FUJI ELECTRIC

MICREX-SX *series*

SPH

USER'S MANUAL

PC CARD INTERFACE MODULE

FEH226

Preface

This User's Manual explains the specifications of the PC Card Interface modules. 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.

Title	Manual No.	Contents
User's Manual Instructions, 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 configuration, the specifications and operations of modules in the MICREX-SX series.
User's Manual D300win <Introduction>, MICREX-SX series	FEH250	Explains the basic operations of D300win, the programming and monitoring for MICREX-SX series.
User's Manual D300win <Reference>, MICREX-SX series	FEH251	Explains the menu and icon of D300win and all of the operations of D300win.
User's Manual D300win V2 <Reference>, MICREX-SX series	FEH254	Explains the menu and icon of D300win V2 and all of the operations of D300win V2.

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 precaution 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 also result in a serious accident.
Both safety instruction categories provide important information. Be sure to strictly observe these instructions.

Warning

- ◊ Never touch any part of charged circuits as terminals and exposed metal portion while the power is turned ON. It may result in an electric shock to theoperator.
- ◊ Turn OFF the power before mounting, dismounting, wiring, maintaining or checking, otherwise, electric shock, erratic operation or troubles might occur.
- ◊ Place the emergency stop circuit, interlock circuit or the like for safety outside the PC. A failure of PC might break or cause problems to the machine.
- ◊ Do not connect in reverse polarity, charge (except rechargeable ones), disassemble, heat, throw in fire or short-circuit the batteries, otherwise, they might burst or take fire.
- ◊ If batteries have any deformation, spilled fluids, or other abnormality, do not use them. The use of such batteries might cause explosion or firing.
- ◊ Do not open the FG terminal with the LG-FG short circuited. (It must be grounded, otherwise it might cause electric shock.)

Safety Precautions

⚠ Caution

◊ Do not use one found damaged or deformed when unpacked, otherwise, failure or erratic operation might be caused.
◊ Do not shock the product by dropping or tipping it over, otherwise, it might be damaged or troubled.
◊ Follow the directions of the operating instructions when mounting the product. If mounting is improper, the product might drop or develop problems or erratic operations.
◊ Use the rated voltage and current mentioned in the operating instructions and manual. Use beyond the rated values might cause fire, erratic operation or failure.
◊ Operate (keep) in the environment specified in the operating instructions and manual. High temperature, high humidity, condensation, dust, corrosive gases, oil, organic solvents, excessive vibration or shock might cause electric shock, fire, erratic operation or failure.
◊ Select a wire size to suit the applied voltage and carrying current. Tighten the wire terminals to the specified torque. Inappropriate wiring or tightening might cause fire, malfunction, failure, or might cause the product to drop from its mounting.
◊ Contaminants, wiring chips, iron powder or other foreign matter must not enter the device when installing it, otherwise, erratic operation or failure might occur.
◊ Connect the ground terminal to the ground, otherwise, an erratic operation might occur.
◊ Periodically make sure the terminal screws and mounting screws are securely tightened.
Operation at a loosened status might cause fire or erratic operation.
◊ Put the furnished connector covers on unused connectors, otherwise, failure or erratic operation might occur.
◊ Install the furnished terminal cover on the terminal block, otherwise, electric shock or fire might occur.
◊ Sufficiently make sure of safety before program change, forced output, starting, stopping or anything else during a run. The wrong operation might break or cause machine problems.
◊ Engage the loader connector in a correct orientation, otherwise, an erratic operation might occur.
◊ Before touching the PC, discharge any static electricity that may have been collected on your body. To discharge it, touch a grounded metallic object. Static electricity might cause erratic operation or failure of the module.
◊ Be sure to install the electrical wiring correctly and securely, observing the operating instructions and manual. Wrong or loose wiring might cause fire, accidents, or failure.
◊ When disengaging the plug from the outlet, do not pull the cord, otherwiase, break of cable might cause fire or failure.
◊ Do not attempt to change system configurations (such as installing or removing I/O modules) while the power is ON, otherwise, failure or erratic operation might occur.
◊ Do not attemp to repair the module by yourself -- contact your Fuji Electric agent. When replacing the batteries, correctly and securely connect the battery connectors, otherwise, fire, accidents or failure might occure.
◊ To clean the module, turn power off and wipe the module with a cloth moistened with warm water. Do not use thinner or other organic solvents, as the module surface might become deformed or discolored.
◊ Do not remodel or disassemble the product, otherwise, a failure might occur.
◊ Follow the regulations of industrial wastes when the device is to be discarded.
◊ The modules covered in these operating instructions have not been designed or manufactured for use in equipment or systems which, in the event of failure, can lead to loss of human life.
◊ If you intend to use the modules covered in these operating instructions for special applications, such as for nuclear energy control, aerospace, medical, or transportation, please consult your Fuji Electric agent.
◊ Be sure to provide protective measures when using the module covered in these operating instructions in equipment which, in the event of failure, may lead to loss of human life or other grave results.
◊ External power supply (such as 24V DC power supply) which is connected to DC I/O should be strongly isolated from AC power supply.

Revisions

***Manual No. is shown on the cover.**

Printed on	*Manual No.	Revision contents
Jan. 2001	FEH226	First edition

Contents

Preface

Safety Precautions

Revisions

Contents

Contents

Contents

Section 1 General

Page

Section 1 General
1-1 General of PC Card Interface Module

The PC Card Interface Module [NP1F-PC2 (will be called PC2 here after)] is installed on the SX Bus of MICREX-SX Series SPH. It utilizes PC Cards on the market and realizes various functions (Communication with other equipment, Data read/write).

The NP1F-PC2 provides the following three functions.

- Ethernet Card communication function
- MODEM Card communication function
- Memory Card communication function

(1) Ethernet Card

When Ethernet Card is installed, communication with other equipment on the Ethernet network is achieved.

1) Universal communication mode (Refer to Section 6)

Communicates with other devices using TCP/IP, UDP/IP Protocols.
Communication with maximum 8 other nodes is possible.

2) Fixed buffer communication mode (Refer to Section 6)

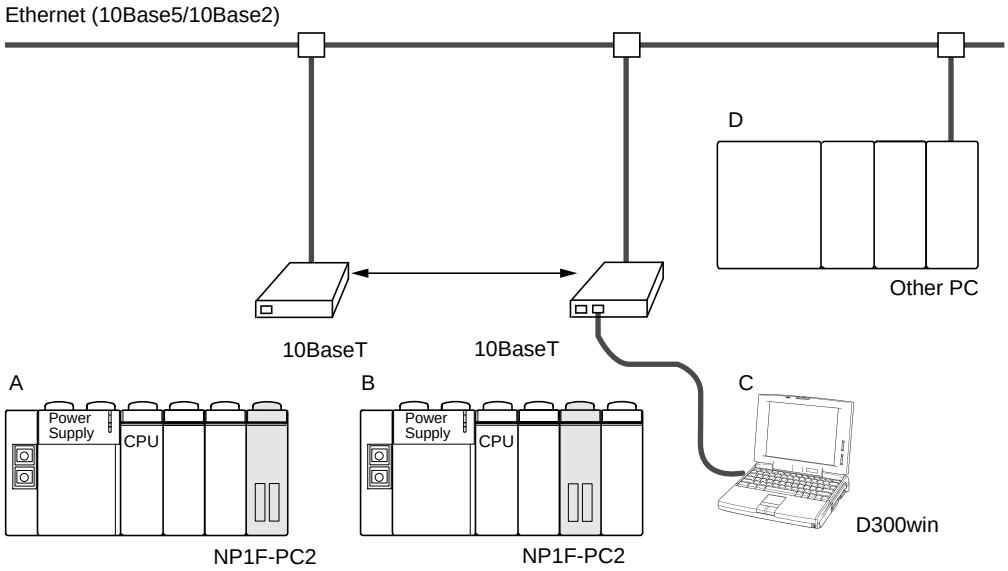
Communication using Fixed Buffer, provided by Mitsubishi Electric's AJ31E71 Ethernet Interface Unit, is possible.
Communication with maximum 8 other nodes is possible.

3) Auto transmission mode (Refer to Section 6)

Data transmission without "Receive Request" is possible by Loader Command auto reception in TCP/IP.
This function is limited to the communication among nodes consist of MICREX-SX.
It is possible to receive from maximum 2 nodes. And transmission to maximum 4 other nodes is possible.

4) Loader command mode

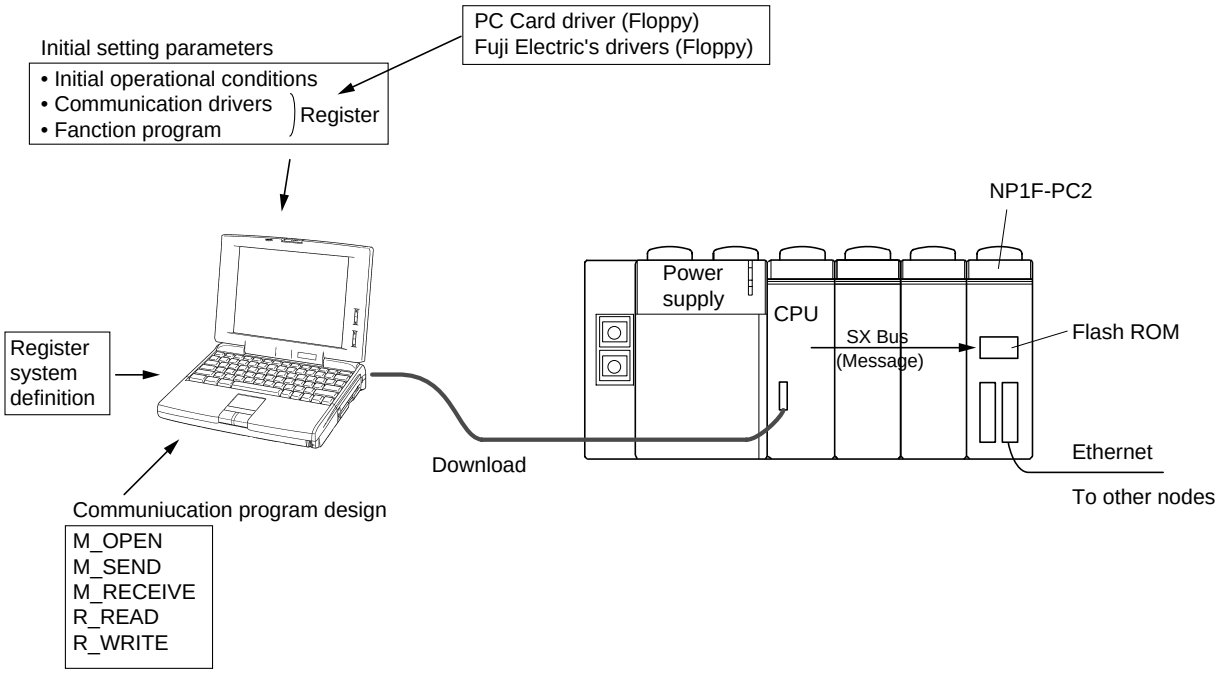
By connecting the Loader (D300win) to Ethernet, loader control of MICREX-SX is possible.
Maximum 2 Loaders can be connected to the Ethernet.



Route	Method
A to B	1), 2), 3)
A to C	4)
A to D	1), 2),

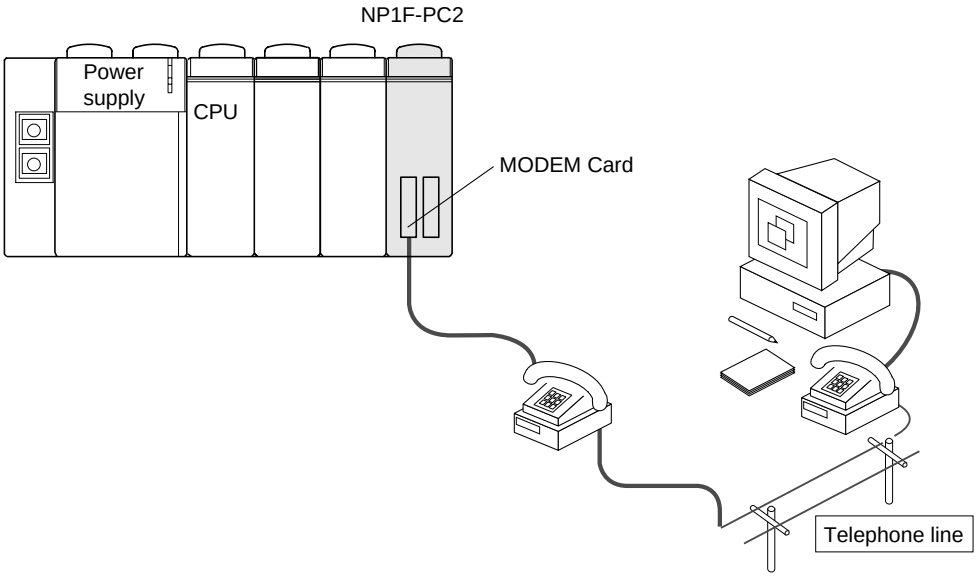
1-1 General of PC Card Interface Module

<Start-up of procedure in PC Card communication>
Procedure to the operation of the system when Ethernet Card is installed is described.

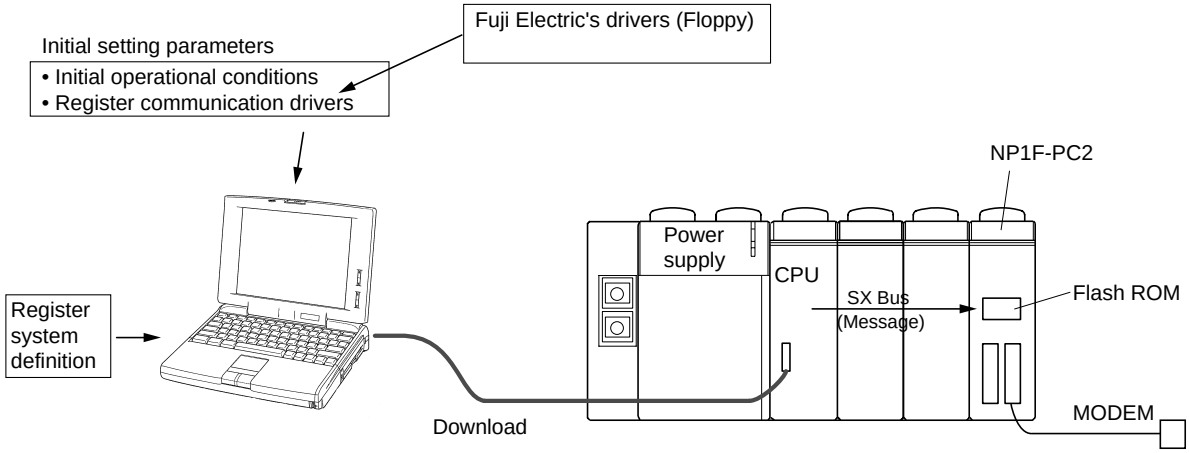


(1) Card selection	Refer to Section 1-2
⇩	
(2) Selection of communication mode	Refer to Section 6-1 to 6-5
⇩	
(3) System definition registration	Refer to Section 4
⇩	
(4) Initial parameter setting	Refer to Section 5-2
⇩	
(5) Communication program design	Refer to Section 6-6
⇩	
(6) Ethernet card installation	Refer to Section 9
⇩	
(7) Download system definition	Refer to Section 5-2-2 (6)
⇩	
(8) Download initial parameters	Refer to Section 5-2-2 (6)
⇩	
(9) Download program	
⇩	
Operation	

(2) MODEM Card communication function (Refer to Section 7)
By installing MODEM Card, communication with other devices via telephone line is realized.



<Start-up of procedure in MODEM Card communication>



(1) Card selection — Refer to Section 1-2

⇩

(2) System definition registration — Refer to Section 4

⇩

(3) Initial parameter setting — Refer to Section 5-3

⇩

(4) Communication program design (when Auto Send function (V2032 or later) is used.)

⇩

(5) MODEM Card installation — Refer to Section 9

⇩

(6) Download system definition — Refer to Section 5-2-2 (6)

⇩

(7) Download initial parameters — Refer to Section 5-2-2 (6)

⇩

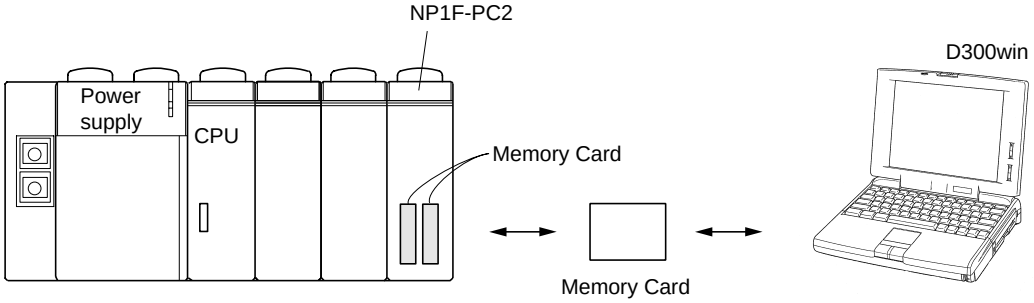
(8) Download program

⇩

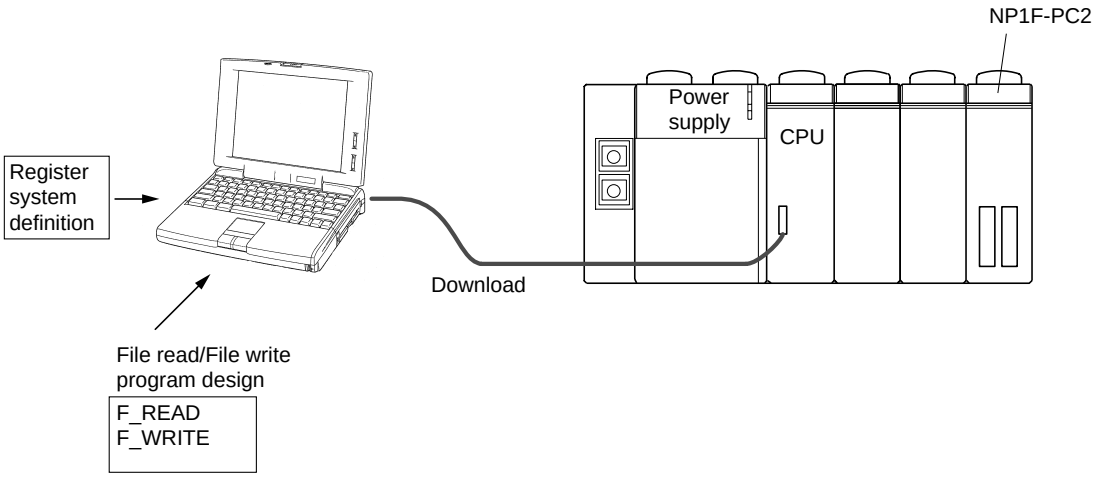
Operation

1-1 General of PC Card Interface Module

(3) Memory Card (Refer to Section 8)
Variables or data are written to or read from memory card installed in this interface module.



<Memory Card communication function>



* Initial setting is unneccessary.

(1) Card selection — Refer to Section 1-2
⇩
(2) System definition registration — Refer to Section 4
⇩
(3) Read/Write program design — Refer to Section 8
⇩
(4) Memory Card installation — Refer to Section 9
⇩
(5) Download system definition program
⇩
Operation

1-2 Compatible Cards

Following table shows the list of PC Cards recommended for use with this module.

	Type	Part No.	Maker	Comment
1	LAN Card	LD-CDY	ELECOM (Laneed)	
		LD-CDS	ELECOM (Laneed) (Note 4)	
		ENW-3501-T	Planet Communications	
		ENW-3503-T	Planet Communications (Note 4)	
		3CCE589ET-JP	3COM	
		LAK-CD021BX	TDK	
2	MODEM Card	DF5633ES	TDK	
3	Memory Card	FLASH-PACKER Series	Sandisk (EPSON)	FLASH ATA(2.5 to 40MB)
		ML-4M-TB4N	Maxell	SRAM (4MB)
		ML-2M-TB4N	Maxell	SRAM (2MB)
		ML-1M-TB4N	Maxell	SRAM (1MB)
		ML-512TB4N	Maxell	SRAM (512KB)
		ML-256TB4N	Maxell	SRAM (256KB)

Note:

1) Caution on Use of LAN Cards
   DOS ODI Driver is used in this unit. There are 2 types of DOS ODI Drivers available, namely SPEC3 and SPEC4. This unit supports SPEC3 ODI Driver only and therefore LAN Card with only SPEC4 ODI Driver can not be used.

2) Caution on Use of Memory Cards
   There are several types of Memory Cards are available such as SRAM, FLASH ATA, MINI etc. The Memory Cards that can be used with this module are SRAM and FLASH ATA with DC 5V rating. The Memory Card with DC 3V specification can not be used.

3) Caution on Use of PC Cards
   When selecting PC Cards, choose one listed in above table. If other cards are used with this module, there are possibilities of malfunctioning. Please exercise special care in selecting LAN Card.

4) When using LD-CDS (Laneed) and ENW-3503-T, it is necessary to update Initialization file. For details, refer to Appendix 2, “Updating Initialization File.”

Section 2 Specifications

Page

Section 2 Specifications
2-1 General Specifications

Item		Specification
Physical environment	Working temp.	0 to 55° C
	Storage temp.	-25 to +70° C
	Relative humidity	20 to 95%RH with no condensation (During Shipment: 5 to 95%RH with no condensation)
	Pollution	Pollution Degrees 2
	Corrosion	No corrosive gases. No contact with organic solvents.
	Altitude	2000m or less above sea level (Transport condition: 70kPa or more)
Mechanical operating condition	Vibration	Amplitude: 0.3mm. Acceleration: 19.6m/s^2 2 hours in each direction, total 6 hours.
	Shock	Peak Acceleration: 147m/s^2 3 times in each direction
Electrical operating condition	Noise	With noise simulator. Rise Time: 1ns, Pulse Width: 1μs, 1.5kV
	Static electric discharge	Contact Discharge: ± 6kV, Discharge in the air: ± 8kV (PC Card performance is as per its specification.)
	Electro-magnetic field	10V/m (80MHz to 1000MHz)
Structure		To be installed in enclosure. IP30
Cooling method		Natural cooling
Insulation characteristic	Insulation method	No insulation
	Insulation	445V AC 1 minute, between I/O connector and earth.
	Insulation resistance	10MΩ or more with 500V DC megger (between I/O connector pins and frame ground)
No. of slots occupied		1 Slot
Internal current consumption		24V DC, 160mA or less (without PC Card installed)
Mass		Approx. 220g (excluding PC Card)
Dimensions		Refer to Section 2-5.

Note: At power up with the NP1F-PC2 installed on the SX Bus, It takes approximately 12 seconds before the SX Bus is operational (longer than normal boot-up).
(It takes a longer time than a system not using the NP1F-PC2.)

2-2 Performance and Functional Specifications

PC Card Socket Specification

Item	Specification
PC Card socket standard	Complies with JEIDA Ver. 4.01 / PCMCIA Rel. 2.01
PC Card standard	2 Slots of Type I, II. (Type III is not supported.)
Card power supply	Supports 5V DC (3.3V DC spec. cards can not be used.)
Accepted card types	• Ethernet Card : Ethernet Card on the market • MODEM Card : MODEM Card on the market • Memory Card : SRAM, FLASH ATA Memory Card

Ethernet Transmission Specifications

Item	Specification
Connector type	10BaseT
Data transmission speed	10Mbps
Transmission method	Base band
Maximum segment length	100m (between node and hub)
No. of nodes	1 Unit/Segment
Node spacing	None
Maximum possible No. of nodes to communicate	8 Station (Port)
Transmission code	Binary / ASCII

<Other PC Cards >

If the PC Cards not listed in the table of recommended products is purchased, points listed in table below should be checked.

Indicated specification item	Example of PC Card spec.	Compatibility with this module and conditions	Comments
Supply voltage	5 ± 0.25V	OK. If 5V specification	
Maximum current consumption	5V DC, 90mA or less	NP1F-PC2: OK. If total current consumption of 2 slots is less than 500mA	
Operating temperature	0 to 60° C	PC Card temperature will rise 10° C due to the heat from the module. Therefore the 10° C de-rating applies. Upper limit of the operating temperature will be governed by the module's operating temperature, 55° C.	If the environmental specification of PC Card is tighter than this module's specification, it must be operated within the specification of the PC Card.
Operating humidity	10 to 90% RH No condensation	OK: Wider range than this module's specification	
Storage temperature	-20 to 70° C	OK: Same range as this module's specification	
Number of insertion/removal	5,000 (Out of office) 10,000 (Office Condition)	Excise care for this specification.	
Vibration/Shock	Vibration: 15Gp-p (Max.) Shock: 50G (Max.)	Specification of this module is achieved by use of brackets supplied with the module.	

Note: Purchase Memory Cards with “Electrostatic Measures” in addition to the specifications above.

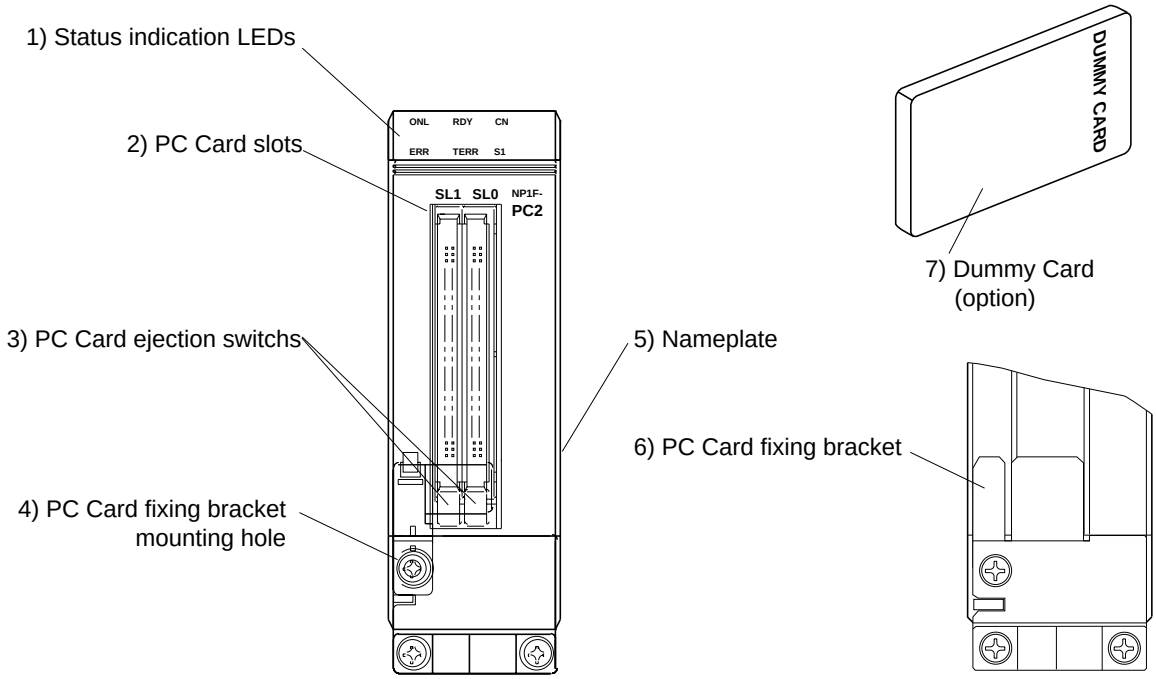
In order to operate this module on the SX Bus, the module has to be registered in the system definition.

	General of function
System structure definition	Register this module in the structure definition of the SX CPU. (Module will not operate without registration of the structure.)

Note: Refer to Section 4: System Definition for details and procedures of registration.

2-4 Name and Function of Each Section

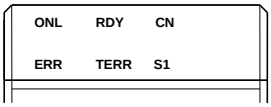
2-4-1 Name of each section



2-4-2 Function of each section

1) Status indication LEDs

These LEDs indicate NP1F-PC2 operation status.



Symbol	Colour	Indicated Status	Conditions
ONL	Green	Normal operation	Lights when the SX Bus connection OK and the module booted up OK.
RDY	Green	Communication preparation	Lights when communication initial process is completed
CN	Green	Connection status	Lights when connection is made. (OR of all connection status)
TERR	Red	Transmission error	Lights when there is abnormal communication.
ERR	Red	Hardware error	Lights when fault is found within the module.
S1	Red	Memory card access status	Lights when Memory Card is being accessed.

2) PC Card slots

Slots for inserting PC Card devices.

3) PC Card ejection switchs

Press this switch to remove PC Card.

4) PC Card mounting hole

Mounting hole for fixing bracket to this module.

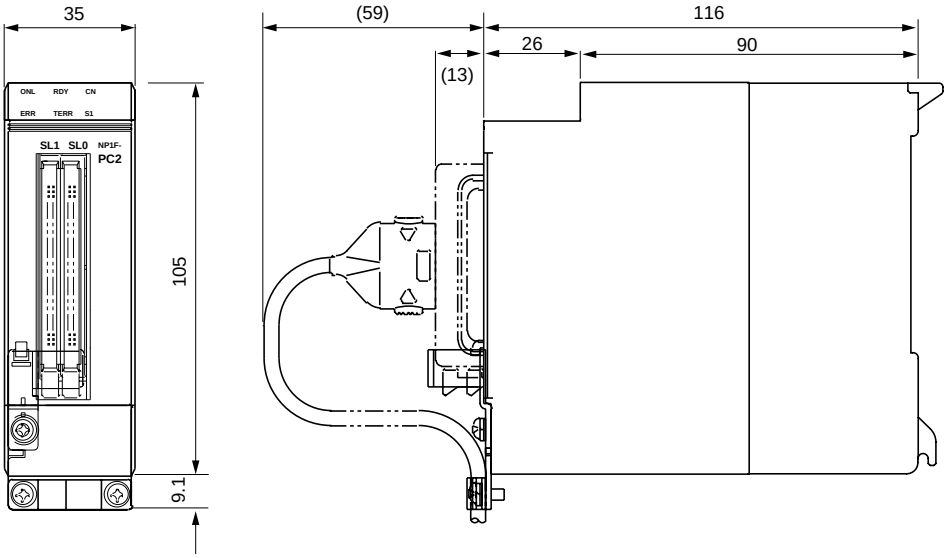
5) Nameplate

6) PC Card fixing bracket (accessory)

Prevent PC Card becoming loose or dislodged.
Use M3 screw provided for mounting. (Tightening Torque: 0.7 ± 0.08N·m)

7) Dummy Card 2 of (accessory)

When PC Cards are not used, insert dummy cards to stop dust from reaching sockets.


35
(59)
116
26
90
(13)
ONL
RDY
CN
ERR
TERR
S1
SL1
SL0
NP1F-
PC2
105
9.1

Section 3 System Configuration

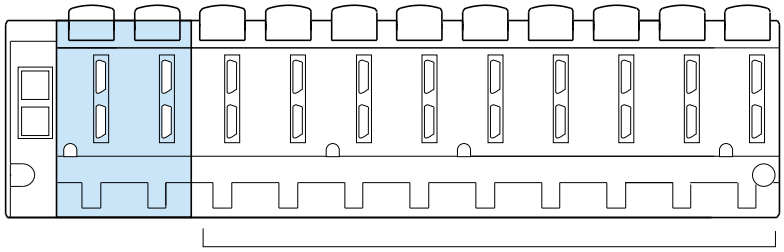
Page

3-1-1 Mounting position

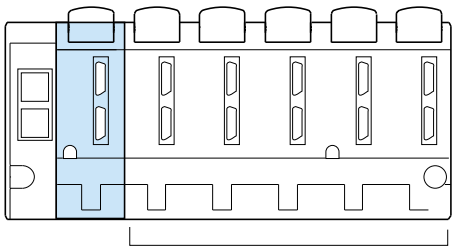
The NP1F-PC2 is a communication module, which is mounted on the SX Bus of MICREX-SX Series. The restriction in mounting this module on the baseboard is explained here.

The module can be mounted any position except 2 left most slots dedicated to the Power supply module.

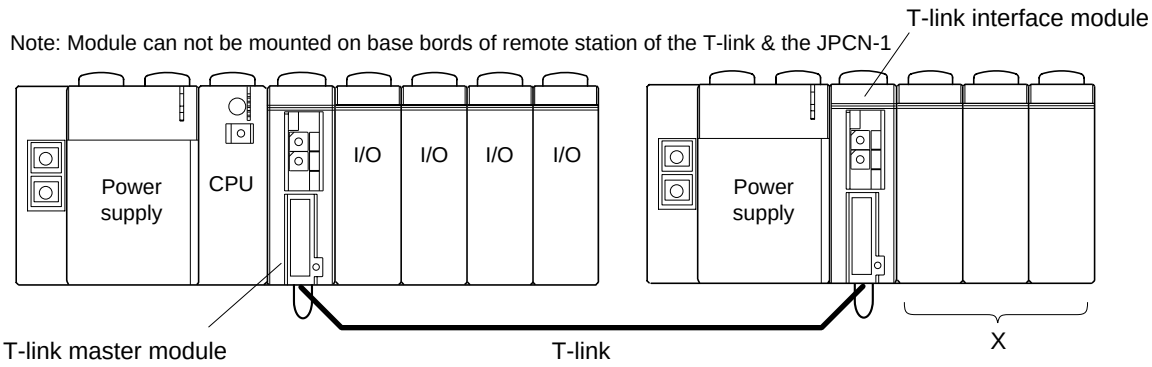
<Base bords excluding 6-slots base bords>



<6-slots base bords>



Note: Module can not be mounted on base bords of remote station of the T-link & the JPCN-1



3-1-2 Number of mountable modules

Maximum 4 units of the NP1F-PC2 can be mounted on any one configuration.
When there are other communication modules mounted on the same configuration, total number of these modules will be 16 units.

FL-net (OPCN-2) module (NP1L-FL1)
P-link module (NP1L-PL1)
PE-link module (NP1L-PE1)
General communication module (NP1L-RS1/2/4)
PC Card interface module (NP1F-PC2)
Memory Card interface module (NP1F-MM1)

] Total 16 units/Configuration

3-2 Precautions for PC Card Mounting

When mounting PC Card into the NP1F-CP2, pay attention to following points.

(1) Communication Card

- Only one communication card (Ethernet or MODEM Card) can be mounted on one module. (2 slots can not have these cards at the same time.)
- Either of slots, SL0 or SL1 may be used to mount communication card.
- When Ethernet or MODEM function is used, these cards must be mounted prior to the module power up. And power must be turned off before removing card.

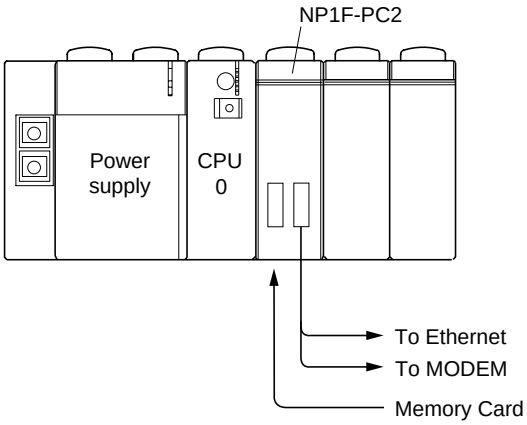
(2) Memory Card

- Two Memory Cards can be mounted at the same time. Only one Memory Card may be mounted.
- When Memory Cards are used, the Autoexec.bat file, located in the Memcard Folder on the “PC Card Interface Driver” floppy disc, must be downloaded.
  For download procedure, refer to Section 5, “Initial Setting.”

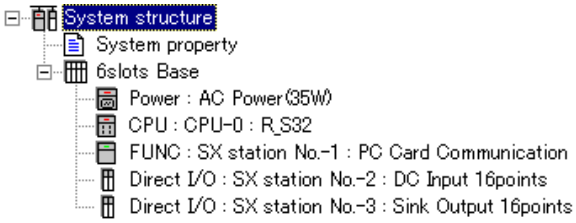
Section 4 System Definition

When building a system with the MICREX-SX Series, it is required that the modules used are registered in the system structure.

<Example of system structure>

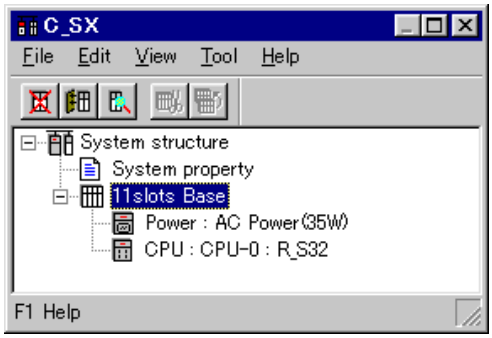


<System structure definition tree screen>
System structure definition tree for the above example will be as follows.

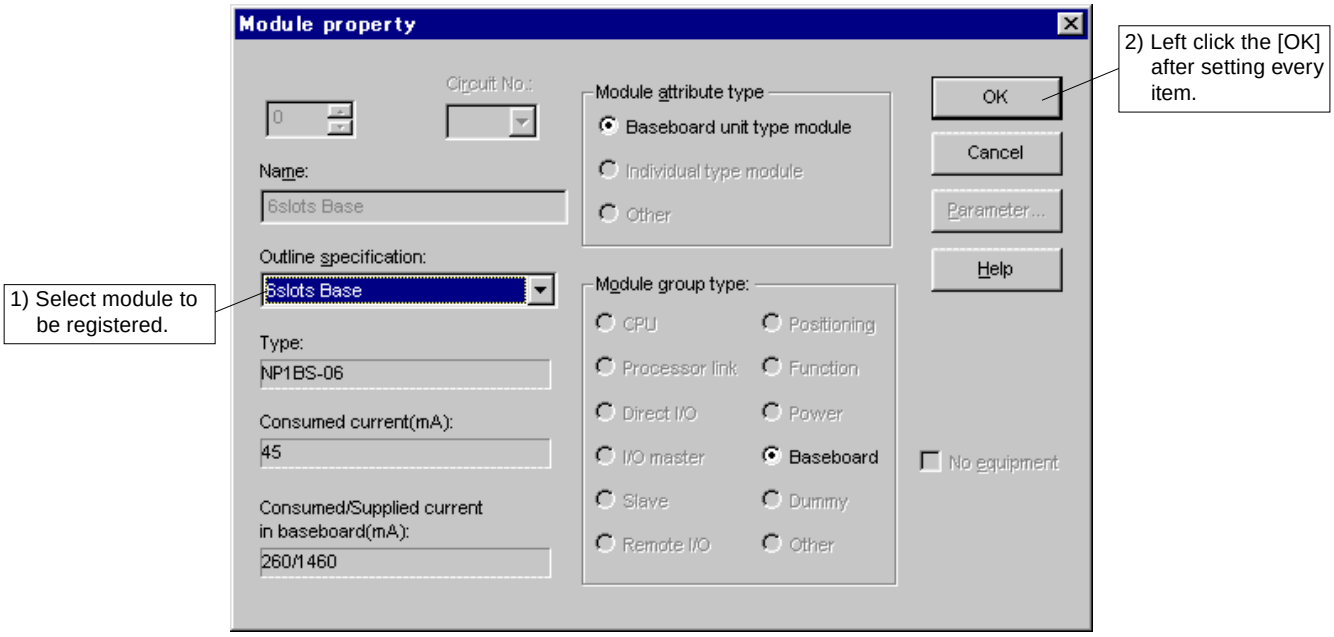


<Procedure>

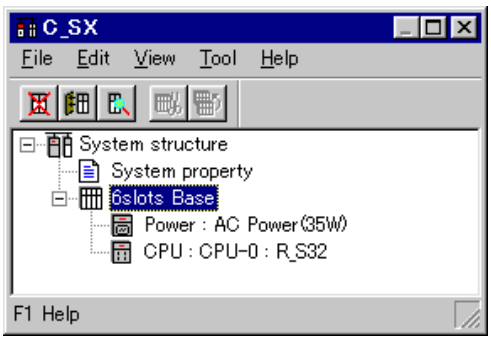
1) Open System Structure Definition window by double clicking [System_Definition] icon.
Initial window shows a Power supply and a CPU Module registered on 11 Slot Base board.
Select base bord and left-click the [Module Property] button.



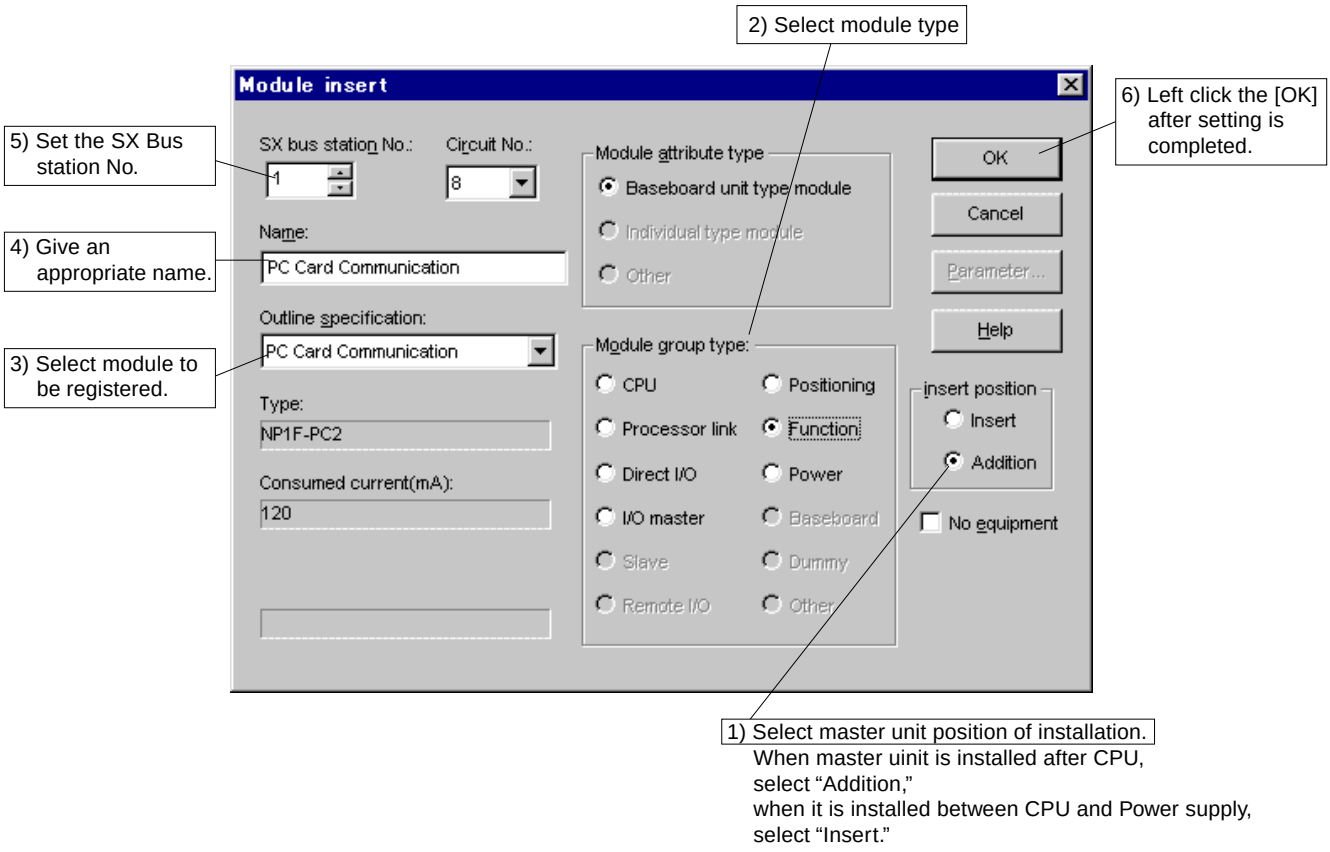
2) [Module Property] dialogue box is displayed as the figure below.



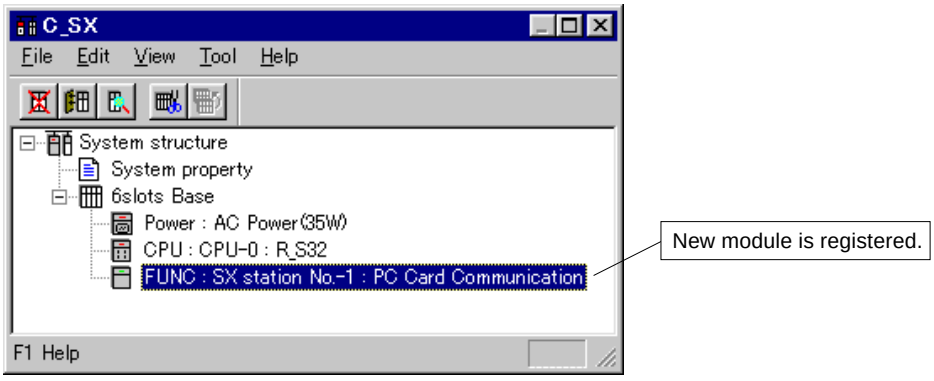
3) New base bord will be registered as the figure below.



4) Following CPU Module, register PC Card I/F Module. Choose CPU and then click Insert button. Module Insertion dialogue box is displayed.



5) The Module will be registered as follows.



Register other modules in similar manner.

Section 5 Initial Setting

Page

Section 5 Initial Setting
5-1 General

When installing various PC Cards into this module, the initial setting of the device must be registered prior to the operation.
Without the registration of the initial setting, normal operation can not be expected.

Initial settings are made in following points.

Function	Setting Part		Setting Point	Detailed Explanation
	NP1F-PC2	Card		
Ethernet	O	X	D300win System Definition Parameter.	Section 5-2
MODEM	O	X	D300win System Definition Parameter.	Section 5-3
Memory Card	O	O * 1	D300win System Definition Parameter, and Operation in Windows.	Section 5-4

Note: * 1 SRAM Cards require initialization and Flash ATA Cards do not.

Refer to the pages related to each card type for details.

5-2 Initial Setting For Ethernet

The communication environment of the Ethernet needs to be registered in the NP1F-PC2.

5-2-1 General of initial setting parameters

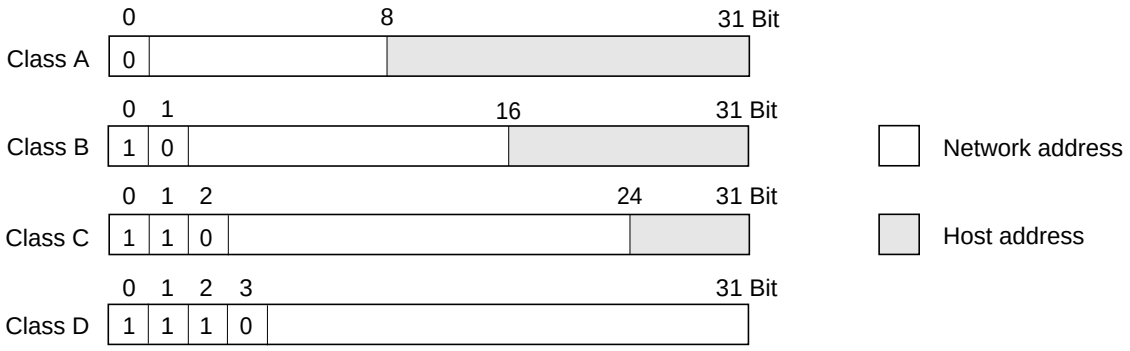
Following table shows parameters for initial setting. Initial setting is made in “Parameter” section of the NP1F-PC2 [Module Property] dialogue box.

Item	Description		Initial Value	Comment
IP Address	IP Address		None	Refer detailed explanation
	Sub-net Mask		255.255.255.0	
Gateway	Default Gateway: IP Address		None	Displays current setting
	New Gateway	IP Address	None	Used to add new gateways
		Net Address	None	
	Installed Gateway		None	Displays current setting
Detail setting	TCP send time-out value		300	Refer detailed explanation
	Max. TCP end timer value		0	
	Response watchdog timer value		300	
	Server FTP command watchdog timer value		3000	
	Client FTP data connection watchdog timer value		30	
	Ethernet communication trace		No Trace	
	Close process at TCP send time-out		Not channel close	
	Close process at response receive time-out		Not channel close	
	Reference No. of own port		256	
	Reference No. of other node's port		256	
Download files	Download initial setting data		None	
	Table of download files		None	

Setting details

(1) IP Address

The Module’s IP Address is set. (IP Address is decided by the Network Manager.)
IP Address is 32-bit address used to identify nodes on Ethernet and made up from Net ID and Host ID. Net ID is network’s address and Host ID is the address of Host (Node) on the network.
IP Addresses are divided into following classes.



IP Address is divided into four 8-bit segments and expressed as 4 decimal numbers separated by dot (.).

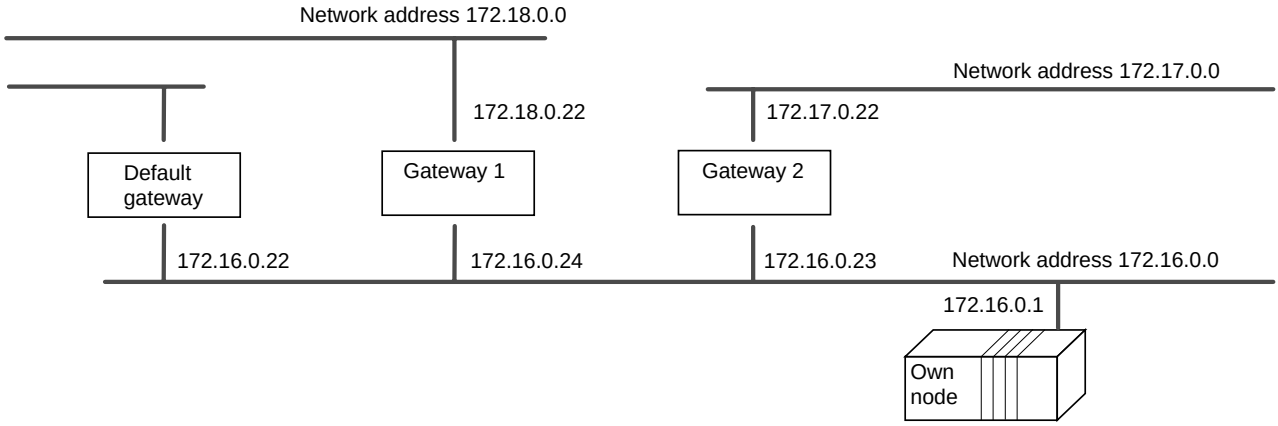
(2) Sub-net Mask
Node control becomes difficult when there are too many nodes on a network. In such a case Sub-net Mask is set and effectively divide the network into multiple Sub-net (or Group). Sub-net uses a part of Host ID. A 32-bit Sub-net Mask is formed by setting Net ID and Sub-net ID part of the mask bit to [1] and Host ID part to [0]. In case of no Sub-net, the mask is set to [0.0.0.0].

Example 1) Class B network is divided into 16 sub-nets.
[255.255.240.0] (FFFFF000)

Example 2) Class B network is divided into 256 sub-nets.
[255.255.255.0] (FFFFFF00)

(3) Gateway
IP Router (Gateway) allows connection of multiple IP Network segments. On IP Network connected through IP Router, it is possible to communicate with network beyond the IP Router. In this case IP Address of the Gateway and IP Address of the network need to be set. If there is no gateway set [0.0.0.0]. When the network address is set to [0.0.0.0], designation to the gateway is ignored.

Example)
When 3 Gateways are present including default gateway, addresses are set as diagram below.



New gateway:
Area to set IP Address and Network Address when adding new gateways.

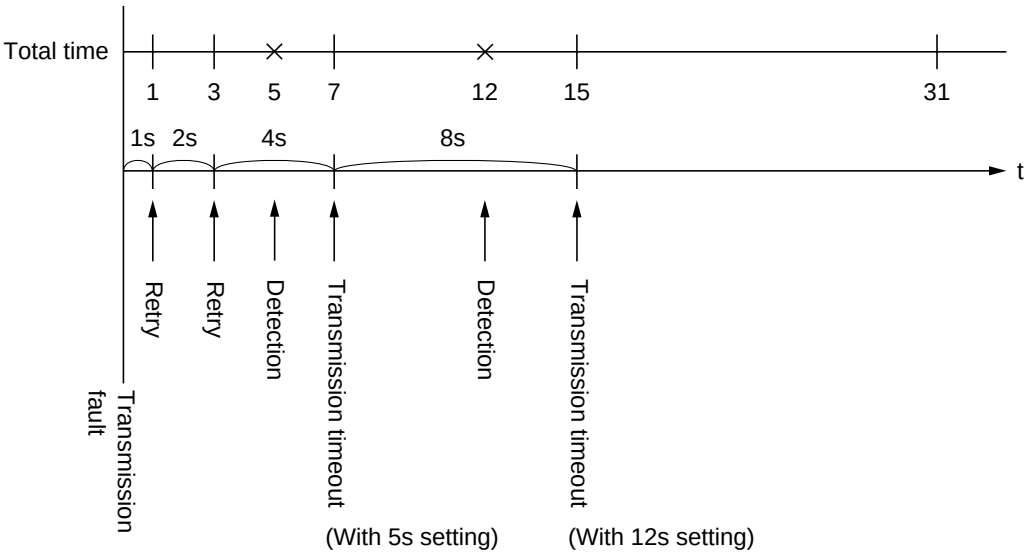
Installed gateway:
Currently installed gateways are displayed here.

Example) Address setting for the Gateway 1 above.
IP Address: 172.16.0.24
Network Address: 172.18.0.0

(4) TCP Send Timeout (0.1s. Unit) [Setting Range: 0 to 300]
This is the preset value of the watchdog timer for TCP Open and data transmission.
In TCP/IP communication, if it fails to Open or Send Data, Ethernet will automatically retries. The interval of the retry increases as it continues to fail the operation (1s, 2s, 4s, 8s etc. exponentially increase).

1) When Open Process could not complete successfully after retries, it ends with fault after the time specified with this parameter elapsed.
2) When Send Process could not complete successfully, it ends with fault at the first retry after the time specified with this parameter elapsed. Therefore the time from commencement of Send process to End with Fault is longer than the timer preset value.

When this timer is set to 12 seconds, internal operation to the error detection is described in following diagram.



(5) Maximum TCP End Timer (0.1s. Unit) [Setting Range: 0 to 600]
Specifies time period to keep socket after Close Process initiated from own station is completed successfully.
When TCP connection is closed from own station, the maximum possible time to wait for the other node to complete close process is twice of TCP Send Timeout.
In case of the other node does not complete close process (FIN signal is not received.), RST Process (Send RST) is executed to the other station in order to force closure.

(6) Response Watchdog Timer (0.1s. Unit) [Setting Range: 0 to 300]
Time to wait for Reply, after sending a command, is set.

(7) Server FTP Command Watchdog Timer (0.1s. Unit) [Setting Range: 0 to 3000]
In Server FTP Communication and client node is logged in, if there are no commands received from the client for fixed time period, the connection will be terminated automatically. This parameter sets the time period.
To keep the connection even if no commands received from the client, set [0] for this parameter.

(8) Client FTP Data Connection Watchdog Timer (0.1s. Unit) [Set Range: 0 to 30]
In Client FTP Communication, waiting time for the server (other node) to open data connection is set with this parameter.

(9) Ethernet Communication Trace (Communication Condition) [Setting: Yes/No]
Sets whether Ethernet Trace is carried out or not. When [Yes] is set, the Ethernet Trace Data is output to the Memory Card installed in PC Card Connector.

(10) Close Process at TCP Send Timeout
(Communication Condition) [Setting: Yes/No]
Select whether the connection should be closed or not, when ACK is not received after retries in TCP/IP Protocol.
Note: This setting is invalid. At TCP Send Timeout, connection will be forced to close.

(11) Close Process at Response Receive Timeout
(Communication Condition) [Setting: Yes/No]
Select whether the connection should be closed or not when Response Receive Timeout occurs.

(12) Reference Number of Own Port [Setting: 0 to 65280]
Sets reference number of own port number in TCP/IP and UDP/IP. The own port number specified in Application FB (User Interface) will be added to the reference number and used as port number in TCP/IP and UDP/IP communication.

For example, if the following setting is made in Parameter Setting screen for PC Card I/F Module,
Reference Number of Own Port = 256
Reference Number of Other Node's Port = 256
and the following M_OPEN FB setting is made,
RPORT_NO = 1
SPORT_NO = 1
the following results:

Port Number of PC Card I/F module:	Reference Number of Own Port + SPORT_NO = 258
Port Number of Other Node (Personal computer):	Reference Number of Other Node's Port + RPORT_NO = 257

(13) Reference Number of Other Node's Port [Setting: 0 to 65280]
Sets reference number of other node's port number in TCP/IP and UDP/IP. The other node's port number specified in Application FB, will be added to the reference number and used as port number in TCP/IP and UDP/IP communication.

(14) Initial setting data download
IP Address, Gateway and Detail Setting are stored in an Initial Value File and downloaded to the module [NP1F-PC2] when project is downloaded to the CPU.
Destination directory is fixed to [D:¥]. (Flash ROM in the [NP1F-PC2] is D drive.)

(15) Table of download files
Displays file names of the driver software for the PC Cards installed in [NP1F-PC2]. These files will be downloaded this module [NP1F-PC2], when the project is downloaded to CPU.

1) Chksum.txt
2) Config.sys
3) Net.cfg
4) PCTCP.exec
5) Autoexec.bat

} Fuji Electric supplied floppy

6) ODI.DOS ……Floppy supplied with Ethernet Card

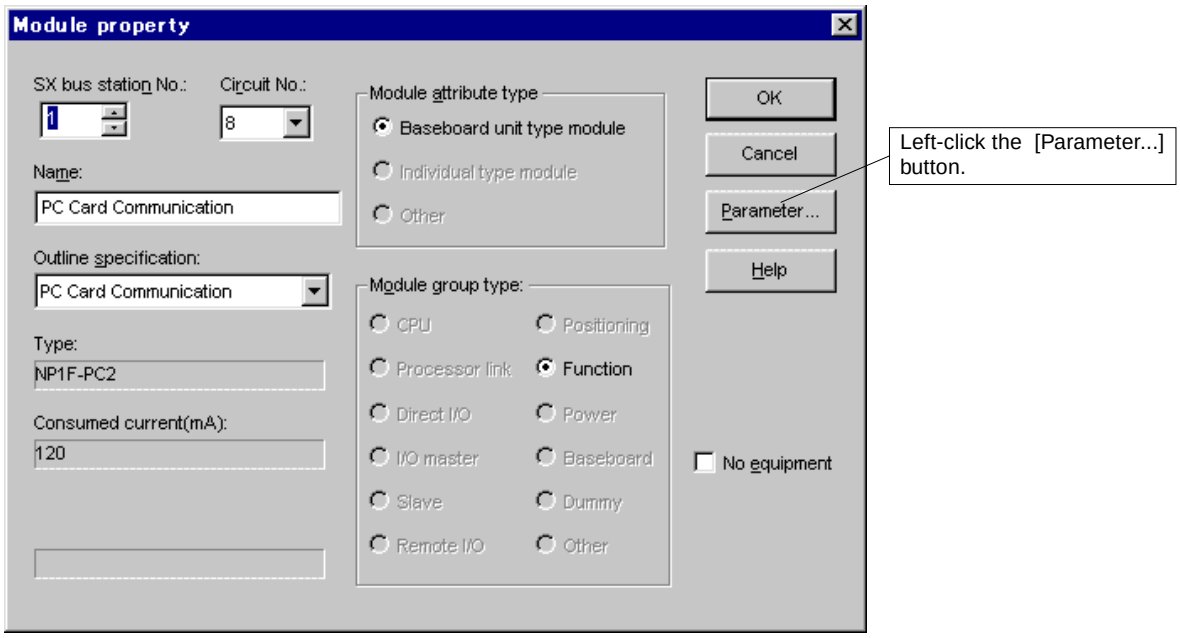
5-2 Initial Setting For Ethernet

5-2-2 Procedure

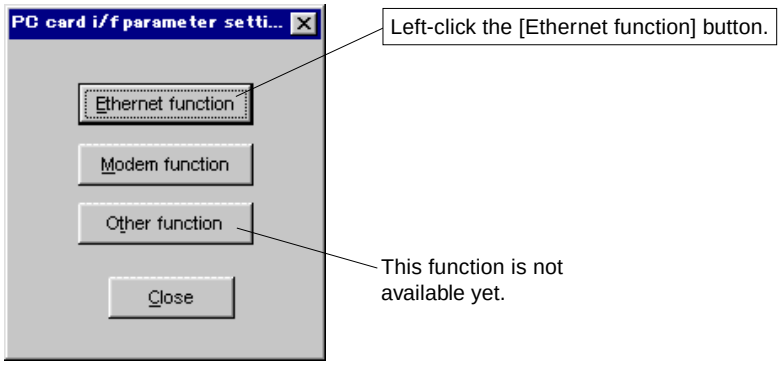
Procedure for setting up the Ethernet Initial Setting is explained.

(1) Ethernet parameter selection

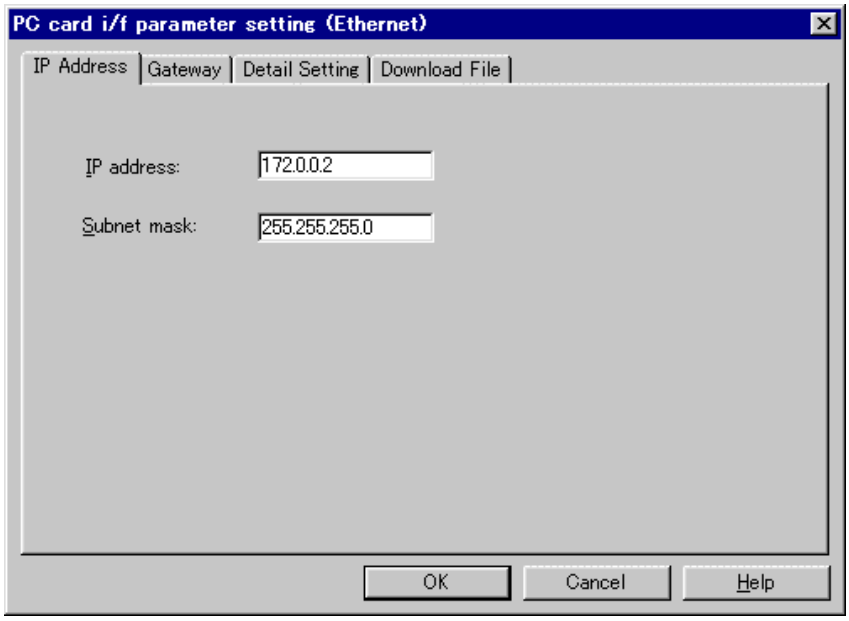
1) Click on [Parameter...] Button on [Module property] dialogue box.



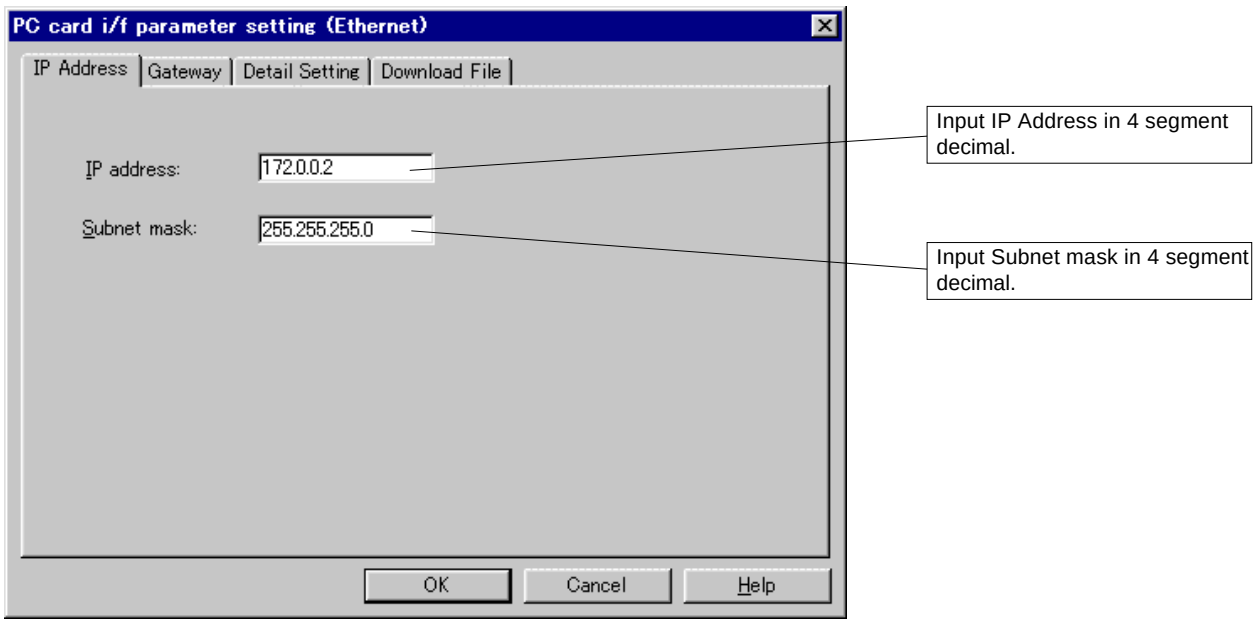
2) [PC Card I/F parameter setting] dialogue box (as the figure below) appears.
(D300win Ver.2.1 and later, with older version, screen for 3 is displayed)



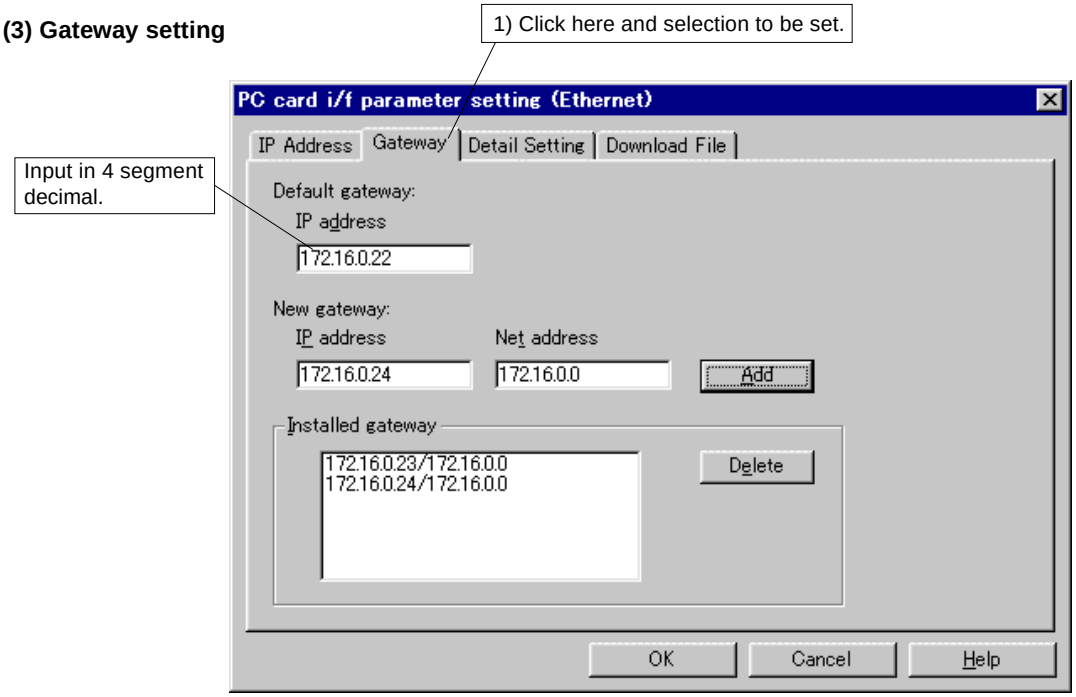
3) Ethernet parameter setting window appears.



(2) IP address setting



(3) Gateway setting



1) Add

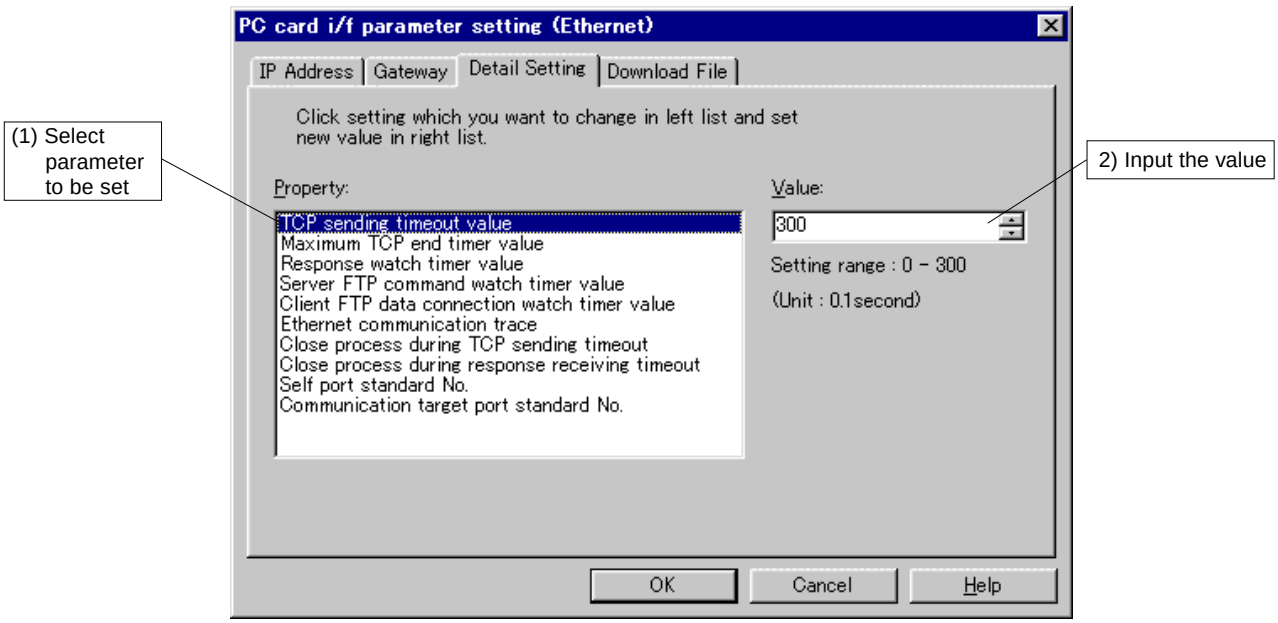
Set [IP address] and [Net address] for the new gateway then click [Add] button.
New address will be added to the list of [Gateways Installed].

2) Delete

Select gateway address to be removed from [Gateways installed] list then click [Delete] button with left mouse button.

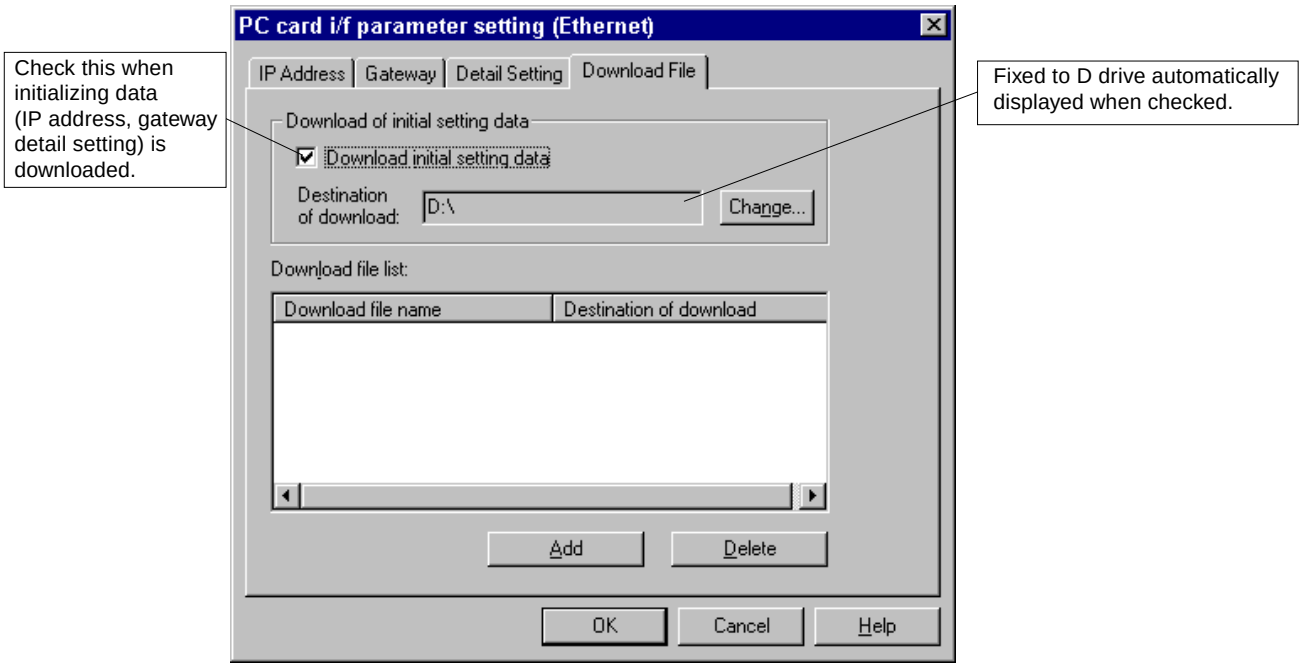
5-2 Initial Setting For Ethernet

(4) Detail setting



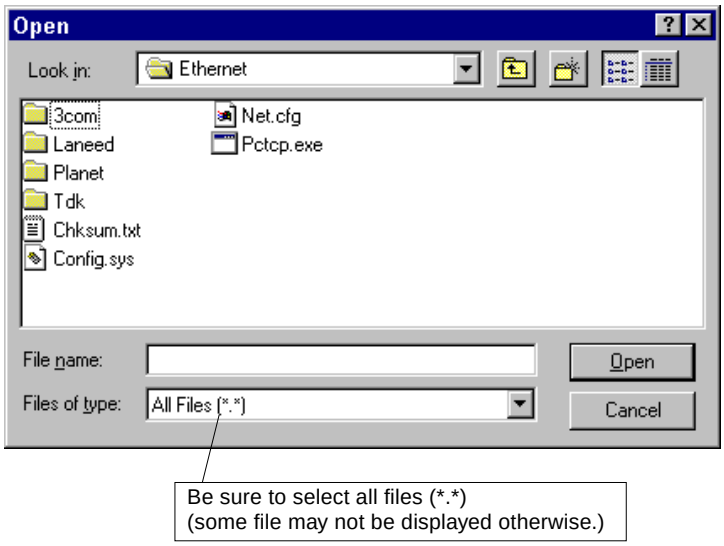
(5) Download files

■ Initializing data

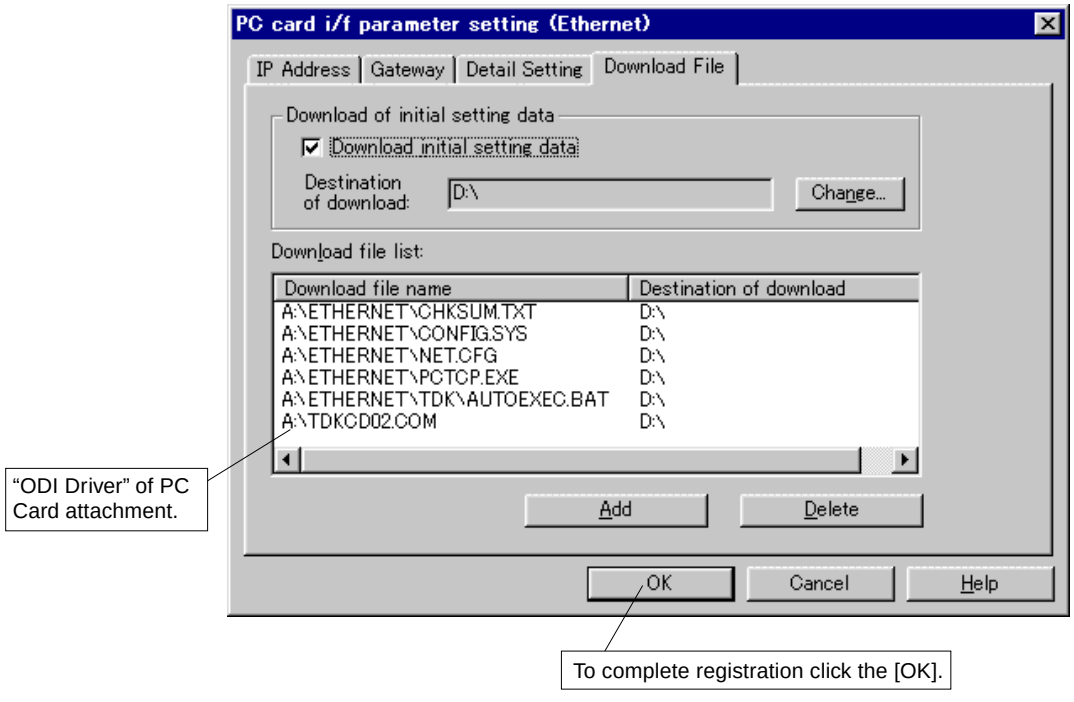


■ **Download Files**

Window shown below is displayed when the [Add] button is clicked on download file setting screen.



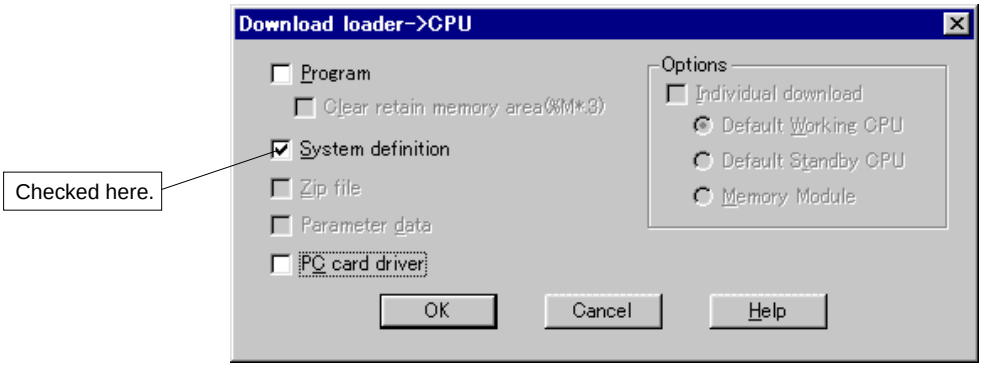
From floppy disk [PC Card Interface Driver] supplied with the unit, select files to be downloaded then click on [Open] button. The directory of the files will registered in Download File List area as shown in the figure below.
The DOS ODI Driver provided by the manufacturer of the Ethernet card must be downloaded. Copy the ODI Driver to a file folder on the hard disk or to the [PC Card Interface Driver] floppy disk prior to the downloading and register it to Download File List.



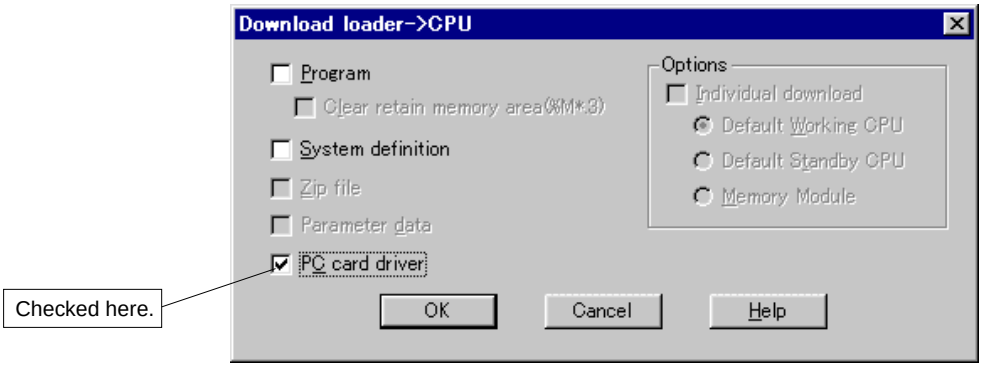
(6) Registering initial setting to NP1F-PC2

[Control] dialogue box

When [Download...] button is clicked, the [Download loader -> CPU] dialogue box will be displayed. (Please make PC a stopped state.)



First, [System Definition] is downloaded to make CPU recognize NP1F-PC2. (After downloading, reset CPU.)



Install PC Card Driver after CPU recognizes NP1F-PC2. Check the PC Card driver, and left-click the [OK] button.

During downloading, download status is displayed. If the downloading does not complete properly due to the error, retry downloading without turning the system power supply.

<Cautions>

- For PC Card Driver downloading, because it takes some time writing into the Interface Module, set Loader timeout to 10,000ms or longer as the temporary measure. After completion of the downloading, set the timeout back to original value (3,000ms).
- When Ethernet or MODEM functions are used, the card must be installed prior to the system power up.

5-3 Initial Setting for MODEM

To communicate through a MODEM installed in NP1F-PC2 Module, the module must be set up for the MODEM. For setting up the unit for the MODEM, Loader (D300win) version 1.2 or later is required.

5-3-1 Table of initial setting parameters

Following table shows items for initial setting.
Initial setting is made on parameter (MODEM function) option of the [Module property] dialogue box.

Item	Description	Initial value	Setting for recommended card	Comment
Port setting	Baud rate	19200bps	19200bps	Set according to the MODEM Card used
	Stop bit	1	1	
	Data length	8	8	
	Parity	None	None	
	Flow control	None	None	
AT command setting		AT&F	AT&F	
		ATE0	ATE0	
		AT¥N3	AT¥N3	
		ATS0=2	ATS0=2	
		ATS7=20	ATS7=20	
Download file	Initial setting download	No		See detail setting
	Download file list	No		

⇧
Use this setting.

Setting details

(1) Baud rate [Setting value: 2400, 4800, 9600, 19200, 38400 bps]
The communication speed is set.

(2) Stop bit [Setting value: 1, 2 bit]
This is the bit(s) to mark the end of a character.

(3) Data length [Setting value: 7, 9 bit]
Number of bits that forms one character.
When 7 is selected, one character is expressed with 7 binary bits.

(4) Parity bit [Setting: None, Odd, Even]
This is a bit added to data for the purpose of error detection.

(5) Flow control [Setting: No, Xon/Xoff]
The Computer or the MODEM has buffer memory where data received is kept temporarily during communication with other devices. The size of the buffer is limited and the MODEM automatically generates signal to request pause or restart of data transmission. This is called Flow Control.

(6) AT command setting
The AT commands for the MODEM control is set.
For the commands supported by the MODEM, refer to the manual supplied with the MODEM.

Meaning of AT commands
(The meaning of the command varies slightly among different models.)

Command	Function	Setting value
AT&F	Initialises memory contents	
ATE0	Selection of command echo function	Does not return echo
AT\N3	Sets MNP mode	Sets with priority to MNP auto selection mode
ATS0=3	Sets Number of rings to auto answer	After 3 rings, MODEM answers.
ATS7=20	Sets waiting time for carrier detection	Carrier waiting time is set to 20s

(7) Download initial setting data
Initial Parameters set in Port Setting and AT Command Setting are downloaded to [NP1F-PC2] when project is downloaded to CPU.
Download destination is specified to [D: \] (fixed). (This is due to the fact that Flash ROM in the [NP1F-PC2] is Drive D.)

(8) Download file list
Driver software for the PC Card supplied with the NP1F-PC2. Download it to the NP1F-PC2 using the D300win.

1) Autoexec.bat
2) Config.sys
3) Const.ini
4) PCTCP.exe
5) Delay.exe

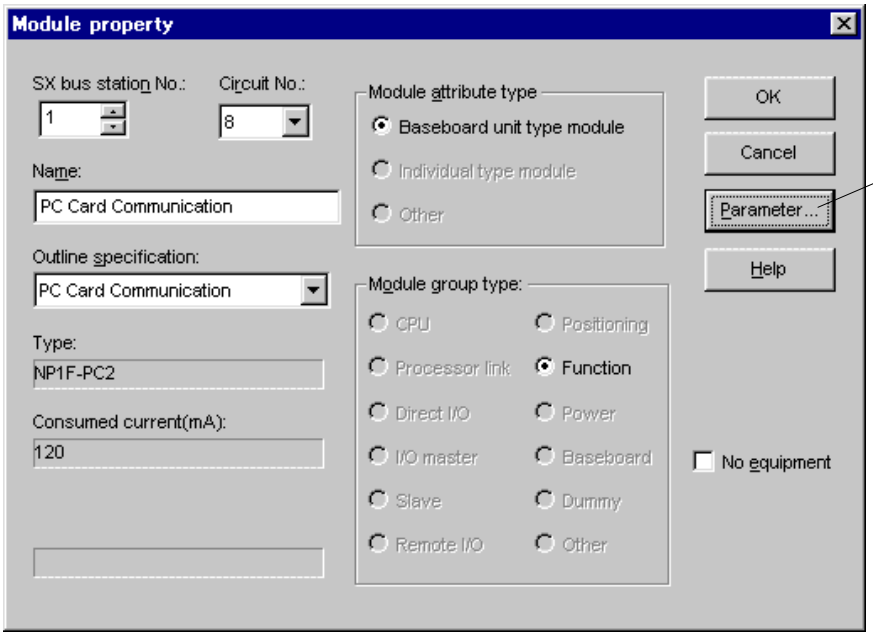
5-3 Initial Setting for MODEM

5-3-2 Procedure

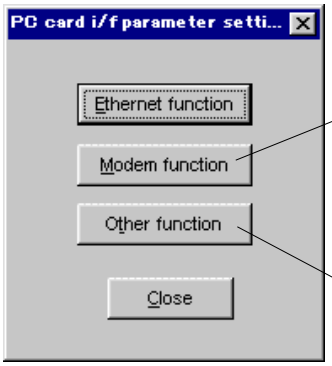
This section describes the Initial Setting Procedure for the MODEM.

(1) MODEM Parameter Selection

1) Click on [Parameter...] button on [Module property] dialogue box.

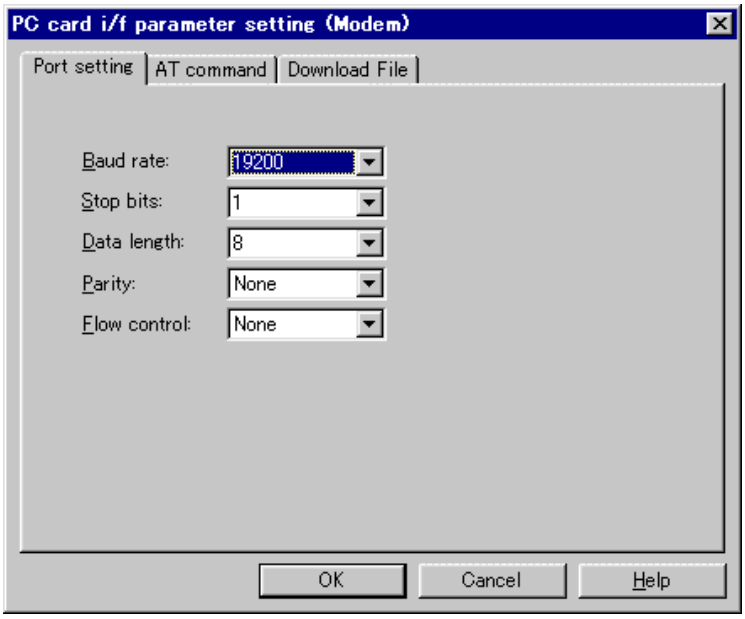


2) [PC card i/f parameter setting] dialogue box is displayed.

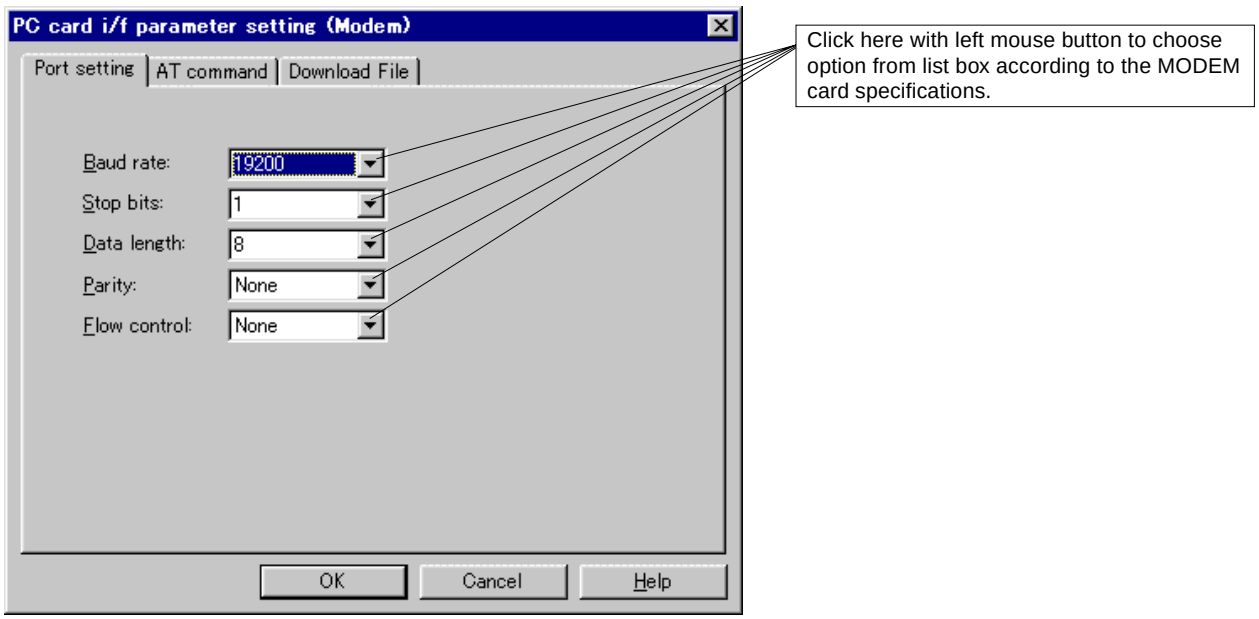


Note: Once MODEM function is selected, the NP1F-PC2 is limited to the MODEM function. (The dialogue box is not displayed hereafter.)
To modify the function, delete the NP1F-PC2 from system configuration definition and then perform registration operation again.

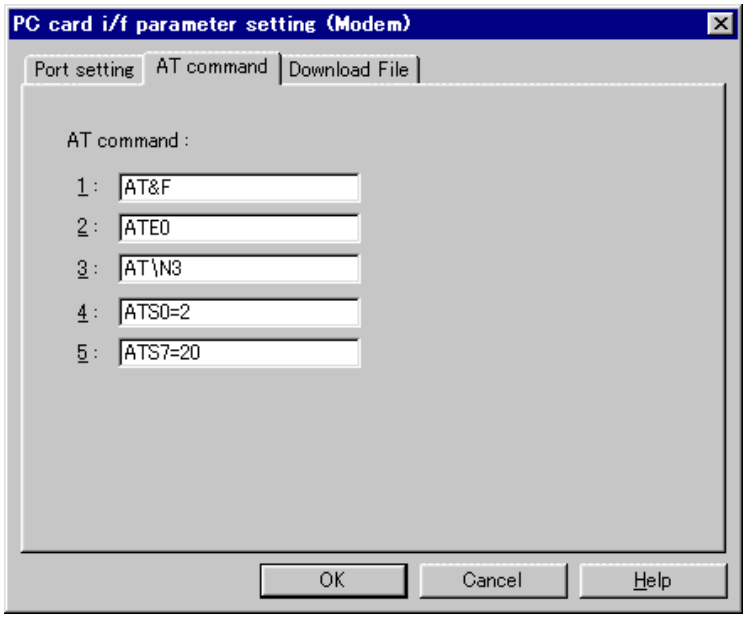
3) [PC card i/f parameter setting (Modem)] screen is displayed as the figure below.



(2) Port setting



(3) AT command setting



There are 5 command boxes. When more than 5 commands are set, write command one after another. In this case drop AT from 2nd command onwards.

Example:

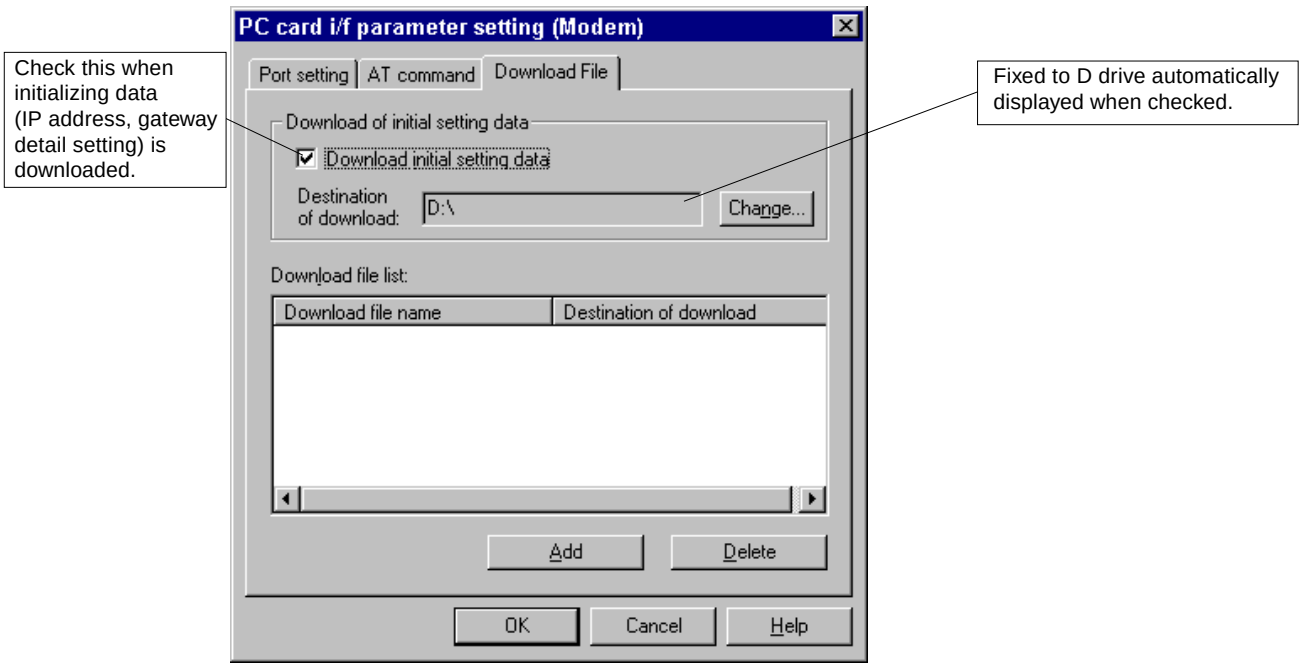
To write [ATV0] after [AT&F], it will be written as

AT&FV0

5-3 Initial Setting for MODEM

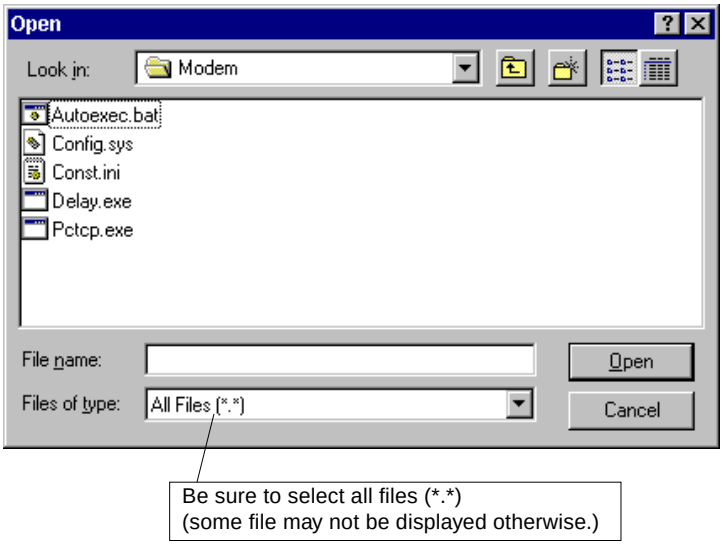
(4) Download files

■ Initializing data

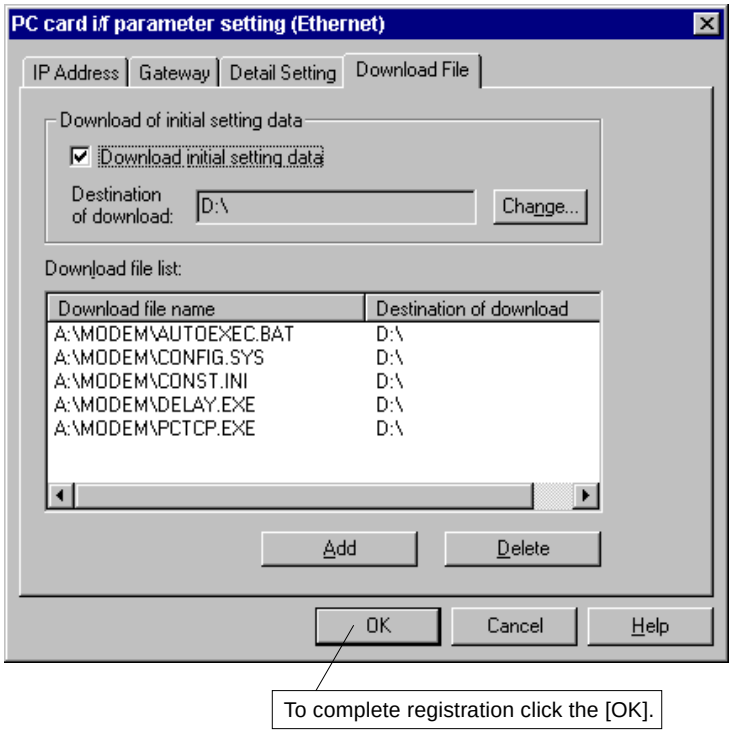


■ Download files

When you Click Add in download file setting screen and choose the MODEM folder, the following screen is displayed.



Choose the files to be downloaded (all files in MODEM folder) from supplied “PC Card Interface Driver” floppy disk and then click the [Open] button with left mouse button. The files to be downloaded are registered in Download File List as shown below.



(5) Registering initial setting to the NP1F-PC2

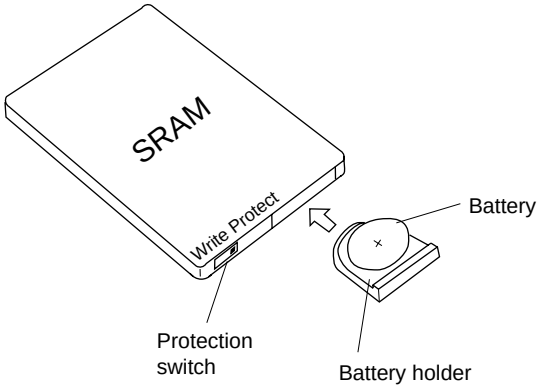
Refer to 5-2-2 (6). Registration procedure is the same as in case of the Ethernet Card.

5-4 Initialization of memory card

An SRAM card to be used must be initialized in MS-DOS FAT (File Allocation Table) format.

5-4-1 Preparation of SRAM memory card

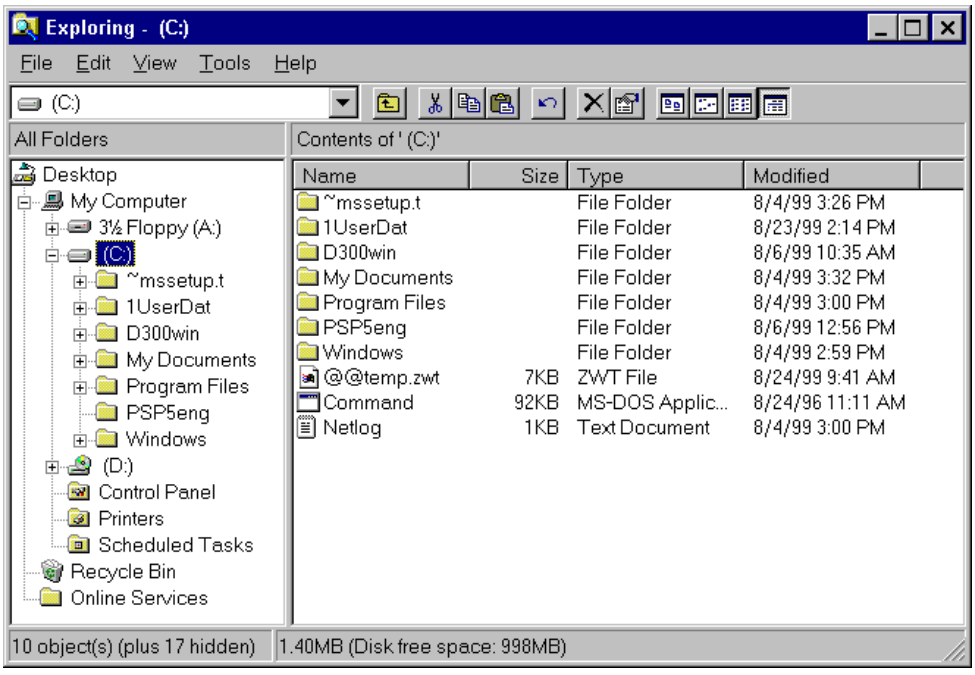
This preparation includes setting a battery in the SRAM memory card and releasing the memory protection switch.



5-4-2 Computer setting for using the SRAM memory card

For a computer with a PC card slot, first the software for using SRAM memory card needs to be registered in the “config.sys” file.

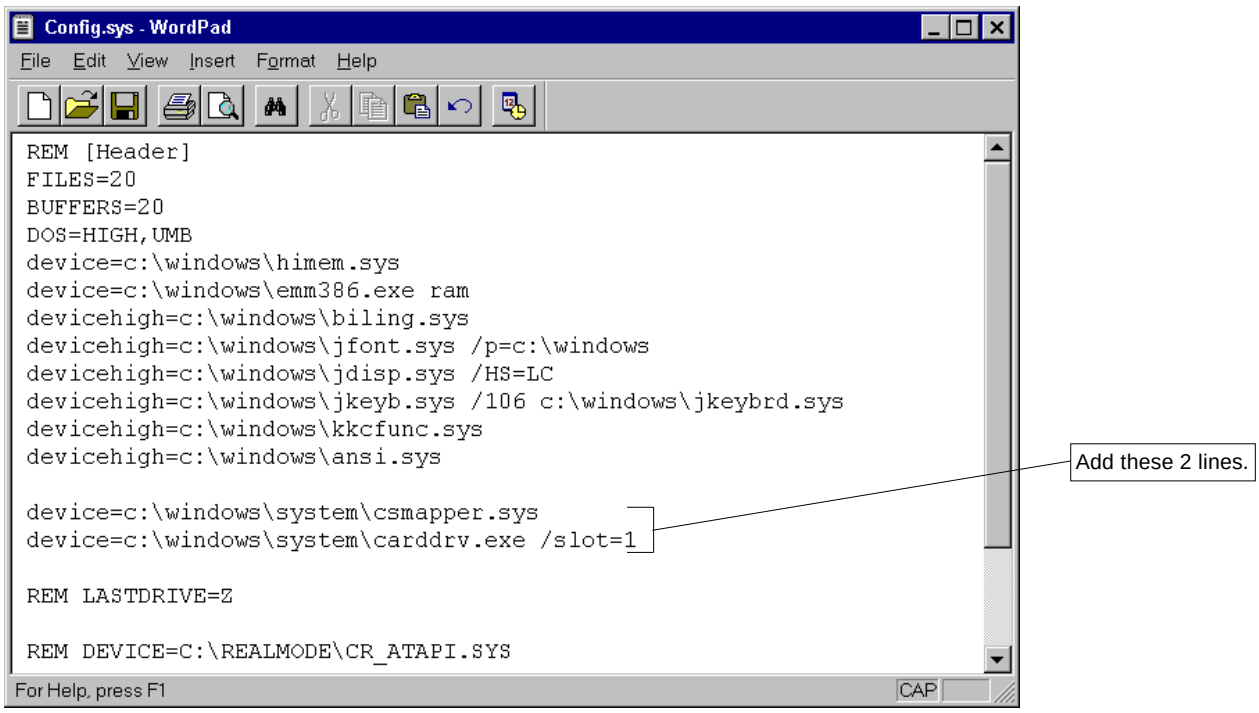
1) Check where “config.sys” file is located.
   In general, the “config.sys” file exists in the route directory for drive C (startup drive).



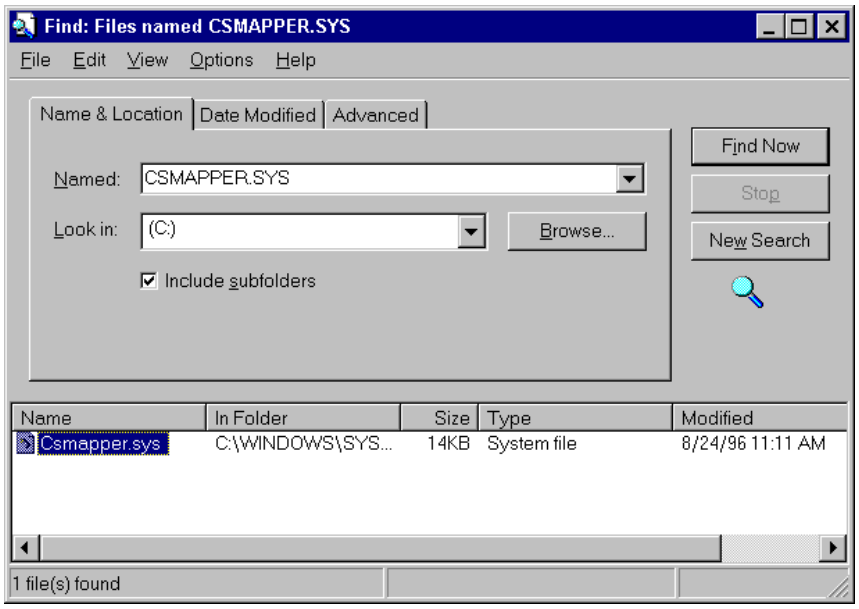
2) Open the “config.sys” file by a text editor such as a memo pad to add the following lines at the end of the file (when Windows is set up in the Windows folder in drive C).

```
device=C:\windows\system\csmapper.sys
device=C:\windows\system\carddrv.exe/slot=n
```

For the value of “n”, input the number of PC card slots (generally “1” or “2”) in the computer used.



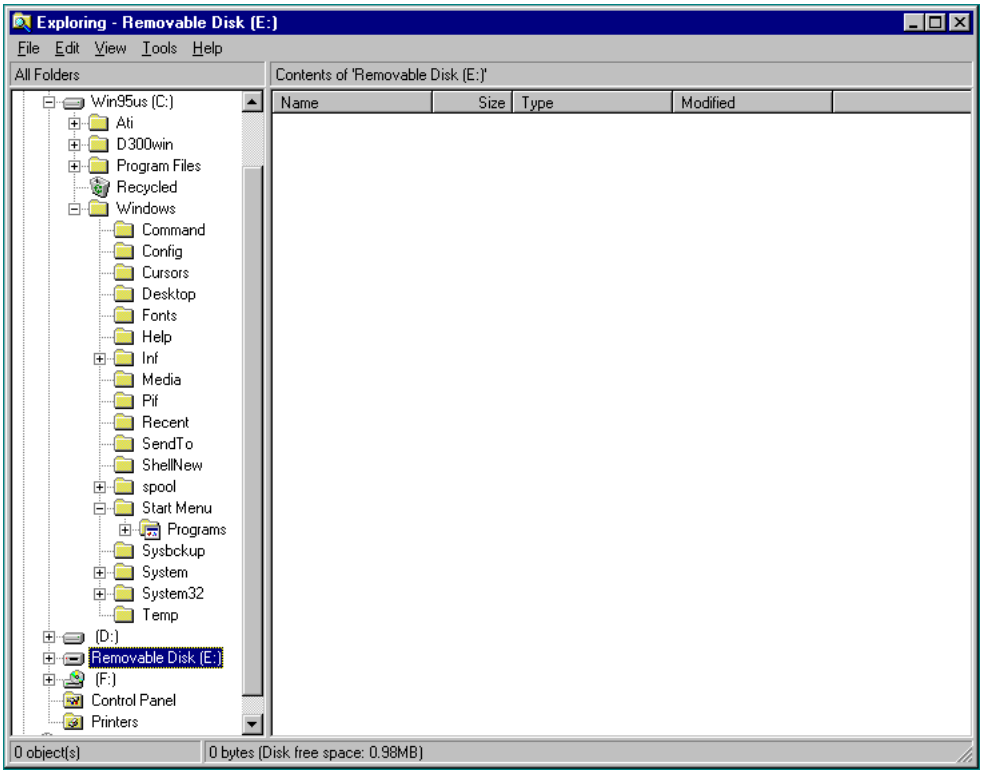
3) Overwrite the changes in “config.sys” file and store the data.
4) Make sure that files “CSMAPPER.SYS” and “CARDDRV.EXE” exist in the “WINDOWS\SYSTEM” folder. For this, right-click the [Start] command of Windows and left-click the [File and Folder] in the [Find] menu, specify “CSMAPPER.SYS” and “(C:)” in the [Named:] and [Look in:] boxes, and left-click the [Find Now] button. It is OK when the display becomes as follows.



5) Restart the computer.
This completes the setup for using the SRAM memory card.

5-4 Initialization of memory card

6) Insert the SRAM card in the PC card slot of the computer, and check the status of the microcomputer by the Explorer. The system is ready when the “Removable disk” is indicated in the window, as shown below. (This removable disk is the SRAM.)



5-4-3 Initialization of SRAM memory card

1) Make sure that the SRAM memory card to be initialized is inserted in the PC card slot of the computer.
2) Open the MS-DOS window by MS-DOS prompt of Windows95, and execute the file format command on the drive where the SRAM memory card is inserted.

<Sample input of format command when SRAM memory card is set on drive D>

Input “format d:” and press the “Enter” key.

C : \ > format ␣ d :

Format command — The drive name in which the memory card is set

After this, operate according to instructions on the screen.

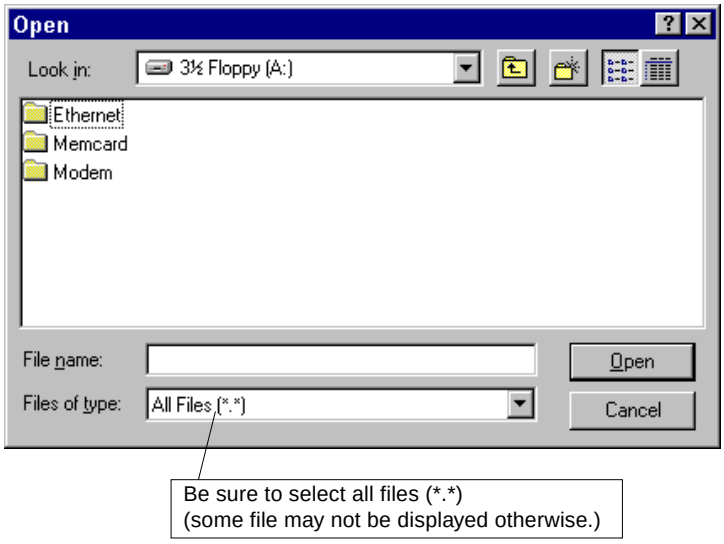
```
C:\WINDOWS>cd:\

C:\>format E:
Insert new diskette for drive E:
and press ENTER when ready...

Checking existing disk format.
Verifying 1.02M
Format complete.

Volume label (11 characters, ENTER for none)?

    1,036,800 bytes total disk space
    1,036,800 bytes available on disk

          512 bytes in each allocation unit.
        2,025 allocation units available on disk.

Volume Serial Number is 2874-19DF


C:\>
```

5-4-4 Download files

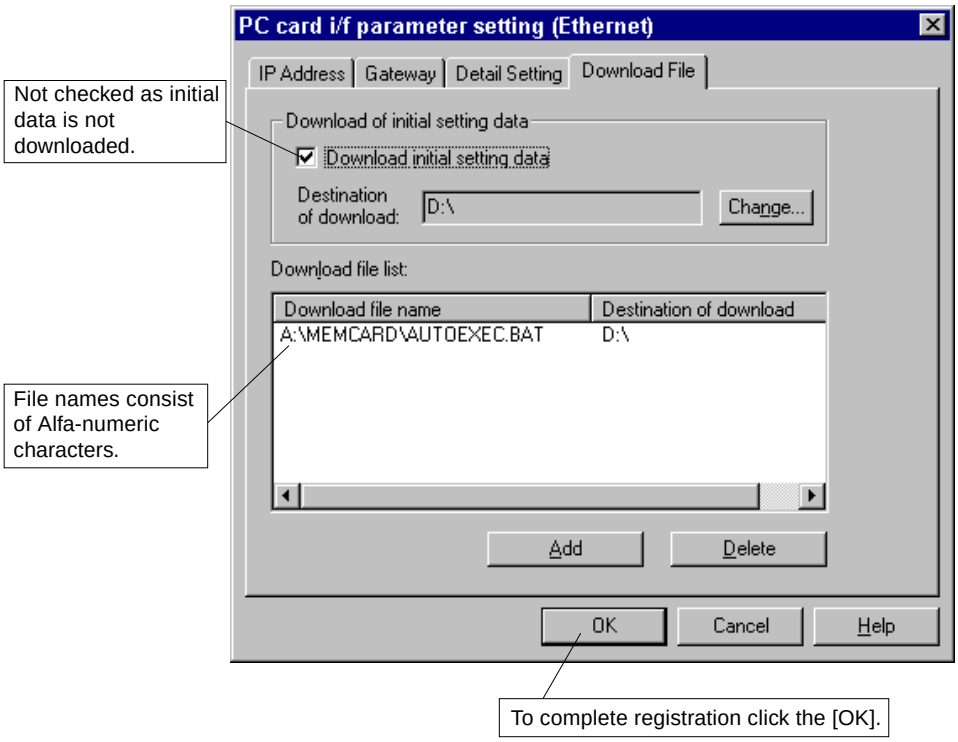
Memory Card is accessed through the driver downloaded to the Interface Module.
The procedure for downloading the files to the module is described in this section.

■ **Procedure**

For downloading memory card driver, [Download File] screen for the Ethernet Function is used. (Refer Section 5-2-2 for the procedure to bring this screen up.)
Click [Add] button on [Download File] screen. Following screen will be displayed.



Select Autoexec.bat file in Memcard Folder on the [PC Card Interface Driver] floppy disk supplied with the module then click on [Open] button. The directory of the file to be downloaded is registered in [Download File List].



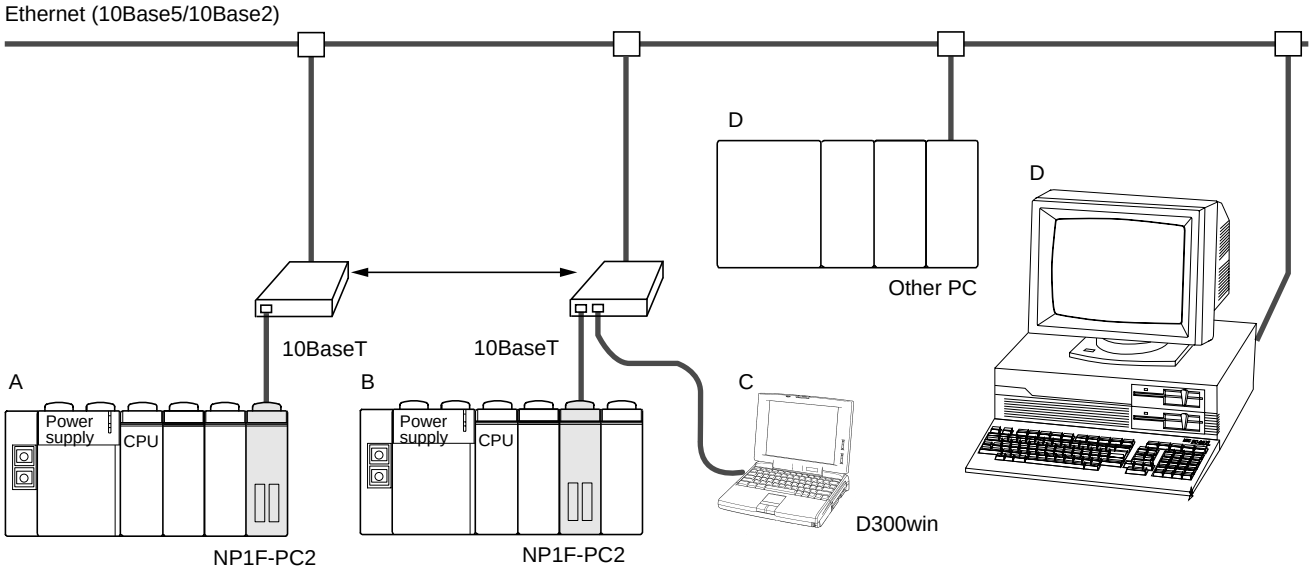
Section 6 Ethernet Function

Page

Section 6 Ethernet Function
6-1 Communication Modes

It is possible to communicate with maximum 8 nodes of Computers and PLC connected to the network.
Communication function supports following protocols.

Communication mode	Application communication mode		Loader command auto-receive mode
	Universal communication mode	Fixed buffer communication mode	Auto transmission mode
Protocol	TCP/IP, UDP/IP	Sub-command set in TCP/IP, UDP/IP data packet	TCP/IP
Communication FB	M_OPEN, M_SEND, M_RECEIVE		R-READ, R_WRITE



Communication subject		Communication mode
Own station	Other node	
MICREX-SA (A)	MICREX-SX (B)	Universal, Fixed buffer, Auto transmission
MICREX-SA (A)	Other PLC (D)	Universal, Fixed buffer
MICREX-SA (A)	D300win (C)	Loader command

6-1 Communication Modes

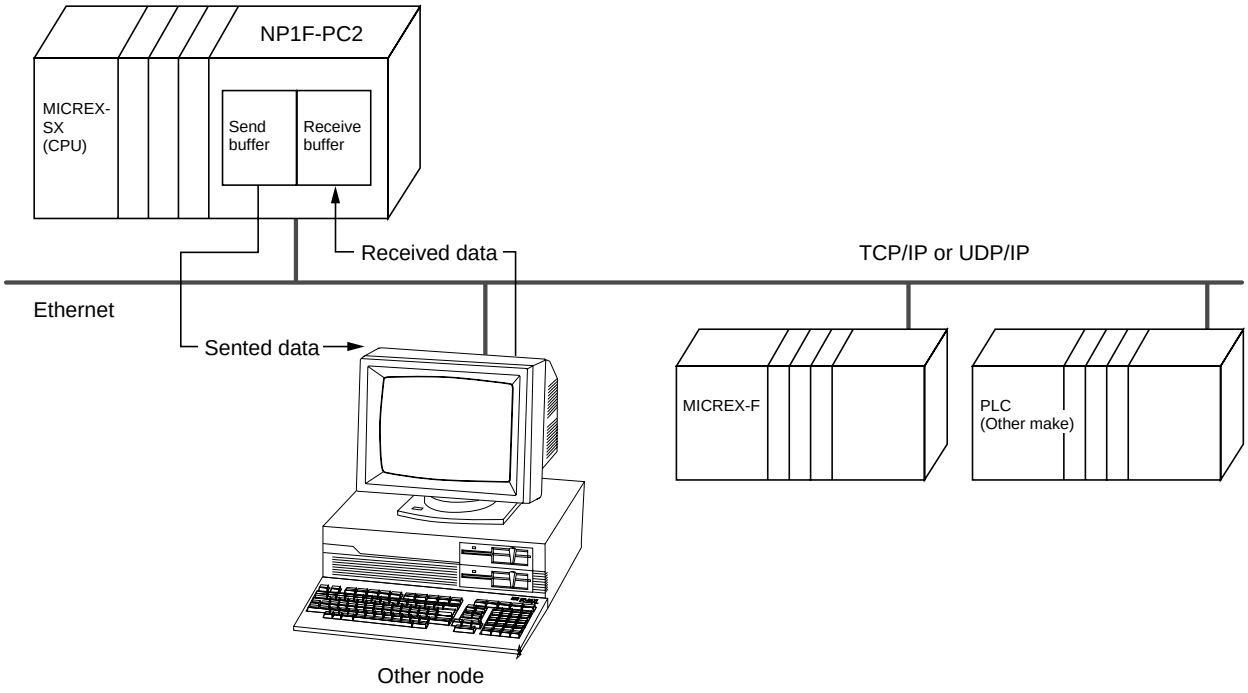
6-1-1 Universal communication mode

Communication with other node takes place using TCP/IP or UDP/IP protocols. Sending data to the other node is achieved by setting the data as variable of Message Send FB (M_SEND) and turning the Send Request signal ON (Data is sent at the positive transition of the Send Request signal.).
Turning Enable Receive signal of Message Receive FB (M_RECEIVE) ON allows reception of data from the other node.

The features of this communication mode are as follows.

- For there is no restriction to the data format, communication with any other node is possible.
- Communication with up to 8 nodes is possible.
- Maximum 1017 words can be sent/received.
- Communication with other Fuji Electric's PC is possible.
- Communication with PLC of other make also is possible.

Note: By adjusting the communication format, it is possible to communicate with Mitsubishi PLC.

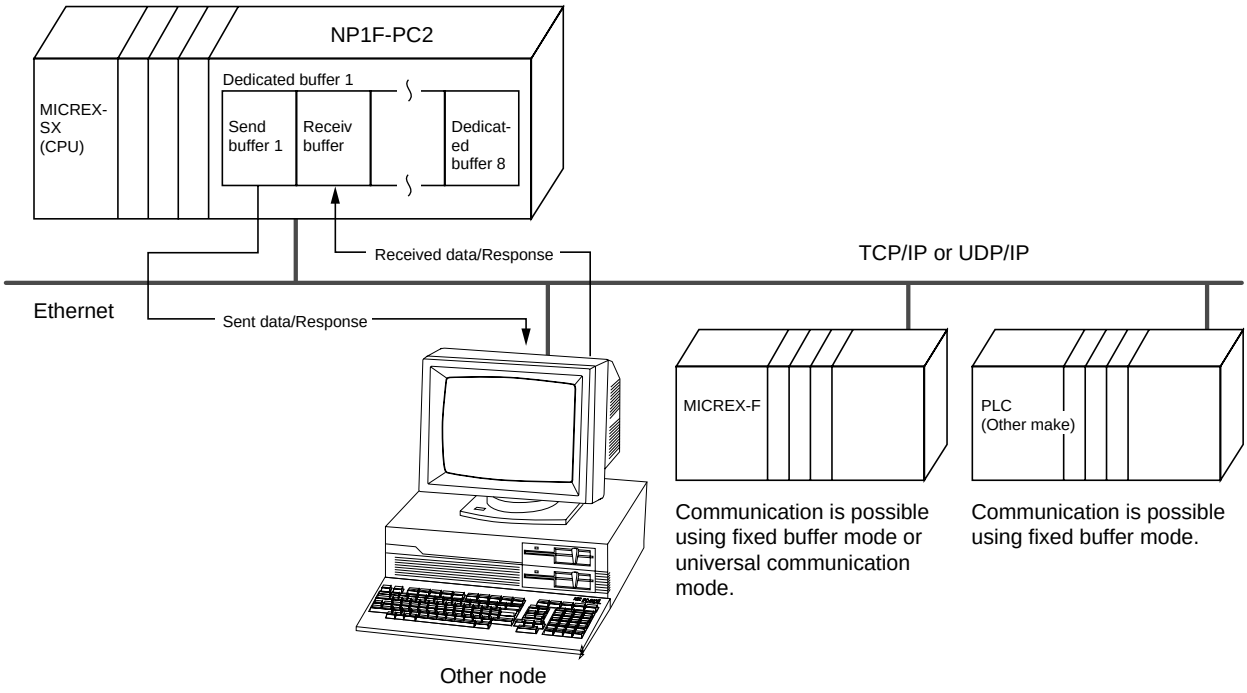


6-1-2 Fixed buffer communication mode

This mode provides the communication using fixed buffer, which Mitsubishi AJ71E71 Ethernet Interface Unit provides.
In this communication mode, sub-commands are embedded in data part of TCP/IP or UDP/IP in Universal Communication mode and provide command-orientated communication and controls handshaking for the communication with specified node.
In this mode communication with other Fuji Electric PC and Mitsubishi PLC is possible.

The features of this communication mode are as follows.

- Communication with up to 8 nodes is possible.
- Maximum 1017 words of data can be sent/received.
- Communication with other Fuji Electric PC is possible.
- Communication with Mitsubishi PLC is possible.



6-1-3 Auto transmission communication mode (For loader command)

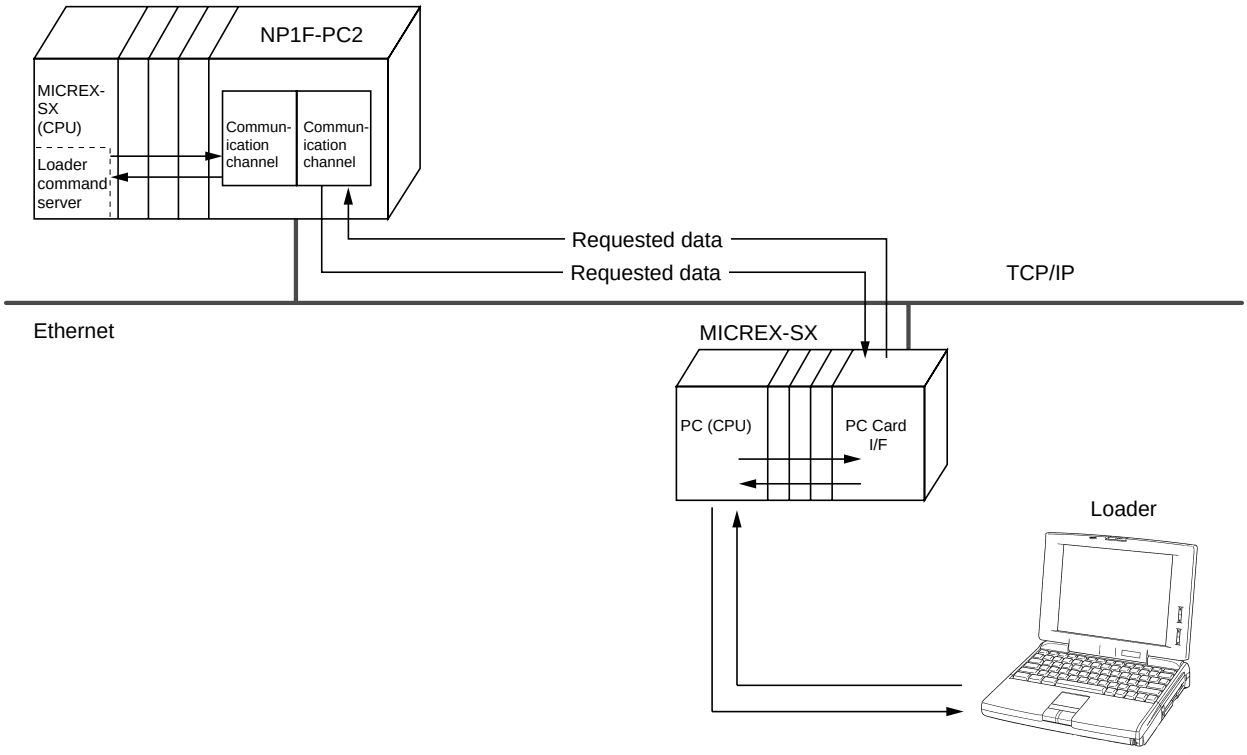
The NP1F-PC2 is provided with Auto Transmission Mode as a communication command (internal mode) on the SX Bus which allows transmission and reception of dedicated Loader commands between the MICREX-SX.
At the system boot up, MICREX-SX processes communication channel open (connection) to the all communication modules on the SX Bus so that PC (CPU) Loader Command Server Port (fixed port number) can receive data from the communication modules. And this enables the reception of loader commands from the network.
When Auto Transmission Communication Mode is specified, communication module automatically transfers relevant data received (at this stage, loader commands only) to the source of open (CPU) without Receive Request (M_RECEIVE) (R_READ, R_WRITE FB's are used.)

As the Ethernet Interface Function, open (connection) process described above enables reception of request from maximum 2 other nodes to Loader Server Port.
In this case, the condition of communication is as follows.

<Condition of communication>

- Communication Mode: Universal
- Communication Protocol: TCP/IP
- Open Method: Unpassive (Passive)
- Own Port No.: Loader Command Server (253) or Loader Interface Server (251)

Note: From other nodes, these port numbers will be the sum of (251 or 253) and port number reference number set as initial parameter (default 256).



Communicable Nodes in Each Communication Mode

Other node \ Own node			MICREX-SX		
			Universal	Fixed buffer	Auto transmission
MICREX-SX	Universal		O	O	-
	Fixed buffer		O	O	-
	Auto transmission		-	-	O
MICREX-F (FLEX-N)	Universal		O	O	-
	Command setting	Read	-	-	-
		Write	-	-	-
	Fixed/common	Fixed buffer	O	O	-
		Common buffer	O	-	-
	File transfer	Server FTP	-	-	-
		Client FTP	-	-	-
		Server TFTP	-	-	-
		Client TFTP	-	-	-
Mitsubishi	Fixed buffer		O	O	-
	Random access buffer		-	-	-
	Data Read/Write in CPU		-	-	-
Personal computer etc.	TCP/IP, UDP/IP		O	O	-
	File transmission	Server FTP	-	-	-
		Client FTP	-	-	-
		Server TFTP	-	-	-
		Client TFTP	-	-	-

6-1 Communication Modes

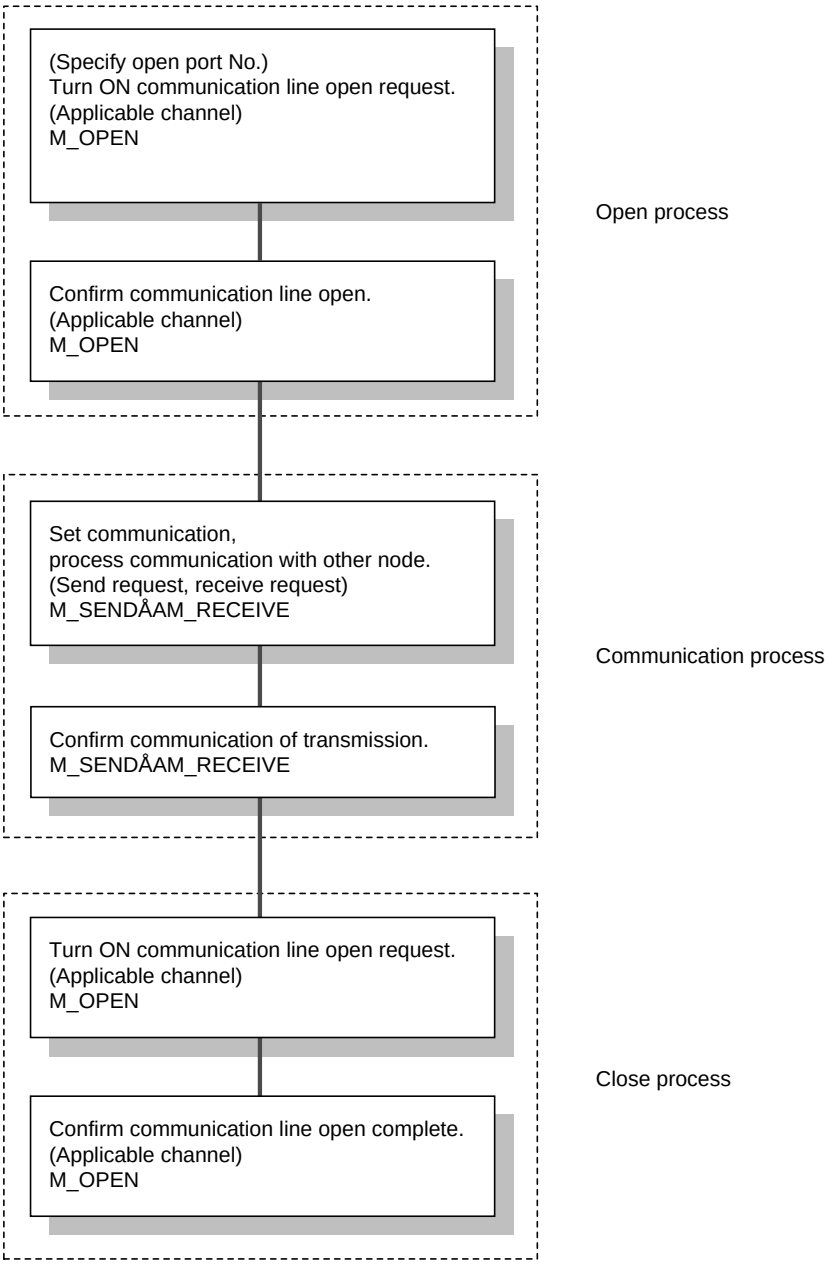
Comparison of communication modes

Communication mode			Universal	Fixed buffer	Auto transmission
MICREX-SX	Required sequence program		Send/Receive program	Send/Receive program	Send/Receive program
	Communication between CPU and this module		Message Send/Receive command (M_SEND, M_RECEIVE) Cannel open command (M_OPEN)	Message Send/Receive command (M_SEND, M_RECEIVE) Cannel open command (M_OPEN)	Remote Read/Write command (R_READ, R_WRITE)
Other node	Required program		Send/Receive program	Send/Receive program	
Communication	Max. data length *1	Binary	2034 Bytes	2038 Bytes	No limit
		ASCII	4068 Bytes	4076 Bytes	
	Protocol		TCP/IP or UDP/IP	TCP/IP or UDP/IP	TCP/IP

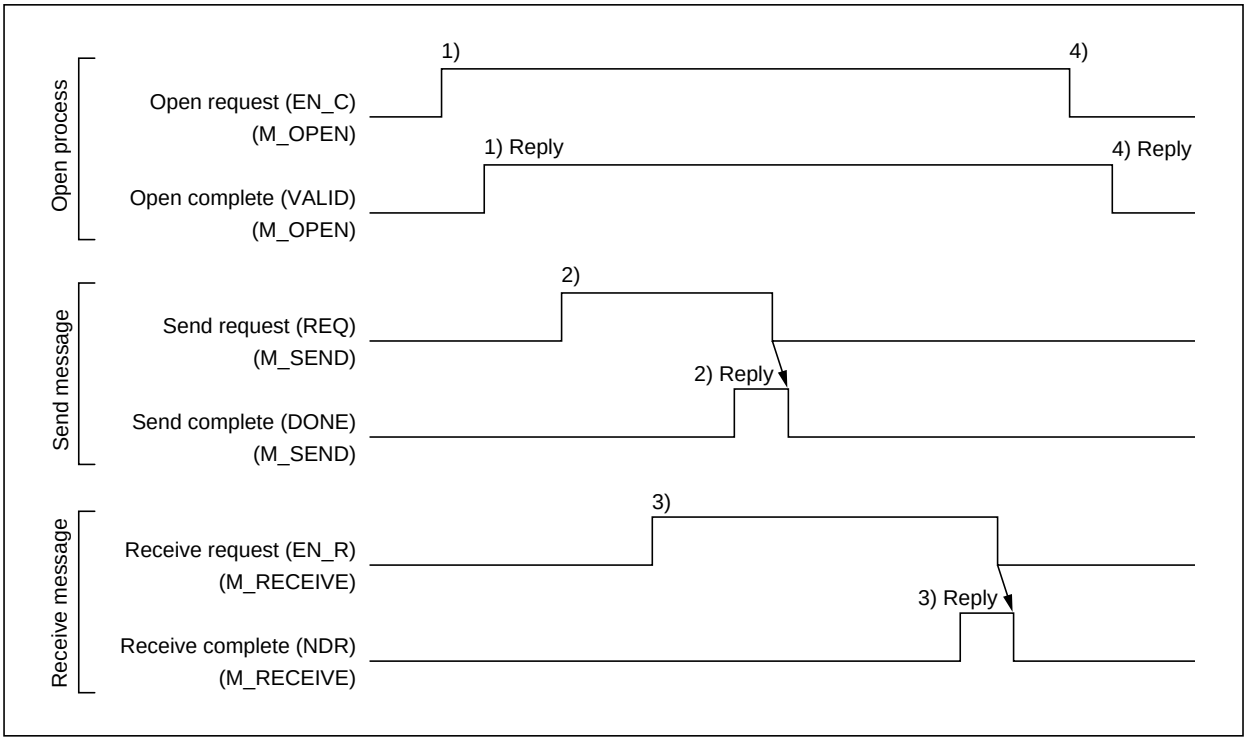
* 1 Maximum length of TCP/IP or UDP/IP data part.

6-2-1 Outline of communication procedure

PC (CPU) uses communication FB [Channel Open (M_OPEN)] and [Message Send/Receive (M_SEND/RECEIVE)] in order to communicate with other configuration through network. PC (CPU) establishes communication line to the other node (configuration) through open process. It can communicate only with the nodes with which the communication channels are opened.
To terminate data communication, close process is executed. The close process disconnects communication line.
As there is limitation in number of communication lines opened at any one time, PC (CPU) repeats open and close processes according to the demand and achieves communication with numerous nodes.
As information to specify the node (IP Address, Station No. Etc.) is set at the time of communication line open, both send and receive data procedures become the same.
In message communication, the node for the communication is specified with the logical address, port number.



6-2 Preparation for Communication with Other Node (Message Communication)



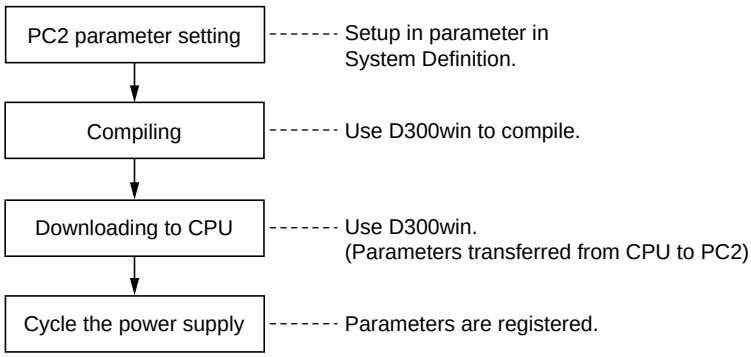
6-2-2 Initial process

(1) Parameters for Initial Setting

The parameters for the initial setting are set up in PC2 Parameter section of [System Definition]. (Refer Section 5 for details.)

(2) Procedure for Initial Process

Procedure of the initial data process is as follows.

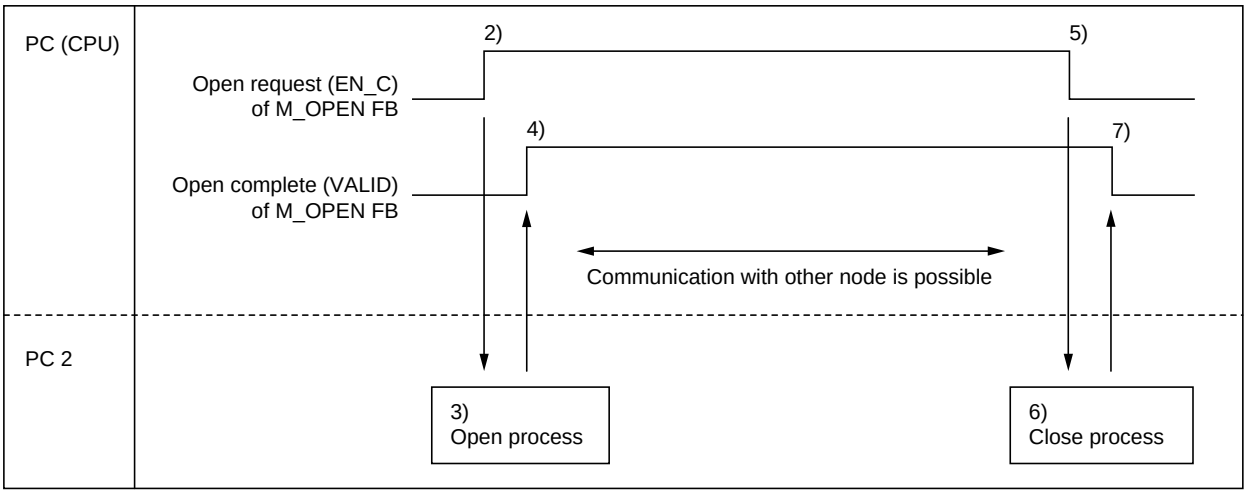


6-2-3 Communication line open and close

The Channel Open (M_OPEN) FB is used for opening of communication line.
Set up the parameters required for open process and execute the FB.
For details of M_OPEN FB, see Section 6-6-1.

• Open and close procedure of the [NP1F-PC2]

In order to execute open process, the initial process must be completed.



1) Write parameters required for Open Process into M_OPEN FB.
2) Turn ON Open Request (EN_C) signal.
3) The module's open process takes place at positive edge of Open Request (EN_C).
4) Open Complete (VALID) signal of M_OPEN FB turns ON when open process complete successfully.
5) Turn OFF Open Request (EN_C) of M_OPEN FB to terminate the communication with the other node.
6) The module's close process takes place when Open Request (EN_C) turns OFF.
7) Open Complete (VALID) turns OFF when close process completes.

- In TCP/IP protocol communication, if close process does not complete after time period specified with TCP End Timer, process RST to the other node in order to force line closure. Open Complete (VALID) turns OFF when close process completes.
  However, if close process is executed during transmission, line is closed after completion of transmission (time specified with TCP Timeout).
- In TCP/IP protocol communication, besides line closure with M_OPEN FB (turn OFF Open Request), the communication line is automatically closed in following cases. In order to reopen the line, turn OFF Open Request (EN_C) once and retry the open process.

Causes of close process	Explanation	Error code (Detail RAS)
TCP send tiomeout error	During TCP protocol communication, ACK was not received after specified number of retries. Transmission timeout.	405□h
Response watchdog tiomeout error	Reception timeout.	302□h
Close request from other node	During TCP protocol communication, "Close" or "ABORT" command was received from other node.	-

Note: Refer to Section 12-2 RAS Details for details.

6-2 Preparation for Communication with Other Node (Message Communication)

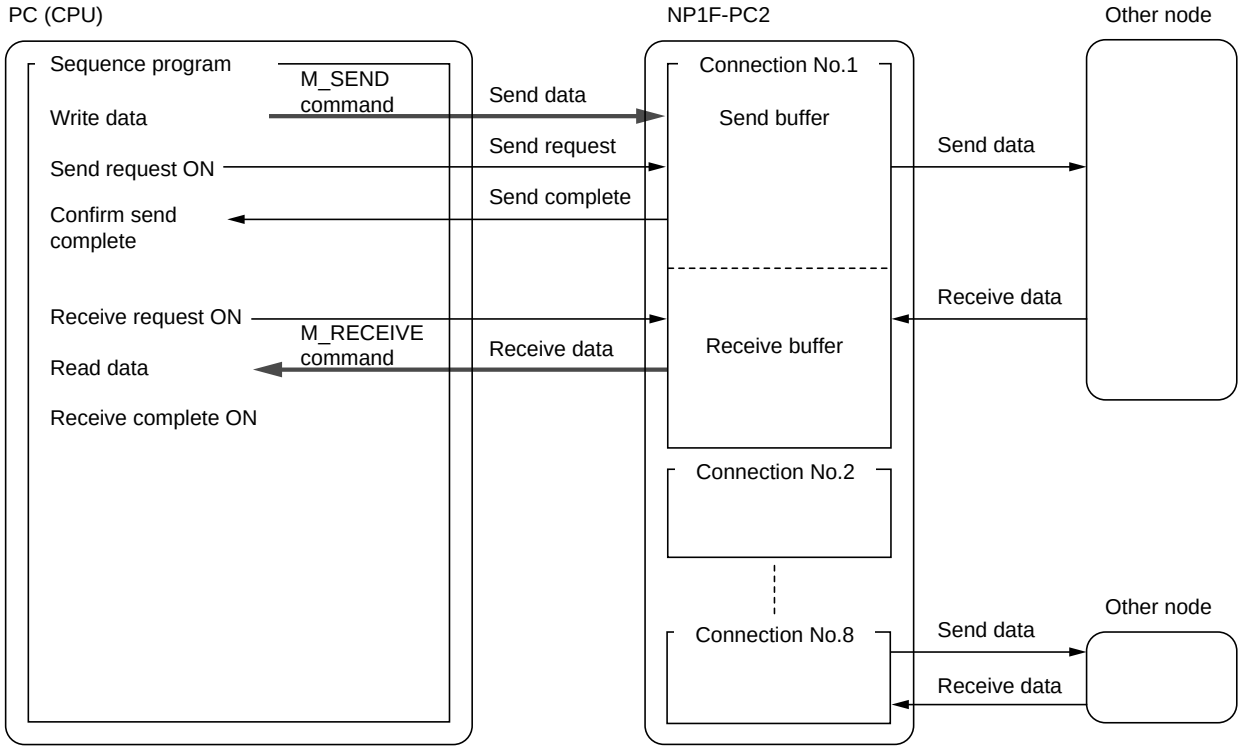
• Relation between Connection Open Method and Communication Parameters

Universal, Fixed Buffer Communication

			Port No. of this module	Other node's IP address	Other node's Port No.	Other node's Ethernet address	User name	Password	Sever FTP user list
TCP	Active	Other node has ARP function	Setting required	Setting required	Setting required	FFFFFFFFFFFF	Not required	Not required	Not required
		Other node does not have ARP function				Setting required			
	Passive	Unpassive	Setting required	Not required	Not required	Not required			
		Fullpassive	Setting required	Setting required	Setting required	Not required			
UDP		Other node has ARP function	Setting required	Setting required	Setting required	FFFFFFFFFFFF			
		Other node does not have ARP function				Setting required			

6-3-1 Outline of communication

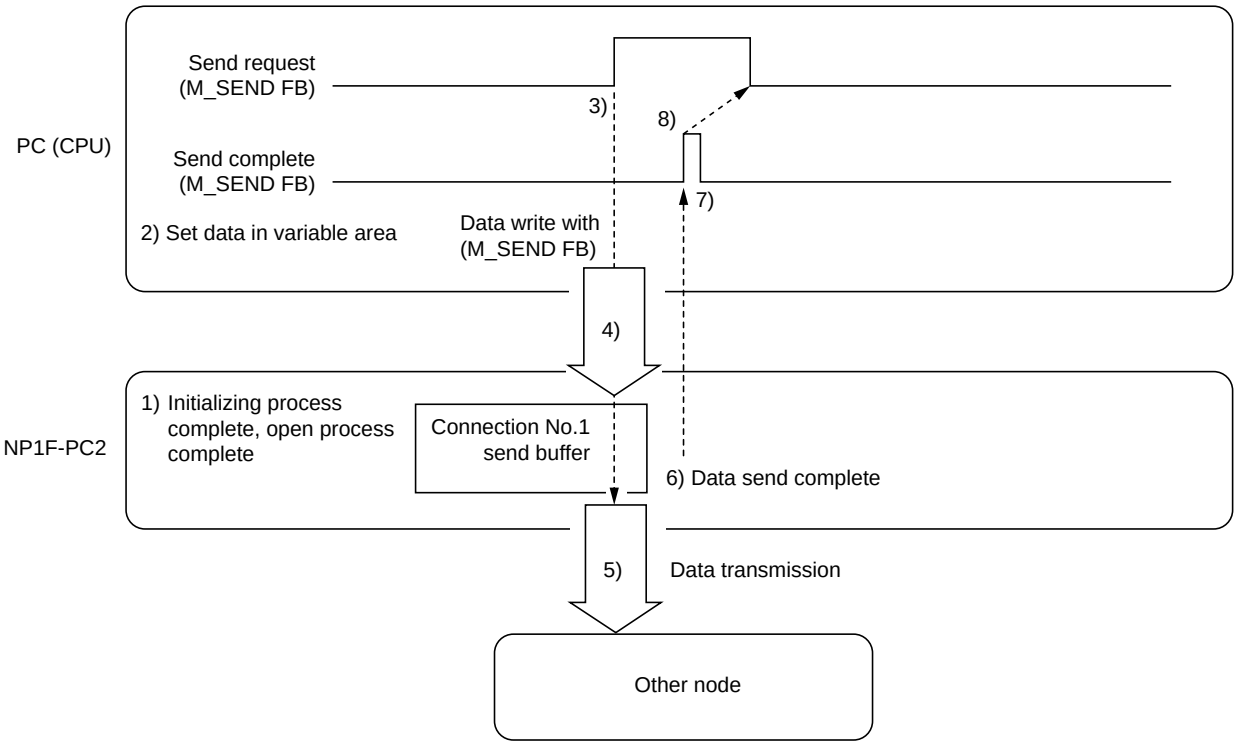
Data flow



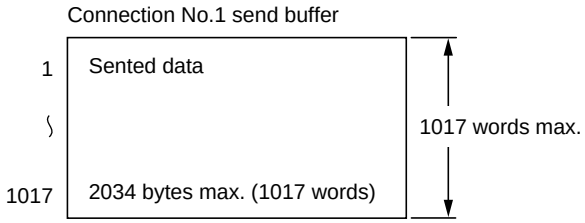
6-3-2 Sending data

Message Send (M_SEND) FB is used to send data to the other node.
The parameters necessary for the transmission is set for the (M_SEND) FB and execute it.
The detail of M_SEND FB is described in Section 6-2-2.

■ **Procedure for sending data**



1) Execute NP1F-PC2 Initializing Process and Open Process with other node.
2) Set up data to be sent in the variable data area specified by M_SEND FB.
3) Turn ON Send Request signal of M_SEND FB.
4) Data is transferred to NP1F-PC2 Send Buffer as follows.



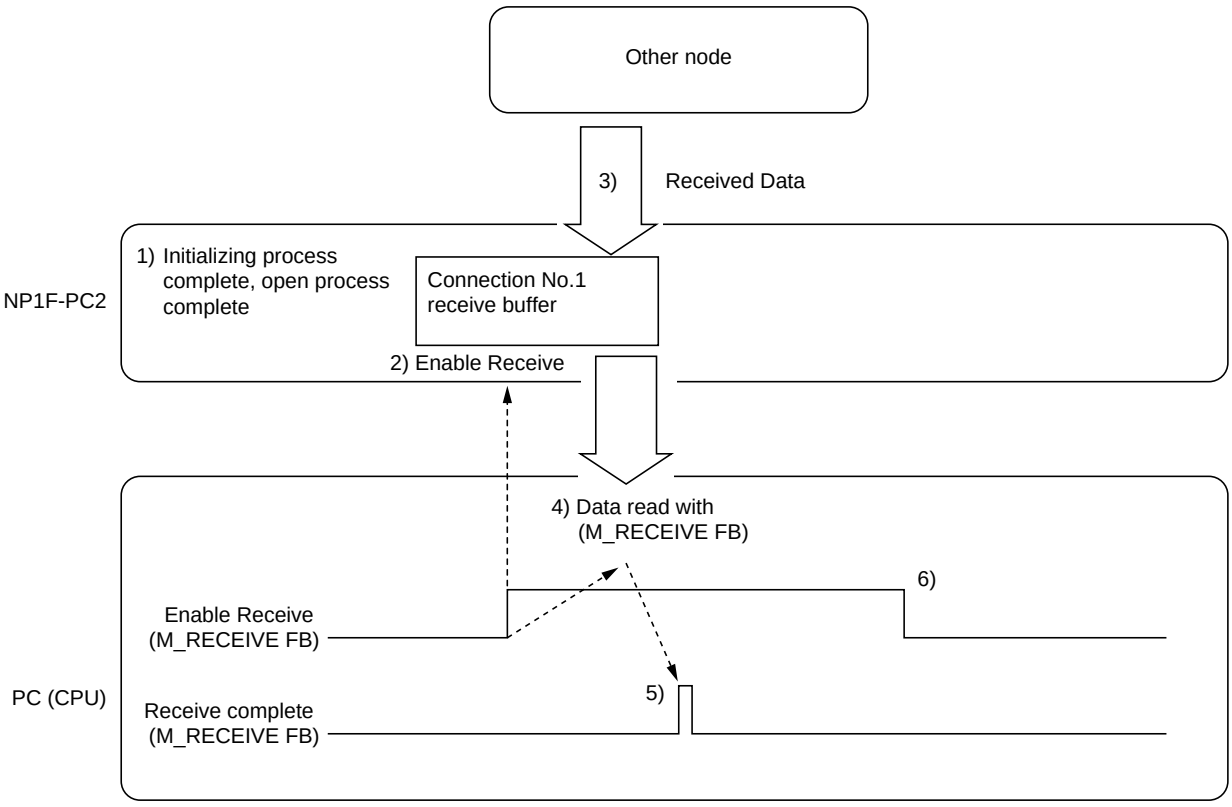
5) Data is sent to the specified node from Send Buffer of the relevant connection.
6) Send Complete signal is returned when data transmission is completed.
7) CPU turns ON Send Complete signal (DONE) of (M_SEND) FB.
8) Turn Send Request signal OFF, after confirming M_SEND FB Send Complete signal is ON.

- If the transmission of data is not completed successfully, the Send Error signal turns ON. In this case, turn Send Request signal OFF then repeat from step 3.

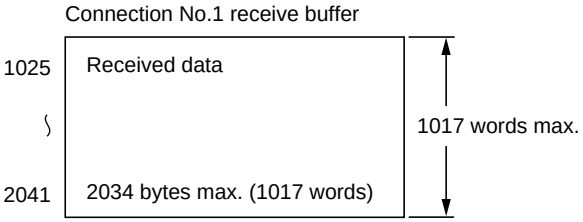
6-3-3 Receiving data

Message Receive (M_RECEIVE) FB is used for receiving data from other node.
Set parameters required for this (M_RECEIVE) FB to receive data then execute the (M_RECEIVE) FB.
(M_RECEIVE) FB is detailed in Section 6-6-3.

■ **Procedure for receiving data**



1) Execute NP1F-PC2 Initializing Process and Open Process with other node.
2) Put in “Waiting” state by turning Enable Receive (EN_R) signal ON.
3) Receive data from other node into the relevant Receive Buffer (decided by the parameter setting at the line open). Received data is stored in Receive Buffer as shown below.

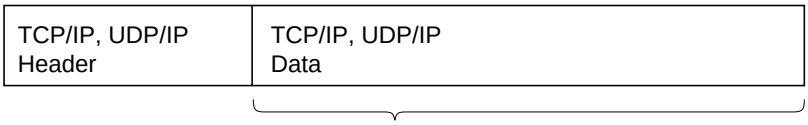


4) Read data out with M_RECEIVE FB.
5) Receive Complete (NDR) pulse is generated when data read is completed with (M_RECEIVE) FB.
6) Turn Enable Receive (EN_R) signal OFF to terminate receiving state.

- When abnormal data is received, Receive Complete (NDR) does not turn ON. And data is not stored in Receive Buffer.

6-3-4 Data format

Data Sent/Received by other node



(1) Header

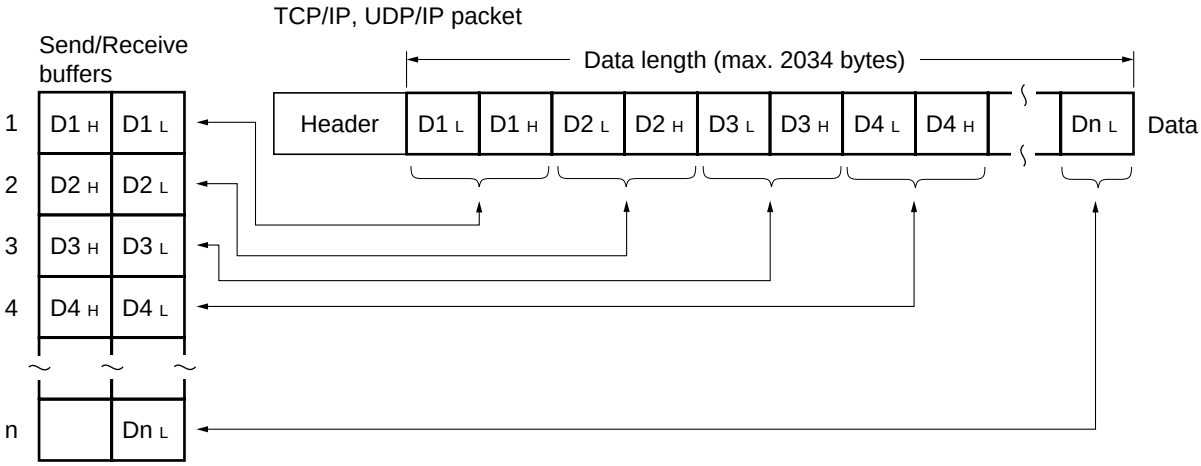
The header is the same as Ethernet TCP/IP or UDP/IP header. When data is sent to other node from the application program, the header is automatically added by the NP1F-PC2. When data is received from other node, the header is removed and only data part is passed to the application program.

(2) Data

The data part is the same as Ethernet TCP/IP or UDP/IP data. When data is sent from the application program, the data transferred into the NP1F-PC2's send buffer is set to the TCP/IP or UDP/IP data and sent out. In binary code transmission, the data in the buffer is sent as it is. And in ASCII code transmission, the data in the buffer is converted to ASCII character and sent.

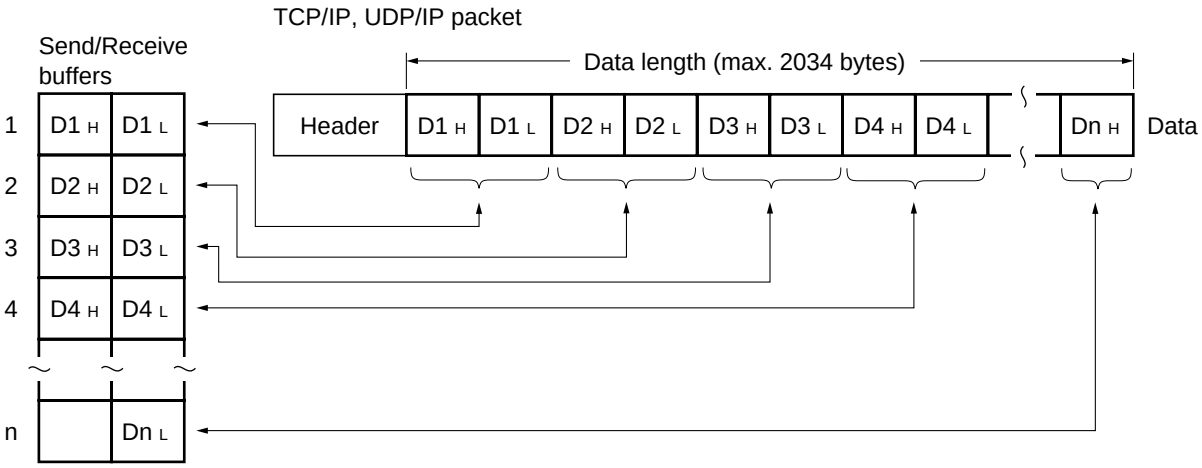
When data is received from other node, data part of TCP/IP or UDP/IP is stored in Receive Buffer of this module. In binary code transmission, data received is stored as it is and in ASCII code transmission the ASCII character data is converted to binary before stored in the buffer.

Binary code (Data swap not specified)



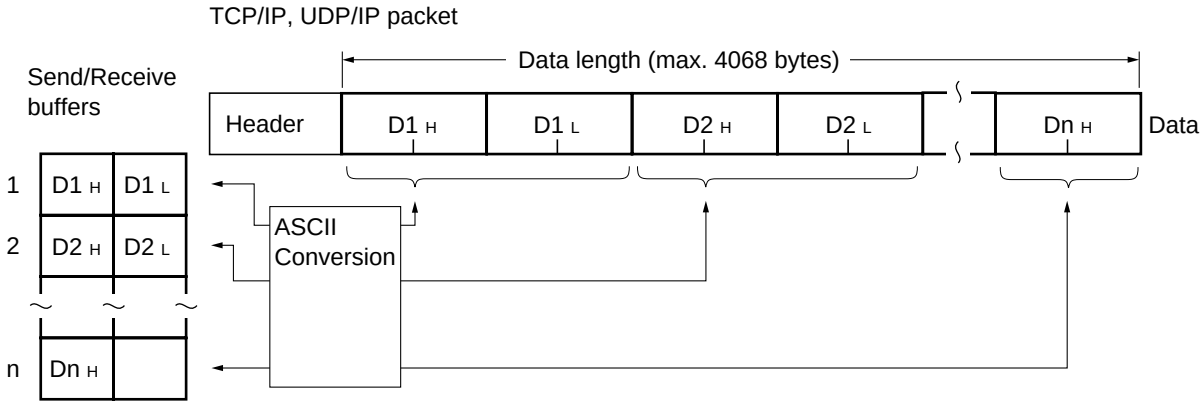
Data length is expressed in decimal as number of data byte stored in Send/Receive buffers (Maximum 2034 bytes). When data length is odd number, up to the lower byte of last data word are the valid data.

Binary code (Data swap specified)



Data length is expressed as number of data bytes stored in the buffer in decimal (Maximum 2034 bytes).
When data length is odd number, the valid data is up to the upper byte of the last data word.

ASCII code

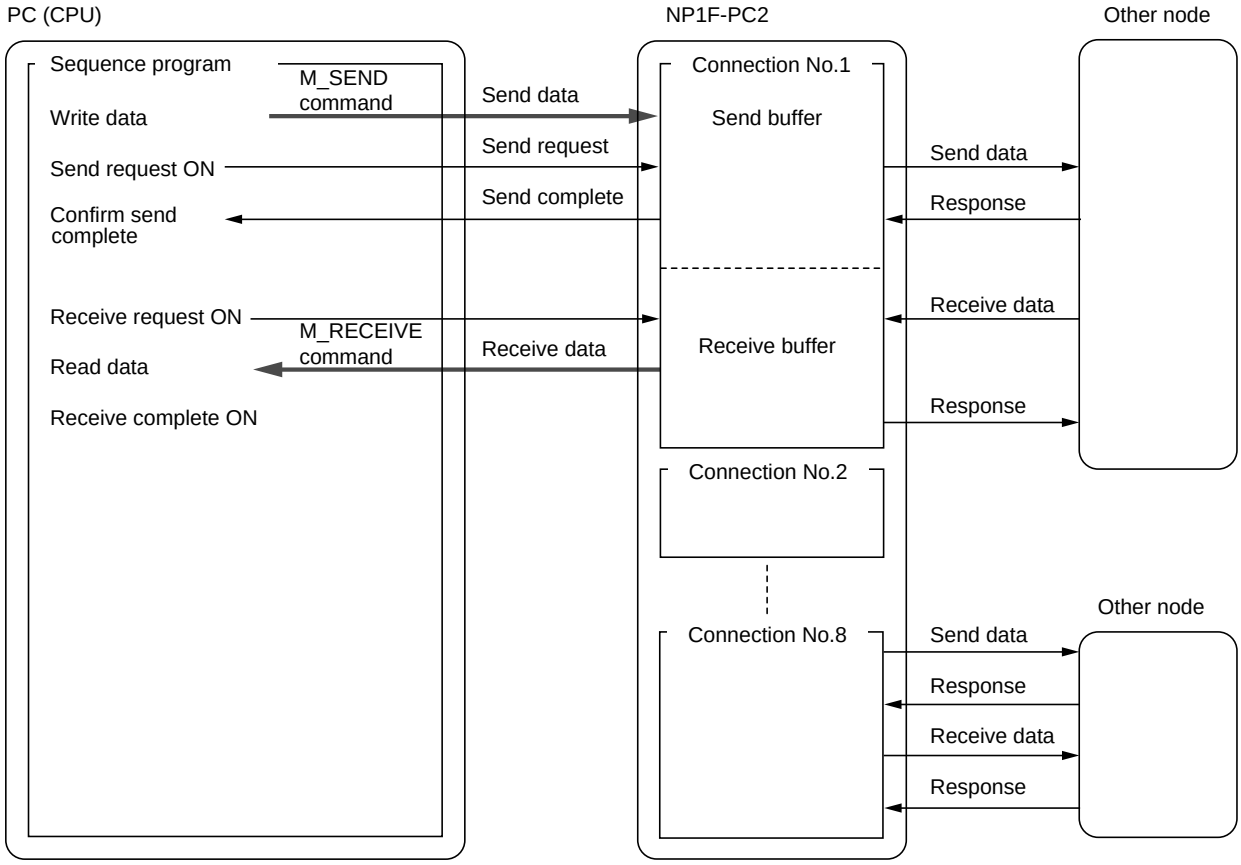


Data length is expressed as number of data bytes stored in the buffer in decimal (Maximum 4068 bytes).
When data length is odd number, the valid data is up to the upper byte of the last data word.

6-4 Fixed Buffer Communication Mode

6-4-1 Outline of communication

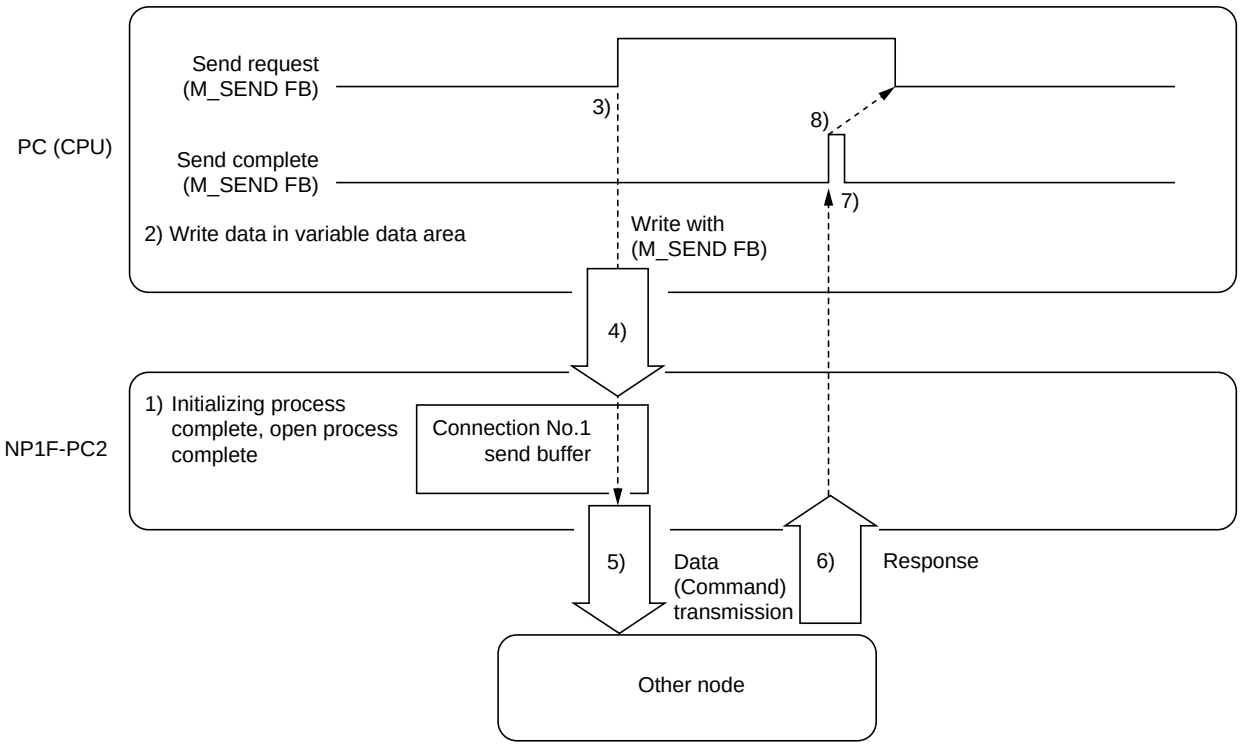
Data flow



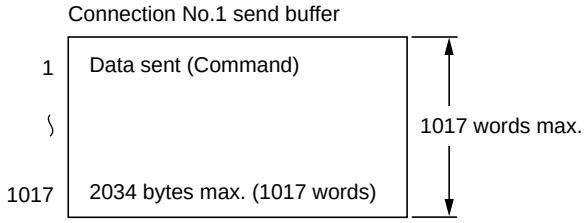
6-4-2 Send procedure

When data is sent to other node, Message Send (M_SEND) FB is used. The parameters required for transmission is set for the (M_SEND) FB then the (M_SEND) FB is executed.
Detail of the (M_SEND) FB is described in Section 6-6-2.

■ **Procedure for Sending Data**



1) Execute Initializing Process of this module and Open Process of the communication line with other node.
2) Set up data in variable data area specified by (M_SEND) FB.
3) Turn ON Send Request signal of (M_SEND) FB.
4) Data is transferred from the variable data area to NP1F-PC2 Send Buffer in following format.



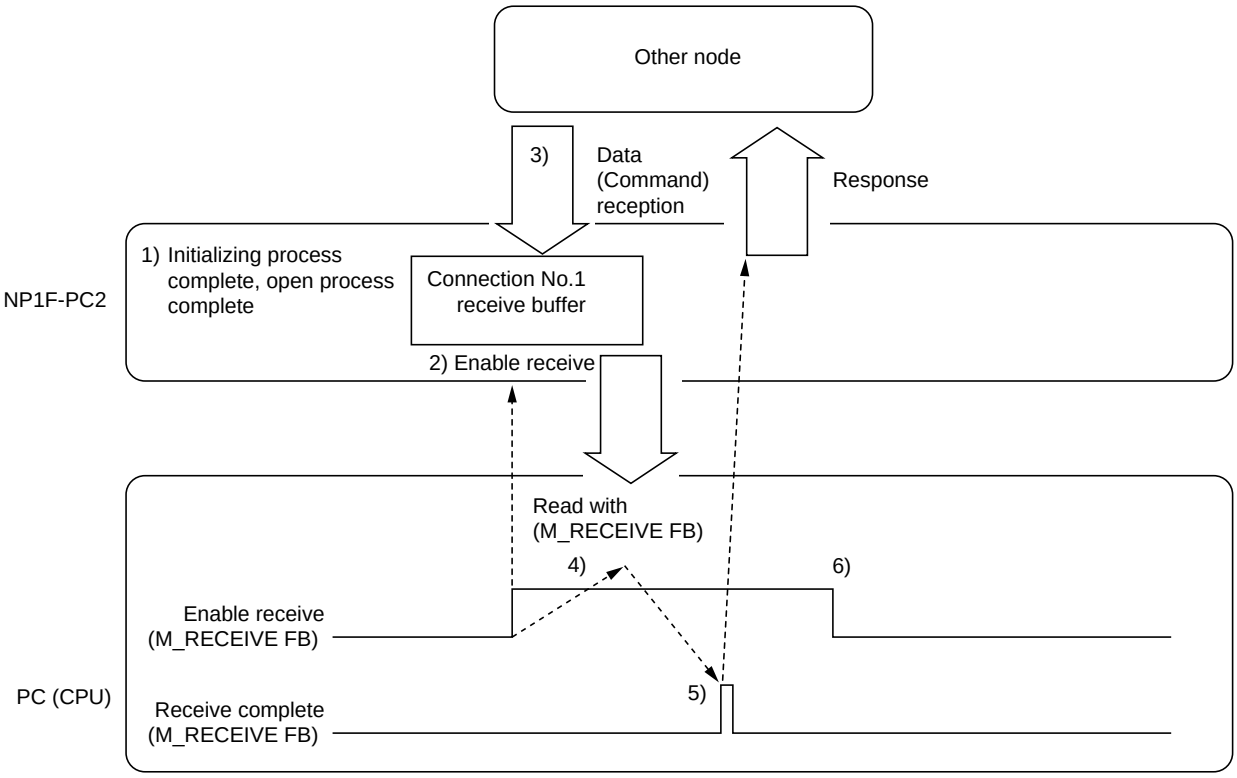
5) Data transmission takes place from relevant Send Buffer to specified node.
6) Specified node returns response when it receives data from NP1F-PC2.
7) CPU turns ON Send Complete (DONE) signal when it receives response from the other node.
8) Confirm Send Complete signal and turn OFF Send Request (REQ).

- When the response is not received or End Code of the response is not 00h, Send Error (ERROR) turns ON and Send Complete (DONE) stays OFF.
  In this case, turn OFF Send Request (REQ) and repeat from step 3.

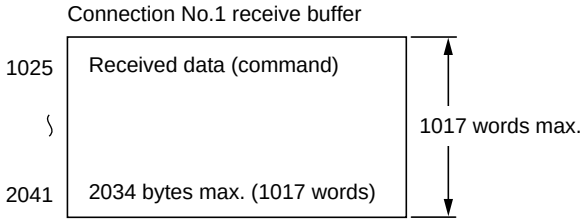
6-4-3 Receive procedure

Message Receive (M_RECEIVE) FB is used for receiving data from other node.
Set parameter required for the FB to receive from other node and execute the (M_RECEIVE) FB.
The detail of M_RECEIVE FB is described in Section 6-6-3.

■ **Procedure for Receiving Data**



1) Execute Initializing Process of NP1F-PC2 and Open Process with other node.
2) Establish “Waiting” state by turning ON the Enable Receive (EN_R) signal.
3) Receive data from other node into relevant Receive Buffer (decided by the parameter setting at the Open Process). Received data is stored in the buffer in following format.

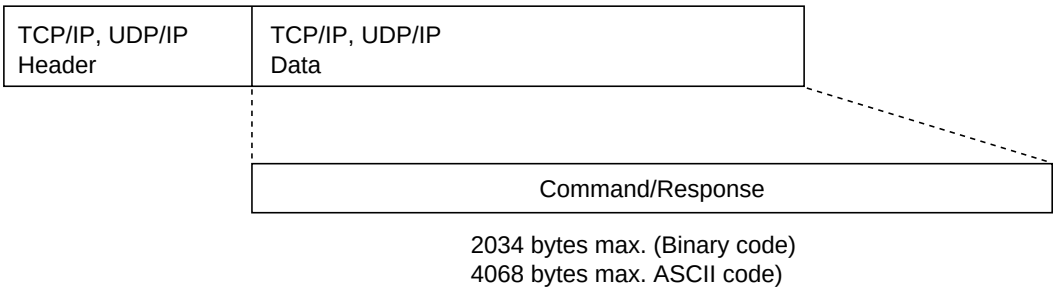


4) Read out data with M_RECEIVE FB.
5) When data read with M_RECEIVE is completed, Receive Complete (NDR) signal is turned ON. NP1F-PC2 returns response to the other node when Receive Complete (NDR) is turned ON.
6) Turn Enable Receive (EN_R) signal to terminate receive condition.

- When abnormal data is received, Receive Complete (NDR) does not turn ON, and data is not stored in Receive Buffer.

6-4-4 Data format

Data Sent/Received by other node



2034 bytes max. (Binary code)
4068 bytes max. ASCII code)

(1) Header

The header is the same as Ethernet TCP/IP or UDP/IP header. When sending data to the other node from application program, the header is automatically added by the NP1F-PC2. The module removes header when data is received from other node and only data is passed to the application program.

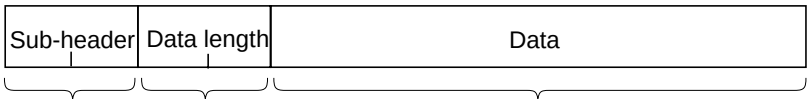
(2) Data

The Data part is equivalent to the Ethernet TCP/IP or UDP/IP data. When data is sent to other node from application program, the (NP1F-PC2) adds sub-header to the data transferred into the Send Buffer before sending it out. In binary communication mode, the data is sent as it is and in ASCII communication mode, it is converted to ASCII character before transmission.

When data is received from other node, the sub-header in TCP/IP or UDP/IP data part is removed and data only is stored in the Receive Buffer. In binary communication mode, the data is stored in the buffer as it is received and in ASCII communication mode, received ASCII character data is converted into binary and stored.

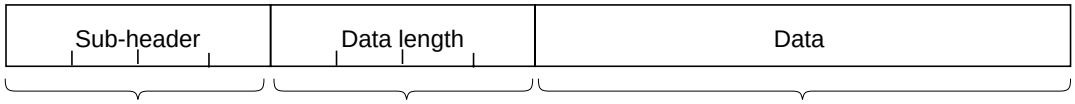
Command format

Binary code communication



2 bytes 2 bytes Variablr length (2034 bytes max.)

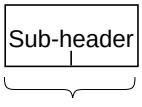
ASCII code communication



4 bytes 4 bytes Variablr length 4068 bytes max.)

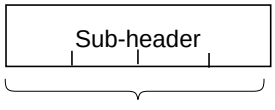
Response format

Binary code communication



2 bytes

ASCII code communication

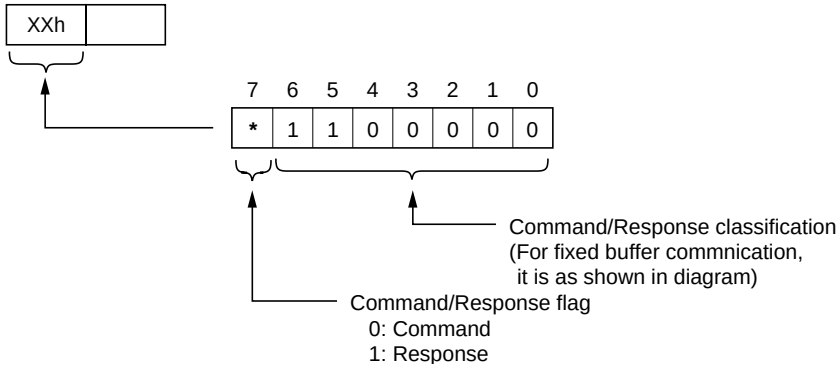


4 bytes

6-4 Fixed Buffer Communication Mode

1) Sub-header

Sub-header format is shown below. This module adds sub-header to command/response and sent it to other node.



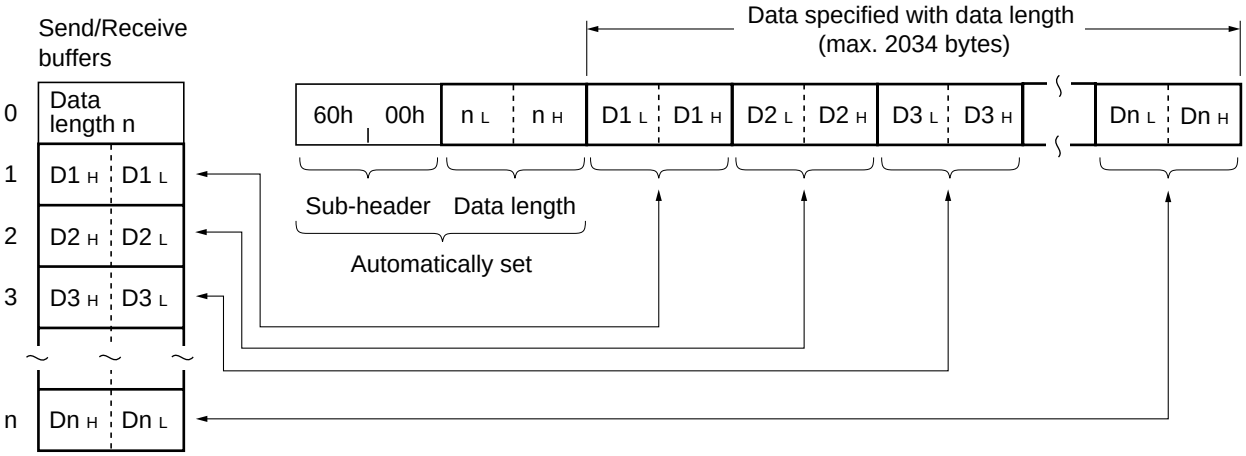
	Command	Responce
Binary code commnication	60h 00h	E0h End code
ASCII code commnication	36h 30h 30h 30h " 6 " " 0 " " 0 " " 0 "	45h 30h End code “ E” “ 0”

End code

Binary	ASCII	Meaning	Comment
00h	30h 30h	Normal complesion	-
50h	35h 30h	Command/Response not defined	Command or Response code in sub-header is incorect.
52h	35h 32h	Data word incorect	Number of data word too large
-	35h 34h	Abnormal ASCII conversion	In ASCII code communication, code that can not be converted to ASCII code is sent from other node.

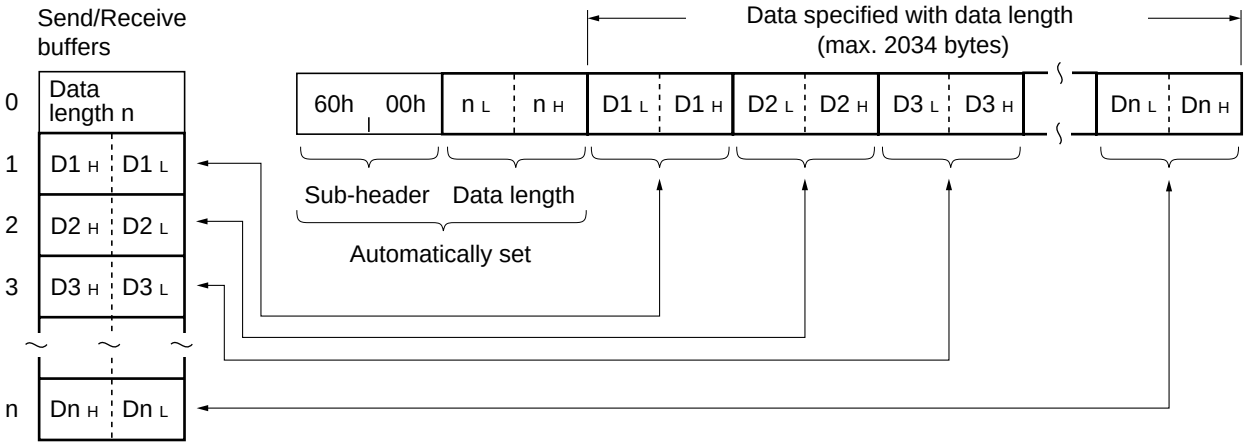
2) Data

Binary code communication (Data swap not specified)



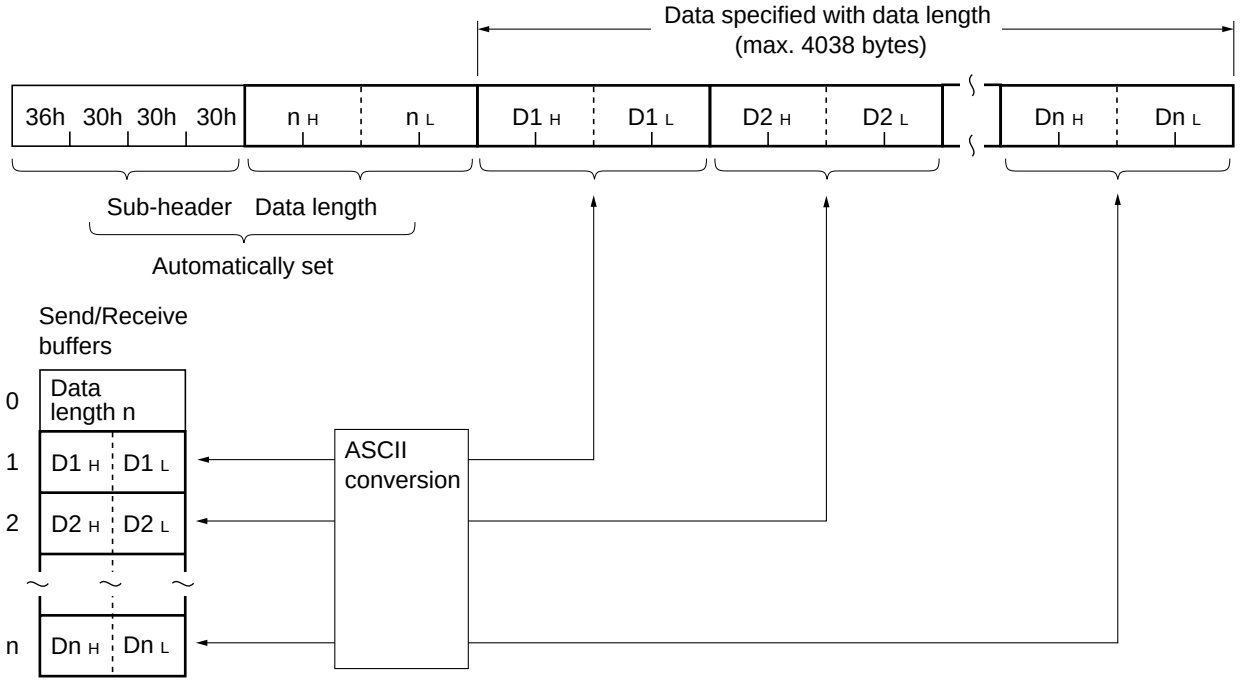
Data length represents number of data words stored in the buffer (maximum 1017 words).

Binary code communication (Data swap specified)



Data length represents number of data words stored in the buffer (maximum 1017 words).

ASCII code communication



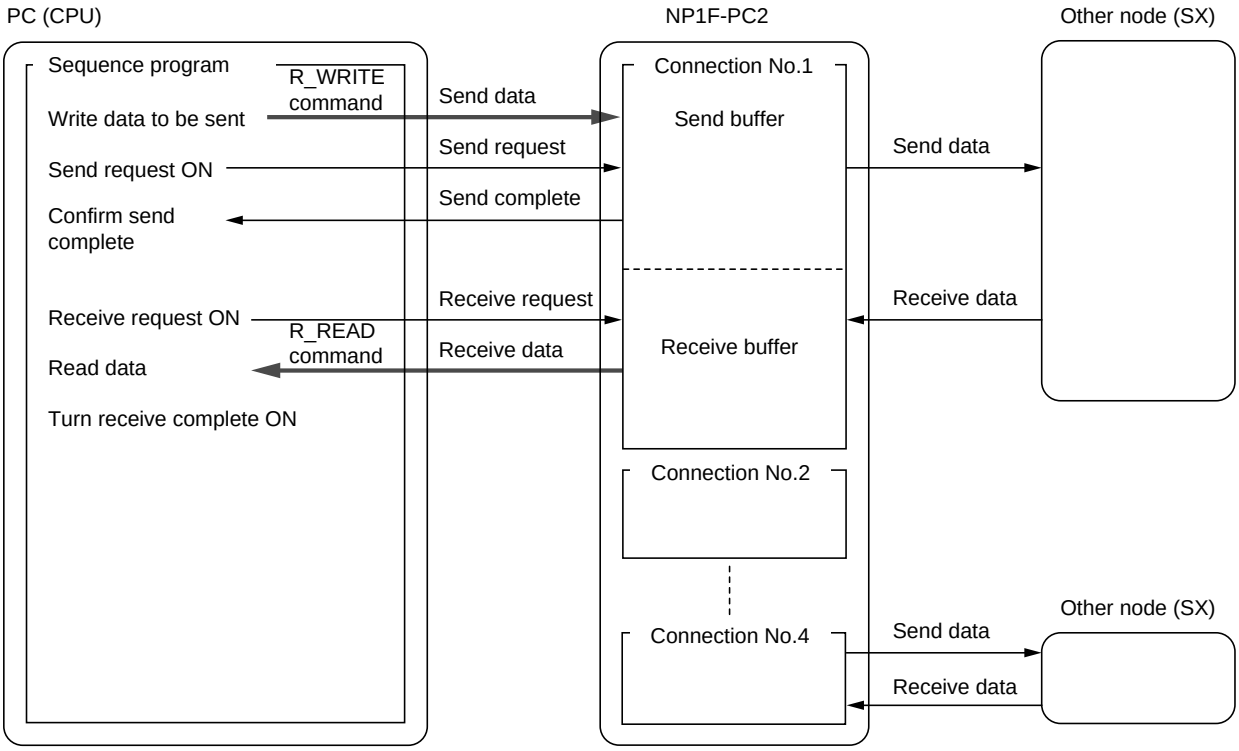
Data length represents number of data words stored in the buffer (maximum 1017 words).

6-5 Auto Transmission Mode

In this mode of communication, using Remote Data Read/Write (R_READ/R_WRITE) FB, it is possible to communicate with other MICREX-SX on Ethernet without executing open process from application program and disregarding to the type of communication interface modules used.

6-5-1 Outline of communication

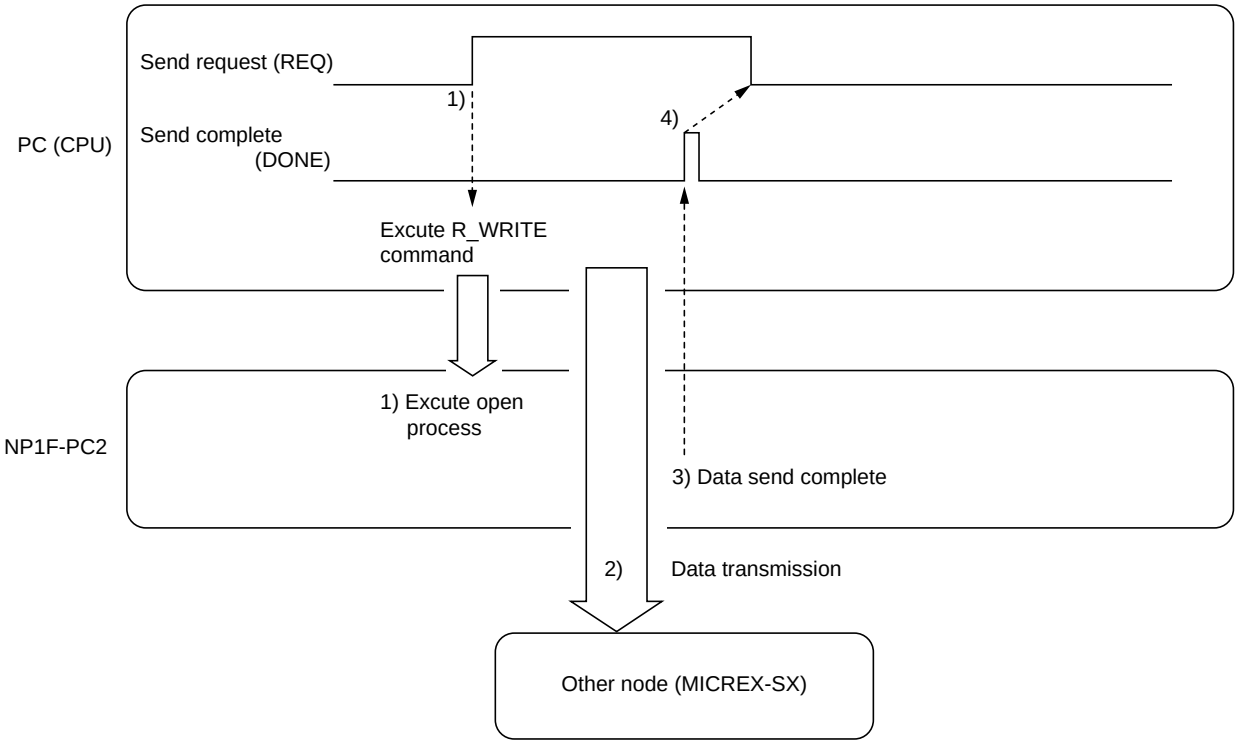
Data flow



6-5-2 Sending data

Procedure for sending data to the other node is explained here. In this example, data is sent through connection No. 1 to the other node.

■ **Procedure for sending data**



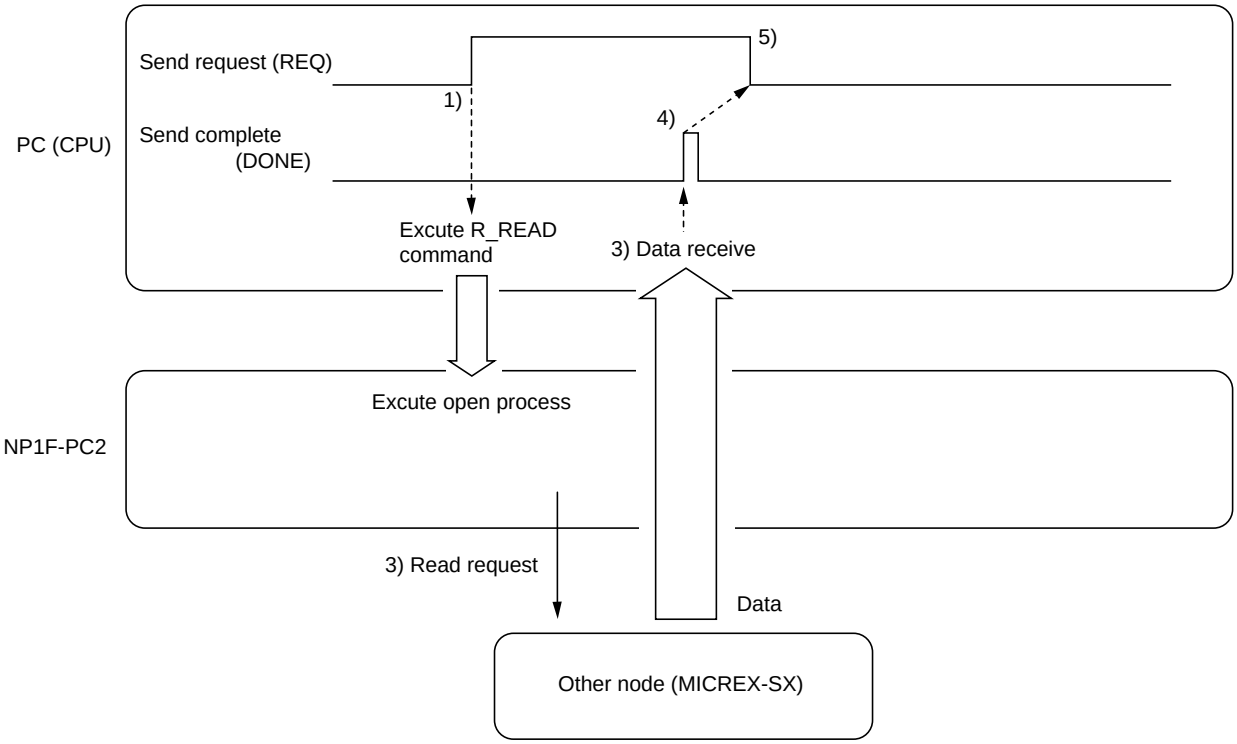
1) By turning ON Send Request (REQ) signal from application program, open process with other node is executed
2) Data transmission from the relevant Send Buffer of the connection to the specified node takes place.
3) Send Complete (DONE) signal turns ON when data transmission is completed.
4) In the application program, turn OFF Send Request (REQ) signal after confirmation of Send Complete (DONE) signal.

• When transmission does not complete normally, Send Error Detect (ERROR) signal turns ON. In this case, turn Send Request (REQ) OFF first, then retry the whole process.

6-5-3 Receiving data

Control for receiving data from other node to this [NP1F-PC2] is explained.

■ **Procedure for receiving data**

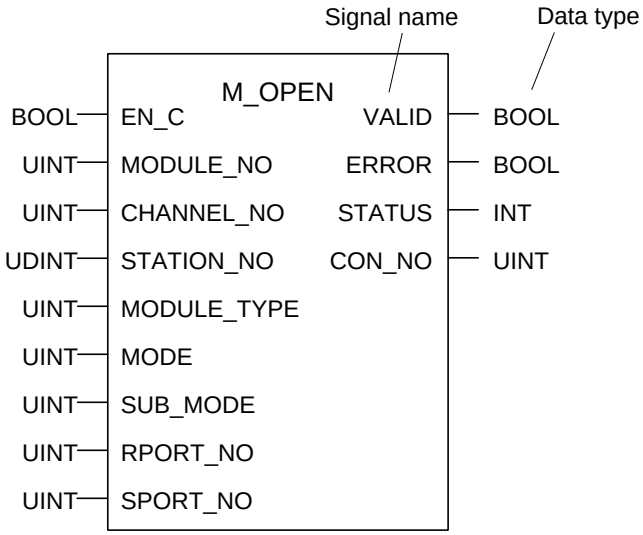


1) Line Open Process is executed when Send Request (REQ) signal is turned ON in the application program.
2) Memory Read Request is executed to the specified node through relevant connection.
3) PC (CPU) receives specified memory data from specified node.
4) Complete (DONE) signal turns ON when data reception is completed.
5) In application program, confirm the Complete (DONE) signal and turn OFF Send Request (REQ) signal.

- When abnormal data is received, Complete (DONE) signal does not turn ON. And data in not stored in the Receive Buffer.

6-6-1 Channel open (M_OPEN)

THE CPU uses Channel Open (M_OPEN) FB to open own ports and establish connection with the stations (equipment) connected to PC2 (maximum 8 channels may be open at the same time for a NP1F-PC2.
When open process complete successfully, connection ID is generated and thereafter this ID is used for the communication using M_SEND and M_RECEIVE.



<Operation>

1) PC2 commences open process specified with “MODULE_NO” at the rising edge (0 to 1) of “EN_C” (Open process does not complete in 1 scan).
2) When open process completes, Connection Number is output to “CON_NO.” And it is possible to use M_SEND and M_RECEIVE.
3) When there is error in open process, “ERROR” turns ON for 1 scan and Error Code is output to “STATUS”.
4) Close process takes place when “EN_C” is turned OFF (Close process requires more than 1 scan).
5) When close process completes, “VALID” turns OFF (There is no abnormal end of close process).

<Note>

1) There is [Passive] open for the receiving and [Active] open for sending. For the communication with other node, there are open processes for sending and receiving data.
   In order to send data to the other node, the other node has to be complete [Passive] open for receiving.
2) During open process, if “EC_C” changes state from 1 to 0, close process is carried out.
3) Reopen after close process requires the other node to execute close process and reopen the channel.

6-6 Commands Used for Communication

<FB terminals>

Classification	Symbol	Name	Data type	Description
INPUT	EN_C	Open Request	BOOL	0 to 1 Open Request (execute command) 1 to 0 Close Request. (Kept 1 during open)
	MODULE_NO	Communication Module No. (SX Bus Stn. No.)	UINT	The SX Bus Station No. of the NP1F-PC2 where Ethernet Card is installed.
	CHANNEL_NO	Channel No. within Module	UINT	Specifies relevant channel No. when there are multiple channels in 1 module. * Fixed to "0."
	STATION_NO * 1	Destination Station No.	UDINT	Station Number of destination on its network. * For Ethernet Card, it is "IP Address."
	MODULE_TYPE	Module Type No.	UINT	Network type of the communication module. * For Ethernet Card, it is fixed to "0001."
	MODE * 2	Communication Mode	UINT	Communication condition of channel to be open.
	SUB_MODE * 3	Communication Sub-mode	UINT	Sets whether destination module has send/receive confirmation.
	RPORT_NO	Sending Port No.	UINT	Sets Port No. for sending data. Communication always takes place through port.
	SPORT_NO	Receiving Port No.	UINT	Sets Port No. for receiving data. Communication always takes place through port. Range of setting is 1 to 127. (Do not assign same port No. for different purposes.)
OUTPUT	VALID	Open Valid	BOOL	This signal is output when open process complete successfully (Level output). When open process is terminated ("EN_C" =0), connection is closed.
	ERROR	Error output	BOOL	This signal is output when open process ends erroneous (Pulse output).
	STATUS * 4	Error Status	INT	When open process ends with error, status of the error is output.
	CON_NO	Connection No.	UINT	When open process completes, Connection No. is allocated. This No. is used in M_SEND, M_RECEIVE.

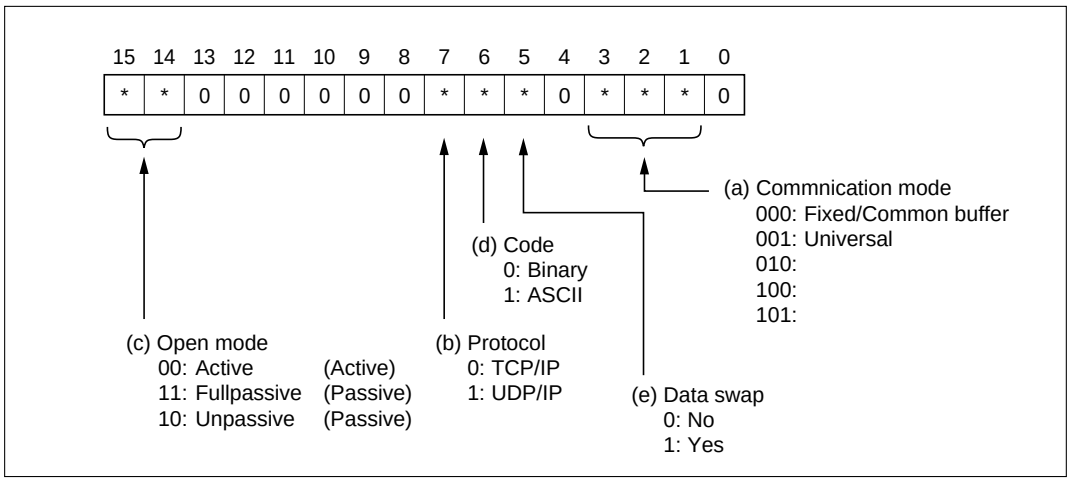
***1 STATION_NO: IP Address of the other node. (2 words)**
Sets communication destination station IP address. IP address is set in Hexadecimal numbers.

Example:
Setting when IP address is 172.16.0.1.

ACh	10h	00h	01h
172	16	0	1

*** 2 MODE: communication mode**

Communication condition of the channel to be open is set in 1 word.
Following diagram shows the format of the word.



(a) Communication mode

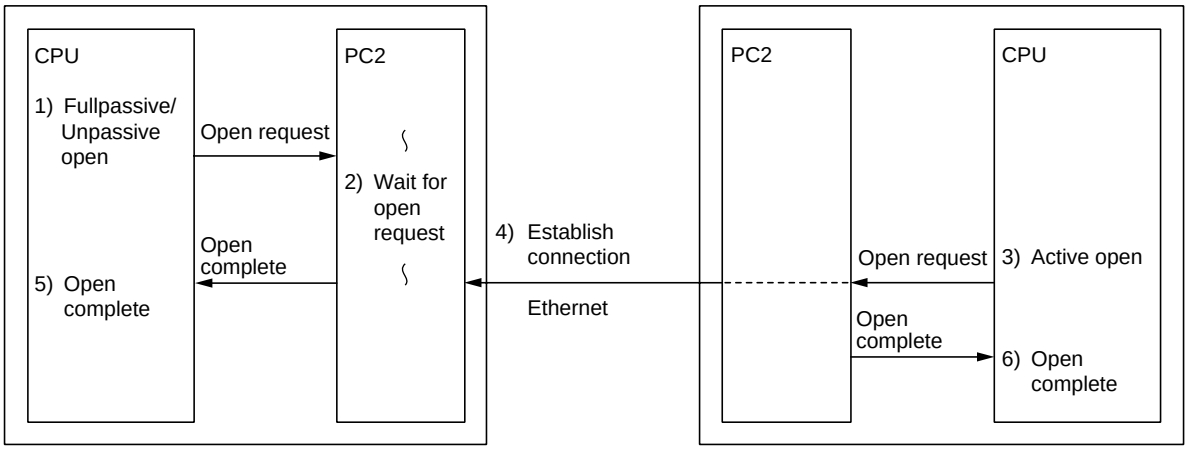
Sets communication mode of the channel.

(b) Communication protocol

Protocol for the connection. Select TCP/IP or UDP/IP.

(c) Open method

For TCP/IP open process, Fullpassive/Unpassive open (Passive Open) nodes are opened first and when this is completed, Active open nodes are opened.



1) Active Open
   Processes active open to the passively opened nodes of TCP connection.
2) Fullpassive Open
   TCP connection is opened passively only to the nodes with their address is specified in communication address set area. The node is in the state of waiting for active open from the nodes specified in communication address set area.
3) Unpassive Open
   TCP connection is opened passively to the all nodes on the network.
   The node is in the state of waiting for the active open from any other nodes on the network.

(d) Transmission code
Select data code (Binary or ASCII) for the communication with other node.

(e) Data swap
For binary code transmission, order of upper byte and lower byte of data is specified.
In ASCII code transmission this specification does not have any effect.

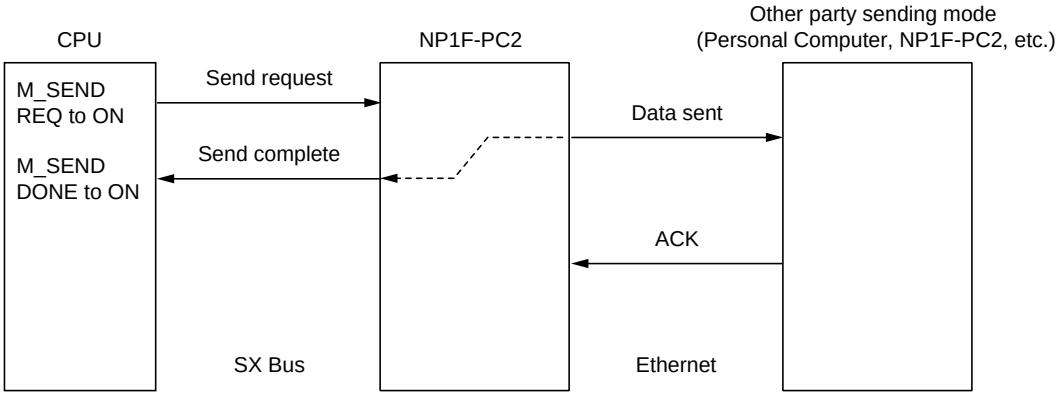
■ **Examples of communication mode Setting (For binary code transmission)**

Communication type \ Communication mode		Universal	Fixed/Common Buffer
TCP	Active	0002h	0000h
	Fullpasive	C002h	C000h
	Unpasive	8002h	8000h
UDP	Active	0082h	0080h
	Fullpasive	C082h	C080h
	Unpasive	8082h	8080h

*** 3 SUB_MODE: communication sub-mode**
0: The other node (Module or Application of other node) does not confirm send/receive.
1: The other node (Module or Application of other node) confirms send/receive.

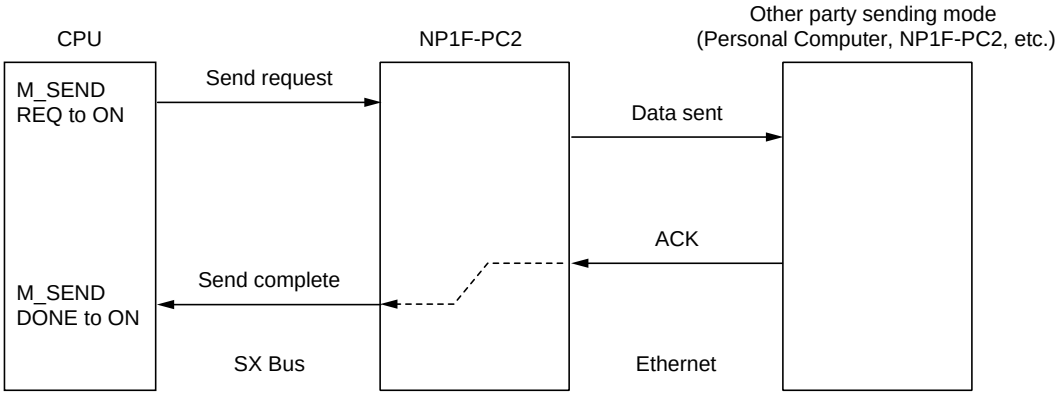
<SUB_MODE operation>

1) SUB_MODE = UINT#0



* When data is sent on Ethernet without waiting ACK from other party node, Send Complete results.

2) SUB_MODE = UINT#1



* When ACK is received from other party node, Send Complete results.

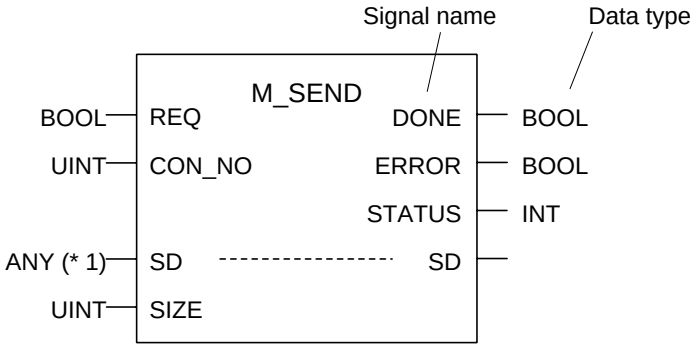
*** 4 STATUS**

Name	Code	Description
Parameter error	177 (B1h)	Module does not exist in the station specified with "MODULE_NO" or code specified with "MODULE_TYPE" does not match the network type of the module.
Channel open error	193 (C1h)	• Wrong number is specified for "STATION_NO." • Wrong value is specified for communication mode. • When communication mode is active (Send), Station No. (IP Address, RPORT_NO) does not exist on the network. • When connection is not established for other reasons.
Port specification error	200 (C8h)	• "SPORT_NO" specification is out of range (1 to 127). • Same "SPORT_NO" is already specified in the resource. • In one module, the same "SPORT_NO" and "RPORT_NO" is registered.
Connection No. Client port No. Full	201 (C9h)	• Attempt is made to open more than 57 ports in a resource. • Attempt is made to open more ports than available in one module.

Refer to Appendix 1-1 for the common status of communication FB's.

6-6-2 Message send (M_SEND)

Message Send (M_SEND) FB is used to send messages to the station with which the communication channel is open.



* 1 All data types excluding BOOL.

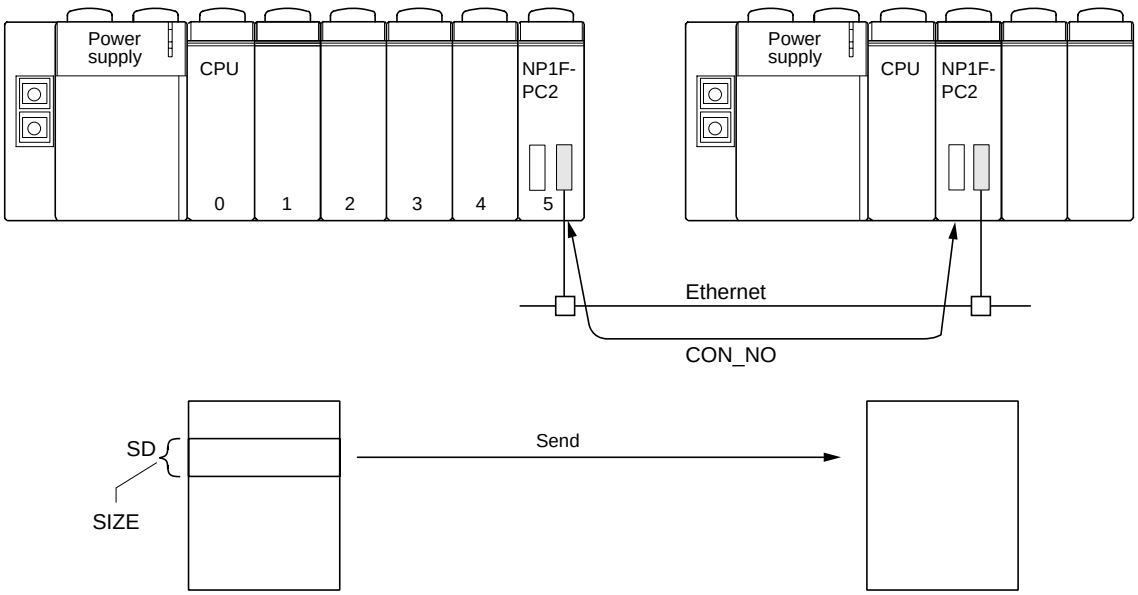
<Operation>

1) At the rising edge (0 to 1) of “REQ,” message is sent to the station with connection number specified with “CON_NO.” (Transmission takes more than 1 scan.)
2) “DONE” is turned ON for 1 scan when transmission is completed.
3) When message transmission is erroneous, “ERROR” turns ON for 1 scan and error code is output to “STATUS.”

<Note>

1) Maximum number of data words for 1 transmission is 1017 words (Universal Mode). For other modes, refer to the relevant section.
2) During transmission (from rising edge of “REQ” to rising edge of “DONE” or “ERROR”), transition of “REQ” (0 to 1) is invalid.
3) Do not change “SD” during transmission. If changed, integrity of data is not guaranteed.
4) Number of data specified with “SIZE” exceeds variable size specified with “SD”, extra data may become undefined.
5) Program the sequence so that “REQ” is set to 1 after “VALID” of M_OPEN turns to 1.

Data flow and signal names of FB are shown below.



Classification	Symbol	Name	Data type	Description
Input	REQ	Send request	BOOL	Excute command at rising edge (0 to 1).
	CON_NO	Connection No.	UINT	Connection No. established with M_OPEN ("CON_NO" of M_OPEN FB)
	SIZE * 1	Send data storage variable size	UINT	Data size of the variables storing data to be sent (Word unit).
Output	DONE	Complete output	BOOL	Output when message transmision is completed successfully. (Pulse output)
	ERROR	Error output	BOOL	Output when message transmision is erroneous. (Pulse output)
	STATUS * 2	Error status	INT	Message transmission error code is output.
Input/Output	SD	Send data storage variable	ANY	Set up transmission data storage variables.

*** 1 SIZE**

Set SIZE to the size of variable specified with SD.
If it is larger than size of SD, data other than the variable (rubbish) will be transmitted.

*** 2 STATUS**

Name	Code	Description
Parameter error	177 (B1h)	"SIZE" is set to 0.
Message send error	195 (C3h)	• Cannot send message to communication module of other station. • Did not receive reply from other station (sending completed but no ACK received).
Channel close	199 (C7h)	• The other station is closed. Note: When this code is received, close relevant channel once and issue open request again.
Port specification error	200 (C8h)	• The other station is not opened.
Buffer overflow	206 (CEh)	• Data exceeded 1017 words (Universal Mode).
Connection No. error	207 (CFh)	• Connection number used is not opened. • Attempt is made to use connection number already used. (2 M_SEND are used in parallel with the same connection number).

Refer to Appendix 1-1 for common status of communication FB's.

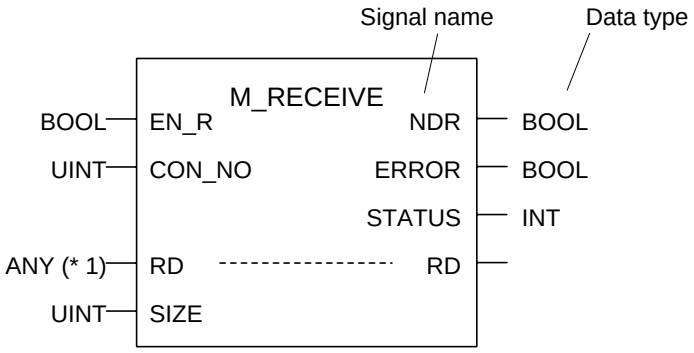
<Caution During M_SEND Use >

1) In Universal Communication Mode in UDP/IP, there is no Send/Receive confirmation and flow control is not used. When Receive Buffer is full, subsequent data received is discarded. And therefore number of words processed by sending station and receiving station does not match.
   When buffer is full, it takes approximately 10 seconds to clear the buffer and receiving operation may paused during this time.
2) In Full Passive Open, when open request is received from station not specified, the connection will be established once then Full Passive station will issue Close Request to the Active station. Therefore at the Active station, open process completes normally and when data is transmitted it will be in error status C7h (Forced Close).
3) When port number does not match between sending receiving station, the sending station executes forced close and error status “C7h: (Forced Close)” is generated.

4) In communication between 2 MICREX-SX, when repeated 1 word data transmission takes place, depending on the timing of M_RECEIVE, 1st and 2nd word received may be combined and response to the CPU become 2 words reception. Therefore when sending 1 word data, allocate 2 words buffer at the receiving station. For data larger than or equal to 2 words, buffer size is the same as the data words.
5) In Universal Communication Mode in UDP/IP, when transmission code is ASCII and data exceeds 1019 bytes, the sending station will divide data in 2 and send. Therefore in the receiving station, 2 Receive Request is required. Also the buffer size needs to be larger than the data size.

6-6-3 Message receive (M_RECEIVE)

Message Receive (M_RECEIVE) FB is used to receive message from the station with which the channel is opened.



* 1 All data types excluding BOOL.

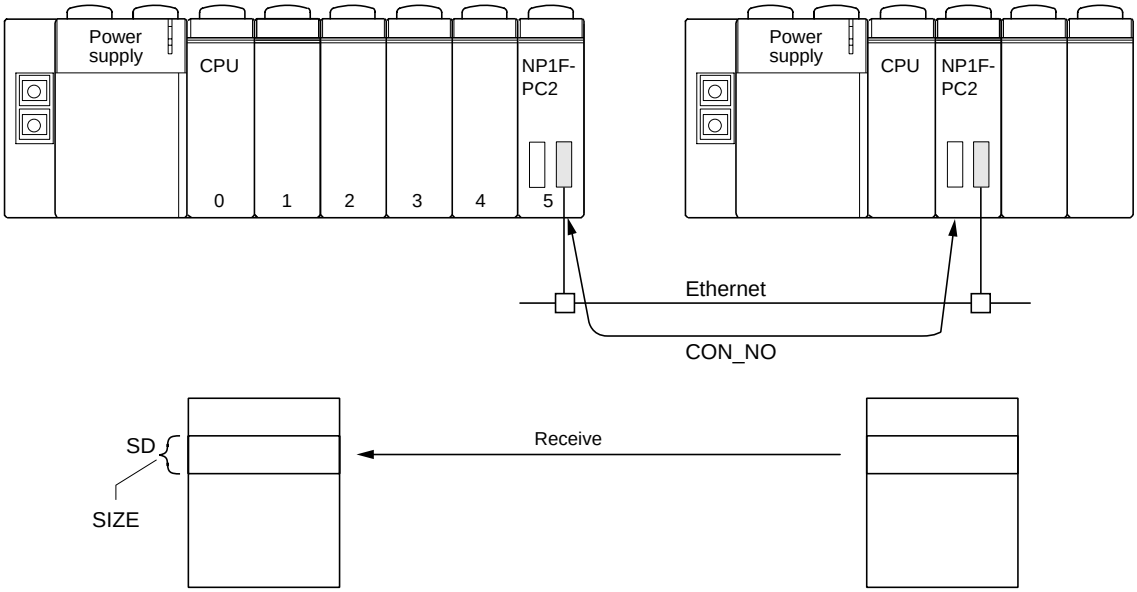
<Operation>

1) At the rising edge (0 to 1) of “EN_R,” message is received from the station specified by “CON_NO.” (Message reception takes more than 1 scan.)
2) When message is received correctly, “NDR” goes “1” for 1 scan.
3) Message reception is not successful, then “ERROR” becomes “1” for 1 scan and error code is output to “STATUS.”

<Note>

1) Maximum number of data words for 1 reception is 1017 words (Universal Communication Mode).
2) Keep “EN_R” to “1” during message reception (from rising edge of “EN_R” to rising edge of “NDR” or “ERROR”). Change of “EN_R” to “0” means that reception is paused.
3) After pausing reception, reception is restarted by changing “EN_R” to “1.” At restart of the reception, change in “CON_NO,” “RD” or “SIZE” does not affect the message reception. The reception is continued with original settings.
4) After completion of message reception, if “EN_R” is kept to “1” in next scan, new process of receiving message will start.
5) Maintain “RD” during message reception. If it is altered, integrity of the received data is not guaranteed.
6) If data size specified with “SIZE” is larger than the size of variable specified with “RD,” received data may be written in other variable area. Without fail set “SIZE” to the size of variable area specified with “RD.”
7) Write sequence program so that “EN_R” is set to “1” after “VALID” of M_OPEN becomes “1.”

Data flow and signal names of FB are shown below



Classification	Symbol	Name	Data type	Description
Input	ENR	Enable receive	BOOL	When this signal is "1", it is possible to receive message.
	CON_NO	Connection No.	UINT	Connection Number established with M_OPEN (CON_NO of M_OPEN FB) is set for this.
	SIZE * 1	Data storage variable area size	UINT	Specifies received data storage variable size (in word unit).
Output	NDR	Complete flag	BOOL	This signal is output when message reception is complete normally. (Pulse Output)
	ERROR	Error output	BOOL	This signal is output when message reception is complete abnormally. (Pulse Output)
	STATUS * 2	Error status	INT	When message reception is erroneous, the error code is output.
Input/Output	RD	Received data storage variable	ANY	Specifies received data storage variable.

*** 1 SIZE**

Make received data size equal to the Received Data Storage Variable size specified with “RD.”
When received data exceeds the size of received data storage, excess data will be discarded.

*** 2 STATUS**

Name	Code	Description
Parameter error	177 (B1h)	"1" is input to "SIZE"
Channel close	199 (C7h)	• Destination station is closed. Note: When this code is generated, close the relevant channel once then issue open request again.
Port specification error	200 (C8h)	• Other station is not open.
Buffer overflow	206 (CEh)	• More data than size specified is received. Data stored in "RD" is valid data.
Connection number error	207 (CFh)	• Connection number used is not open. • Attempt is made to use connection number already in use. (2 M_RECEIVE are used in parallel.)

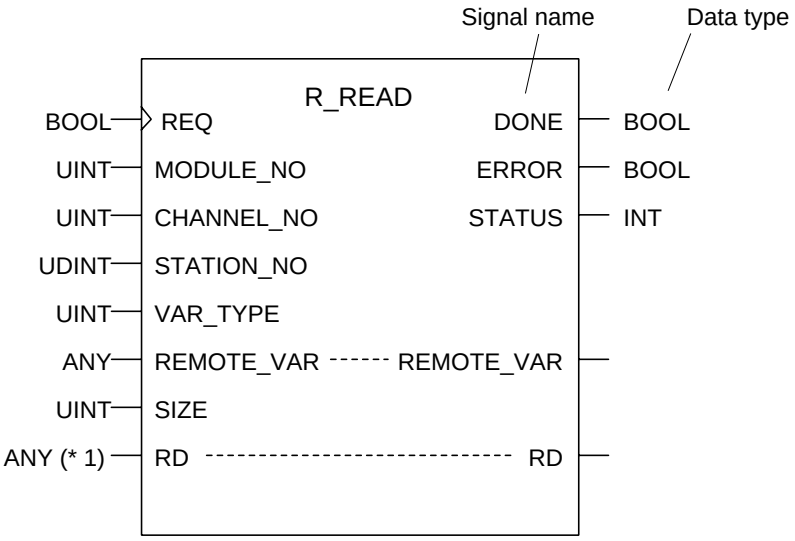
Refer to Appendix 1-1 for common status of communication FB’s.

<Caution During M_RECEIVE Use >

This is the same as M_SEND. Please refer to “Caution During M_SEND Use” on page 6-31 and 6-32.

6-6-4 Remote data read (R_READ)

Through the Ethernet, Remote Data Read (R_READ) FB is used to read data directly from specified address in the MICREX-SX connected to the network.



* 1 Data type other than BOOL or array of BOOL.

<Operation>

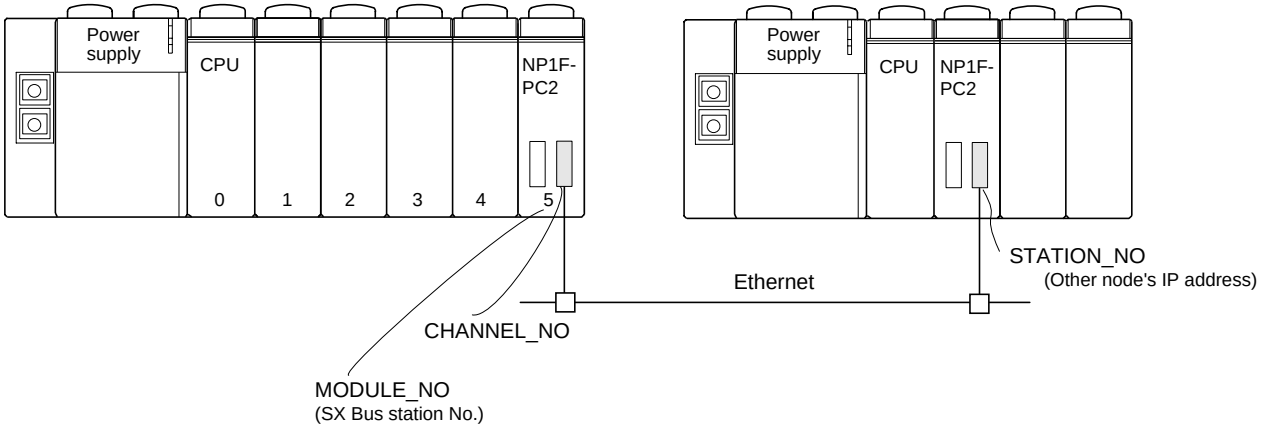
1) At the rising edge (0 to 1) of “REQ,” data specified with “MODULE_NO,” “CHANNEL_NO,” “STATION_NO” and “REMOTE_VAR” is read into the variable area specified with “RD.” The process does not complete in 1 scan.
2) Reading process completes successfully, “DONE” becomes “1” for 1 scan.
3) When reading process is erroneous, “ERROR” becomes “1” for 1 scan and the error code is output to “STATUS.”

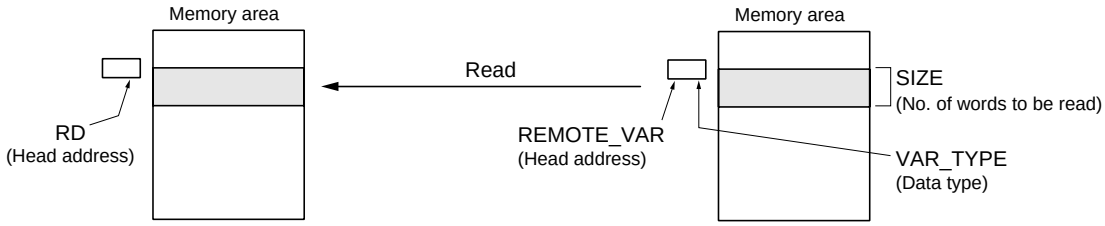
<Note>

1) Rising edge of “REQ” is effective but during reading (from rising edge of “REQ” to “DONE” or “ERROR” pulse), transition of “REQ” (0 to 1) has no effect.
2) Do not change “RD” during reading. If it is changed integrity of data read is not guaranteed.
3) There is no restriction for the “SIZE” when “VAR_TYPE” is “0.” Otherwise it is limited by the communication module through which data is read.
4) If the number specified with input “SIZE” exceeds size of variable area specified with “RD,” received data may change the content of other variable area. Be sure to specify the same size for “SIZE” and “RD.”
5) When “R_READ” command is used repeatedly, set TCP End Timer Value to “0” in initial setting.

6-6 Commands Used for Communication

Data flow and signal names of FB are shown below.





Classification	Symbol	Name	Data type	Description
Input	REQ	Send request	BOOL	At rising edge (0 to 1), transmission starts (Execution of command).
	MODULE_NO	Communication module No. (SX Bus station No.)	UINT	Destination PC2's station number on the SX Bus.
	CHANNEL_NO	Channel No.	UINT	When module has multiple channel, specify channel to be used. * Fixed to "0."
	STATION_NO	Destination's network station No.	UDINT	Destination's network station No. is set. Ethernet Card: 00000000h to FFFFFFFFh.
	VAR_TYPE	Variable specification (Data type)	UINT	Object Ethernet network data type. Fixed to "0" with this module.
	SIZE	Read data size (No. of words)	UINT	Number of data word to be read. (16 bit: 1 word)
Output	DONE	Complete output	BOOL	Output when result of command execution is normal. (Pulse output)
	ERROR	Error output	BOOL	Output when result of command execution is abnormal. (Pulse output)
	STATUS * 2	Error status	INT	Error code is output when command execution resulted in error.
Input/Output	REMOTE_VAR * 1	Access object variable (Read head address)	ANY	Sets the head address of the data to be read.
	RD	Storage variable (Storage head address)	ANY	Sets the head address of data storage area.

*** 1 REMOTE_VAR**

REMOTE_VAR data format depends on VAR_TYPE (refer User's Manual FEH200 for commands).
In case of transmission through the [NP1F-PC2], VAR_TYPE is fixed to "0" and therefore the format is as follows.

15 0

CPU No.
Mwmory type
Address (Lower)
Address (Upper)

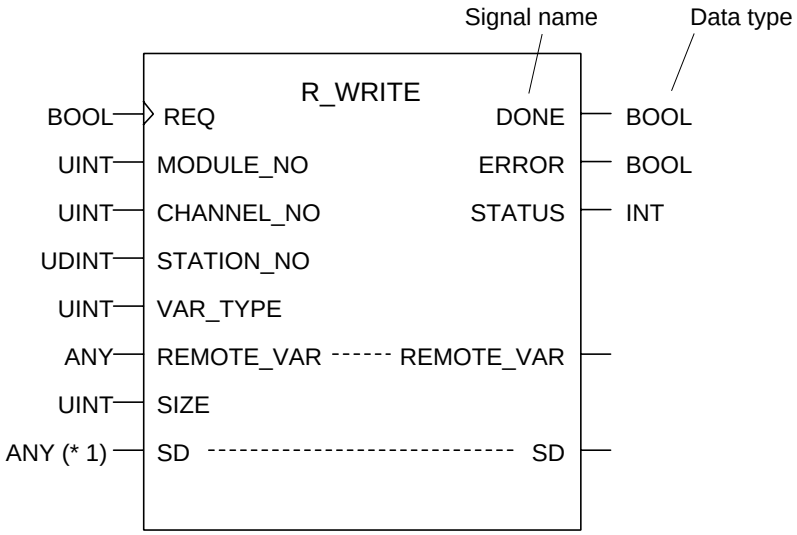
*** 2 STATUS**

Name	Code	Description
Parameter error	177 (B1h)	• "SIZE" was set to 0. • Input value to "VAR_TYPE" is out of specification. • No module exist in station specified with "MODULE_NO."
Channel open error	193 (C1h)	Abnormal value is set for channel No.
Station No. error	195 (C3h)	Abnormal value is set for station No.
Client port No. FULL	201 (C9h)	Atempt to open more port than available was made. (The [NP1F-PC2] has maximum 4 ports.)
Transmission size over	206 (CEh)	When "VAR_TYPE" is set to other than "0," transmission size exceeded limit of modules en route.

Refer to Appendix 1-1 for common status of communication FB's.

6-6-5 Remote data write (R_WRITE)

Through the Ethernet, Remote Data Write (R_WRITE) FB is used to write own CPU data directly into specified address of other MICREX-SX connected to the network.



* 1 Data type other than BOOL or array of BOOL.

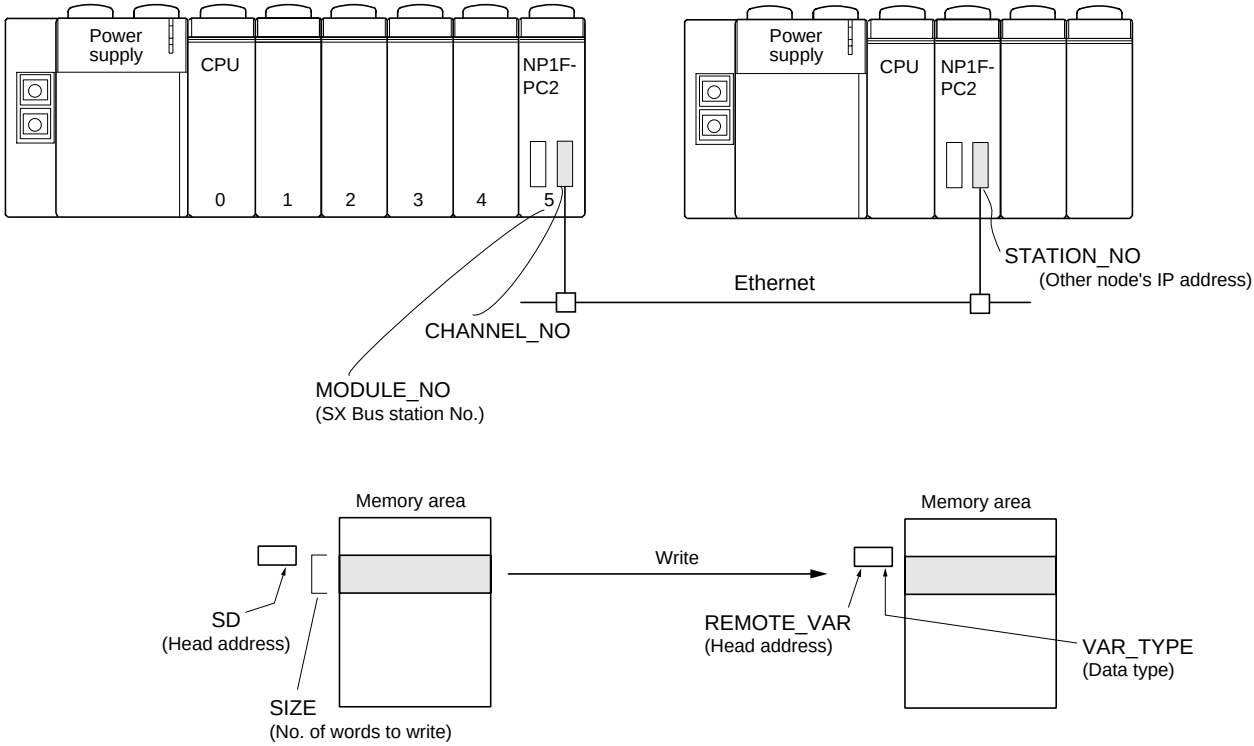
<Operation>

1) At the rising edge (0 to 1) of “REQ,” data specified with “SD” is written into the area specified with “MODULE_NO,” “CHANNEL_NO,” “STATION_NO” and “REMOTE_VAR” (The process takes more than 1 scan).
2) When data write is completed successfully, “DONE” becomes “1” for 1 scan.
3) When data write does not end normally, “ERROR” becomes “1” for 1 scan and error code is output to “STATUS.”

<Note>

1) Rising edge of “REQ” is effective but during writing (from rising edge of “REQ” to “DONE” or “ERROR” pulse), the transition of “REQ” (0 to 1) has no effect.
2) Do not change “SD” during writing. If it is changed integrity of data read is not guaranteed.
3) There is no restriction for the “SIZE” when “VAR_TYPE2 is “02.” Otherwise it is limited by the communication module through which data is read.
4) If the number specified with input “SIZE” exceeds size of variable area specified with “SD,” excess data may be undefined. Be sure to specify the same size for “SIZE” and “SD.”
5) When “R_WRITE” command is used repeatedly, set TCP End Timer Value to “0” in initial setting.

Data flow and signal names of FB are shown below.



Classification	Symbol	Name	Data type	Description
Input	REQ	Send request	BOOL	At rising edge (0 to 1), transmission starts (Execution of command).
	MODULE_NO	Communication module No. (SX Bus station No.)	UINT	Destination PC2's station number on the SX Bus.
	CHANNEL_NO	Channel No.	UINT	When Module has multiple channel, specify channel to be used. * Fixed to "0."
	STATION_NO	Destination's network station No.	UDINT	Destination's network station No. is set. Ethernet Card: 00000000h to FFFFFFFFh.
	VAR_TYPE	Variable specification (Data type)	UINT	Object Ethernet network data type. Fixed to "0" with this module.
	SIZE	Write data size (No. of words)	UINT	Number of data word to be written in. (16 bit: 1 word)
Output	DONE	Complete output	BOOL	Output when result of command execution is normal. (Pulse output)
	ERROR	Error output	BOOL	Output when result of command execution is abnormal. (Pulse output)
	STATUS * 2	Error status	INT	Error code is output when command execution resulted in error.
Input/Output	REMOTE_VAR * 1	Access object variable (Write head address)	ANY	Sets the head address of the area to be written in.
	RD	Source variable (Source head address)	ANY	Sets the head address of data source area.

*** 1 REMOTE_VAR**

The format of REMOTE_VAR is depends on VAR_TYPE (refer to The User's Manual for Commands: FEH200). When the communication is through the [NP1F-PC2], VAR_TYPE is fixed to “0” and the format is as follows.

15 0

CPU No.
Mwmory type
Address (Lower)
Address (Upper)

*** 2 STATUS**

Name	Code	Description
Parameter error	177 (B1h)	• "SIZE" was set to 0. • Input value to "VAR_TYPE" is out of specification. • No module exist in station specified with "MODULE_NO."
Channel open error	193 (C1h)	Abnormal value is set for channel No.
Station No. error	195 (C3h)	Abnormal value is set for station No.
Client port No. FULL	201 (C9h)	Atempt to open more port than available was made. (The [NP1F-PC2] has maximum 4 ports.)
Transmission size over	206 (CEh)	When "VAR_TYPE" is set to other than "0," transmission size exceeded limit of modules en route.

Refer to Appendix 1-1 for common status of communication FB's.

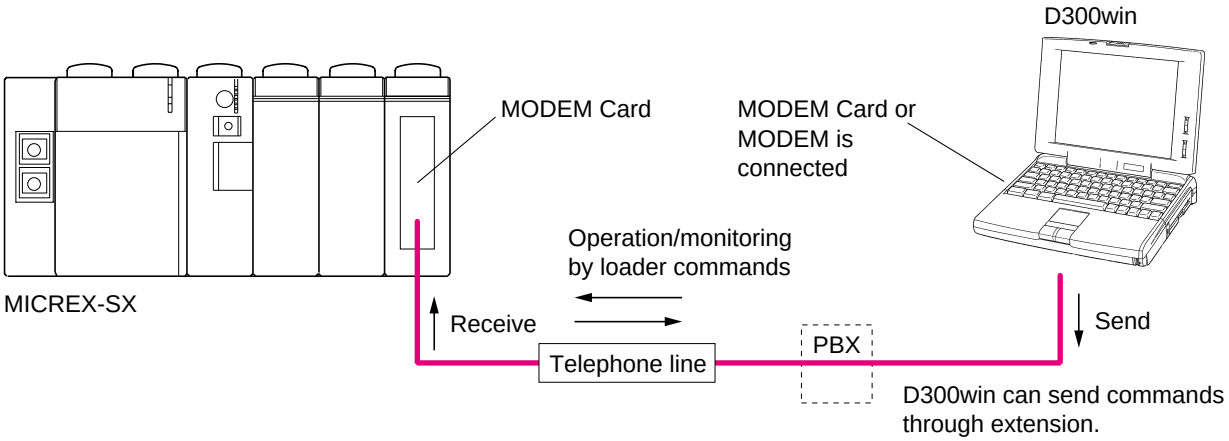
Section 7 MODEM Card Function

Page

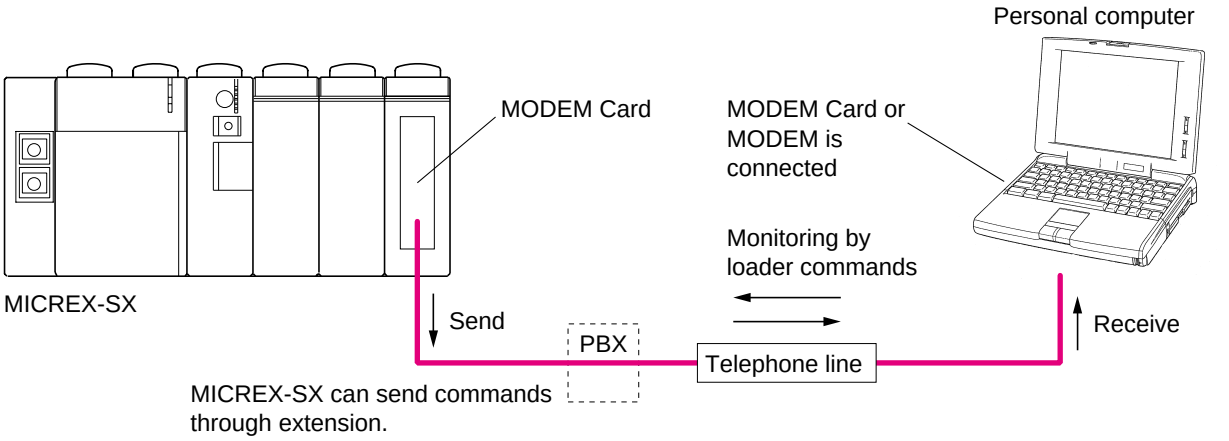
Section 7 MODEM Card Function
7-1 MODEM Function

The NP1F-PC2 offers the following 3 different MODEM Card functions:

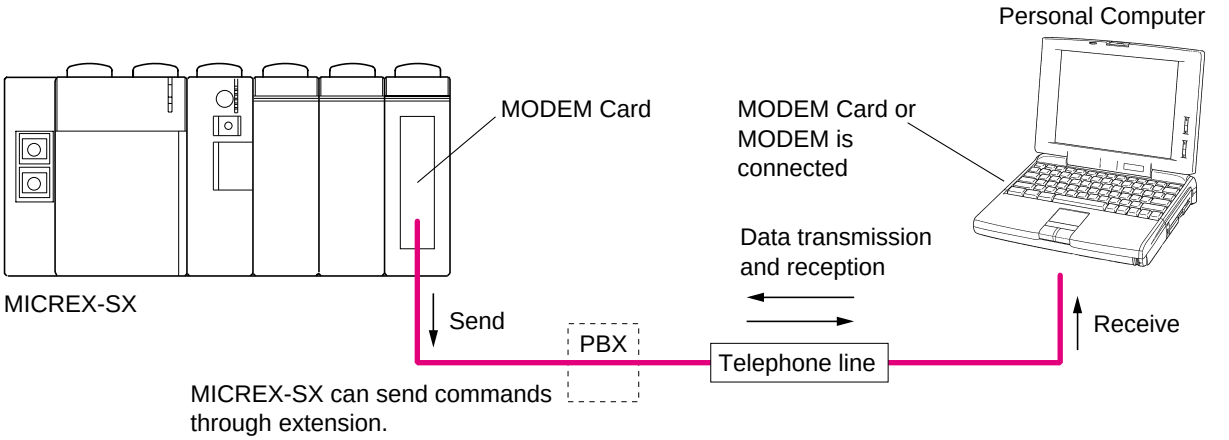
1) Dial receive + remote loader function
This function relays Loader commands from D300win connected through a telephone line to CPU Module. This function is operated from D300win.



2) Dial call + remote loader function (Supported from V2032)
This function makes a call to a Personal Computer connected through a telephone line. Upon completion of online connection, it relays Loader commands from the Personal Computer to CPU Module.



3) Dial call + data transmission and reception from application (Supported from V2032)
This function makes a call to a Personal Computer connected through a telephone line. Upon completion of online connection, data transmission and reception is possible. Data transmission and reception can be performed at the same time.



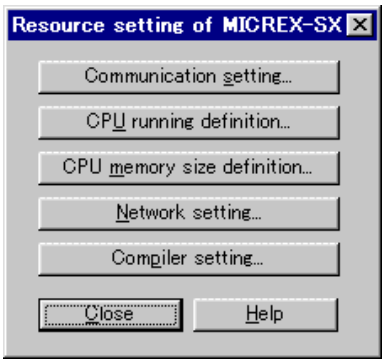
7-2 Operating Procedure for Dial Receive + Remote Loader Function

(1) Setup on the NP1F-PC2 side

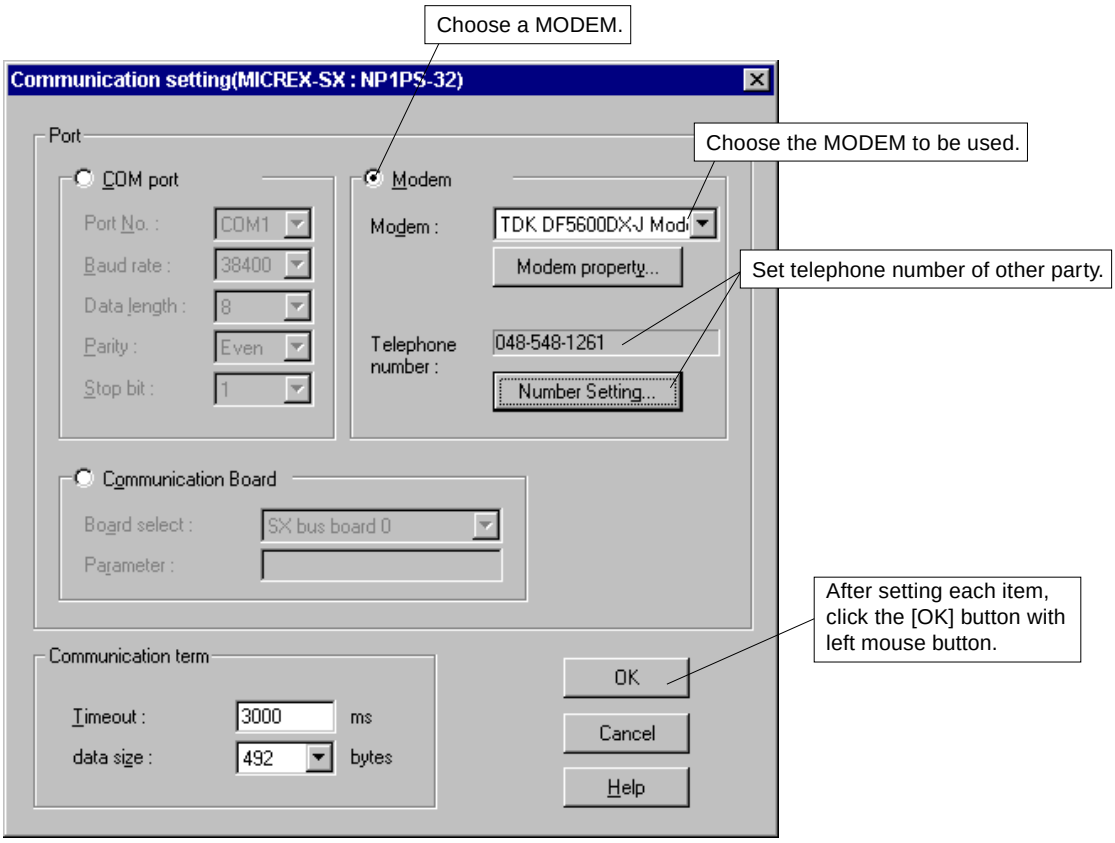
1) Download System Configuration Definition on MICREX-SX side from D300win.
2) Prepare a file for MODEM Card initialization and then download it from D300win.
3) Attach MODEM Card and then activate MICREX-SX System.

(2) D300win 0peration

1) When you click Resource Icon on D300win Project Tree with right mouse button and then choose Set menu, the following MICREX-SX Resource Setting dialogue box is displayed.

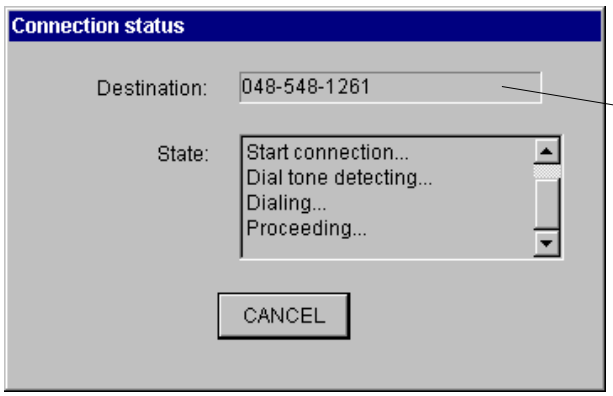


2) Communication Setting dialogue box of Resource is displayed. Make setting as follows:



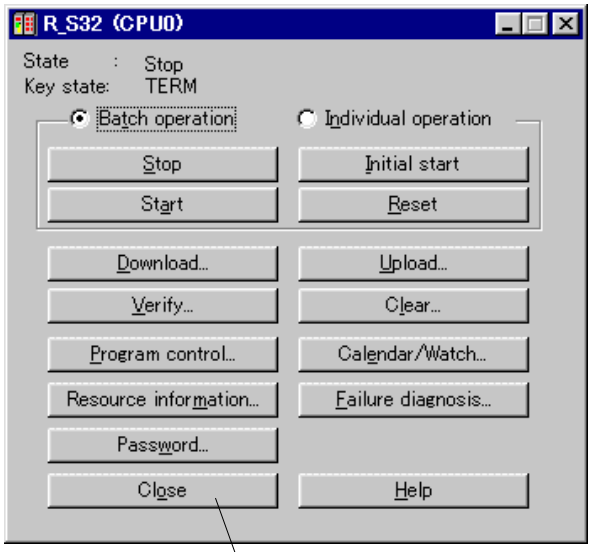
7-2 Operating Procedure for Dial Receive + Remote Loader Function

3) When you click Control Icon on D300win, dial call is started automatically. The following screen is displayed.



Telephone number of other party is displayed.

4) When connection is established between CPU through a telephone line, Control dialogue box is displayed. Subsequent operations are the same as usual operations.



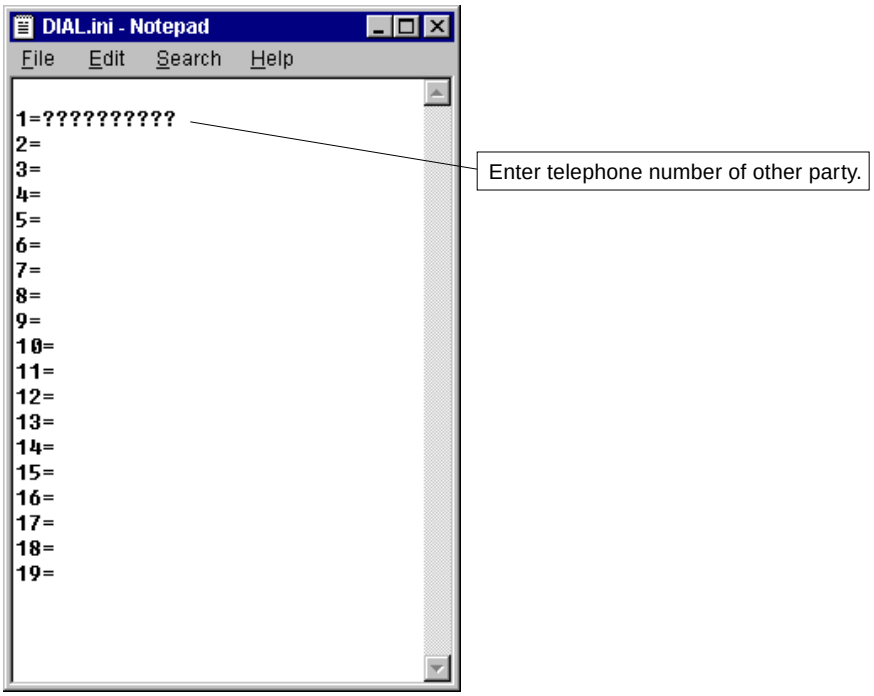
When you click Close button with left mouse button, Control dialogue box closes and then the line is disconnected automatically.

7-3 Auto Send Function

This function allows MICREX-SX side to send a call and send and receive data and Loader commands to/from a Personal Computer or other intelligent equipment connected to the telephone line. The following describes the procedure for making a dial call from MICREX-SX side.

< Setting Procedure >

1) Download System Configuration Definition of MICREX-SX from D300win.
2) Prepare a file for MODEM Card initialization and then download it from D300win. In this case, open file “DIAL.ini” using a commercial editor and then register telephone number of other party. Up to 20 numbers can be registered.

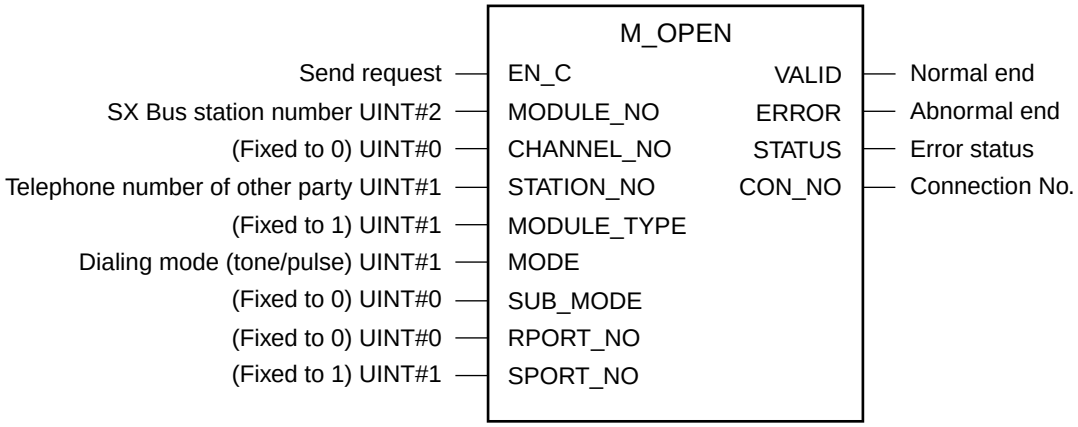


3) After registration of telephone number, save file “DIAL.ini” in overwrite manner.
4) Prepare a file for MODEM Card initialization and then download it from D300win.
5) Make a dial call using an application.
6) Download application program from D300win to CPU Module.
7) Attach MODEM Card, activate MICREX-SX System, then start operation.

7-3-1 Dial call program

M_OPEN FB is used for dial call. M_SEND FB/M_RECEIVE FB is used for data communication after line connection.

<M_OPEN FB>



<Pin Description>

Pin name	Purpose	Setting
EN_C	Connects variable for Send Request signal. When this pin is set to ON, a dial call is made. To disconnect line, set this pin to OFF.	
MODULE_NO	Sets the SX Bus station number of the NP1F-PC2.	
CHANNEL_NO	Fixed to 0.	UINT#0
STATION_NO	Sets telephone number of other party or the ID number of the telephone number registered in "DIAL.ini".	UDINT#1 to UDINT#20
MODULE_TYPE	Fixed to 1.	UINT#1
MODE	Sets the dialing mode. Tone: 0 Pulse: 1	UINT#0 (Tone) UINT#1 (Pulse)
SUB_MODE	Fixed to 0.	UINT#0
RPORT_NO	Fixed to 0.	UINT#0
SPORT_NO	Fixed to 1.	UINT#1
VALID	When the line is connected and becomes busy, this pin is set to ON. During the time interval from the time when EN_C pin is set to ON to when this pin is set to ON, the Dialing status results.	
ERROR	If line connection fails, this pin is set to ON only in one scan.	
STATUS	The error status is displayed. Set to 0 under normal condition.	
CON_NO	The connection number is output. This pin is connected to CON_NO of M_SEND or M_RECEIVE.	

<Error Codes>

Error code	Cause of error
164	Different SX Bus station number (For example, the NP1F-PC2 does not exist on the station number specified by "MODULE_NO.")
193	Invalid setting of "STATION_NO" This error also results if normal connection could not be made.
198	Other party is busy.
201	Two or more line connections were attempted with a single NP1F-PC2.

Section 8 Memory Card Function

Page

Section 8 Memory Card Function

8-1 General

When various memory cards are installed in the NP1F-PC2, access to the files on the memory card using Loader Commands is possible.
Memory Card can be inserted or removed with module's power on and the Card Service automatically recognizes the type of the memory card.

With memory card installed, following service is available.

(1) Data Read/Write from program
Reading data from the memory card and writing data to it is executed from application program using System FB.

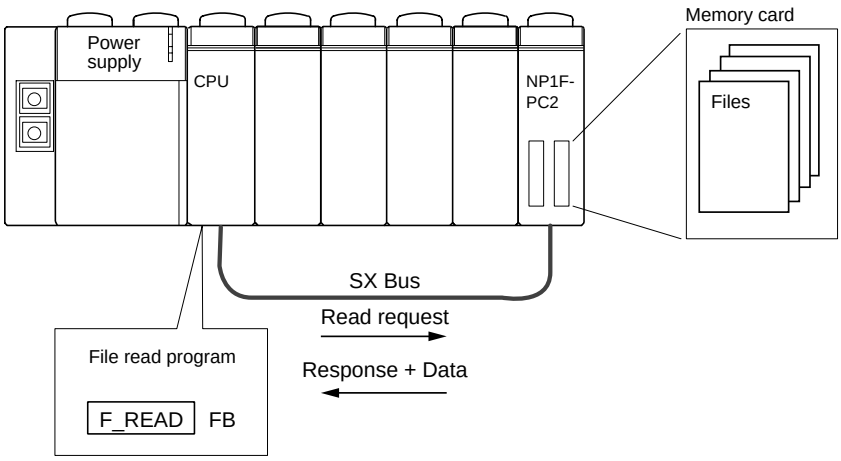
Note: Types of the data handled with memory card are variable and data files, not program files.

8-2 Data Read/Write from Program

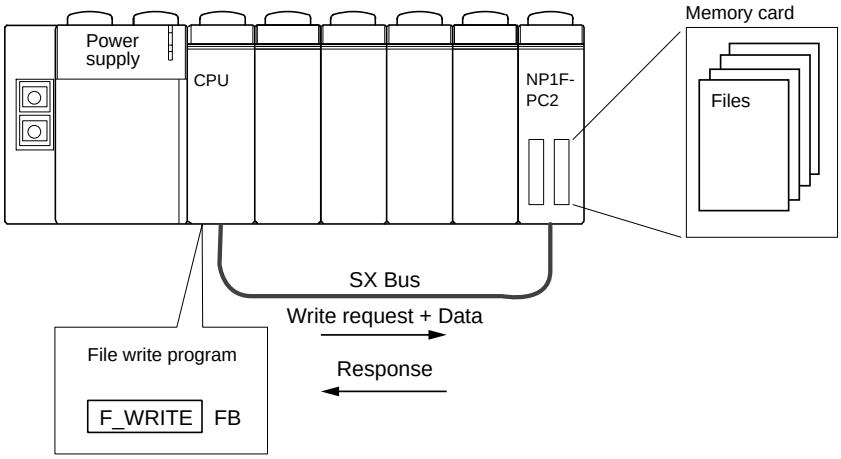
Using Application Program System FB, [FILE DATA READ/WRITE], data (variables/ data) is read from or written into the memory card.

8-2-1 Data flow

(1) Data read from memory card to PC (CPU)
Using “File Data Read Command (F_READ)” in an application program, the specified file data on the Memory Card is read into the specified variable area of PC (CPU) memory.

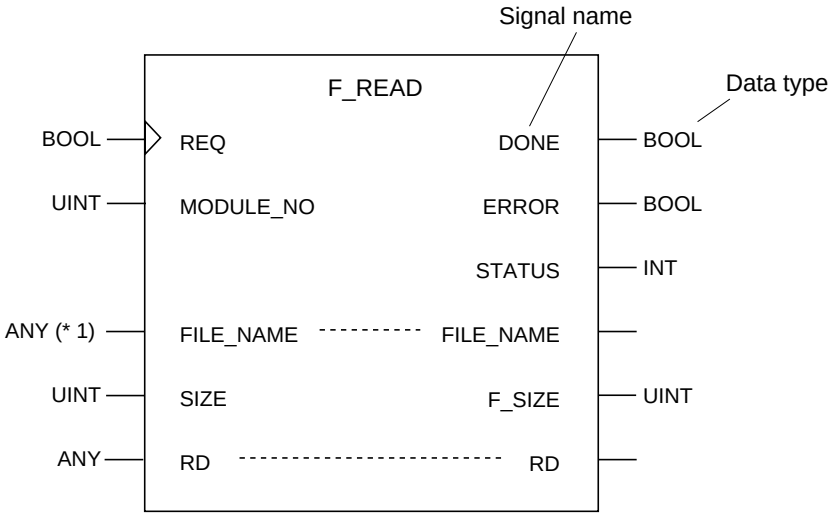


(2) Data write from PC (CPU) to memory card
Using “File Data Write Command (F_WRITE)” in an application program, data in specified variable in CPU memory is written into the specified file on the Memory Card.



8-2-2 File data read (F_READ)

File Data Read (F_READ) FB reads files stored in the memory cards installed in NP1F-PC2 into CPU memory.



* 1: Set the file name in characters converted to Shift JIS Code.

<Operation>

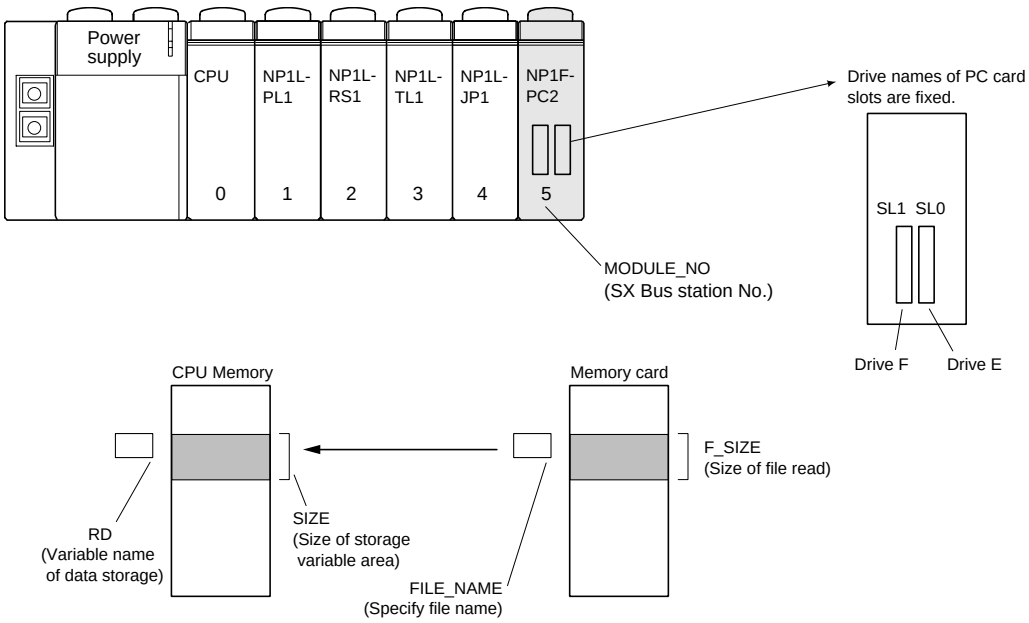
1) At the rising edge (0 to 1) of “REQ,” the file specified with “FILE_NAME” on the memory card installed in the NP1F-PC2 specified with “MODULE_NO” into the variable area specified with “RD.” (File read operation takes more than 1 scan.)
2) When read operation completes successfully, “DONE” becomes “1” for 1 scan.
3) When the operation is erroneous, “ERROR” becomes “1” for 1 scan and the error code is output to “STATUS.”

<Note>

1) “FILE_NAME” is specified in Shift JIS Code. (For converting a string to Shift JIS Code, function “STR_TO_SJ” is used.)
2) “REQ” is effective at the rising edge, but during file read (from rising edge of “REQ” to “DONE” or “ERROR” pulse) the transition of “REQ” (0 to 1) is invalid.
3) Do not change “RD2 during file read. If not the integrity of the data read is not guaranteed.
4) If number of data specified with “SIZE” exceeds variable area size specified with “RD,” data read may rewrite the other variable area. Make sure the number specified with “SIZE” is less than the variable area size.

8-2 Data Read/Write from Program

Data flow and signal names are shown below.

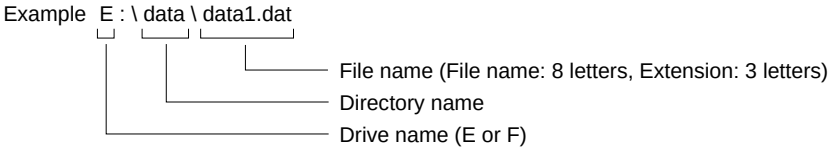


Classification	Symbol	Name	Data type	Description
Input	REQ	Send request	BOOL	At rising edge (0 to 1) starts transmission (execute command).
	MODULE_NO	Communication module No. (SX Bus station No.)	UINT	The SX Bus station No. of the NP1F-PC2 Module where the memory card with subject file is installed.
	SIZE	File size	UINT	File size to be read from the memory card. (in word unit)
Output	DONE	Complete output	BOOL	Output when execution result is normal. (Pulse Output)
	ERROR	Error output	BOOL	Output when execution result is abnormal. (Pulse Output)
	STATUS	Error status	INT	Error code is output when execution is erroneous.
	F_SIZE	Read variable size (No. of words read)	UINT	Stores file size read from memory card. (in word unit)
Input/Output	RD	Storage variable	ANY	Storage variable name in CPU memory for the data read.
	FILE_NAME * 1	File name	ANY	File name of subject file to be read. (Character Code: Shift JIS Code) Example: "E:\data\M0219.dat"

BOOL: Bit
UINT: Unsigned Integer
INT: Integer
ANY: Data other than BOOL

*** 1 File name setting**

- In FILE_NAME, drive name, directory and file name may be specified.



* File name must be maximum 64 characters.

- File name is in Shift JIS Code (alphanumeric) characters. (For converting a string to Shift JIS Code, function “STR_TO_SJ” is used.)

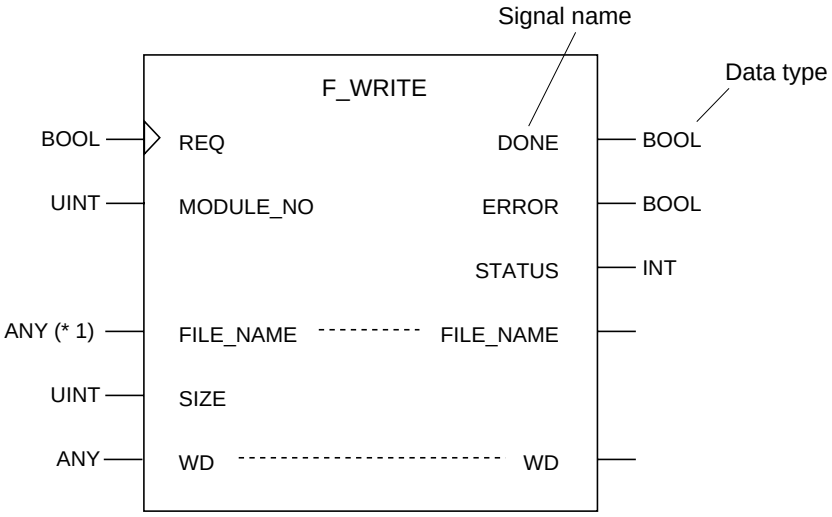
<STATUS>
File Read “F_READ” command status codes are listed here.

Name	Code	Error description
File name error	101 (65h)	File specified does not exist.
File access error	102 (66h)	SUM Check Error occurred during file read.
Input parameter error	177 (B1h)	• Number specified with "MODULE_NO" is out of range of SX Bus Station No. (1 to 254). • String variable length exceeds 64 characters.

Refer to Appendix 1-1 for the details of common status of communication FB.

8-2-3 File data write (F_WRITE)

File Data Write (F_WRITE) FB writes CPU memory data into a file in the memory cards installed in NP1F-PC2.



* 1: Set the file name in characters converted to Shift JIS Code.

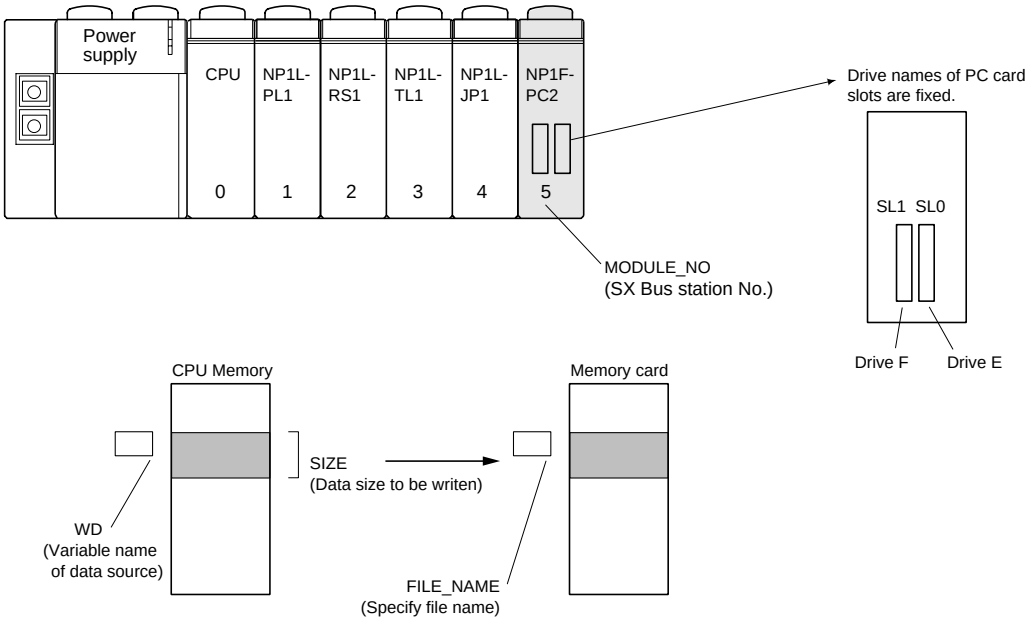
<Operation>

1) At the rising edge (0 to 1) of “REQ,” the data specified with “WD” is written into the file specified with “FILE_NAME” on the memory card installed in the NP1F-PC2 specified with “MODULE_NO.”
2) When write operation completes successfully, “DONE” becomes “1” for 1 scan.
3) When the operation is erroneous, “ERROR” becomes “1” for 1 scan and the error code is output to “STATUS.”

<Note>

1) “FILE_NAME” is specified in Shift JIS Code. (For converting a string to Shift JIS Code, function “STR_TO_SJ” is used.)
2) “REQ” is effective at the rising edge, but during file writing (from rising edge of “REQ” to “DONE” or “ERROR” pulse) the transition of “REQ” (0 to 1) is invalid.
3) Do not change “WD” during file writing. If not, the integrity of the data written is not guaranteed.
4) When the same name is filed, the file is overwrited.
5) If number of data specified with “SIZE” exceeds variable area size specified with “WD,” excess data written in is undefined. Make sure to specify the number less than the variable area for “SIZE.”

Data flow and signal names of the FB are shown below.



Classification	Symbol	Name	Data type	Description
Input	REQ	Send request	BOOL	At rising edge (0 to 1) starts transmission (execute command).
	MODULE_NO	Communication module No. (SX Bus station No.)	UINT	The SX Bus station No. of the NP1F-PC2 Module where the memory card is installed.
	SIZE	Write data size	UINT	Data size to be written into the memory card. (in word unit)
Output	DONE	Complete output	BOOL	Output when execution result is normal. (Pulse output)
	ERROR	Error output	BOOL	Output when execution result is abnormal. (Pulse output)
	STATUS	Error status	INT	Error code is output when execution is erroneous.
Input/Output	WD	Source variable	ANY	Data source variable name in CPU memory.
	FILE_NAME * 1	File name	ANY	File name of data to be written. (Character code: Shift JIS code)

BOOL: Bit
UINT: Unsigned Integer
INT: Integer
ANY: Data other than BOOL

*** 1 File name setting**

- In FILE_NAME, drive name, directory and file name may be specified.

Example E : \ data \ data1.dat

File name (File name: 8 letters, Extension: 3 letters)
Directory name
Drive name (E or F)

* File name must be maximum 64 characters.

- File name is in Shift JIS Code (alphanumeric) characters. (For converting a string to Shift JIS Code, function “STR_TO_SJ” is used.)

<STATUS>
The File Write “F_WRITE” command status codes are listed.

Name	Code	Error description
File name error	101 (65h)	• Specified directory does not exist. • If specified file name does not exist, a new file is created.
File access error	102 (66h)	Write error to the memory card occurred.
No space	105 (69h)	Writing is not possible due to the space shortage.
Input parameter error	177 (B1h)	• Number specified with "MODULE_NO" is out of range of the SX Bus station No. (1 to 254). • String variable length exceeds 64 characters.

Refer to Appendix 1-1 for the details of common status of communication FB.

Section 9 PC Card Installation

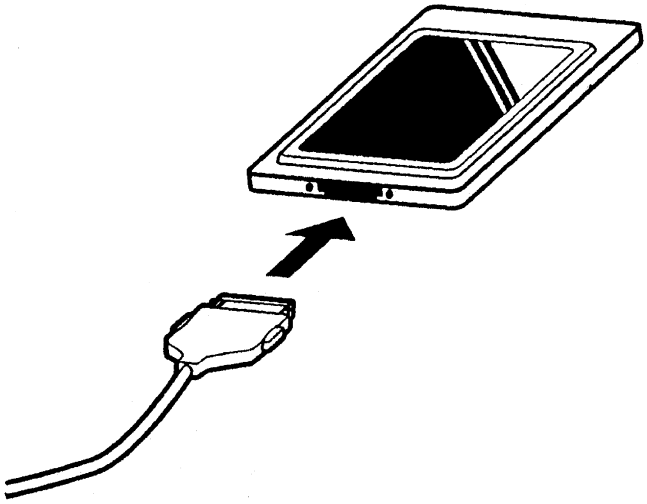
Page

Section 9 PC Card Installation
9-1 PC Card Installation

9-1-1 PC Card installation procedure

(1) Attaching communication adaptor to card

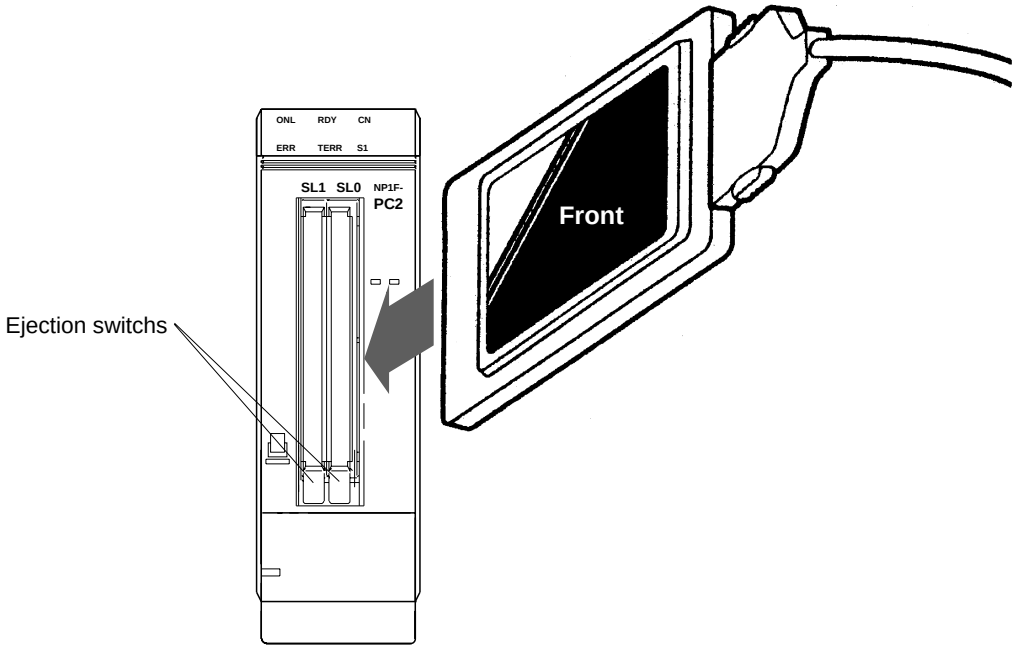
Connect adaptor supplied with communication card as following diagram. Pay attention to the polarization of the connector and make sure the connector is fully engaged.



(2) Installing PC Card into NP1F-PC2

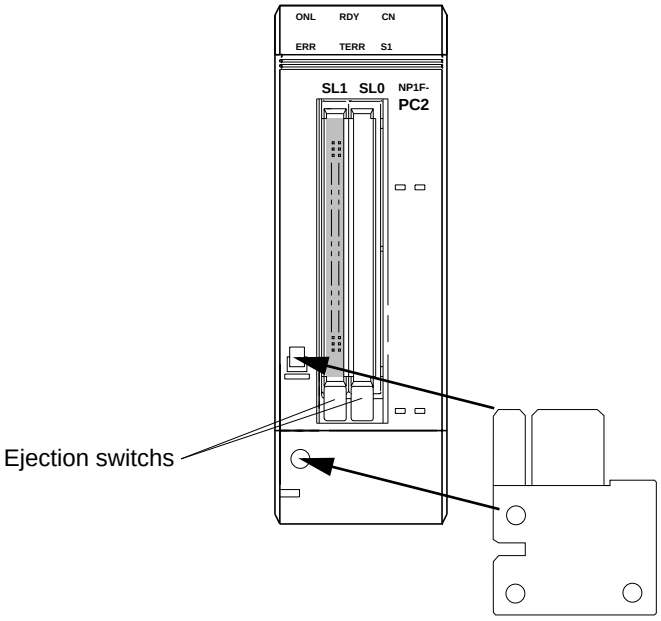
Install the communication card in the PC Card slots as shown in following diagram. Push the card in until eject button is flush with the card.

Memory Cards are installed in the same way.



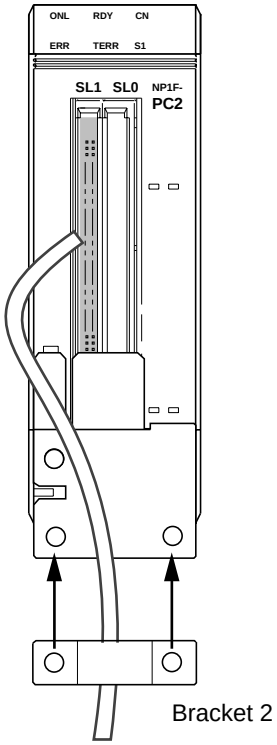
9-1 PC Card Installation

(3) Bracket 1 installation (for fixing card)
Fix the bracket 1 with the screw to the position indicated with arrows in following diagram.



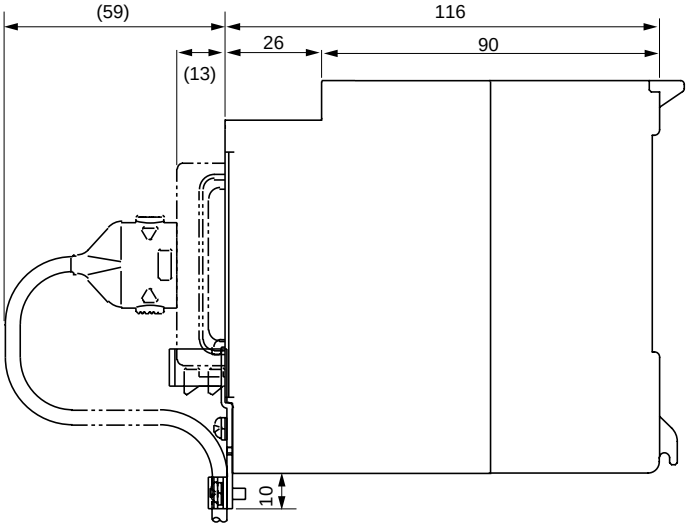
Bracket 1

(4) Bracket 2 installation (for fixing cable)
As shown in following diagram, put through the cable between bracket 1 and 2 and fix the bracket 2 with screws.



Bracket 2

Note: When a communication card is installed, give allowance for the cable bending and the cable fixing brackets.



9-1-2 Procedure for PC Card removal

1) Undo the screws and remove the bracket 2.
2) Undo the screw and remove the bracket 1.
3) Press the eject button and remove the card.

Note:

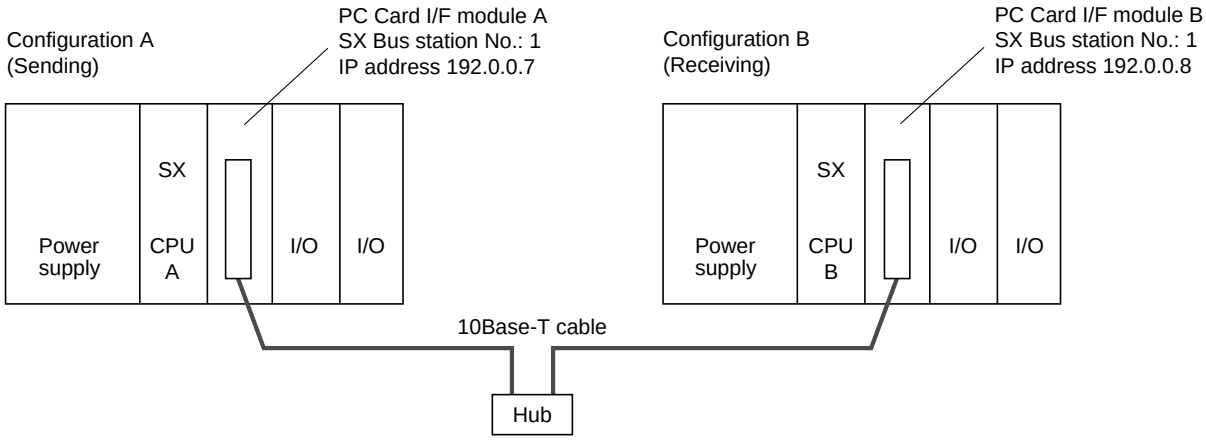
1) Do not press the eject button without removing brackets.
2) Do not remove the card with the module power ON.

Section 10 Application Examples

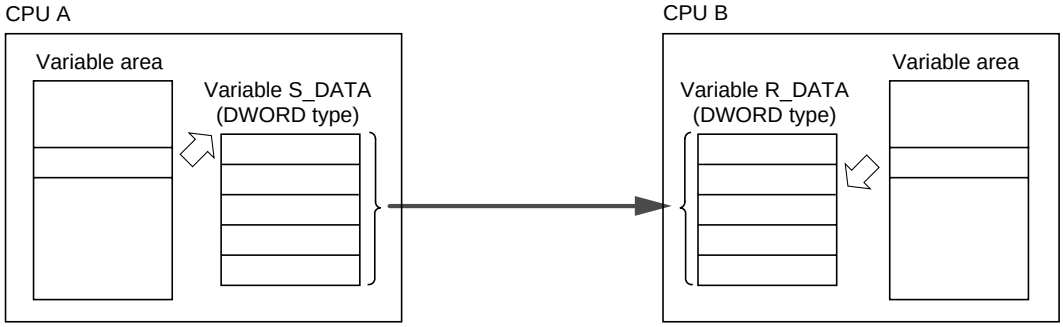
10-1-1 Universal communication mode

For the system shown in the diagram below, an example of program to send data from A to B using Channel Open "M_OPEN," Message Send "M_SEND" and Message Receive "M_RECEIVE" commands is explained.

<System structure>



<Program image>



■ Details of program

1) Execute "M_OPEN" at receiving CPU: B to open a channel.
2) Execute "M_RECEIVE" at receiving CPU: B to put it in waiting state.
3) Execute "M_OPEN" at sending CPU: A to open a channel.
4) Execute "M_SEND" at sending CPU: A to send data to CPU B.

■ Send program

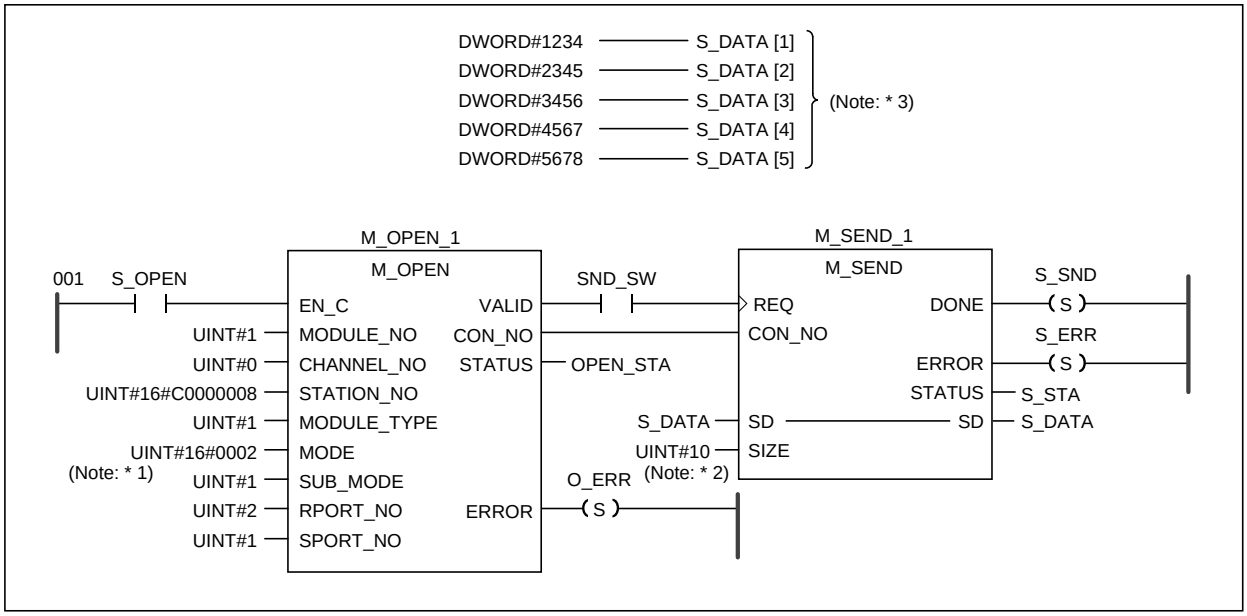
<Variable declaration>

```
VAR  ( * AUTOINSERT * )
      S_SEND  :      BOOL;
      S_ERR   :      BOOL;
      SND_SW  :      BOOL;
      M_OPEN_1:      M_OPEN;
      M_SEND_1:      M_SEND;
      OPEN_STA:      INT;
      S_STA   :      INT;
      O_ERR   :      BOOL;
      S_DATA  :      ARRAY1;
      S_OPEN  :      BOOL;
END_VAR
```

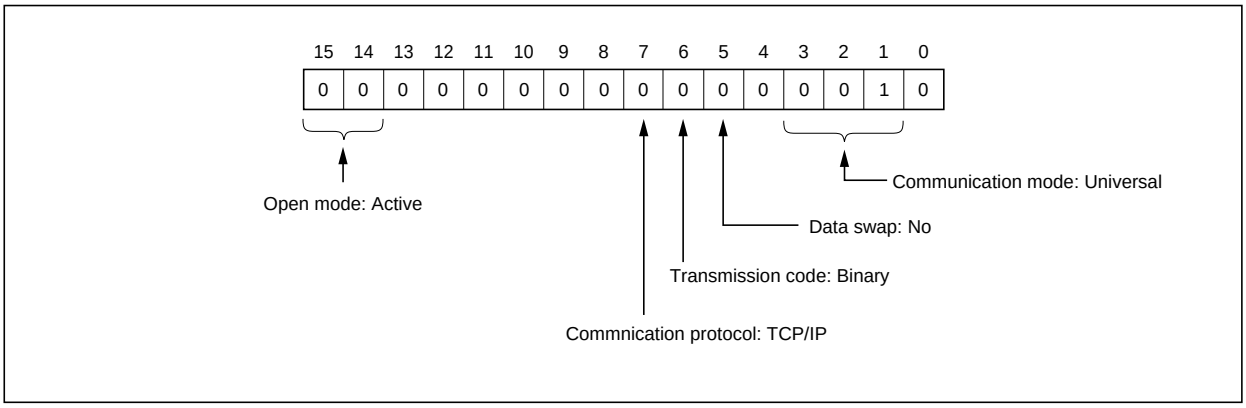
<Data type declaration>

```
TYPE
   ARRAY1    : ARRAY1  [1..5]  OF DWORD;
END_TYPE
```

<Code>



Note * 1: Communication mode is as shown in following diagram.

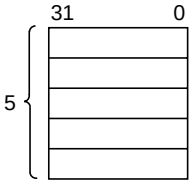


Note * 2: Point to be noticed about data size.

- Sending data size is specified with number of words. Therefore, DWORD type data size is twice of the area size.

Example:

Data transmission size for “S_DATA (ARRAY (1..5) OF WORD)”



It is 10 words. (Not 5 words!!)

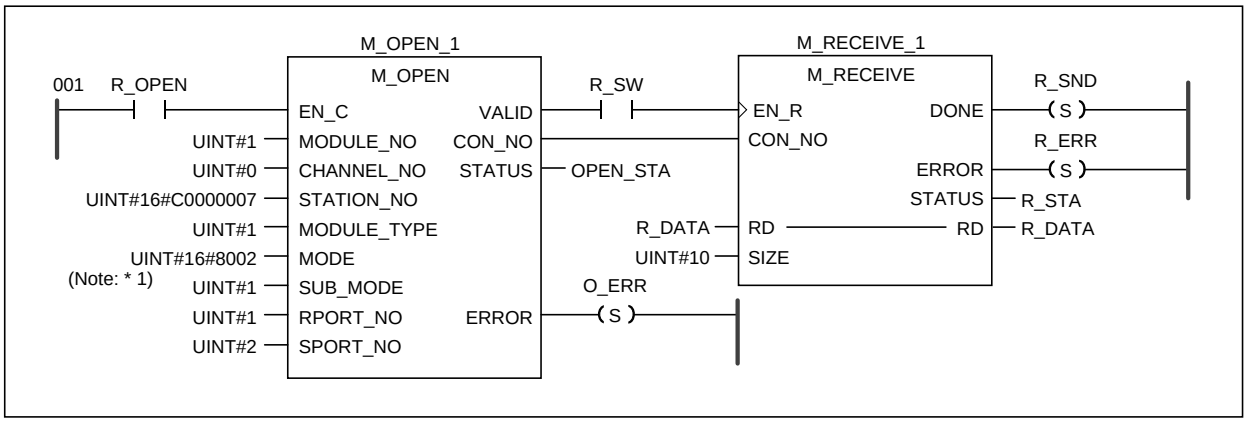
Note * 3: S_DATA [1 to 5] are the contents of S_DATA array.

■ Receive program

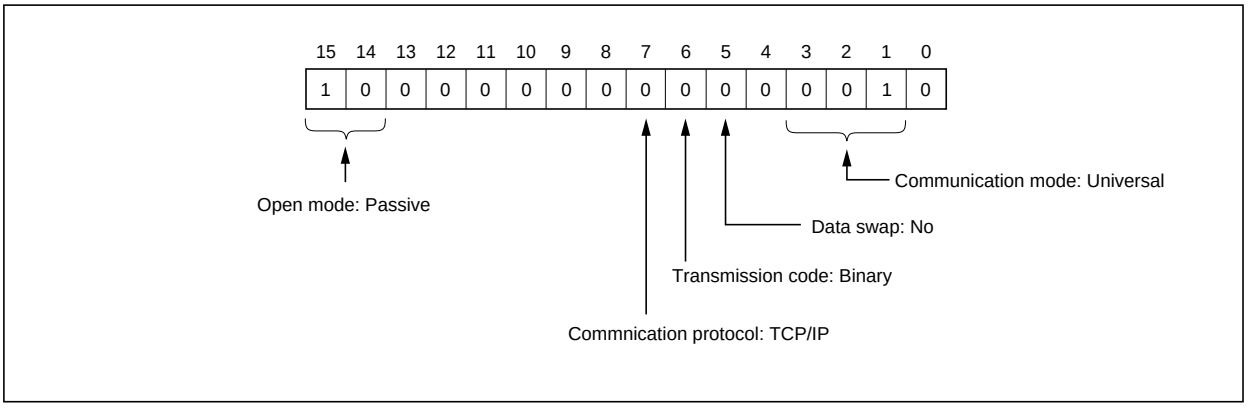
<Variable declaration>

```
VAR ( * AUTOINSERT * )
      R_SW     :       BOOL;
      R_SND    :       BOOL;
      R_ERR    :       BOOL;
      M_RECEIVE_1: M_RECEIVE;
      R_STA:   INT;
      R_DATA   :       ARRAY1
      O_ERR    :       BOOL;
      M_OPEN_1:        M_OPEN;
      OPEN_STA:        INT;
      R_OPEN:          BOOL;
END_VAR
```

<Code>



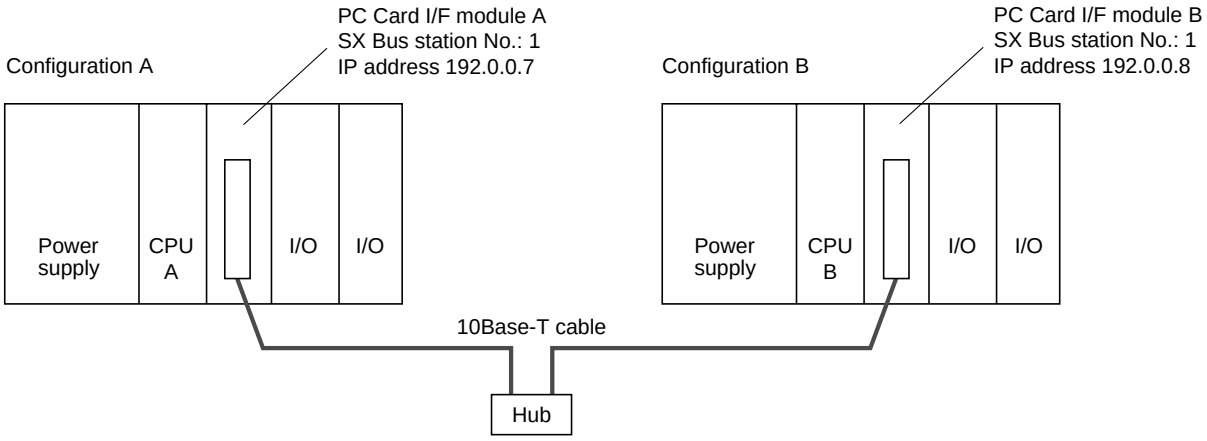
Note 1: Communication mode is as shown in following diagram.



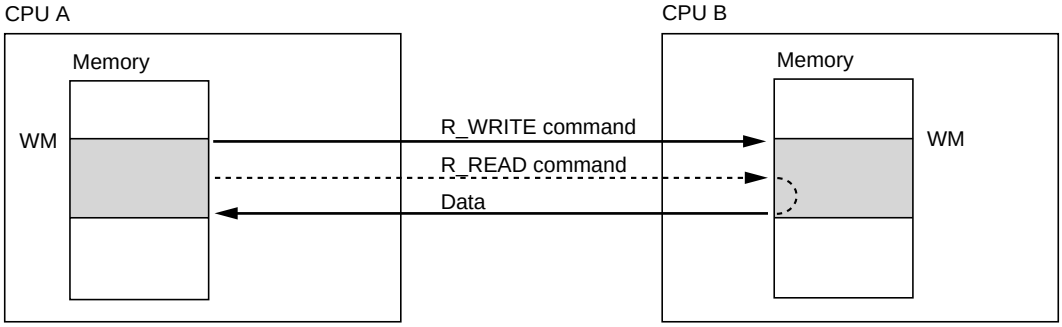
10-1-2 Auto transmission mode

For the system shown in the diagram below, an example of program to send data from A to B and B to A using Remote Data Read “R_READ” and Remote Data Write “R_WRITE” commands is explained.

<System structure>



<Program image>



■ **Details of program**

After power up, following operations are repeated automatically.

1) “R_WRITE” in CPU A program writes data in CPU A memory area specified with WT_DATA into CPU B memory area specified with R_VAR.
2) “R_READ” in CPU A program reads data in CPU B memory area specified with R_VAR into CPU A memory area specified with RD_DATA.
3) At the completion of each command a counter is incremented.
4) With “DT_CHG,” data is changed for every transmission.

<Data type declaration>

```
TYPE
      ARY4      : ARRAY [1..4] OF UINT;
      ARY14     : ARRAY [1..4] OF UINT;
      ARY260    : ARRAY [1..7952] OF UINT;
      ARY512    : ARRAY [1..512] OF UINT;
      _NAME     : ARRAY [0..63] OF WORD;
      ARY4096 : ARRAY [1..5000] OF INT;
END_TYPE
```

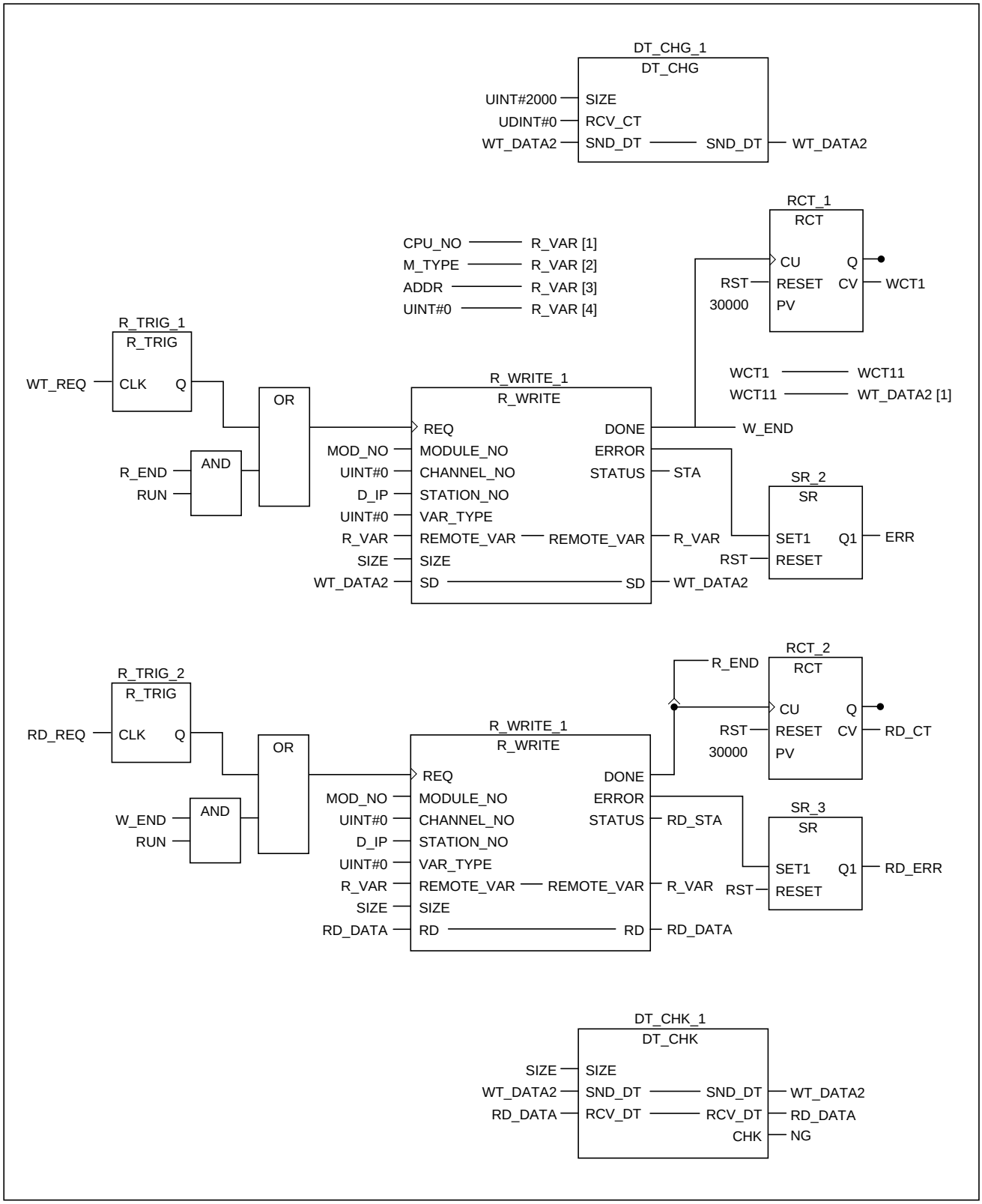
```
TYPE    BIT_ARY3:
      ARRAY [1..48] OF BOOL;
END_TYPE

TYPE    BIT_ARY4:
      ARRAY [1..500] OF BOOL;
END_TYPE

TYPE    WORD_ARRAY:
      ARRAY [1..500] OF UINT;
END_TYPE


TYPE    BIT_2ARY:
      ARRAY [1..4,1..3] OF BIT_ARY3;
END_TYPE
```

<Variable declaration>

```
VAR ( * AUTOINSERT * )
      SR_2      :       SR;
      STA       :       INT;
      ERR       :       BOOL;
      SIZE      :       UINT:=100;
      R_WRITE_1:        R_WRITE;
      RST       :       BOOL;
      M_TYPE    :       UINT:=1;
      ADDR      :       UINT:=2048;
      D_IP      :       UDINT:=16#c000000a;
      MOD_NO:           UINT:=1;
      CPU_NO    :       UINT:=0;
      RCT_1     :       RCT;
      WCT       :       INT;
      R_READ_2:         R_READ;

      WT_REQ    :       BOOL;=TRUE;
      RD_REQ    :       BOOL;
      RCT_2     :       RCT;
      SR_3      :       SR;
      RD_STA    :       INT;
      RD_CT     :       INT;
      RD_ERR    :       BOOL;
      DT_CHG_1:         DT_CHG;
      DT_CHK_1:         DT_CHK;
      NG        :       BOOL;
      R_TRIG_1:         R_TRIG;
      W_END     :       BOOL;
      R_END     :       BOOL;
      R_TRIG_2:         R_TRIG;
      RUN       :       BOOL;=TRUE;
      R_VAR     :       ARY4;
      RD_DATA           AT %MW1.2048:  ARY260;
      WT_DATA2          AT %MW1.2048:  ARY260;
      WCT1      :       INT;
      WCT11     :       INT;
END_VAR
```

10-1 Ethernet

<Code>



■ **“DT_CHG_1” FB**

Variables and Code of the “DT_CHG_1” used in the code are shown below.

<Variable declaration>

```
VAR ( * AUTOINSERT * )
      LP_CNT   :      INT;
END_VAR
VAR_INPUT ( * AUTOINSERT * )
      SIZE     :      UINT;
 END_VAR
VAR_IN_OUT ( * AUTOINSERT * )
      SND_DT   :      ARY4096;
      RCV_DT   :      ARY4096;
END_VAR
VAR_OUTPUT ( * AUTOINSERT * )
      CHK      :      BOOL;
END_VAR
```

<Code>

```
CHK:= BOOL#0
FOR LP_CNT:= INT#1 TO UINT_TO_INT (SIZE) BY 1 DO
      IF SND_DT [LP_CNT] <> RCV_DT [LP_CNT] THEN
          CHK:= BOOL#1;
      END_IF;
 END_FOR;
```

■ **“DT_CHG_1” FB**

Variables and Code of the “DT_CHK_1” used in the code are shown below.

<Variable declaration>

```
VAR ( * AUTOINSERT * )
      LP_CNT   :      INT;
END_VAR
VAR_INPUT ( * AUTOINSERT * )
      SIZE     :      UINT;
      RCV_CT   :      UDINT;
 END_VAR
VAR_IN_OUT
      SND_DT   :      ARY4096;
END_VAR
```

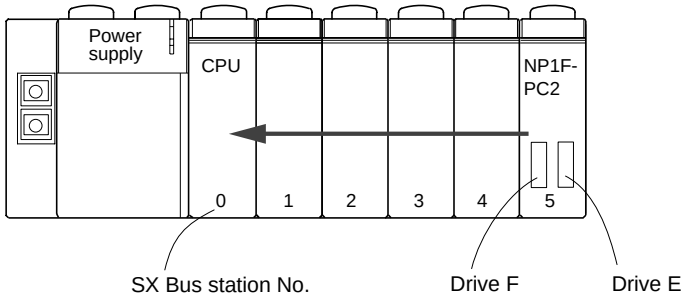
<Code>

```
SND_DT [1] :UDINT_TO_INT (RCV_CT) +1;

FOR LP_CNT:= INT#1 TO UINT_TO_INT (SIZE) BY 1 DO
      SND_DT [LP_CNT] :=UDINT_TO_INT (RCV_CT) +LP_CNT;
      END_FOR;
```

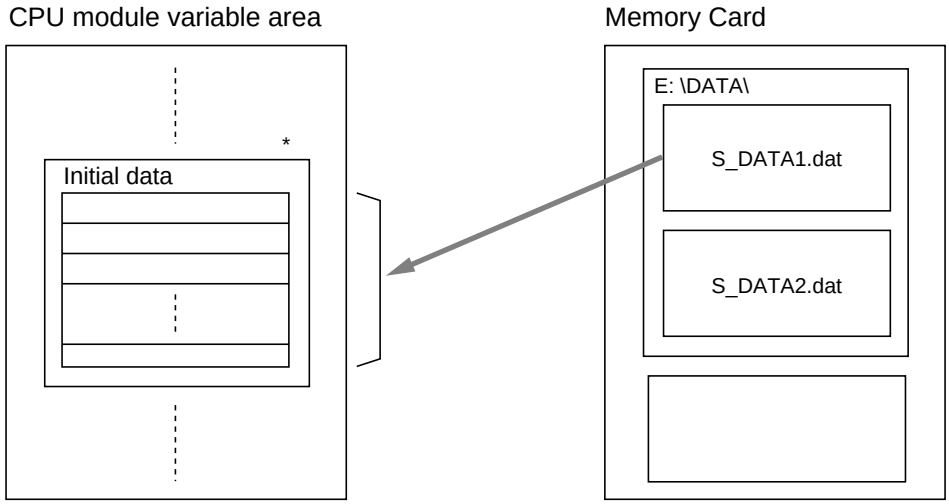
10-2 Memory Card

10-2-1 Data read from CPU program

Following example program reads data stored on the memory card installed in right slot of NP1F-PC2 using File Data Read “F_READ” command.



<Program image>



* Variable “Initial data” is an array of Type INT.

■ **Details of program**

1) With “SEL_FILE,” select two files in the memory card [S_DATA1.dat] and [S_DATA2.dat].
2) Read file data in with “RD_SW.”

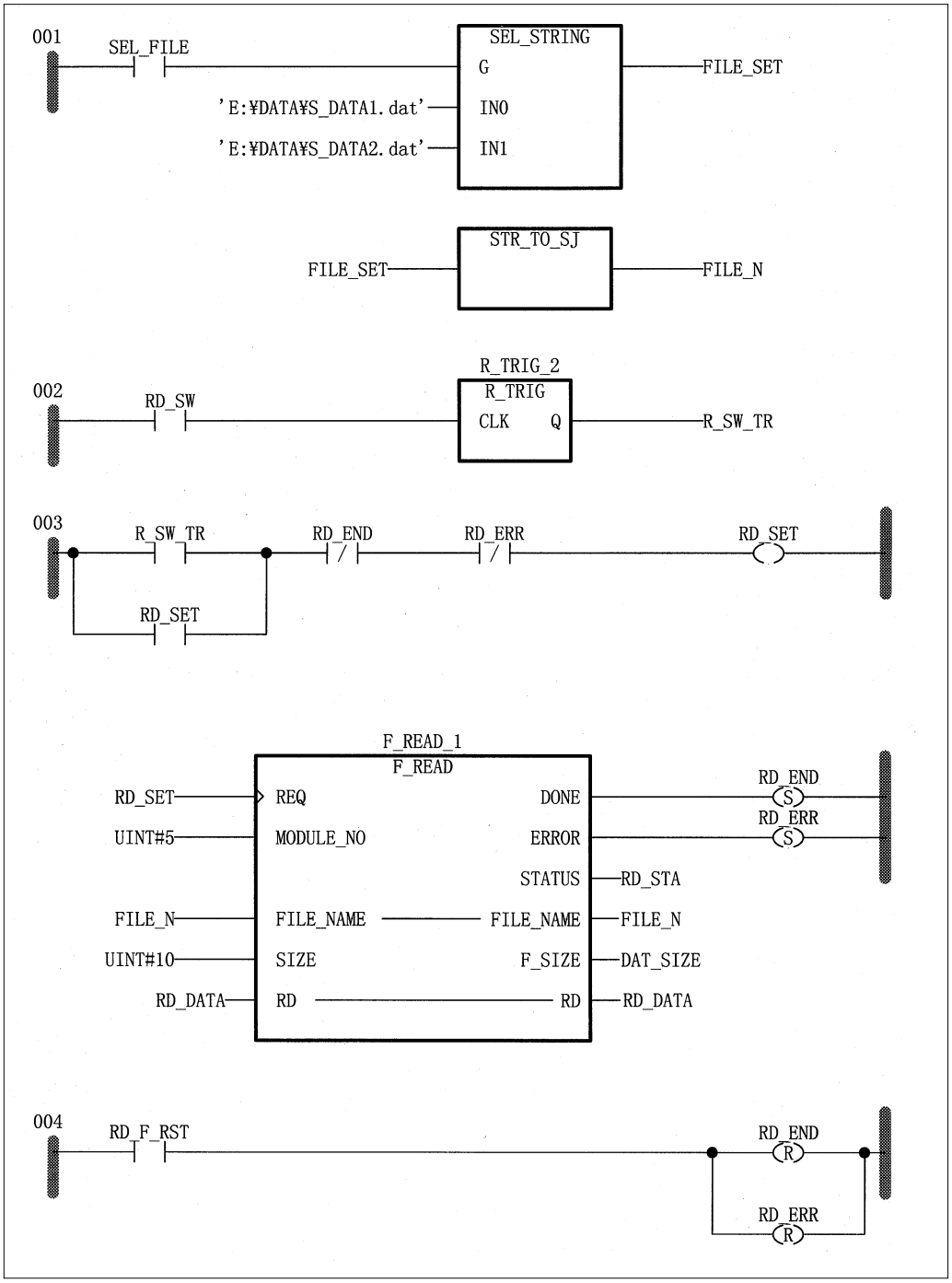
<Variable declaration>

```
VAR
    FILE_SET   :   STRING;     ( * SELECTED FILE NAME * )
    FILE_N     :   FILE_N;     ( * FILE NAME * )
    R_TRIG_2   :   R_TRIG;
    R_SW_TR    :   BOOL;       ( * READ SWITCH TRANSITION * )
    RD_END     :   BOOL;       ( * READ COMPLETE * )
    RD_ERR     :   BOOL;       ( * READ ERROR * )
    RD_SET     :   BOOL;       ( * READ SET SWITCH * )
    F_READ_1   :   F_READ;
    RD_STA     :   INT;        ( * READ FB STATUS CODE * )
    DAT_SIZE   :   UINT;       ( *  READ DATA SIZE * )
    RD_DATA    :   FST_DATA;   ( *  READ DATA * )
    SEL_FILE   AT %IX1.0.15 :  BOOL;  ( * FILE NAME SELECT * )
    RD_SW      AT %IX2.0.0  :  BOOL;  ( * READ SWITCH * )
    RD_F_RST   AT %IX2.0.2  :  BOOL;  ( * READ FB FLAG RESET * )
END_VAR
```

<Data type declaration>

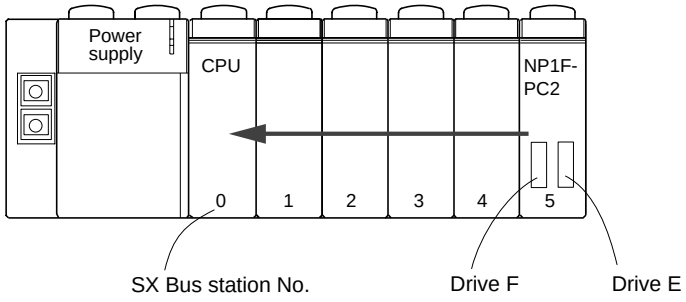
```
TYPE
      ( * FILE NAME DEFINITION * )
      FILE_N: ARRAY [0..63] OF WORD;
END_TYPE

TYPE
      ( * INITIAL DATA DEFINITION * )
      FST_DATA: ARRAY [0..10] OF INT;
END_TYPE

TYPE
      ( * INITIAL READ DATA DEFINITION * )
     R_FS_DAT: ARRAY [0..10] OF INT;
END_TYPE
```

<Code>

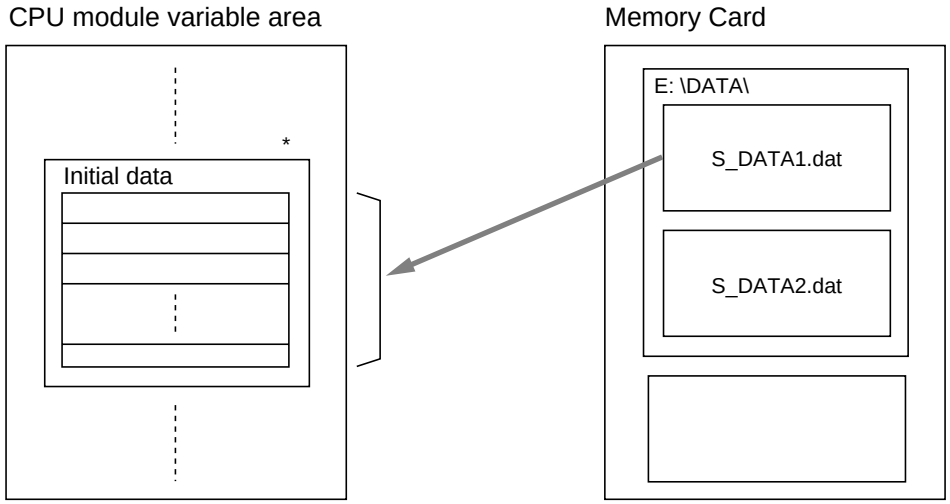

001
SEL_FILE
SEL_STRING
G
FILE_SET
'E:¥DATA¥S_DATA1.dat'
IN0
'E:¥DATA¥S_DATA2.dat'
IN1
STR_TO_SJ
FILE_SET
FILE_N
R_TRIG_2
002
RD_SW
R_TRIG
CLK
Q
R_SW_TR
003
R_SW_TR
RD_END
RD_ERR
RD_SET
RD_SET
F_READ_1
F_READ
RD_SET
REQ
DONE
RD_END
S
UINT#5
MODULE_NO
ERROR
RD_ERR
S
STATUS
RD_STA
FILE_N
FILE_NAME
FILE_NAME
FILE_N
UINT#10
SIZE
F_SIZE
DAT_SIZE
RD_DATA
RD
RD
RD_DATA
004
RD_F_RST
RD_END
R
RD_ERR
R

10-2-2 Data write from CPU program

Following example program writes data into the memory card installed in right slot of NP1F-PC2 using File Data Write "F_WRITE" command.

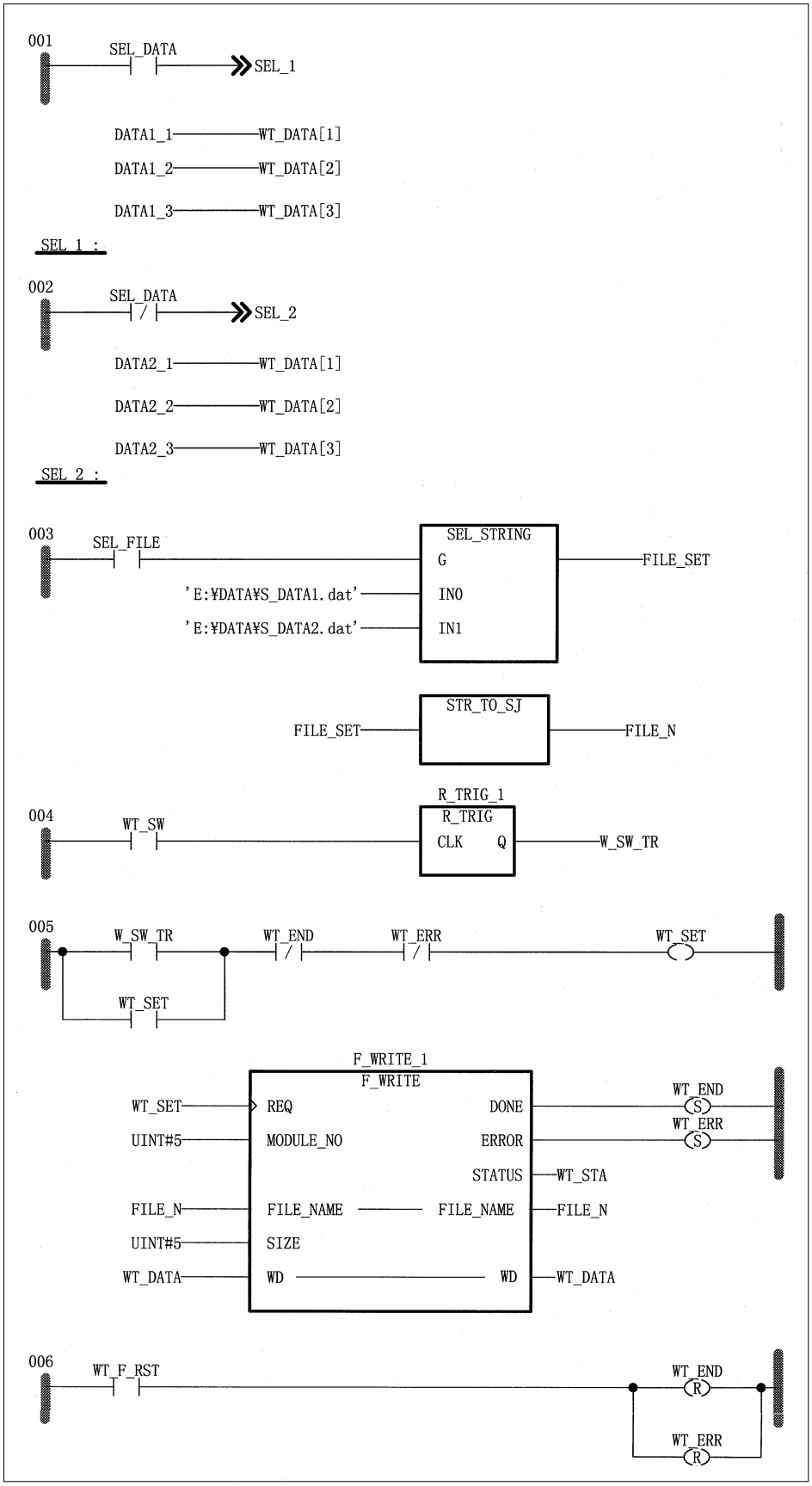


<Program image>



* Variable "Initial data" is an array of Type INT.

<Code>


001
SEL_DATA
SEL_1
DATA1_1
WT_DATA[1]
DATA1_2
WT_DATA[2]
DATA1_3
WT_DATA[3]
SEL_1 :
002
SEL_DATA
SEL_2
DATA2_1
WT_DATA[1]
DATA2_2
WT_DATA[2]
DATA2_3
WT_DATA[3]
SEL_2 :
003
SEL_FILE
SEL_STRING
G
FILE_SET
'E:¥DATA¥S_DATA1.dat'
IN0
'E:¥DATA¥S_DATA2.dat'
IN1
STR_TO_SJ
FILE_SET
FILE_N
R_TRIG_1
004
WT_SW
R_TRIG
CLK
Q
W_SW_TR
005
W_SW_TR
WT_END
WT_ERR
WT_SET
WT_SET
F_WRITE_1
F_WRITE
WT_SET
REQ
DONE
WT_END
S
UINT#5
MODULE_NO
ERROR
WT_ERR
S
STATUS
WT_STA
FILE_N
FILE_NAME
FILE_NAME
FILE_N
UINT#5
SIZE
WT_DATA
WD
WD
WT_DATA
006
WT_F_RST
WT_END
R
WT_ERR
R

10-2 Memory Card

<Variable declaration>

```
VAR
    FILE_SET    :   STRING;     ( * SELECTED FILE NAME * )
    FILE_N      :   FILE_N;     ( * FILE NAME * )
    W_SW_TR     :   BOOL;       ( * WRITE SWITCH TRANSITION * )
    WT_END      :   BOOL;       ( * WRITE COMPLETE * )
    WT_ERR      :   BOOL;       ( * WRITE ERROR * )
    WT_SET      :   BOOL;       ( * WRITE SET SWITCH * )
    WT_STA      :   INT;        ( * WRITE FB STATUS CODE * )
    R_TRIG_1    :   R_TRIG;
    F_WRITE_1   :   F_WRITE;
    WT_DATA     :   FST_DATA;   ( *  WRITE DATA * )
    SEL_DATA    AT %IX1.0.5   : BOOL; ( * SELECT SETTING DATA* )
    SEL_FILE    AT %IX1.0.15  : BOOL; ( * SELECT FILE NAME * )
    WT_SW       AT %IX1.0.0   : BOOL; ( * WRITE SWITCH * )
    WT_F_RST    AT %IX1.0.2   : BOOL; ( * WRITE FB FLAG RESET * )
END_VAR

VAR
    DATA1_1 :   INT   := 123;
    DATA1_2 :   INT   := 222;
    DATA1_3 :   INT   := 333;
    DATA2_1 :   INT   := 555;
    DATA2_2 :   INT   := 666;
    DATA2_3 :   INT   := 777;
END_VAR
```

<Data type declaration>

```
TYPE
    ( * FILE NAME DEFINITION * )
    FILE_N: ARRAY [0..63] OF WORD;
END_TYPE

TYPE
    ( * INITIAL DATA DEFINITION * )
    FST_DATA: ARRAY [0..10] OF INT;
END_TYPE

TYPE
    ( * INITIAL READ DATA DEFINITION * )
    R_FS_DAT: ARRAY [0..10] OF INT;
END_TYPE
```

Section 11 RAS

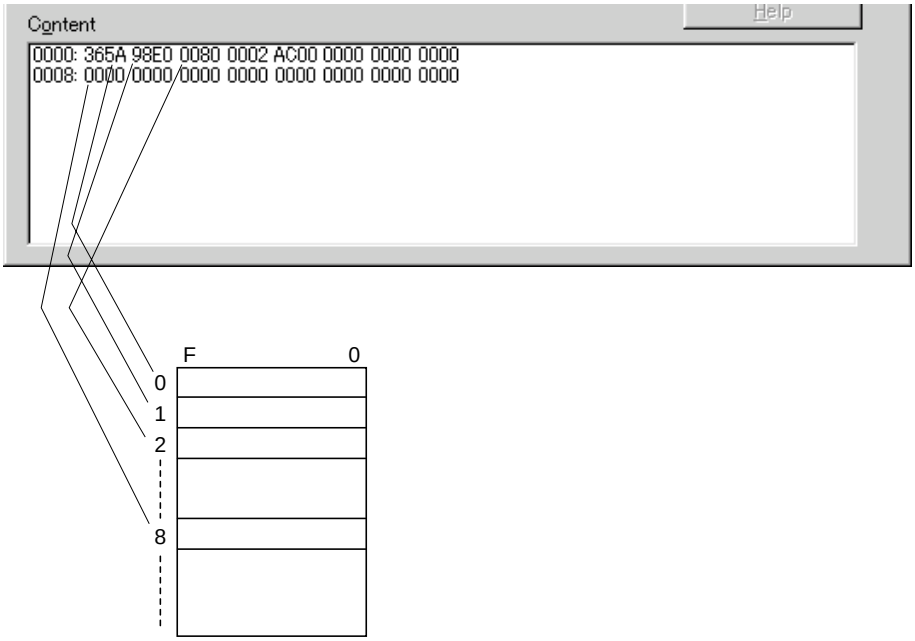
Page

Section 11 RAS
11-1 RAS Type

Status of the module is monitored using D300win. (Refer to the D300win Manual (FEH254) for the operation.)
Following status of the module can be monitored.

1) Module summary status information
2) SX Bus transmission information
3) Error logging function
4) SX Bus message communication count information
5) Initializing information
6) Communication channel information
7) Channel communication state information
8) Node connect information
9) Type information (Module type)

The screen examples used in the description of RAS, the relation between the memory contents displayed on the screen and the memory map is as follows.

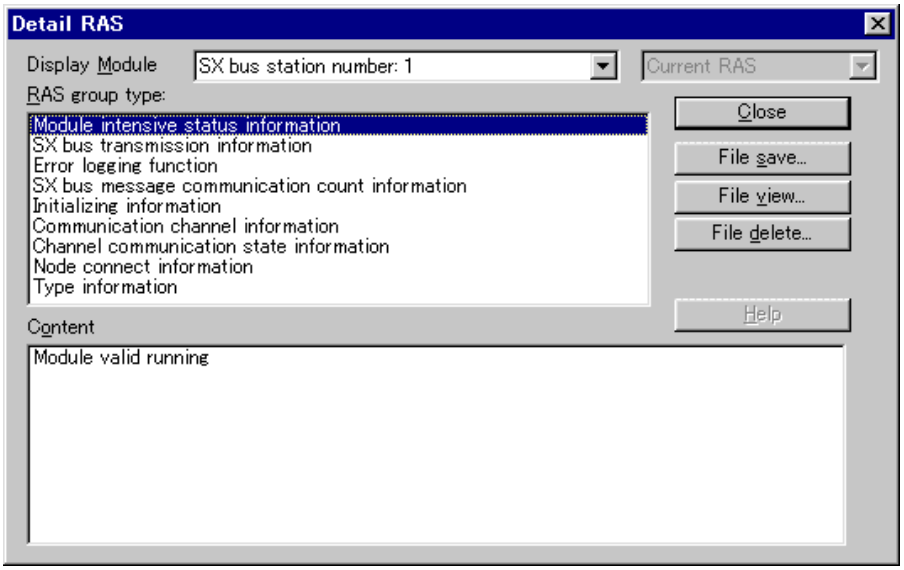


11-2 RAS Details

(1) Module summary status information
Summarized module status is displayed.

Status	Contents	Details
Normal	Running normally	
Serious fault	Watchdog timeout error	
	CPU fault	Command exception, Interrupt exception, OS error
	ROM fault	Sum check error
	RAM fault	Read after write error
	SX Bus fault	SBM fault, Timeout
	Flash ROM fault	Write error to the flash ROM
Minor fault	External interface fault	PC Card Slot 0 fault (Memory card access)
		PC Card Slot 1 fault (Memory card access)
		PC Card communication error (during transmission)
	Setup fault	Parameter error (communication initialization error)

■ **Screen example**

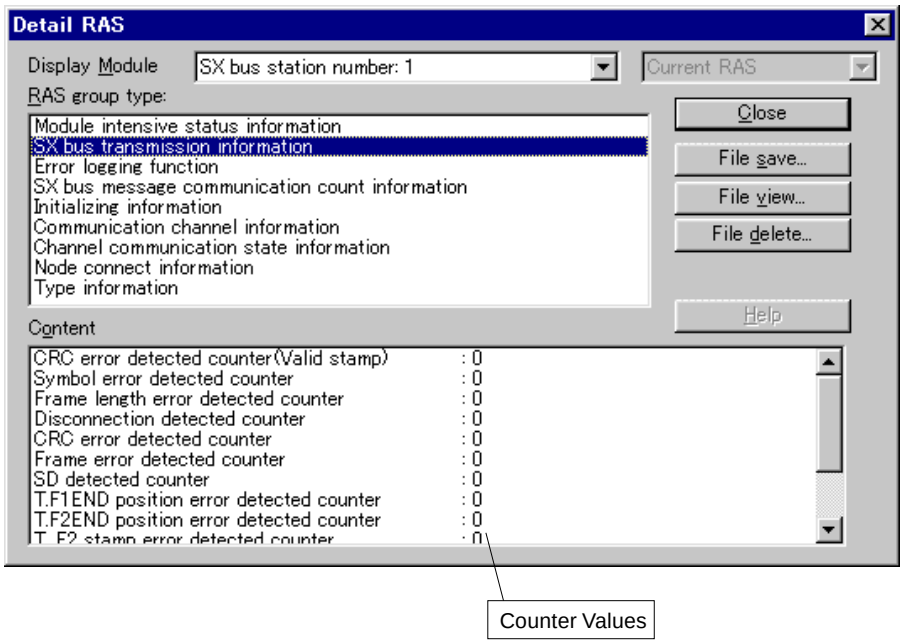


(2) SX Bus transmission information

The SX Bus transmission information is displayed.

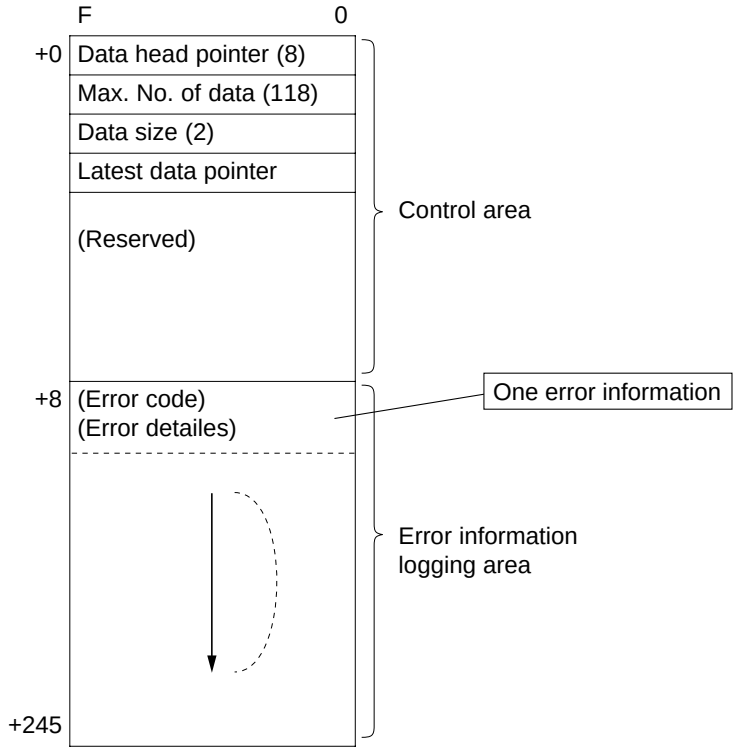
	7	0	
+0	Transmission status 1		CRC error detected counter (Valid stamp)
	Transmission status 2		Symbol error detected counter
+1	Transmission status 3		Frame length error detected counter
	Transmission status 4		Disconnection detected counter
+2	Transmission status 5		CRC error detected counter
	Transmission status 6		Frame error detected counter
+3	Transmission status 7		SD detected counter
	Transmission status 8		T.F1 END position error detected counter
+4	Transmission status 9		T.F2 END position error detected counter
	Transmission status 10		T.F2 stamp error detected counter
+5	Transmission status 11		BufferFull1 detected counter
	Transmission status 12		BufferFull2 detected counter
+6	Transmission status 13		BufferFull3 detected counter

■ **Screen example**



(3) Error logging function
Error history is displayed.

Logged information of various errors detected by the module is shown.
The error log is stored cyclic in the buffer memory area with size to store maximum data points. The latest data pointer points the position of latest error information.



1) Data head pointer
This is the pointer to the first error information in the buffer for the error logging. The offset from the head address of the error logging buffer memory is stored.

2) Maximum number of data
Defines maximum number of error information to be stored.

3) Data size
Number of words for each error log.

4) Latest data pointer
This is the pointer to the latest error log. It stores offset from the head address of the error logging buffer memory.

5) Error information

1) Data area where the error information is stored forms a ring buffer.

Error information < Max. No. of data

Address	Contents	
8	XXXX H	← Oldest error information
9	XXXX H	
10	XXXX H	
11	XXXX H	
12	XXXX H	← Latest error information
13	XXXX H	(Latest data point)
14	0000 H	
15	0000 H	
~	~	
246	0000 H	
247	0000 H	

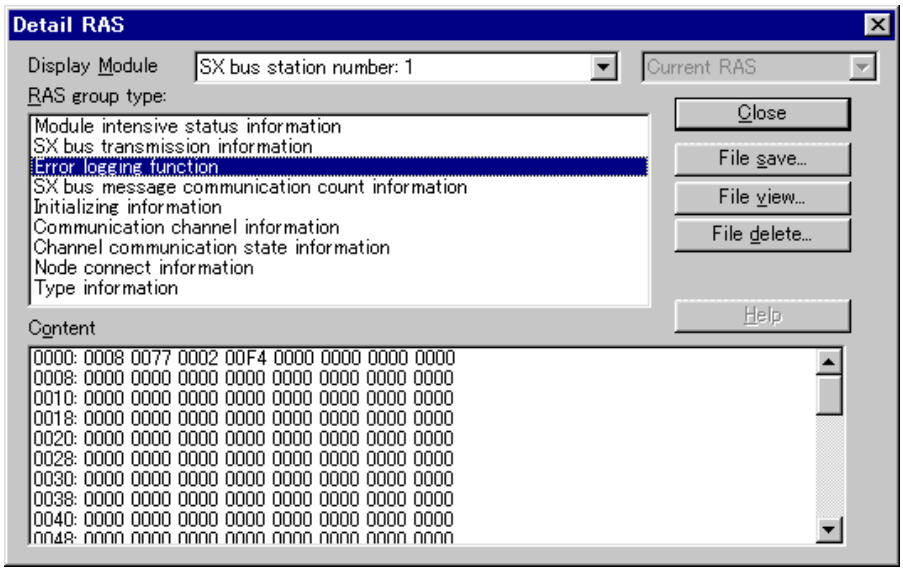
Error information > Max. No. of data

Address	Contents	
8	XXXX H	
9	XXXX H	
10	XXXX H	
11	XXXX H	
12	XXXX H	← Latest error information (Latest data point)
13	XXXX H	
14	XXXX H	← Oldest error information
15	XXXX H	
~	~	
246	XXXX H	
247	XXXX H	

2) At the start of communication, buffer is cleared to 0 and Latest Data pointer points the head address of data area.

3) For the details of error code, refer to the List of Error code.

■ **Screen example**



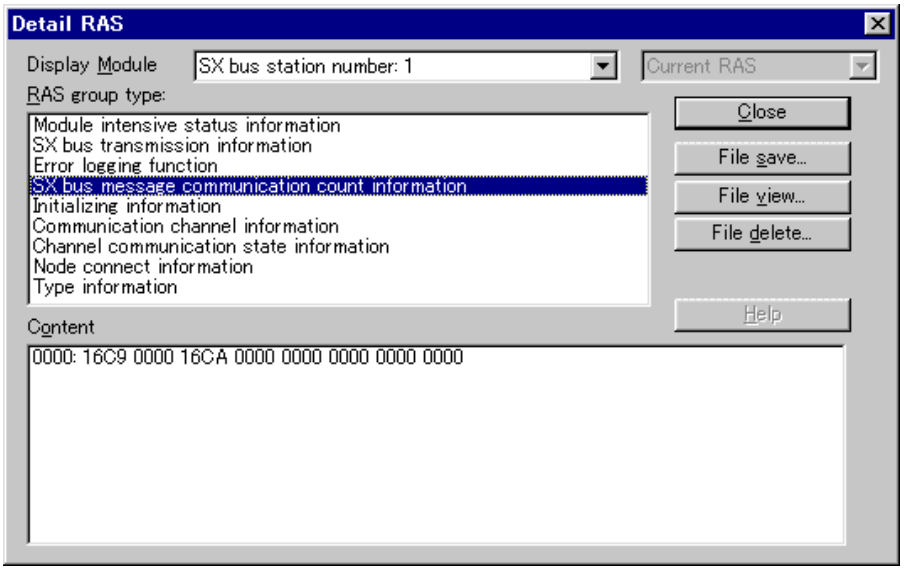
11-2 RAS Details

(4) SX Bus message communication count information

In the SX Bus message communication, cumulative status of communication is stored.

	F		0	
+0		SX Bus send Count	(L) (H)	SX Bus send normal complete aggregate
+2		SX Bus receive Count	(L) (H)	SX Bus receive normal complete aggregate
+4		SX Bus send error Count	(L) (H)	SX Bus send error aggregate
+6		SX Bus NACK Count	(L) (H)	SX Bus received NACK aggregate

■ **Screen example**



(5) Initializing information
Initial basic data of this module is stored.

■ **Ethernet communication**

	F		0
+0		Own module	(L)
		Ethernet address	
+2			(H)
		Own module	(L)
+4		IP address	(H)
+5		Reserved	
+15			

1) Own module Ethernet address
Own module's Ethernet Address is stored as binary data.

Example:
When Ethernet Address is 00401A10100Fh, data is stored as following diagram.

Address	Contents
0	100Fh
1	1A10h
2	0040h

2) Own module IP address
Own module's IP Address stored in Initializing Data File at the start of communication is stored as binary data.

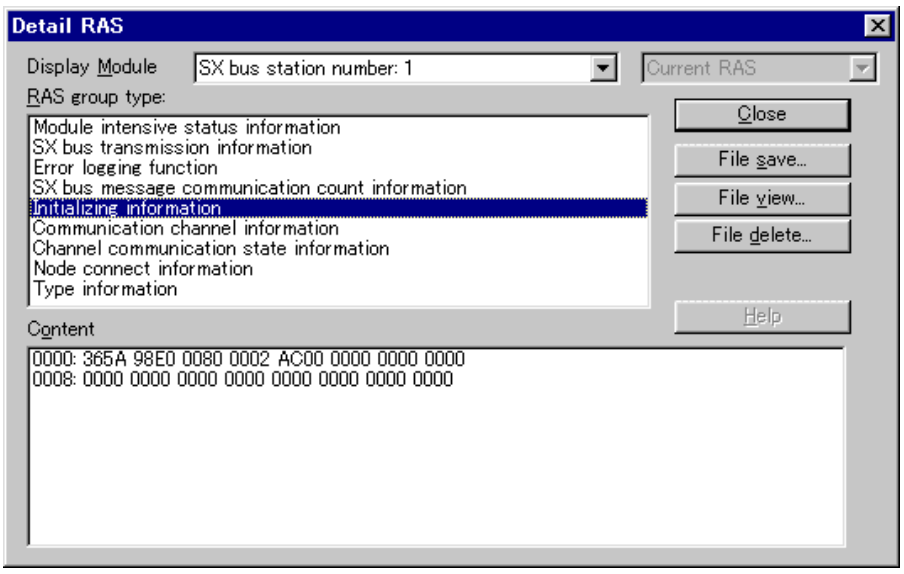
Example:
When IP Address is 172.16.0.1, data is stored as following diagram.

172	16	0	1
ACh	10h	00h	01h
172	16	0	1

⇩

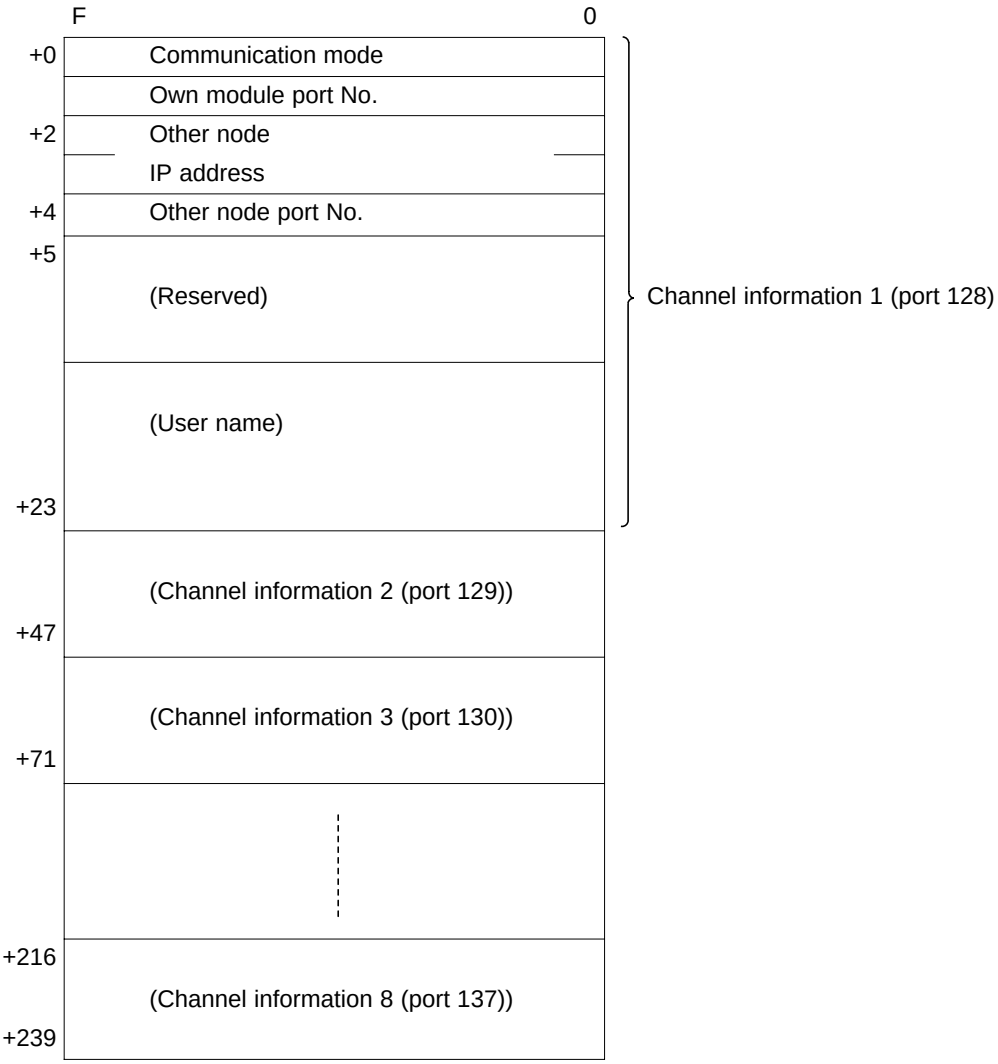
Address	Contents
3	0001h
4	AC10h

■ **Screen example**



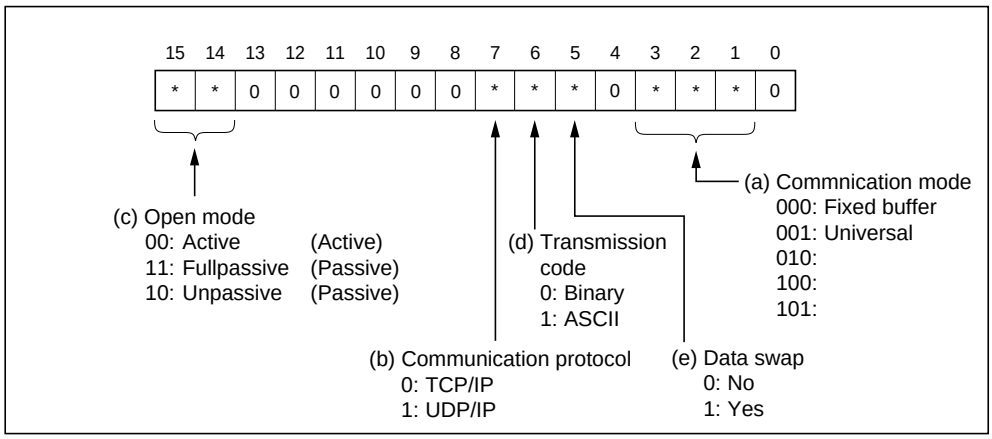
(6) Communication channel information

Information of each channel, specified with this module's Client Port Number, at the open is stored.



1) Communication mode

Communication condition of a channel at the time of open is stored as bit information in 1 word. The detail of coding is shown in following diagram.



a) Communication mode

Indicates communication mode of a channel at the time of open.

b) Communication protocol

Indicates communication protocol used is either TCP/IP or UDP/IP.

c) Open Method

Indicates open method of the channel.

• Active open

In TCP connection Process active open to the node which is in passive open condition.

• Fullpassive open

TCP connection is passively opened to the node specified with communication address. It waits an active open from the node with specified address.

• Unpassive open

TCP connection is opened to the every node on the network. It waits an active open from any other nodes on the network.

d) Transmission code

Indicates data coding, when communicating with other nodes, is either Binary or ASCII.

e) Data swap

When data swap is specified in any binary code communication mode, order of upper and lower byte of transmission data is reversed.

This specification has no effect in ASCII Code communication.

f) Example of communication mode setting (In case of binary code)

Communication mode / Communication method		Universal	Fixed/Common buffer
TCP	Active	0002h	0000h
	Fullpassive	C002h	C000h
	Unpassive	8002h	8000h
UDP	Active	0082h	0080h
	Fullpassive	C082h	C080h
	Unpassive	8082h	8080h

2) Own module port number

Own module's port number specified at the time of channel open is stored.

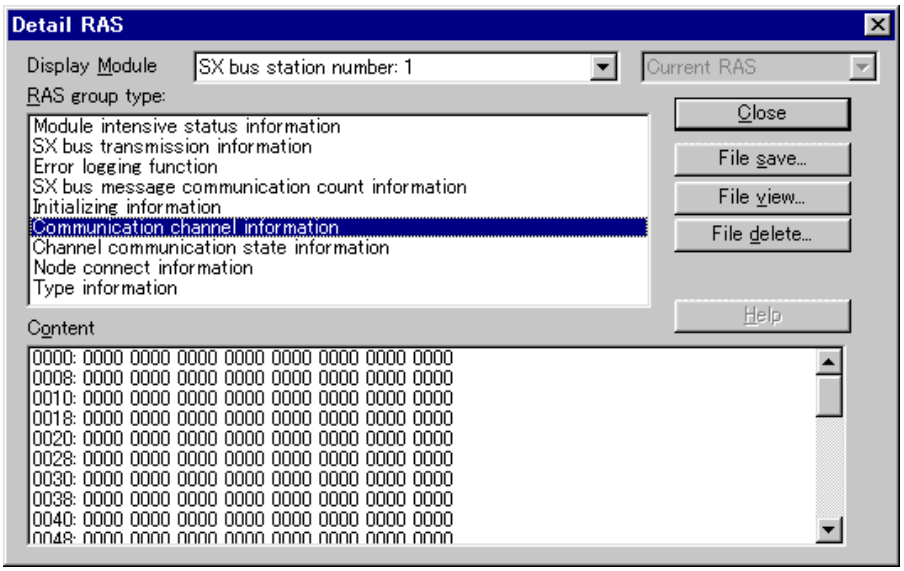
3) Other node IP address

Other node's IP address specified at the time of channel open is stored.
In case of Full/Unpassive open, other node's IP address is stored at the time of establishing connection.

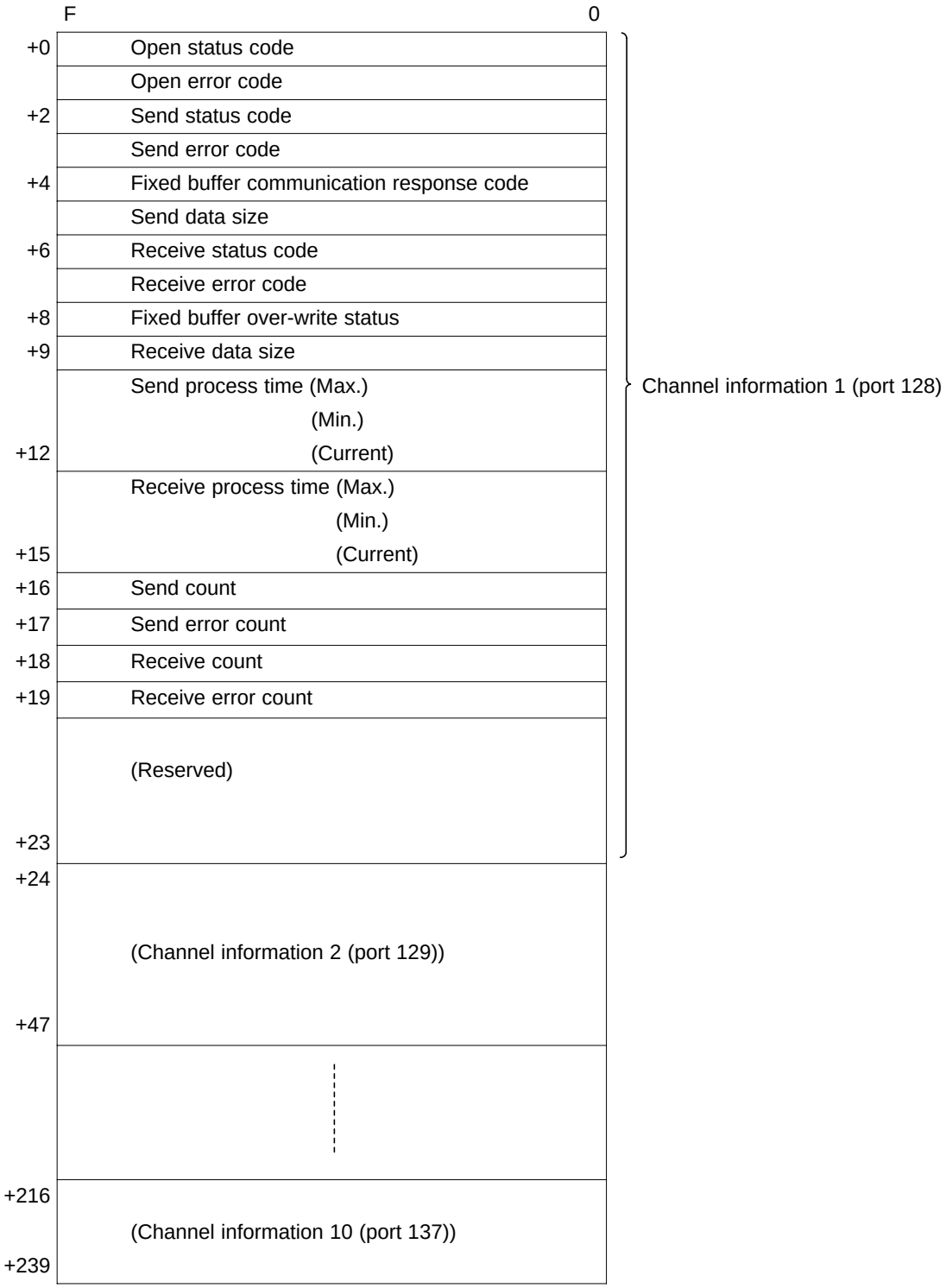
4) Other node port number

Other node's port number specified at the time of channel open is stored.
In case of Full/Unpassive open, other node's port number is stored at the time of establishing connection.

■ **Screen example**



(7) Channel communication state information (RAS type: 0x44)
Communication status of each channel, specified with this module's client port number, is stored.



11-2 RAS Details

1) Open status code
Channel open status is stored with following code.
0: Not opened
1: Open process in progress
2: Open complete
3: Close process in progress
8xh: Forced close

2) Open error code
Error code of the error occurred during open process is stored in binary. Refer to the list of error codes for details of open error code.
Error code is cleared in following cases.
• Next open process is completed successfully.
• Communication is started.

3) Send status code
Send status is stored with following code.
0: No send request or send complete
1: Sending

4) Send error code
Error code of the error occurred during send process is stored in binary. Refer to the list of error codes for details of send error code.
Error code is cleared in following cases.
• Next open process is completed successfully.
• Communication is started.
• Send process is completed successfully.

5) Fixed buffer communication response code
End Code of the response received during fixed buffer communication is stored in binary. Refer the section, which explains fixed buffer communication for details of end code of response.
End code is cleared in following cases.
• Next open process is completed successfully.
• Communication is started.

6) Send data size
When data send is requested, specified data size is stored in binary form. If there were no request for sending data, [0] is stored.
Send Data Size is cleared in following cases.
• Next open process is completed successfully.
• Communication is started.

7) Receive status code
Receive status is stored with following code.
0: No receive request or receive complete
1: Receiving
2: Received data ready (When there is no receive request.)

8) Receive error code

Error code of the error occurred during receive process is stored in binary. Refer to the list of error codes for details of send error code.

Error code is cleared in following cases.

- Next open process is completed successfully.
- Communication is started.
- Receive process is completed successfully.

9) Fixed buffer overwrite status

In fixed buffer communication mode, whether data is received from other node after receiving data once and before sending response (Overwrite) or not is indicated with following code.

0: No overwrite
1: Overwrite occurred

Fixed buffer overwrite status is cleared in following cases.

- Next open process is completed successfully.
- Communication is started.

10) Receive data size

At the time of data receive confirmation following data receive request, number of bytes of the data transferred to the received data storage area is stored in binary code.

11) Send process time

Maximum, minimum and current send process time for each communication mode is stored.

a) It is stored in 55ms unit and in binary code.
b) Time from send request to the own module's send completion is stored.
c) It is cleared at next open process or communication start.

12) Receive process time

Maximum, minimum and current receive process time for each communication mode is stored.

a) It is stored in 55ms unit and in binary code.
b) Following value is stored as receive process time.

• Universal communication

Time from own module's receive completion to the confirmation call from the user program.

• Fixed buffer communication

Time from own module's receive completion, then confirmation of call from the user program to the completion of response return.

• Client FTP/TFTP communication

Time from receive request to the own module's receive completion.

c) It is cleared at next open process or communication start.

13) Send count

Aggregate of successfully completed data transmission to the other node is stored.

a) It is increased when Command/Response exchange is completed normally.
b) It is increased when following communication took place successfully.

- Data transmission to the other node in fixed buffer communication
- Common Buffer data read from other node in common buffer communication
- Data transmission to the other node in universal communication
- File transmission to the other node in server FTP/TFTP communication
- File transmission to the other node in client FTP/TFTP communication

c) It is cleared at next open process or communication start.

14) Send error count

Aggregate of communication error in sending data to the other node is stored.

a) It is increased when Command/Response exchange is completed with error.
b) It is cleared at next open process or communication start.

15) Receive count

Aggregate of successfully completed data reception from the other node is stored.

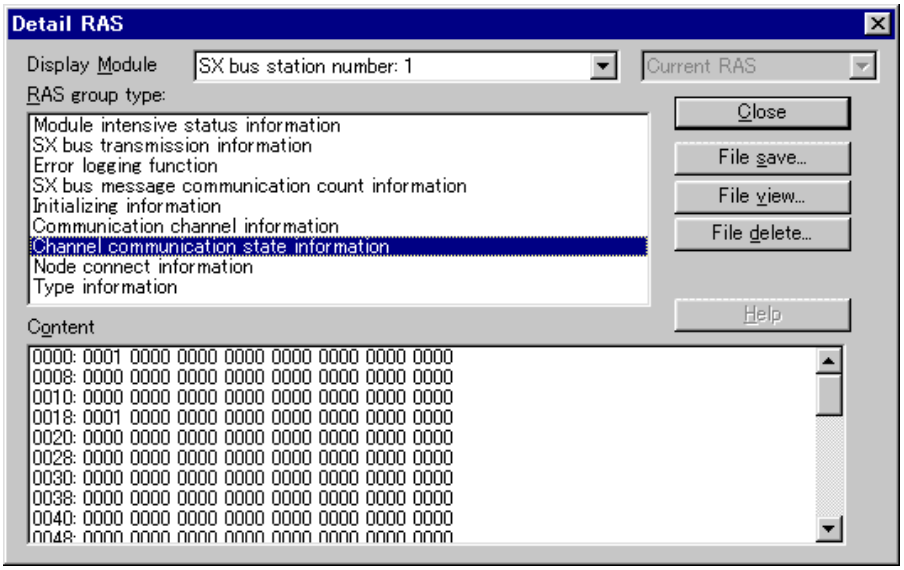
a) It is increased when Command/Response exchange is completed normally.
b) It is increased when following communication took place successfully.
• Data transmission from the other node in fixed buffer communication
• Common buffer data write from other node in common buffer communication
• Data reception from the other node in universal communication
• File transmission from the other node in server FTP/TFTP communication
• File transmission from the other node in client FTP/TFTP communication
c) It is cleared at next open process or communication start.

16) Receive error count

Aggregate of communication error in receiving data from the other node is stored.

a) It is increased when Command/Response exchange is completed with error.
b) It is cleared at next open process or communication start.

■ **Screen example**



(8) Node connect information

Communication status of each communication channel specified with this module's client port number is summarized here. With this information, communication status of each channel and occurrence of communication errors are checked.

1) Connection status

Channel open status is stored in following code.

0: Not Open
1: Open process in progress
2: Open complete
3: Close process in progress
8xh: Forced close

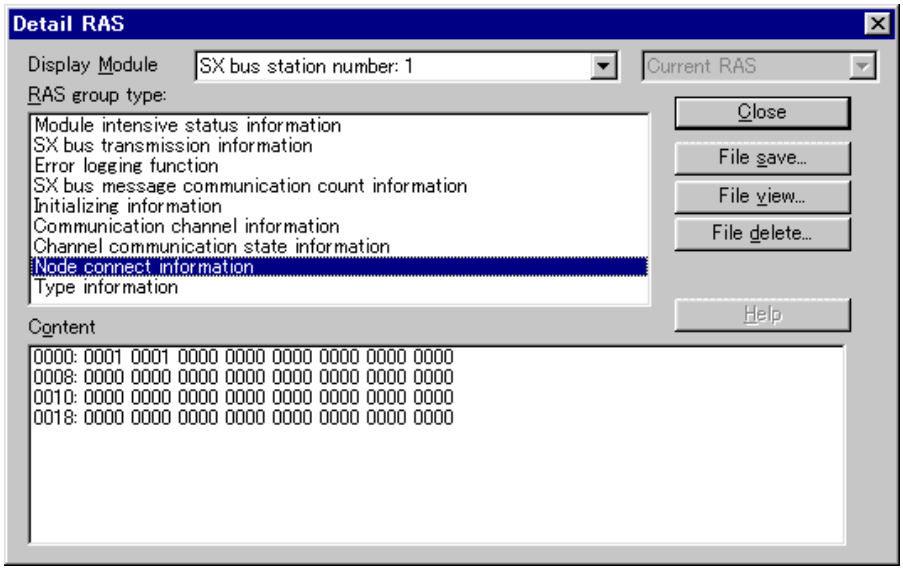
2) Error code

a) Error code of the problems occurred during open process, data sending and Data receiving is stored in binary code. Refer to the list of error code for the details.
The error codes are cleared by following actions.

- Next open process is completed successfully.
- Communication is started (System reset and restarting).

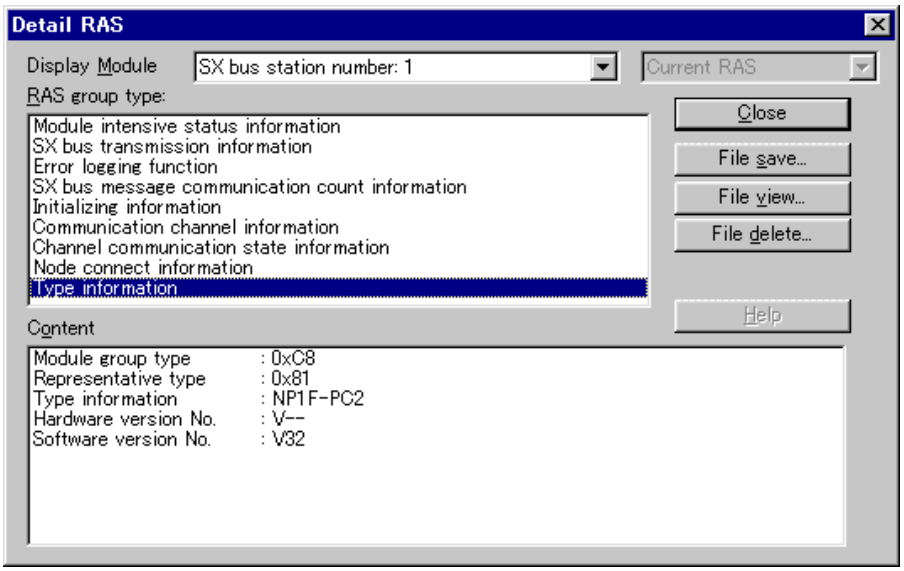
	F 0
+0	Connection status (Port 128)
	Connection status (Port 129)
	Connection status (Port 130)
	Connection status (Port 131)
	Connection status (Port 132)
	Connection status (Port 133)
	Connection status (Port 134)
	Connection status (Port 135)
	Connection status (Port 136)
+9	Connection status (Port 137)
	(Reserved)
+15	
+16	Error code (Port 128)
	Error code (Port 129)
	Error code (Port 130)
	Error code (Port 131)
	Error code (Port 132)
	Error code (Port 133)
	Error code (Port 134)
	Error code (Port 135)
	Error code (Port 136)
	Error code (Port 137)
+26	(Reserved)
+31	

■ **Screen example**



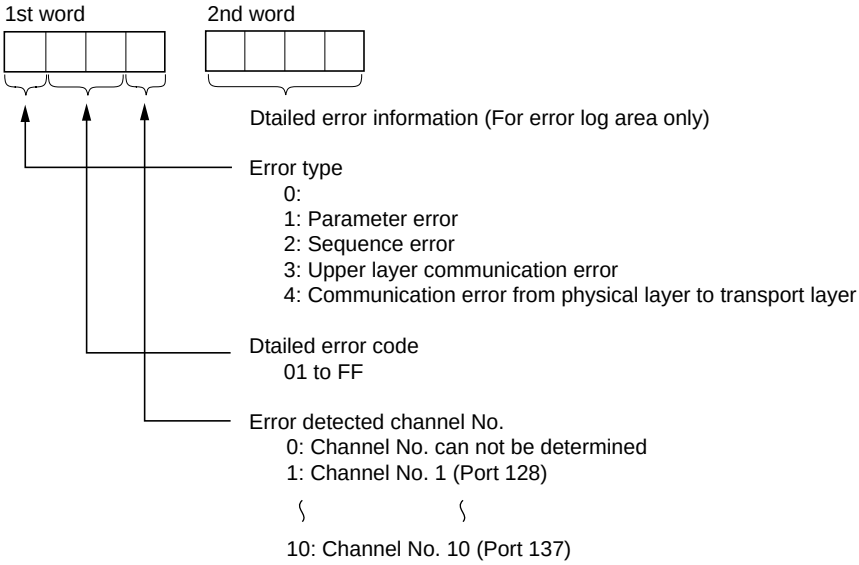
(9) Type information

Module type information is displayed.



In this section error codes stored in Error Logging Information, Channel Communication Status Information and Node Connection Information are detailed.

11-3-1 Error code basic format



11-3-2 List of error codes (Ethernet Card)

(1) Parameter error

Error code	Error description	Initial error	Open error	Send/Receive error	Error log area
1010h	Own module IP address is set to 00000000h or FFFFFFFFh.	O			
1020h	Initializing error is occurred due to the error in parameters for initializing process.	O			
1030h	Number of bits in sub-net mask is smaller than number of bits in net mask obtained from this module's IP address.	O			
1040h	One of following errors exists in default gateway IP address or specification of gateway. 1) FFFFFFFFh is set for default gateway IP address. 2) FFFFFFFFh is set for network address of the gateway. 3) Gateway IP address is set to 00000000h or FFFFFFFFh.	O			
1050h	Trace file open error	O			
1060h	Ethernet Card error Ethernet Card driver is not started.	O			
110xh	Other node's IP address is set to 00000000h, FFFFFFFFh or the same address as own module.		O		
111xh	Own module's port number is set to in the range of 0000h to 00FFh.		O		
112xh	Other node's port number is set to in the range of 0000h to 00FFh.		O		
113xh	Combination of [Own module's port No.] and [Other node's port No.] specified is the same as a connection already established.		O		
114xh	No user name is specified in client FTP communication mode.		O		
115xh	Specified other node's IP address is the IP address of a network that cannot be communicated.		O		
116xh	There is an error in specifying communication mode.		O		
120xh	Data length exceeds the regulation. (Universal communication: > 2034 bytes) (Fixed buffer communication: > 1017 words) (Client FTP/TFTP communication: > 3966 bytes)			O	
121xh	No file name is specified in client FTP/TFTP communication mode.			O	
122xh	Storage area for received data is less than received data size.			O	

(2) Sequence error

Error code	Error description	Initial error	Open error	Send/Receive error	Error log area
201xh	Own module has not completed communication starting process.		O	O	
202xh	Relevant channel's open process is not completed.			O	
203xh	Send/Receive attempt is made during close process.			O	
204xh	In Server/Client FTP communication mode, attempt to open more channel than it is allowed (maximum 4 channels).		O		

(3) Communication error in upper layer

Error code	Error description	Initial error	Open error	Send/Receive error	Error log area
301xh	Response from other node for fixed buffer transmission is other than 00h (Normal Completion). Response end code is stored in error detail.			O * 1	O *
302xh	Response is not returned within the interval of response watchdog timer.			O	O
303xh	Amount of data specified with data length is not received within the period of response watchdog timer. Actual data size is less than the data length specified. Remainder of datagram divided in TCP/UDP level is not received within the period of response watchdog timer.			O	O
304xh	An error message as the response is received from server during file transfer in client FTP/TFTP communication. Error code of message received is stored in detailed error information.		O * 1	O * 1	O *

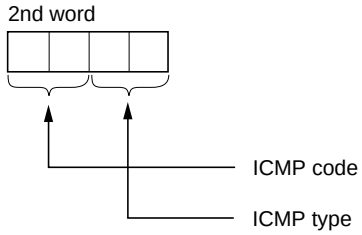
Note:
* 1 Detailed error information is stored in fixed buffer response code of channel communication status.
* Detailed error information is available.

(4) Communication error from physical layer to transport layer

Error code	Error description	Initial error	Open error	Send/Receive error	Error log area
401xh					
402xh	Connection could not established in TCP connection open process.		O		O
403xh	Sum-check error in received data in TCP protocol.				O
404xh	Sum-check error in received data in UDP protocol.				O
405xh	TCP send timeout occurred in TCP/IP protocol. ACK was not received from other node in TCP/IP protocol.		O		O
406xh	Invalid IP address (Network Address) is used. (When IP address of IP packet sent to the other node is the same value, ICMP error packet is received.) ICMP TYPE=3, CODE=0				O *
407xh	Invalid IP Address (Host Number) is used. (When IP address of IP packet sent to the other node is the same value, ICMP error packet is received.) ICMP TYPE=3, CODE=1				O *
408xh	Invalid port number is used. (When port number of IP packet sent to the other node is not registered at the other node, ICMP error packet is received.) ICMP TYPE=3, CODE=2,3				O *
409xh	When IP packet is discarded due to the increased traffic, ICMP error packet is received. ICMP TYPE=4				O *
40Axh	ICMP error packet is received when there is assembling timeout error in other node. ICMP TYPE=11				O *
40Bxh	ICMP error packet that is not supported by the system is received. ICMP error packet type and code are stored in detailed error information.				O *
40Cxh	Ethernet address corresponding to the IP address specified does not exist. (IP Address is searched with ARP function but no response is received.) Invalid IP address (Network No, Host No.) or invalid port number is used. (TYPE 3 ICMP error packet is received.)		O		O
40Dxh					
40Exh	Head check sum of received IP Packet is abnormal.				O
40Fxh	Can not open because the internal buffer used for open is full.		O		
410xh	Send error has occurred.			O	O
411xh	Could not obtain send buffer.			O	O
412xh	TCP Send Timeout occurred during TCP/IP communication. Window size of the other node is too small to send.			O	O

Note:
* Detailed error information is available.

When error code is in the range 406xh to 40Bxh, the type and code of received ICMP message are stored as detailed error information in following format.



Appendix 1 Common Status Code

Appendix 1 Common Status Code

Common status codes of message handling FB used in this module are listed in following table.

<Common status codes>

Error Code	Name	Cause	Countermeasures
160	Message destination specification error	No module exists at the specified the SX Station No.	• Check input terminals used for setting destination.
165	Busy: Message	Receive message could not be sent because other station on the SX-Bus was busy.	• Execute the FB after while. • If this error occurs frequently, the load to send messages on the other station may be too high, reduce the load.
170	Busy: Message send	Message could not be sent because resources required for sending message in own CPU was busy.	• Execute the FB after while. • If this error occurs frequently, the load to send messages on the own CPU may be too high, reduce the load.
197	Busy: Network send	In communication through network of communication modules, destination module is busy and message can not be sent.	• Execute the FB after while. • If this error occurs frequently, the load to send messages on own CPU may be too high, reduce the load.

Appendix 2 Updating Initialization File

Appendix 2 Updating Initialization File

<General>
When using LAN Card LD-CDS (Laneed) and LAN Card ENW-3503-T (Planex Communications Inc.), it is necessary to update Initialization file supplied with this module. (For updating, use a commercial editor.)
To update the file, update the contents of Initialization file for each individual LAN Card and then download it from D300win. (For download procedure, refer to Section 5, Initial Setting.)

<LD-CDS (Laneed)>

1) The contents of the AUTOEXEC.bat

```
PATH=C:\;D:\
ver/r
D:
DCD16/IRQ:9/PIO:300
LSL
LDCDS                    <- Specify the DOS ODI driver supplied with the LAN Card.
IF NOT EXIST PCTCP. EXE GOTO BASIC
PCTCP
GOTO END
:BASIC
PCCDIF
:END
```

2) The contents of the NET.cfg

```
LINK DRIVER 3C589
        INT 5
        PORT 300
        FRAME ETHERNET _ II

LINK DRIVER LDCDY
        PORT 300
        INT 9
        FRAME ETHERNET _ II

LINK DRIVER TDKCD02
        PORT 300
        INT 5
        FRAME ETHERNET _ II

LINK DRIVER ENW35
        PORT 300
        INT 9
        FRAME ETHERNET _ II

LINK DRIVER LDCDS    <- Specify the DOS ODI driver supplied with the LAN Card.
        PORT 300
        INT 9
        FRAME ETHERNET _ II
```

Add the ETHERNET Card settings.

3) Downloading the file

Download the following file from the DOS/V floppy disk supplied with ENW-3503-T (Planex Communications Inc.), together with the files mentioned 1) and 2).

<ENW-3503-T (Planex Communications Inc.)>

1) The contents of the AUTOEXEC.bat

```
PATH=C:\;D:\
ver/r
D:
DCD16/IRQ:9/PIO:300
LSL
LE10ODI        <- Specify the DOS ODI driver supplied with the LAN Card.
IF NOT EXIST PCTCP. EXE GOTO BASIC
PCTCP
GOTO END
:BASIC
PCCDIF
:END
```

2) The contents of the NET.cfg

```
LINK DRIVER 3C589
        INT 5
        PORT 300
        FRAME ETHERNET _ II

LINK DRIVER LDCDY
        PORT 300
        INT 9
        FRAME ETHERNET _ II

LINK DRIVER TDKCD02
        PORT 300
        INT 5
        FRAME ETHERNET _ II

LINK DRIVER ENW35
        PORT 300
        INT 9
        FRAME ETHERNET _ II

LINK DRIVER LE10ODI  <- Specify the DOS ODI driver supplied with the LAN Card.
        PORT 300
        INT 9
        FRAME ETHERNET  _ II
```

Add the ETHERNET Card settings.

3) Downloading the file

Download the following file from the DOS/V floppy disk supplied with ENW-3503-T (Planex Communications Inc.), together with the files mentioned 1) and 2).

Fuji Electric Co.,Ltd.
ED & C · Drive Systems Company

Gate City Ohsaki, East Tower, 11-2, Osaki 1-chome, Shinagawa-ku, Tokyo, 141-0032, Japan
Phone: +81-3-5435-7135 ~ 8
Fax: +81-3-5435-7456 ~ 9
URL http://www.fujielectric.co.jp/kiki/