FD	FD
Connection ID (H)	00	00
Fixed to 11h	11	11
Fixed to 00h	00	00
Fixed to 00h	00	00
Fixed to 00h	00	00
Fixed to 00h	00	00
Fixed to 00h	00	00
Command	04	04
Mode	07	07
Fixed to 00h	00	00
Fixed to 01h	01	01
No. of bytes of data part (L)	00	00
No. of bytes of data part (H)	00	00

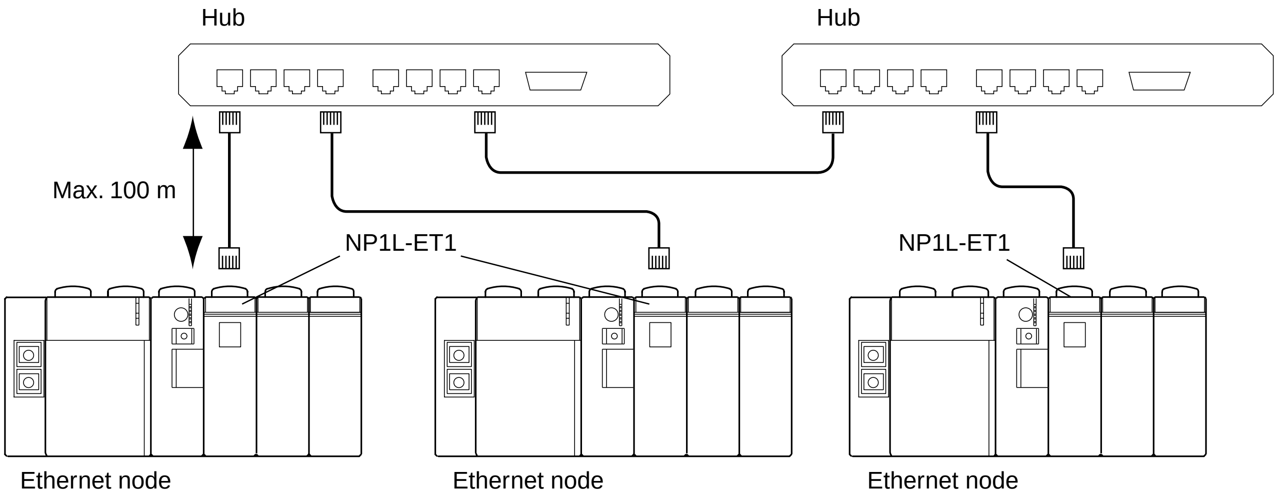
Appendix 2 Wiring for Ethernet

Page

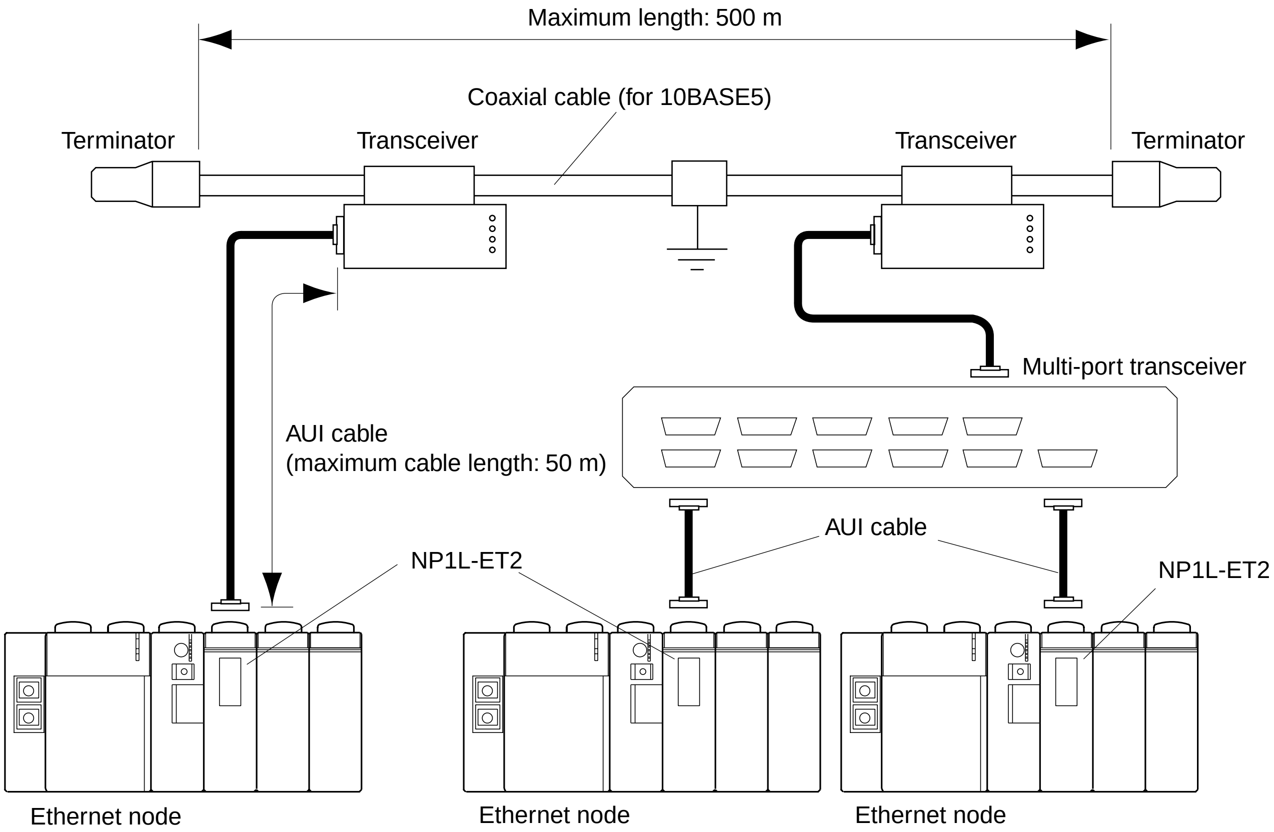
Appendix 2 Wiring for Ethernet

Appendix 2-1 Connection Configuration for Ethernet

Appendix 2-1-1 10BASE-T/100BASE-TX connection configuration

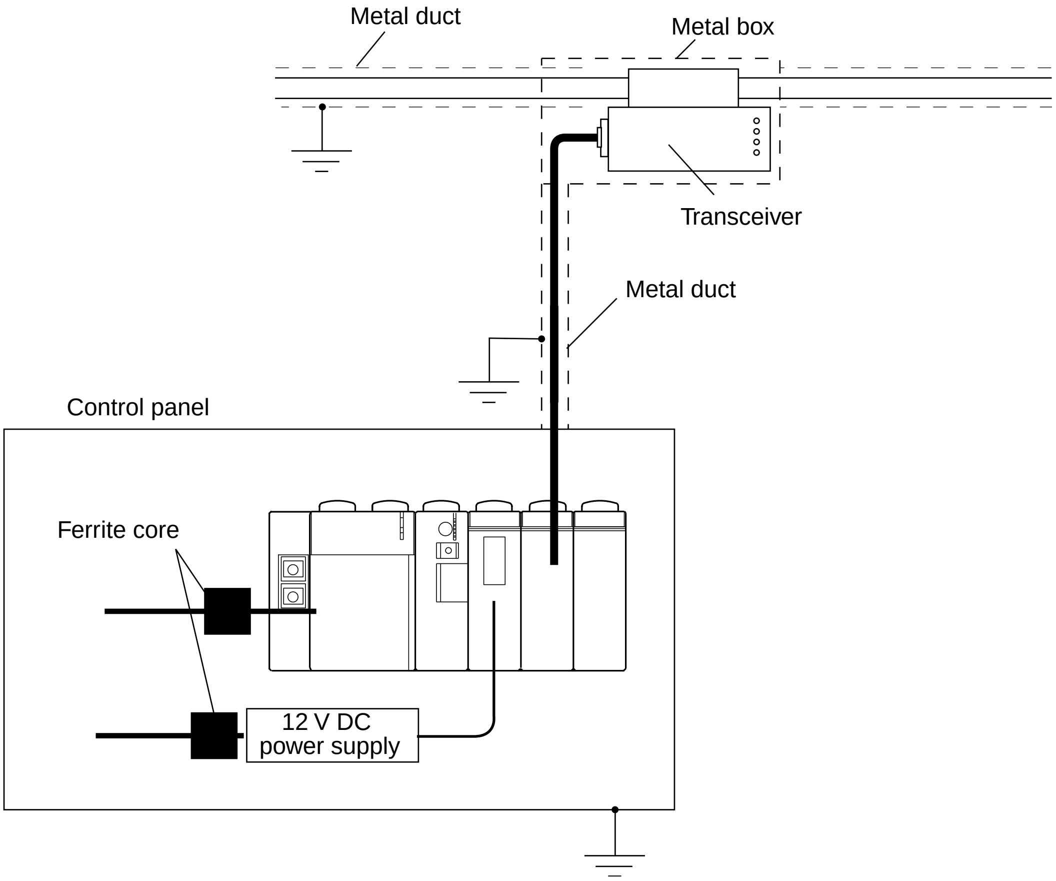


Appendix 2-1-2 10BASE5 connection configuration



Appendix 2-2 Layout and Wiring for Ethernet

When Ethernet system is used in an environment where the influence of noise is unavoidable, the following layout and wiring are necessary.



Note: The wiring and layout explained above may not be able to completely remove noise. So, for TCP sending, be sure to set a proper timeout value for sending operation, taking retry into consideration. For UDP/IP communication, be sure to execute retransmission by user application program.

Fuji Electric Systems Co.,Ltd.

Gate City Ohsaki, East Tower, 11-2, Osaki 1-chome, Shinagawa-ku, Tokyo, 141-0032,Japan
Phone: +81-3-5435-7280
Fax: +81-3-5435-7425
URL http://www.fesys.co.jp/eng/

Information in this manual is subject to change without notice.