S10mini

HITACHI

S10mini
HARDWARE MANUAL

OPTION
J.NET

SME-1-104（D）

S10mini
HARDWARE MANUAL

OPTION
J.NET

First Edition, December 1998, SME-1-104(A) (out of print)
Second Edition, October 2008, SME-1-104(D)

BI-KB-KO<IC-IC> (FL-MW20, AI8.0)

SAFETY PRECAUTIONS

Be sure to read this manual and all other attached documents carefully before installing, operating inspecting or conducting maintenance on this unit. Always use this unit properly. Be sure to carefully read the information about the device, the safety information and precautions before using this unit. Be sure that the person(s) responsible for maintenance receives and understands this manual completely.

This manual divides the safety precautions into DANGERs and CAUTIONs.

⚠ DANGER : Failure to observe these warnings may result in death or serious injury.

⚠ CAUTION : Failure to observe these cautions may result in injury or property damage.

Failure to observe any **⚠ CAUTION** may lead to serious consequences.

All of these DANGERs and CAUTIONs provide very important precautions and should always be observed.
Additional safety symbols representing a prohibition or a requirement are as follows:

🛇 : Prohibition. For example, “Do not disassemble” is represented by:

🛇

❗ : Requirement. For example, if a ground is required, the following will be shown:

⏚

1. Installation

⚠ **CAUTION**

- Use this product under the environmental conditions specified in the catalogs and manual.
  Utilizing this product in a hot, damp, or dusty atmosphere or in an atmosphere of corrosive gas, vibration or impact may lead to a malfunction, shock hazard or fire.
- Install this product according to the procedure outline in the manual.
  Imperfect installation may lead to a part drop, failure or malfunction.
- Do not put any wire chip or other foreign matter into this product.
  This may cause a malfunction, failure or fire.

2. Wiring

⏚ **REQUIREMENT**

- Be sure to ground this product with FG.
  Failure to ground this product may lead to a malfunction or shock hazard.

⚠ **CAUTION**

- Connect this product to a power supply with the same ratings.
  Connecting this product to a power supply exceeding its voltage rating may lead to a fire.
- Wiring must be conducted by a qualified technician.
  Miswiring may lead to failure, shock hazard or fire.
- Wiring by transceiver cables and coaxial cables must be separated from wiring from high-voltage equipment.　Combining these two systems of wires in one place may lead to a malfunction.

3. Handling precautions

⚠ DANGER

- Do not touch any terminal while this product is live, as this may lead to a shock hazard.
- Configure an emergency stop circuit, interlocking circuit and related circuitry outside the programmable controller.
  A programmable controller failure may lead to a general breakdown or an accident.

⚠ CAUTION

- Make sure that everything is safe before changing programs, running or stopping this product while on the fly or producing forced output.
  Mishandling may lead to product breakdown or an accident.
- Turn on the product according to the correct power - on procedure.
  Mishandling may lead to product breakdown or an accident.
- Do not use a transceiver, mobile phone, or the like near this module. Using such a device nearby may lead to a malfunction or a module failure due to electrical noise.

4. Maintenance

🛇 PROHIBITION

Do not disassemble or remodel this product, as this may lead to a malfunction, failure or fire.

⚠ CAUTION

Power off this product before attaching or detaching any module or unit as this may lead to a malfunction, failure or shock hazard.

WARRANTY AND SERVICING

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

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

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

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

   - The malfunction was caused by handling or use of the product in a manner not specified in the product specifications.
   - The malfunction was caused by a unit other than that which was delivered.
   - The malfunction was caused by modifications or repairs made by a vendor other than the vendor that delivered the unit.
   - The malfunction was caused by a relay or other consumable which has passed the end of its service life.
   - The malfunction was caused by a disaster, natural or otherwise, for which the vendor is not responsible.

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

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

   - Instruction for installation and adjustments, and witnessing trial operations.
   - Inspections, maintenance and adjustments.
   - Technical instruction, technical training and training schools.
   - Examinations and repairs after the warranty period is concluded.
   - Even if the warranty is valid, examination of malfunctions that are caused by reasons outside the above warranty scope.

This manual provides information for the following hardware product:

<Hardware product>
J.NET (LQE040)

<Changes added to this manual>

Description of added changes	Page
Supplementary, “Replacing or adding on the module” is newly added.	Z-1

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

Revision record

Revision No.	Revision Record (revision details and reason for revision)	Month, Year	Remarks
A	First Edition	December 1998	
D	Supplementary, “Replacing or adding on the module” is newly added.	October 2008	

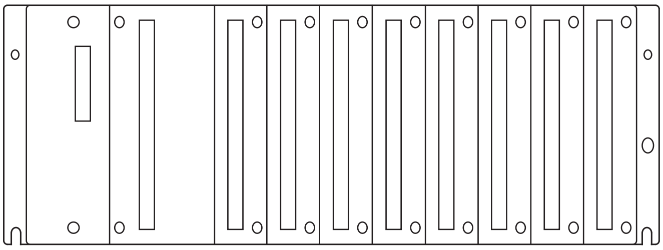
PREFACE

We greatly appreciate your purchase of the J.NET module — an option for the CPU. This manual describes how to operate the J.NET module. We request that you read this manual carefully to ensure the correct use of the J.NET module.

CONTENTS

1 BEFORE USE

1.1 CPU Mount Base

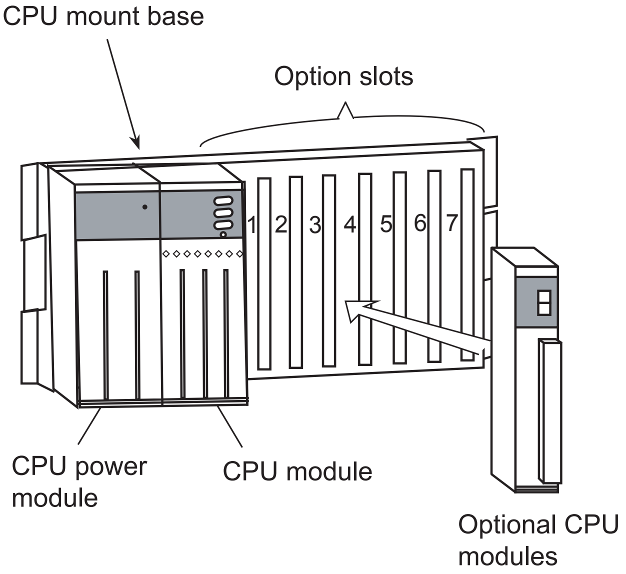


The CPU mount base is classified as one of the following three types:

- 2-slot mount base (type: HSC-1020)
- 4-slot mount base (type: HSC-1040)
- 8-slot mount base (type: HSC-1080)

For the 8-slot mount base, for example, up to eight modules can be mounted.

1.2 Mounting Optional Modules



CPU extension mount base: HSC-1080

PS slot: Slot into which the CPU power supply (LVQ000) is inserted.

CPU slot: Slot into which a CPU module (LQP000, LQP010) is inserted.

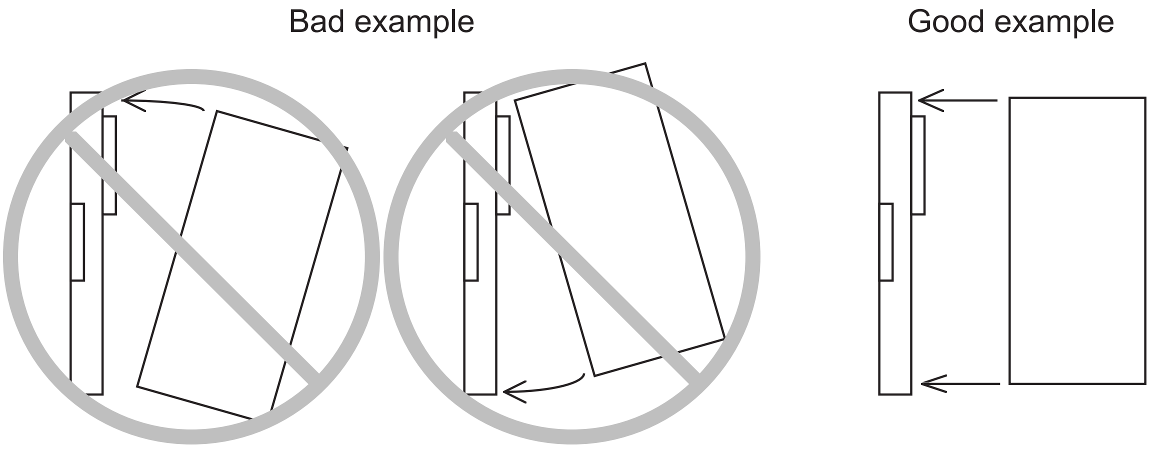
Slot 0 to slot 7: Slots into which CPU optional modules or I/O modules are inserted.

CAUTION

- Be sure to insert J.NET modules sequentially into the slots, starting from the leftmost slot, without creating any empty slots in between.
- When only one J.NET module is inserted, set it as the main module.

Note the following when mounting optional modules:

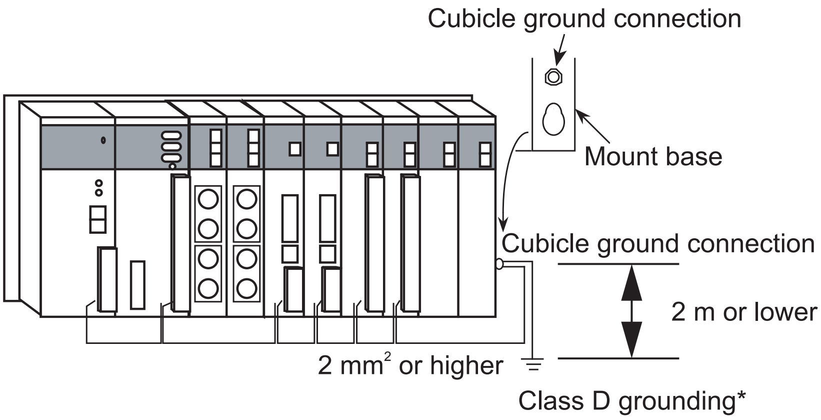
- As shown in the figure below, mount an optional module upright on the CPU extension mount base. (As shown in the bad example below, mounting the optional module aslant on the mount base may cause a connector pin to bend. If the pin is bent, the optional module may malfunction.)



CAUTION

If the mount base is installed overhead because of the cabinet structure, you may use a stepladder or something similar to install the optional module. However, never install it aslant.

1.3 Grounding



REQUIREMENT

- In frame ground (FG) cabling, connect the FG terminal of each module with external terminals to the cubicle ground terminal of the mount base. The grounding cable wired from the mount base's ground terminal must be grounded by Class-D grounding. It must be up to 2 meters long.
- Use a ground wire whose cross-sectional area is 2 mm^2 or greater.

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

2 SPECIFICATIONS

2.1 Use

The J.NET module (type: LQE040) performs data communication between station units (e.g., J.STATION); it does not support message write and read services. This module can be connected to networks conforming to field network level 1 for JEMA standard programmable controllers.

2.2 Specifications

2.2.1 System specifications

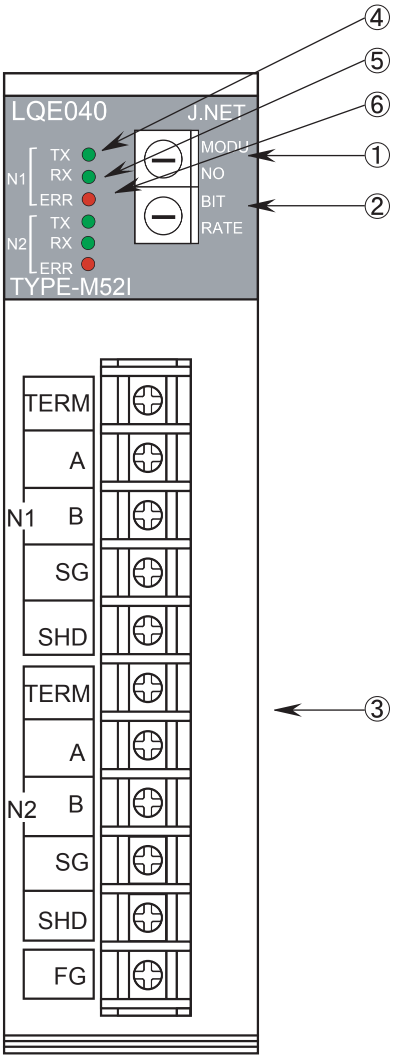
Item	Specifications
Type	LQE040
Number of networks	2 networks/module
Maximum number of J.NET modules that can be mounted in the CPU	2 (Insert J.NET modules sequentially into the slots, starting from the leftmost slot, without creating any empty slots in between.)
Weight	260 g

2.2.2 Line specifications

Item		Specifications
Transmission method		Serial transmission (bit serial transmission)
Electrical interface		RS-485
Number of stations		Up to 31 stations/network (62 stations/module)
Connection cable	Line type	Two pairs of twisted-pair shield cables Recommended cable: KPEV-SB 2P 0.5mm^2 (Hitachi Cable, Ltd.)
	Distance	The distance depends on the transmission rate as follows: Transmission rate ≤1.0 Mbps: Up to 240 m Transmission rate ≤ 0.5 Mbps: Up to 480 m Transmission rate ≤ 0.25 Mbps: Up to 800 m Transmission rate ≤ 0.125 Mbps: Up to 1000 m
	Terminal block	11-point terminal block (M3 × 11)

3 NAMES AND FUNCTIONS OF EACH PART AND CABLING

3.1 Names and Functions of Each Part



① Module No. setting switch
Up to two J.NET modules can be mounted on one CPU unit. Use this switch to set up the main module and sub-module. (For details on T/M, see Subsection 6.1.2.)
Set up the main module and sub-module according to the table shown below.

Setting No.	Main module/sub-module
0	Main module
1	Sub-module
8, 9	Use of T/M

② Bit rate setting switch
The bit rate setting switch is used to set a transmission rate. The table below shows the relationship between setting Nos. and transmission rates.

Setting No.	Transmission rate
0	1.0 Mbps
1	0.5 Mbps
2	0.25 Mbps
3	0.125 Mbps
8 to F	Use of T/M

③ Terminal blocks for interface
Nn: Network No.
TERM: Terminals for a terminating resistor. Short these terminals with a jumper when the J.NET module is used at an end of the network.
A, B: Used to connect a transmission/reception data line.
SG: Used for signal ground terminal.
SHD: Used for shield ground terminal.
FG: Used for frame ground terminal.

④ LED for transmission
Comes on when the J.NET module starts transmitting data in each network.

⑤ LED for reception
Comes on when the J.NET module starts receiving data in each network.

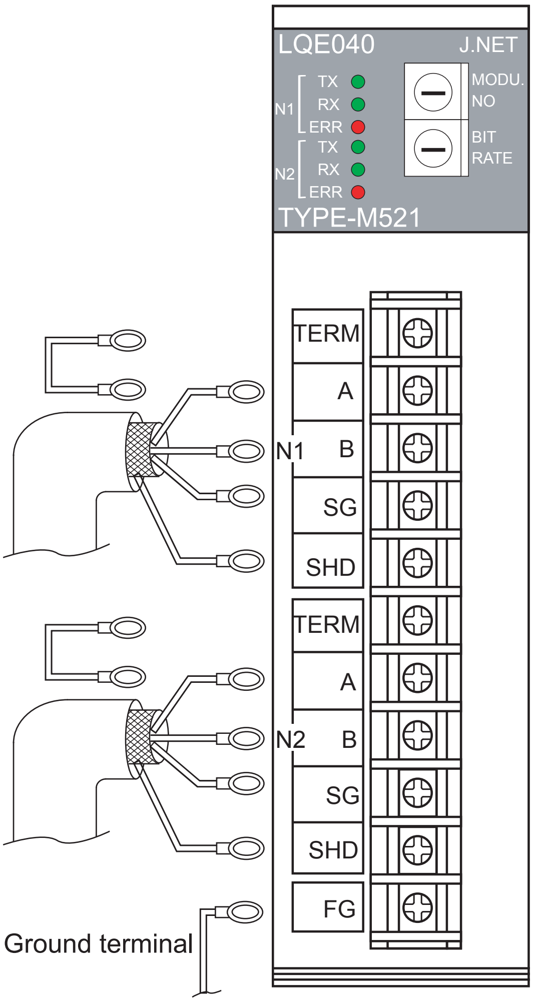
⑥ LED for errors
Comes on when a hardware error is detected in the J.NET module (see Subsection 6.3.2).

⚠ CAUTION

Do not operate the module No. switch and bit rate switch when the J.NET module is in operation. Failure to observe this rule may lead to a module malfunction.

3.2 Cabling

3.2.1 Interface signals and cabling method



Network 1 (N1), Network 2 (N2)

Signal name	
Abbreviation	Name
A	Send/receive data Linkage data
B	
SG	Grounding for signal Signal Ground
SHD	Grounding for shield SHielD ground
TERM	Terminating resistor for transmission/reception TERMinal registor

Others

Signal name	
Abbreviation	Name
FG	Grounding for frame Frame Ground

Interface signal voltage levels

Designation	Mark	Space
Interpretation	1/OFF	0/ON
Output condition	-6 to -1.5 V	1.5 to 6 V
Input condition	-0.2 V or lower	0.2 V or higher

The input condition represents the electric potential of A viewed from B.

Short TERM terminals with jumpers when the J.NET module is used at an end of a network. (Short-circuit the TERM and A-terminals of the J.NET module with jumpers.) The terminating resistor (120 ohms) is internally connected.

⚠ **CAUTION**

Two shield terminals (SHD) and one frame ground (FG) terminal are internally connected. Be sure to ground the FG terminal.

3.2.2 Cable specifications

The J.NET module requires two pairs of twisted-pair shield cables.
A polyethylene-insulated vinyl sheath cable for instrumentation should be used as the cable for J.NET and J.STATION.

Interface Cable Specifications for KPEV-SB 2P 0.5 mm^2 of Hitachi Cable, Ltd.
(Recommended Cable Type)

Item	Unit	Measured value
Maximum conductor resistance (20°C)	Ω/km	34.0
Withstand voltage	VAC/minute	1000
Minimum insulation resistance (20°C)	MΩ • km	2500
Electrostatic capacity (1 kHz)	PF/m	60
Characteristics impedance (1 MHz)	Ω	110

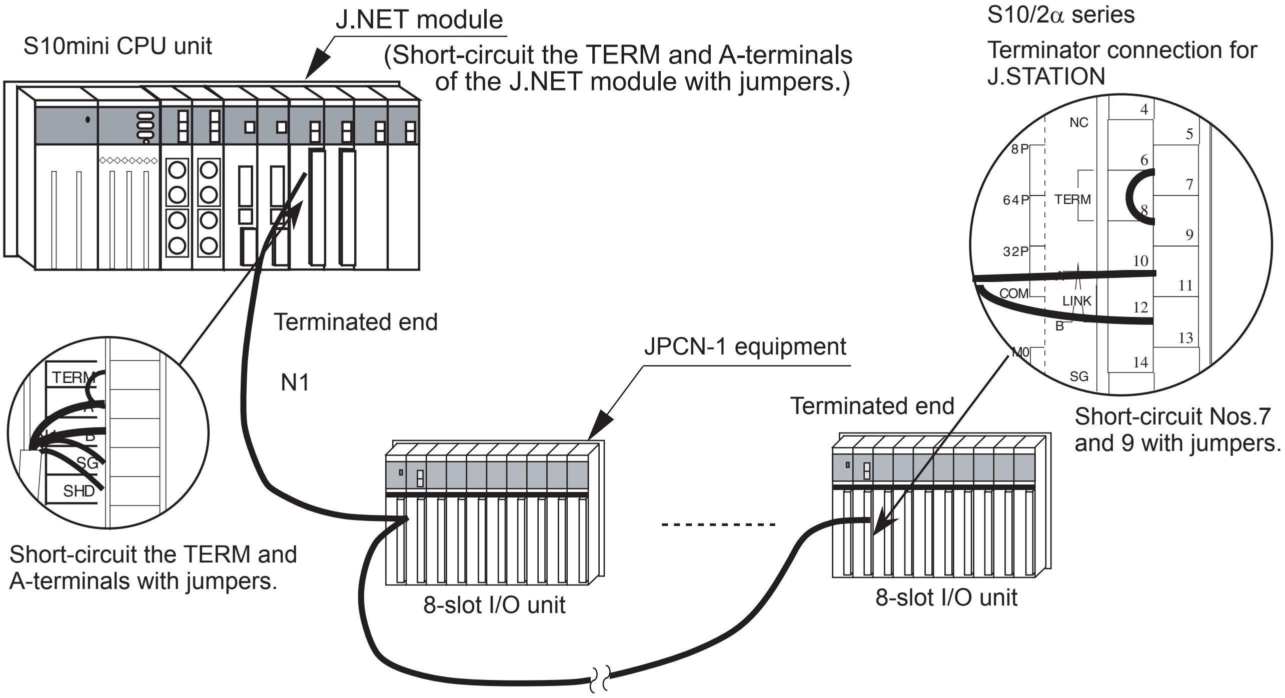
Note

The 1-MHz characteristics impedance of the above cable is 110 ohms but J.NET and J.STATION have 120-ohm internal terminating resistors in consideration of other transmission rates.
Short TERM terminals with a jumper when the J.NET module is used at an end of the network.
120-ohm terminating resistors are connected in J.NET and J.STATION.

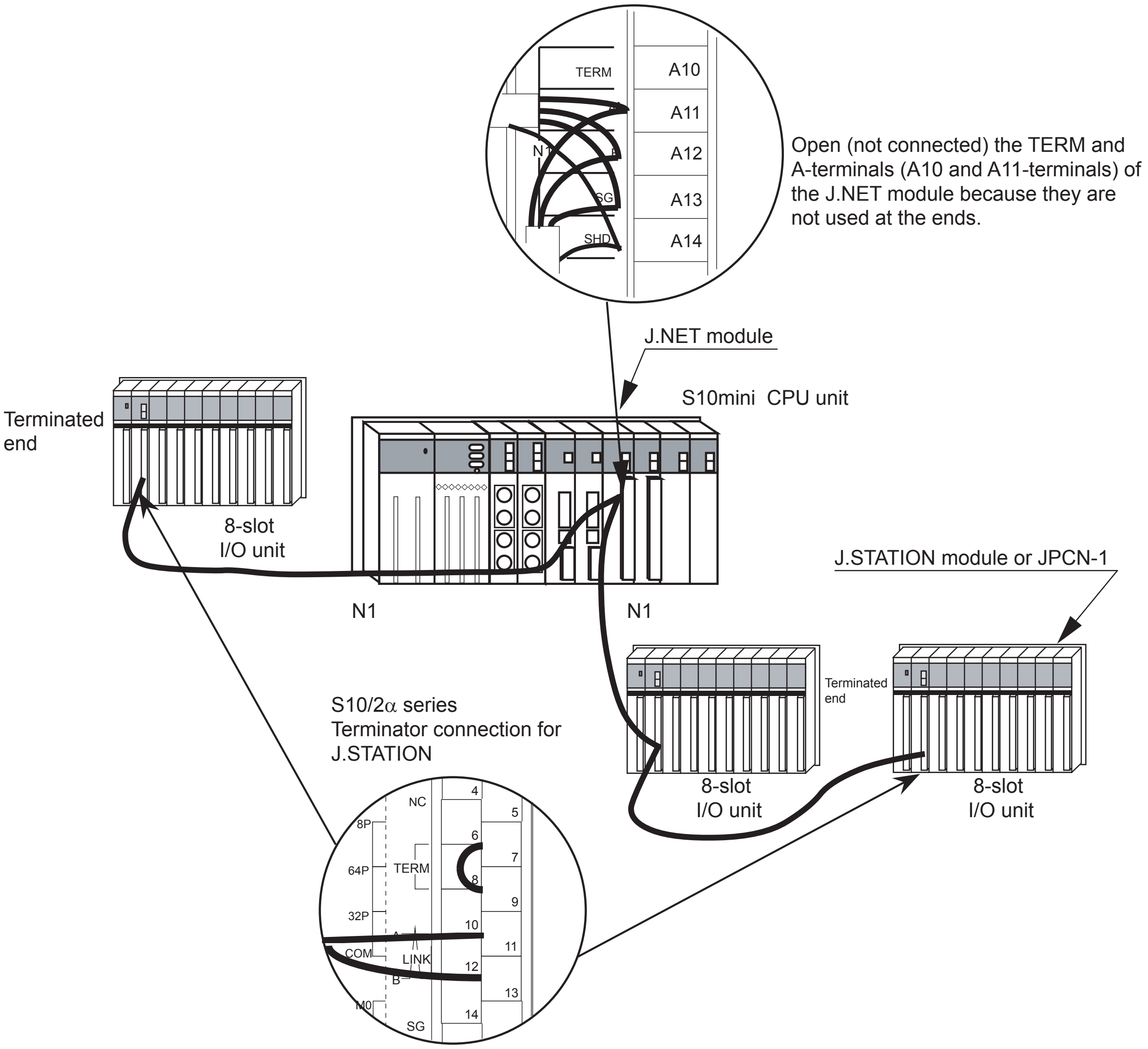
3.2.3 Examples of cabling

Connect terminating resistors to the devices connected to both ends of the cable. Wire the J.NET module as a completely different network because it has two independent networks, N1 and N2. Examples of N1 cabling are given below. Examples of N2 cabling are the same as these examples.

- Connection when the J.NET module is used at an end of N1


S10mini CPU unit
J.NET module
(Short-circuit the TERM and A-terminals of the J.NET module with jumpers.)
S10/2α series
Terminator connection for J.STATION
NC
8P
64P
TERM
32P
A
COM
LINK
B
M0
SG
4
5
6
7
8
9
10
11
12
13
14
Terminated end
N1
JPCN-1 equipment
Terminated end
Short-circuit Nos.7 and 9 with jumpers.
TERM
A
B
SG
SHD
Short-circuit the TERM and A-terminals with jumpers.
8-slot I/O unit
8-slot I/O unit

- Connection when the J.NET module is not used at an end of N1



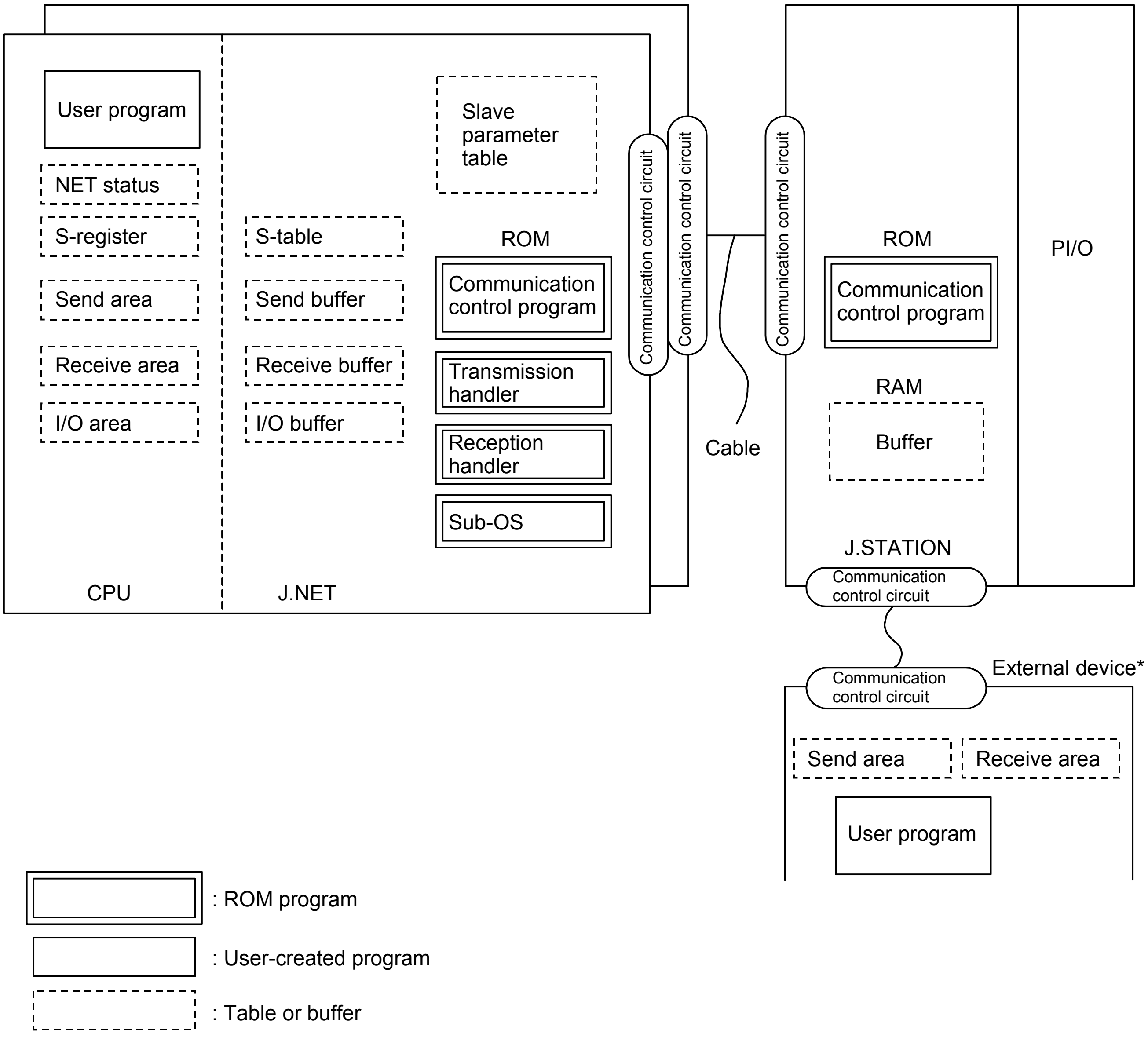
Open (not connected) the TERM and A-terminals (A10 and A11-terminals) of the J.NET module because they are not used at the ends.

Short-circuit Nos.7 and 9 with jumpers.

4 USER GUIDE

4.1 Software Configuration of J.NET System

The software configuration of the J.NET system is shown in the figure below. The communication control programs, transmission handler, reception handler, and sub-OS in the figure need not be loaded into memory by the user because they are ROM programs.

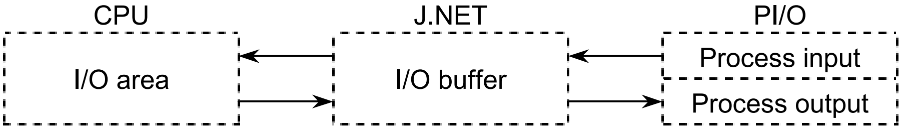


* The external device can be connected as required (RS-232C).

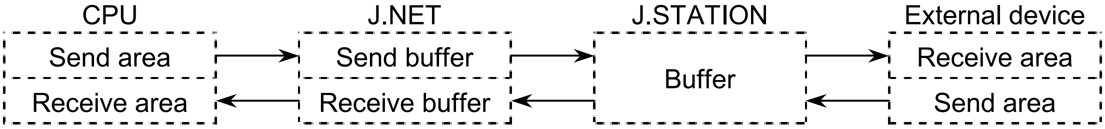
- Communication control program

  The main functions of the communication control program are as follows:

  • The communication control program performs I/O communication with PI/O units.



  • The program transmits and receives data to and from an external device.



  • When data reception from the external device is completed, the program sends an interrupt to the CPU to start the sub-OS.

- Transmission and reception handlers

  The main functions of the transmission and reception handlers are as follows:

  • The transmission handler requests the communication control program to transmit data and the reception handler requests the program to receive data. These handlers are initiated in the user program.

  • The handlers set data transmission/reception information in the NET status table, S-register, and S-table.

- Sub-OS

  The sub-OS is initiated by an interrupt from the communication control program. The main function of the sub-OS is as follows:

  • The sub-OS starts the reception task when data reception is completed. (For reception task registration, see Subsection 5.4.5, “Setting LGB.”)

- NET status table, S-register, and S-table

  The communication control program sets data transmission/reception information and error information in the NET status table, S-register, and S-table. The user program transmits and receives data and handles errors according to this information.

- I/O area

  The table below shows the I/O areas that can be used for I/O communication.

Name	Symbol range	Number of points
External input	XW000 (X000) to XWFF0 (XFFF)	256 words (4,096 points)
External output	YW000 (Y000) to YWFF0 (YFFF)	256 words (4,096 points)
Internal register	RW000 (R000) to RWFF0 (RFFF)	256 words (4,096 points)
Global link register	GW000 (G000) to GWFF0 (GFFF)	256 words (4,096 points)
Transfer register	JW000 (J000) to JWFF0 (JFFF)	256 words (4,096 points)
Receive register	QW000 (Q000) to QWFF0 (QFFF)	256 words (4,096 points)
Event register	EW400 (E400) to EWFF0 (EFFF)	192 words (3,072 points)
Extended internal register	MW000 (M000) to MWFF0 (MFFF)	256 words (4,096 points)
Function work register	FW000 to FWBFF	3,072 words
Extended memory	/100000 to /4FFFFF	2M words

4.2 User-created Programs

This section describes the user-created programs that constitute the J.NET system.
The user need not create the programs explained below if J.STATION is connected and only I/O communication is executed.

4.2.1 User program

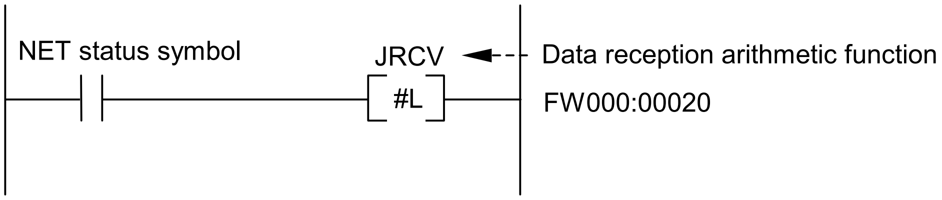
The transmission and reception handlers are initiated by the user program. The user program is divided into the following three types:

- Ladder program: The ladder program is also called the sequence program. It consists of an A-contact (—| |—), B-contact (—|/|—), and output coil (—○—), etc.
- C-mode program: The C-mode program is coded in computer programming languages (e.g., C and assembly language); it is executed in the form of "task, P-coil." This program requires the Compact Process Monitor System (CPMS) and extended memory.
- BASIC program: The BASIC program is coded in FA-BASIC; it is executed in the form of "task, P-coil." This program requires the FA-BASIC compiler and extended memory.

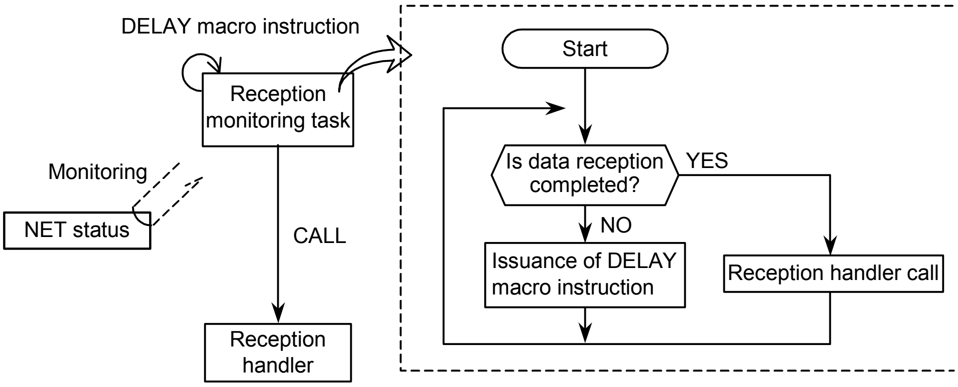
The ladder program calls the arithmetic function to start the transmission and reception handlers. The C-mode program and BASIC program call a subroutine to start the handlers.

4.2.2 User program reception processing

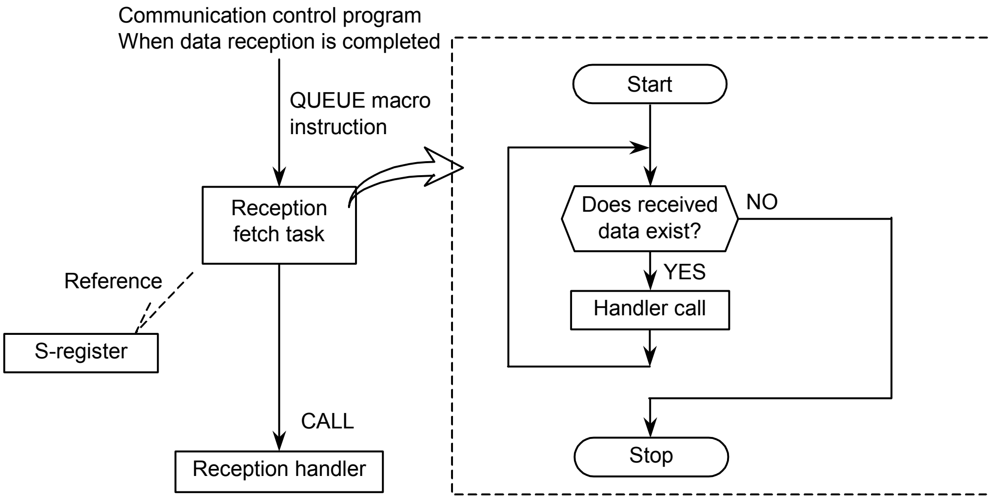
When data reception is completed, this fact is reflected in the associated NET status. If the user program is the ladder program, it starts the reception handler (arithmetic function) using the NET status as the condition. A reception data fetch delay is short enough to be born in the sequence cycle.



When the user program is a C-mode program, it monitors the NET status and initiates the reception handler (subroutine) when data reception is completed.



In the method depicted above, the reception completion monitoring interval is specified in the DELAY macro instruction for the Compact Process Monitor System (CPMS). For this reason, reception data fetch is delayed by the processing delay time specified in the DELAY macro instruction or because the processing level of the reception monitoring task is low (usually, a low processing level is set for the task).
When the user program is a C-mode program or BASIC program, a user task must be created and registered in the system. The user task is started by the sub-OS when data reception is completed. This eliminates the need to monitor completion of data reception by the user program. Received data can be fetched by calling the reception handler as a subroutine in the task started by the sub-OS. Register the task numbers of sub-OS started tasks with the Windows personal computer. Also register such tasks with the CPMS PSE debugger function. (Refer to the SOFTWARE MANUAL OPERATION CPMSE DEBUGGER For Windows [manual number SAE-3-126].)



4.3 JEMA Standard Services

The table below shows the correspondence between JEMA standard communication services, J.NET communication commands, and functions.

JEMA standard service name	J.NET command name	Function
Initialization service	Automatically issued in the system program	Service used to exchange information when the J.NET module and station are started. The J.NET module automatically exchanges information according to the user-set station information when power is turned on or when RESET is executed.
I/O service	Automatically issued in the system program	Service used to exchange I/O data between the J.NET module and station. Where a J.NET I/O area is defined, the J.NET module automatically performs I/O communication.
Data read service	POLLING command	Service that the J.NET module uses to ask whether there is a request that the station wishes to issue. After the J.NET module has exchanged initialization setting service information with the station, it automatically issues this command when data communication is idle.
	PUT command	Service that the station uses to write data to the J.NET module. After the J.NET module has exchanged initialization service information with the station, the station issues the PUT request (data read service) to write data to the J.NET module.
	GET command	Service that the station uses to read data from the J.NET module. After the J.NET module has exchanged initialization service information with the station, the station issues the GET request (data read service) to read data from the J.NET module.
Data write service	READ command	Service that the J.NET module uses to read data from the station. The user can issue a READ command (arithmetic function and subroutine) as required to read data from the station.
	WRITE command	Service that the J.NET module uses to write data to the station. The user can issue a WRITE command (arithmetic function and subroutine) as required to write data to the station.
Reset service	Reset command	Service that the J.NET module uses to restore the station to its initial condition. The user can issue a reset command (arithmetic function and subroutine) as required to restore the station to the initial condition.
Broadcast service	Broadcast command	Service that the J.NET module uses to transmit broadcast messages to all the stations connected to the network. The user can issue a broadcast command (arithmetic function and subroutine) as required to transmit broadcast messages to all the stations.
Message write service	Not supported	Service used to write message data to a station. The J.NET module does not support this service.
Message read service	Not supported	Service used to read message data from a station. The J.NET module does not support this service.

The PUT and GET services of the initialization, I/O, data read, and data write services set station information in the J.NET internal parameter table. They automatically perform information communication between the J.NET module and station when resetting is made or when power is turned on.

For information on how to use the READ, WRITE, reset, and broadcast services, see Section 4.7, “Handlers.”

4.4 NET Status

The NET status table stores communication information (binary) for each station. The user must register the starting address of the NET status table in any of the following eight areas.

X000 to XFFF
Y000 to YFFF
J000 to JFFF
Q000 to QFFF
G000 to GFFF
R000 to RFFF
E400 to EFFF
M000 to MFFF

The NET status table is configured as shown in the table below and requires a capacity of successive 128 points for each of N1 and N2. (For example, when X500 is specified, table elements X500 to X57F are occupied. In the table below, the portion “X5” of each element number replaces △△.)

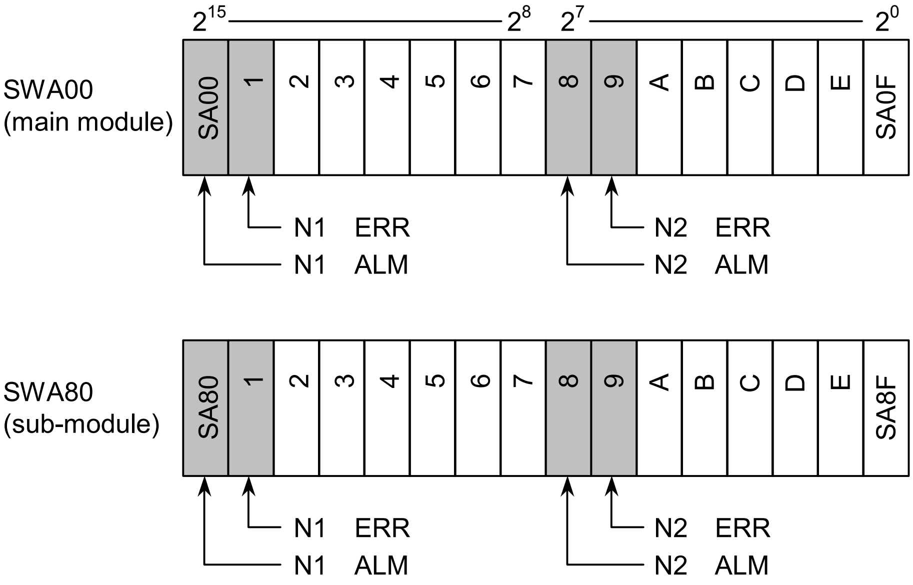
△△: Registered symbol

Station ID	Transmission enable flag	Data reception flag	Response reception flag	Error flag
Unoccupied	–	–	–	–
1	△△01	△△21	△△41	△△61
2	△△02	△△22	△△42	△△62
⋮	⋮	⋮	⋮	⋮
30	△△1E	△△3E	△△5E	△△7E
31	△△1F	△△3F	△△5F	△△7F

Contents of bits	0	Being transmitted	Data not received	Response not received	No errors detected
	1	Transmission enable	Data received	Response received	Error detected

4.5 S-register

The S-register stores information on errors that occur in each network. This register is set if an error is detected even in one of the stations (sub-stations) connected to each network (N1 or N2).



Note: ALM, communication error; ERR, hardware error.

Symbol	Bit	Description
SA00	0	N1 of the main module is providing data communications service normally.
	1	A communication error was detected in N1 of the main module.
SA01	0	N1 of the main module is in normal operation.
	1	A hardware error was detected in N1 of the main module.
SA08	0	N2 of the main module is providing data communications service normally.
	1	A communication error was detected in N2 of the main module.
SA09	0	N2 of the main module is in normal operation.
	1	A hardware error was detected in N2 of the main module.
SA80	0	N1 of the sub-module is providing data communications service normally.
	1	A communication error was detected in N1 of the sub-module.
SA81	0	N1 of the sub-module is in normal operation.
	1	A hardware error was detected in N1 of the sub-module.
SA88	0	N2 of the sub-module is providing data communications service normally.
	1	A communication error was detected in N2 of the sub-module.
SA89	0	N2 of the sub-module is in normal operation.
	1	A hardware error was detected in N2 of the sub-module.

Any other bits not listed above are unused.

4.6 S-table

The S-table is an address table that stores the number of bytes received during data reception, number of bytes received during response reception, and communication error codes. (For details on communication error codes, see Subsections 6.3.3 and 6.3.4.)

Table 4-1 S-table Assignment Table

Net No.	Station ID	Main module		
		Number of bytes received during data reception	Number of bytes received during response reception	Error code
N1	(255)	/A40000	/A40040	/A40080
	1	02	42	82
	2	04	44	84
	⋮	⋮	⋮	⋮
	30	/A4003C	/A4007C	/A400BC
	31	3E	7E	BE
N2	(255)	/A40100	/A40140	/A40180
	1	102	142	182
	2	104	144	184
	⋮	⋮	⋮	⋮
	30	/A4013C	/A4017C	/A401BC
	31	13E	17E	1BE

Net No.	Station ID	Sub-module		
		Number of bytes received during data reception	Number of bytes received during response reception	Error code
N1	(255)	/AC0000	/AC0040	/AC0080
	1	02	42	82
	2	04	44	84
	⋮	⋮	⋮	⋮
	30	/AC003C	/AC007C	/AC00BC
	31	3E	7E	BE
N2	(255)	/AC0100	/AC0140	/AC0180
	1	102	142	182
	2	104	144	184
	⋮	⋮	⋮	⋮
	30	/AC013C	/AC017C	/AC01BC
	31	13E	17E	1BE

4.7 Handlers

The J.NET module provides two types of handlers (arithmetic functions and subroutines) for use in user programs.

4.7.1 Arithmetic functions

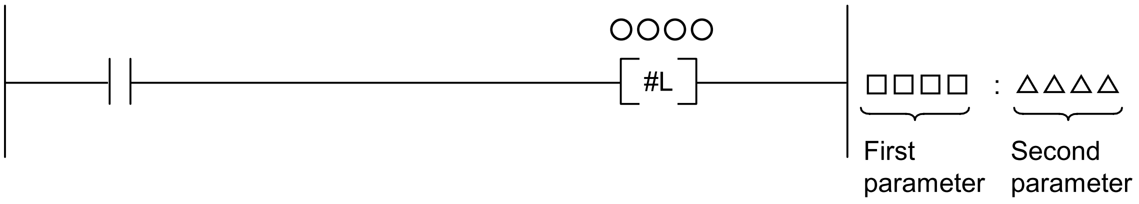
Arithmetic functions are classified into the following four types:

Name	Function	Remarks
JCMD	Service request arithmetic function. This function issues the data write service (READ/WRITE command), RESET service, and broadcast service.	RESET command, broadcast command, READ command, WRITE command
JRSP	Service check arithmetic function. This function captures the data received as a result of READ command issuance into a specified area.	READ command only
JSND	Data transmission arithmetic function. This function transmits data from the RS-232C port of a J.STATION to the external device.	Issuable only to J.STATION
JRCV	Data reception arithmetic function. This function captures the data that the RS-232C port of the J.STATION received from the external device.	Issuable only to J.STATION

⚠ **CAUTION**

Register the arithmetic functions that you wish to use in the J.NET system. (Refer to SOFTWARE MANUAL, OPTION J.NET For Windows® [manual number SAE-3-146]).

Basic form of arithmetic function:



○○○○: Arithmetic function name (JCMD, JRSP, JSND, JRCV)
□□□□: First parameter (starting address of send or receive area)
△△△△: Second parameter (number of send/receive area bytes)

- Enter a symbol (e.g., FW000) as the first parameter.
  Set the starting address of the send or receive area.
  Numeric data (e.g., extended memory address) cannot be entered directly.

- Set the data length (number of bytes) of the send/receive area.
  The second parameter setting range depends on the arithmetic function to be used.

JCMD	4 to 254 bytes
JRSP	4 to 254 bytes
JSND	4 to 516 bytes
JRCV	4 to 516 bytes

Send/receive area

	2^{15} — 2^8	2^7 — 2^0
0	MDL	SID
2	NET	SVC
4	Data 1	Data 2
6	Data 3	
		Data N

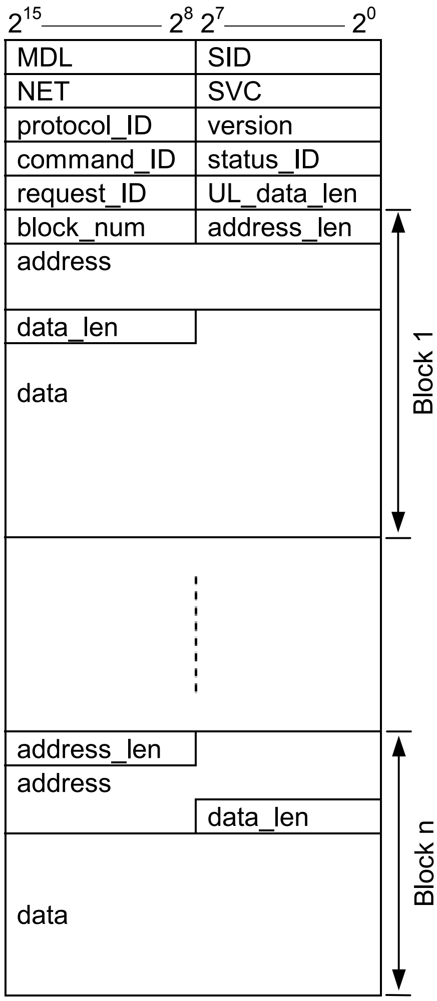
MDL: Module No. (= 0 for main module; = 1 for sub-module)
SID: Station ID (/01 to /1F)
NET: Net No. (= 0 for N1; = 1 for N2)
SVC: Service code

SVC	Service name
/31	Reset service
/32	Data write service
/33	Broadcast service

Data 1 to data N: 1 to 250 (JCMD, JRSP)
1 to 512 (JSND, JRCV)

For the WRITE or READ command, the send/receive area is configured as shown on the next page because data in the area further requires detailed setting.

Send/receive area of READ or WRITE command:



MDL: Module No. (= /00 for main module; = /01 for sub-module)
SID: Station ID (/01 to /1F)
NET: Net No. (= /00 for N1; = /01 for N2)
SVC: Service code

SVC	Service name
/32	Data write service

protocl_ID: Indicates protocol registration or non-registration. Set /00.
version: Indicates the protocol version. Set /00.
command_ID: Indicates the command identification code defined for each protocol.

Code	Command name
/11	WRITE command
/12	READ command

status_ID: Indicates a response to the command. Set /00 on command transmission.
request_ID: Indicates data for identifying the sequence of the command that retains the contents of the service until the service is completed.
UL_data_len: Set the number of UL_data bytes.
block_num: Set the number of data blocks following block_num.
address_len: Set the number of bytes of the address field.

7	6 5 ——— 0
Attribute	Byte length of address field

Attribute	Meaning
/00	The address field is a symbol (character string).
/01	The address field is a numeric value.
/10	Not used. (Do not set this attribute.)
/11	

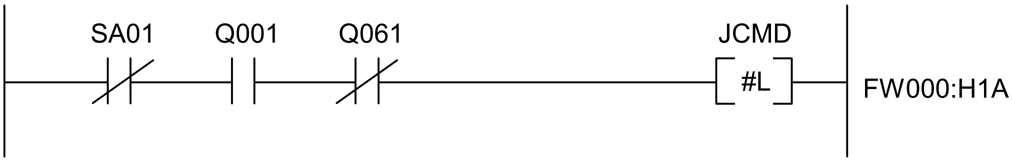
address: Set the transmission/reception destination address. Store this address, starting from the lowest-order byte.
data_len: Set the number of bytes of the data field.
data: Indicates transmitted or received data setting.

JCMD: WRITE Command

Function Transmits the command specified by the parameter.

Status Information The transmission/reception status and error information are stored in the S-register, NET status table, and S-table.

Program Example 1 An example of issuing the WRITE command when the NET status table is Q000 to Q07F and 8 bytes are written to sub-station addresses /24000 to /24007 is given below.



SA01: N1 ERR
Q001: Data can be transmitted to station 1 (SID = 1).
Q061: An error was detected in station 1 (SID = 1).

Send area setting data

Starting address specified in the first parameter

Number of bytes specified in the second parameter

	2^{15} — 2^8	2^7 — 2^0
FW000	① mdl(00)	② sid(01)
1	③ net(00)	④ svc(32)
2	⑤ protocol_ID(00)	⑥ version(00)
3	⑦ command_ID(11)	⑧ status_ID(00)
4	⑨ request_ID(00)	⑩ UL_data_len(0F)
5	⑪ block_num(01)	⑫ address_len(44)
6	⑬ address(00)	⑭ address(40)
7	⑮ address(02)	⑯ address(00)
8	⑰ data_len(08)	⑱ data1(12)
9	⑲ data2(34)	⑳ data3(56)
A	㉑ data4(78)	㉒ data5(9A)
B	㉓ data6(BC)	㉔ data7(DE)
C	㉕ data8(F0)	㉖ 00

Notes:
1. The values in parentheses in the above figure are examples of set values.
2. The set values of the shaded bytes are fixed.

① mdl: Set a module No. (= 00 for main module; = 01 for sub-module)

② sid: Set a station ID.

③ net: Set a net No.

④ svc: Set a service code (= 32H).

④ to ⑨: Set the fixed data shown in the left figure. (Values in ⑩ and after depend on transmitted data.)

⑩ UL_data_len: Set the number of bytes following block_num in ⑪.

⑪ block_num: Set the value 01 because there is only one block.

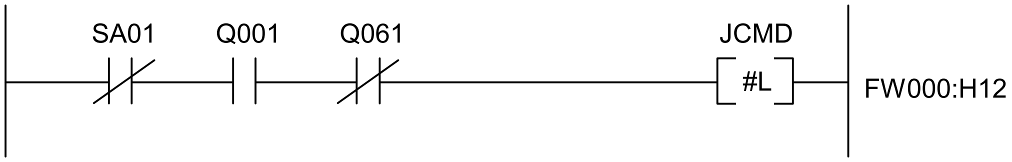
⑫ address_len: Set the value 44H because address /00024000 is a 4-byte numeric value.

⑬ to ⑯ addresses: Set the values 00, 40, 02, and 00 in this order, starting from the lowest-order byte, because the address is /00024000.

⑰ data_len: Set the number of bytes to be transmitted (number of bytes in ⑱ to ㉕).

JCMD: READ Command

Program Example 2 An example of issuing the READ command when the NET status table is Q000 to Q07F and 8 bytes are read from sub-station addresses /24000 to /24007 is given below.



SA01: N1 ERR
Q001: Data can be transmitted to station 1 (SID = 1).
Q061: An error was detected in station 1 (SID = 1).

Send area setting data

Starting address specified in the first parameter

Number of bytes specified in the second parameter

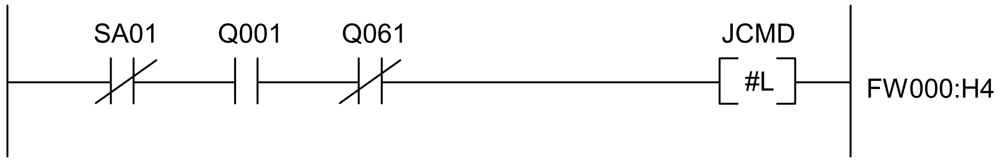
	2^{15} — 2^{8}	2^{7} — 2^{0}
FW000	① mdl(00)	② sid(01)
1	③ net(00)	④ svc(32)
2	⑤ protocol_ID(00)	⑥ version(00)
3	⑦ command_ID(12)	⑧ status_ID(00)
4	⑨ request_ID(00)	⑩ UL_data_len(07)
5	⑪ block_num(01)	⑫ address_len(44)
6	⑬ address(00)	⑭ address(40)
7	⑮ address(02)	⑯ address(00)
8	⑰ data_len(08)	⑱ 00

Notes:
1. The values in parentheses in the above figure are examples of set values.
2. The set values of the shaded bytes are fixed.

① mdl: Set a module No.
② sid: Set a station ID.
③ net: Set a net No.
④ svc: Set a service code (= 32H).
④ to ⑨: Set the fixed data shown in the left figure. (Values in ⑩ and after depend on transmitted data.)
⑩ UL_data_len: Set the number of bytes following block_num in ⑪.
⑪ block_num: Set the value 01 because there is only one block.
⑫ address_len: Set the value 44H because address /00024000 is a 4-byte numeric value.
⑬ to ⑯ addresses: Set the values 00, 40, 02, and 00 in this order, starting from the lowest-order byte, because the address is /00024000.
⑰ data_len: Set the number of bytes to be received.

JCMD: RESET Command

Program Example 3 An example of issuing the RESET command when the NET status table is Q000 to Q07F is given below.



SA01: N1 ERR
Q001: Data can be transmitted to station 1 (SID = 1).
Q061: An error was detected in station 1 (SID = 1).

Send area setting data

Starting address specified in the first parameter

Number of bytes specified in the second parameter

	2^{15} — 2^8	2^7 — 2^0
FW000	① mdl(00)	② sid(01)
1	③ net(00)	④ svc(31)

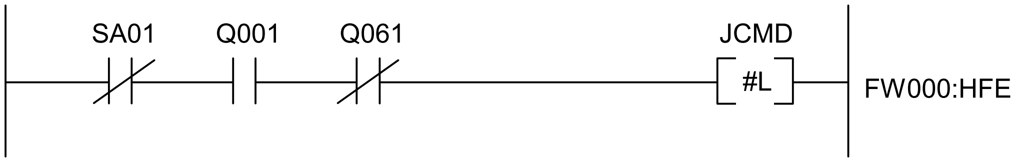
Notes:

1. The values in parentheses in the above figure are examples of set values.
2. The set values of the shaded bytes are fixed.

① mdl: Set a module No.
② sid: Set a station ID.
③ net: Set a net No.
④ svc: Set a service code (= 31H).

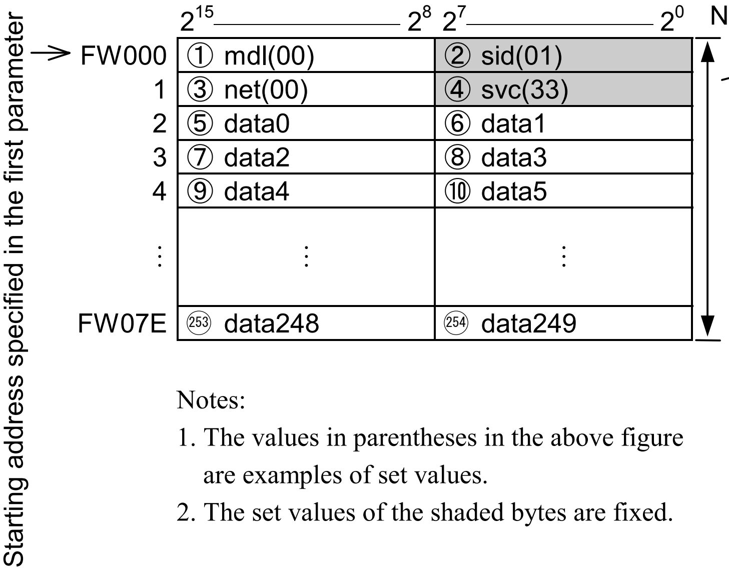
JCMD: Broadcast Command

Program Example 4 An example of issuing the Broadcast command when the NET status table is Q000 to Q07F is given below.



SA01: N1 ERR
Q001: Data can be transmitted to station 1 (SID = 1).
Q061: An error was detected in station 1 (SID = 1).

Send area setting data


Starting address specified in the first parameter
Number of bytes specified in the second parameter

	2^{15} — 2^8	2^7 — 2^0
FW000	① mdl(00)	② sid(01)
1	③ net(00)	④ svc(33)
2	⑤ data0	⑥ data1
3	⑦ data2	⑧ data3
4	⑨ data4	⑩ data5
⋮	⋮	⋮
FW07E	㉕③ data248	㉕④ data249

① mdl: Set a module No.
② sid: Set station ID 01.
③ net: Set a net No.
④ svc: Set a service code (= 33H).

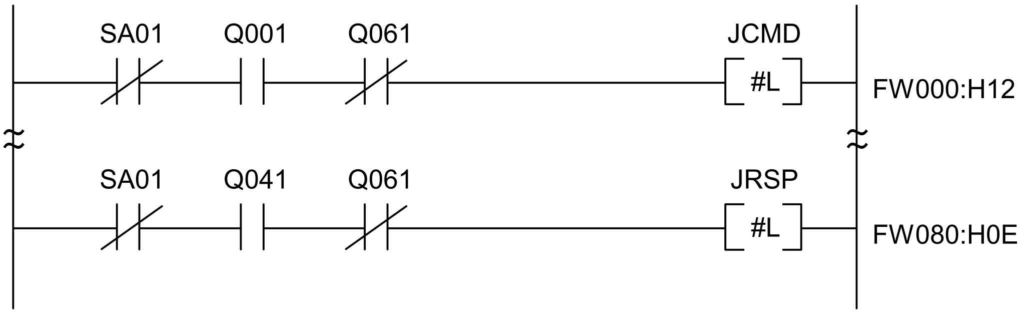
Notes:
1. The values in parentheses in the above figure are examples of set values.
2. The set values of the shaded bytes are fixed.

JRSP

Function — Captures the station data received in the J.NET buffer as a result of JCMD READ command issuance into a user-specified area.

Status Information — The transmission/reception status and error information are stored in the S-register, NET status table, and S-table.

Program Example — An example of issuing JRSP when the NET status table is Q000 to Q07F and response data from the sub-station in reply to the JCMD READ command request is captured into FW080 to FW086 is given below.



SA01: N1 ERR
Q001: Data can be transmitted (SID = 1).
Q041: Response reception completion in station 1 (SID = 1).
Q061: An error was detected in station 1 (SID = 1).

Send area setting data

Starting address specified in the first parameter

	2^{15} — 2^8	2^7 — 2^0
→ FW080	① mdl(00)	② sid(01)
1	③ net(00)	④ svc(32)
2	⑤ data0	⑥ data1
3	⑦ data2	⑧ data3
4	⑨ data4	⑩ data5
5	⑪ data6	⑫ data7
6	⑬ data8	⑭ 00

Number of bytes specified in the second parameter

① mdl: Set a module No.
② sid: Set a station ID.
③ net: Set a net No.
④ svc: Set a service code (= 32H).

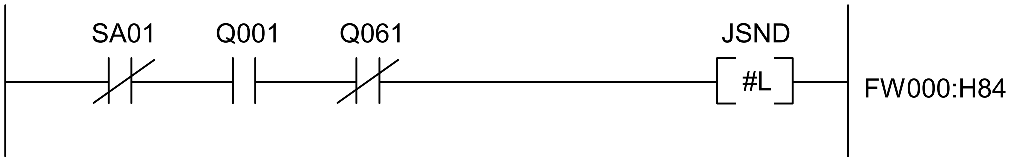
Notes:
1. The values in parentheses in the above figure are examples of set values.
2. The set values of the shaded bytes are fixed.

JSND

Function Transmits as many bytes of data as specified from the send area specified by the parameter to the external device connected to J.STATION.

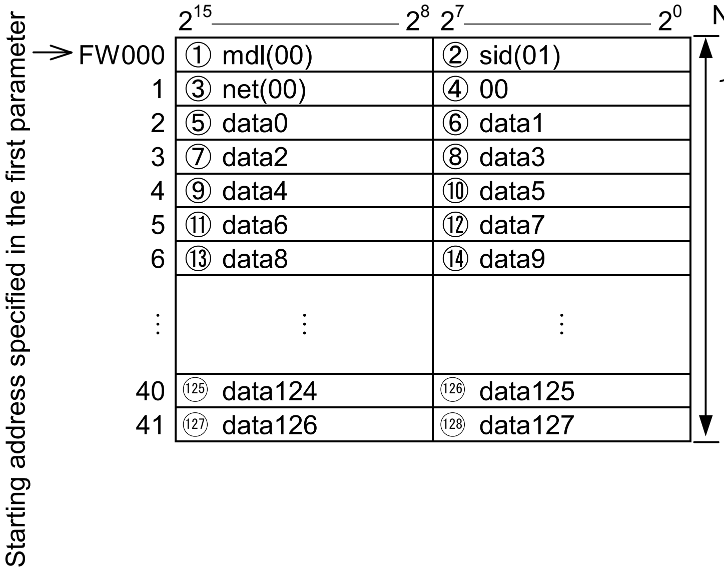
Status Information The transmission/reception status and error information are stored in the S-register, NET status table, and S-table.

Program Example An example of transmitting data of 128 bytes from FW000 to the external device connected to J.STATION when the NET status table is from Q000 to Q07F and LGB is set is given below. (Because the byte number specified by the second parameter has mdl, etc. attached to it, be sure to increment the transmitted data length by 4 bytes.)



SA01: N1 ERR
Q001: Data can be transmitted to station 1 (SID = 1).
Q061: An error was detected in station 1 (SID = 1).

Send area setting data


Starting address specified in the first parameter
Number of bytes specified in the second parameter

	2^{15} — 2^{8}	2^{7} — 2^{0}
FW000	① mdl(00)	② sid(01)
1	③ net(00)	④ 00
2	⑤ data0	⑥ data1
3	⑦ data2	⑧ data3
4	⑨ data4	⑩ data5
5	⑪ data6	⑫ data7
6	⑬ data8	⑭ data9
⋮	⋮	⋮
40	⑿⑤ data124	⑿⑥ data125
41	⑿⑦ data126	⑿⑧ data127

① mdl: Set a module No.
② sid: Set a station ID.
③ net: Set a net No.

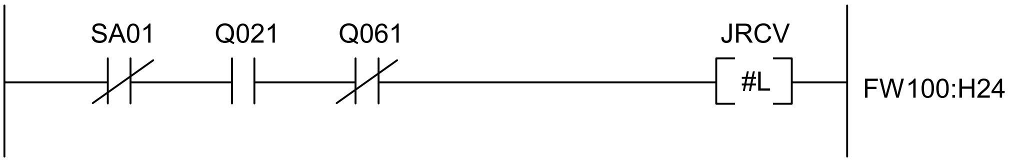
Note: The values in parentheses in the left figure are examples of set values.

JRCV

Function Transmits as many bytes of data as specified minus 4 from the external device connected with J.STATION to the receive area specified by parameters.

Status Information The transmission/reception status and error information are stored in the S-register, NET status table, and S-table.

Program Example An example of capturing the 32 bytes of data received from the external device connected to J.STATION into FW100 when the NET status table is Q000 to Q07F and LGB is set is given below. (Because the byte number specified by the second parameter has mdl, etc. attached to it, be sure to increment the received data length by 4 bytes.)



SA01: N1 ERR
Q021: Data reception completion in station 1 (SID = 1).
Q061: An error was detected in station 1 (SID = 1).

Send area setting data

Starting address specified in the first parameter

Number of bytes specified in the second parameter

	2^{15} — 2^8	2^7 — 2^0
→ FW100	① mdl(00)	② sid(01)
1	③ net(00)	④ 00
2	⑤ data0	⑥ data1
3	⑦ data2	⑧ data3
4	⑨ data4	⑩ data5
5	⑪ data6	⑫ data7
6	⑬ data8	⑭ data9
⋮	⋮	⋮
40	㉝ data28	㉞ data29
41	㉟ data30	㊱ data31

① mdl: Set a module No.
② sid: Set a station ID.
③ net: Set a net No.

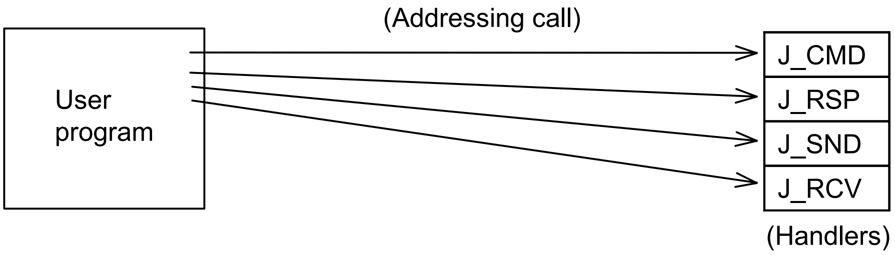
Note: The values in parentheses in the left figure are examples of set values.

4.7.2 Subroutines

Like arithmetic functions, subroutines are classified into the following four types:

Name	Function	Link address
J_CMD	Service request subroutine, which issues the data write service (READ/WRITE command), RESET service, and broadcast service.	/A0040C
J_RSP	Service subroutine, which captures the data received as a result of READ command issuance into a specified area.	/A00412
J_SND	Data transmission subroutine, which transmits data from the RS-232C port of J.STATION to the external device.	/A00400
J_RCV	Data reception subroutine, which captures the data that the RS-232C port of J.STATION received from the external device.	/A00406

Code the user program in such a programming language as C, 68000 assembly language, etc. The handlers (subroutines) of the J.NET module are called by specifying an address from the user program. For this reason, the handlers (subroutines) cannot be created (linked) when they are included in the user program.



J_CMD

Function Transmits the command specified by parameters.

Link Procedure

C	Assembly language

long (* j_cmd)();
long rtn;
struct para *para;
j_cmd=0xa0040c;
rtn=(*j_cmd)(para);
	movea.l #$a0040c,a0
	movea.l #para,-(sp)
	jsr (a0)
	addq.l #4,sp
	para:
	Parameter

CAUTION

- The assembly language guarantees the contents of registers other than the DO register that stores return codes. (In C, these registers are transparent to the user.)
- The J_CMD subroutine uses 128 bytes of the user program stack.

Parameters

```
    struct para {
        unsigned char mdl;      /* Module No. (= 0 for main module; = 1 for sub-module) */
        unsigned char sid;      /* Station ID (/01 to /1F) */
        unsigned char net;      /* Net No. (= 0 for N1; = 1 for N2) */
        unsigned char svc;      /* Service code */
        unsigned long adr;      /* Starting address of transmitted data */
        unsigned short len;     /* Number of bytes of transmitted data (/01 to /FA) */
    } ;
```

Service code	/31: Reset service
	/32: Data write service
	/33: Broadcast service
Transmitted data	JEM-F3008 data section

Return Codes

=0: Normal termination

=/FFFFFFFF: Abnormal termination

Error information is stored in the error code area of the S-table and the error flag of the NET status.

=/80000000: Input parameter error

An input parameter error results when the mdl, sid, net, svc, or len parameter is outside the range or when the station number of the slave parameter table (SVPT) is undefined.

Note

long (*f) (): Declares function f to be returned as the function value of the pointer to a double-precision integer.

J_RSP

Function Captures as many bytes of data as specified into the area specified by parameters (valid when a response is received from the station as a result of J_CMD command [READ] issuance).

Link Procedure

C	Assembly language

long (* j_rsp)();
long rtn;
struct para *para;
j_rsp=0xa00412;
rtn=(*j_rsp)(para);
	movea.l #$a00412,a0
	movea.l #para,-(sp)
	jsr (a0)
	addq.l #4,sp
	para:
	Parameter

CAUTION

- The assembly language guarantees the contents of registers other than the DO register that stores return codes. (In C, these registers are transparent to the user.)
- The J_RSP subroutine uses 128 bytes of the user program stack.

Parameters

```
struct para {
        unsigned char mdl;      /* Module No. (= 0 for main module; = 1 for sub-module) */
        unsigned char sid;      /* Station ID (/01 to /1F) */
        unsigned char net;      /* Net No. (= 0 for N1; = 1 for N2) */
        unsigned char svc;      /* Service code */
        unsigned long adr;      /* Starting address of received data */
        unsigned short len;     /* Number of bytes of received data (/01 to /FA) */
} ;
```

Service code /32: Data write service

Return Codes

=0: Normal termination

If there is still uncaptured data after data capturing, the response reception flag of the NET status remains set (data received).

=1: No received data is in the receive buffer.

=/001A0000: The last data appears in the buffer when the received data was being captured.

This return code is set when the number of bytes specified by parameter is greater than or equal to the number of bytes actually received.

=/FFFFFFFF: Abnormal termination

Error information is stored in the error code area of the S-table and the error flag of the NET status.

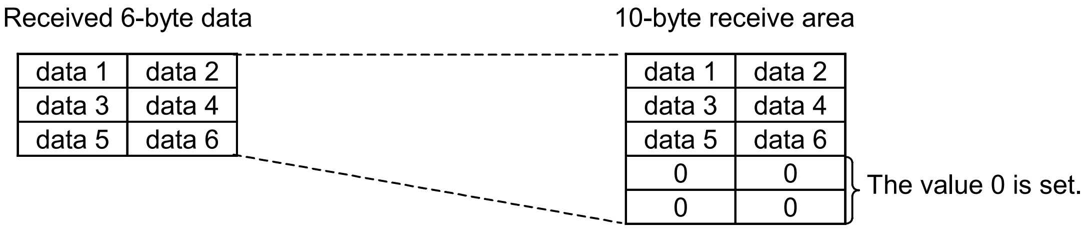
=/80000000: Input parameter error

An input parameter error results when the mdl, sid, net, svc, or len parameter is outside the range or when the station number of the slave parameter table (SVPT) is undefined.

Note

If the number of bytes specified by the parameter is greater than the number of actually received bytes in received data capturing, the value 0 is set after the received data in the receive area.

An example when 10 bytes are specified as the bytes received and 6-byte data is actually received is given below.



Received 6-byte data

data 1	data 2
data 3	data 4
data 5	data 6

10-byte receive area

data 1	data 2
data 3	data 4
data 5	data 6
0	0
0	0

The value 0 is set.

J_SND

Function Transmits as many bytes of data as specified from the area specified by parameters to the external device connected to J.STATION.

Link Procedure

C	Assembly language

long (* j_snd)();
long rtn;
struct sr_para *padr;
j_snd=0xa00400;
rtn=(*j_snd)(padr);
	movea.l #$a00400,a0
	movea.l #sr_para,-(sp)
	jsr (a0)
	addq.l #4,sp
	sr_para:
	Parameter

⚠ CAUTION

- The assembly language guarantees the contents of registers other than the DO register that stores return codes. (In C, these registers are transparent to the user.)
- The J_SND subroutine uses 128 bytes of the user program stack.
- Data transmission by the J_SND subroutine affects the I/O service refresh cycle because it is synchronized with the I/O service.

Parameters

```
struct sr_para {
        unsigned char mdl;      /* Module No. (= 0 for main module; = 1 for sub-module) */
        unsigned char sid;      /* Station ID (/01 to /1F) */
        unsigned char net;      /* Net No. (= 0 for N1; = 1 for N2) */
        unsigned char fu;       /* Unoccupied */
        unsigned long adr;      /* Starting address of transmitted data */
        unsigned short len;     /* Number of bytes of transmitted data (/01 to /200) */
} ;
```

Return Codes

=0: Normal termination

=/FFFFFFFF: Abnormal termination

Error information is stored in the error code area of the S-table and the error flag of the NET status.

=/80000000: Input parameter error

An input parameter error results when the mdl, sid, net, or len parameter is outside the range or when the station number of the slave parameter table (SVPT) is not registered.

J_RCV

Function Captures as many bytes of data as specified from the external device connected to J.STATION into the receive area specified by parameters.

Link Procedure

C	Assembly language

long (* j_rcv)();
long rtn;
struct sr_para *padr;
j_rcv=0xa00406;
rtn=(*j_rcv)(padr);
	movea.l #$a00406,a0
	movea.l #sr_para,-(sp)
	jsr (a0)
	addq.l #4,sp
	sr_para:
	Parameter

CAUTION

- The assembly language guarantees the contents of registers other than the DO register that stores return codes. (In C, these registers are transparent to the user.)
- The J_RCV subroutine uses 128 bytes of the user program stack.

Parameters

```
struct sr_para {
        unsigned char mdl;      /* Module No. (= 0 for main module; = 1 for sub-module) */
        unsigned char sid;      /* Station ID (/01 to /1F) */
        unsigned char net;      /* Net No. (= 0 for N1; = 1 for N2) */
        unsigned char fu;       /* Unoccupied */
        unsigned long adr;      /* Starting address of received data */
        unsigned short len;     /* Number of bytes of received data (/01 to /200) */
} ;
```

Return Codes

=0: Normal termination

If there is still uncaptured data after data capturing, the response reception flag of the NET status remains set (data received).

=1: No received data is in the receive buffer.

=/001A0000: The last data appears in the buffer when received data was being captured. This return code is set when the number of bytes specified by parameter is greater than or equal to the number of bytes actually received.

=/FFFFFFFF: Abnormal termination

Error information is stored in the error code area of the S-table and the error flag of the NET status.

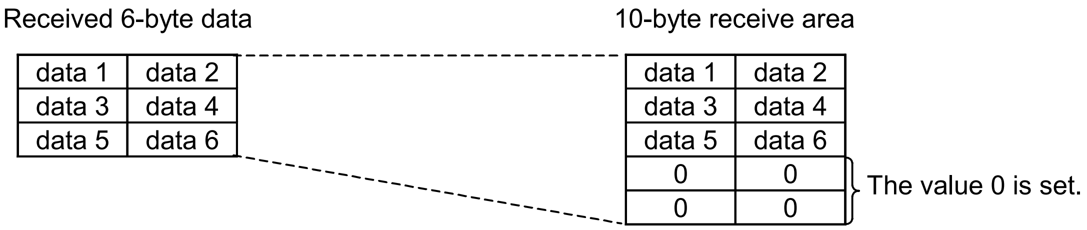
=/80000000: Input parameter error

An input parameter error results when the mdl, sid, net, or len parameter is outside the range or when the station number of the slave parameter table (SVPT) is not registered.

Note

If the number of bytes specified by the parameter is greater than that of actually received bytes in received data capturing, the value 0 is set after the received data in the receive area.

An example when 10 bytes are specified as the bytes received and 6-byte data is actually received is given below.



Received 6-byte data

data 1	data 2
data 3	data 4
data 5	data 6

10-byte receive area

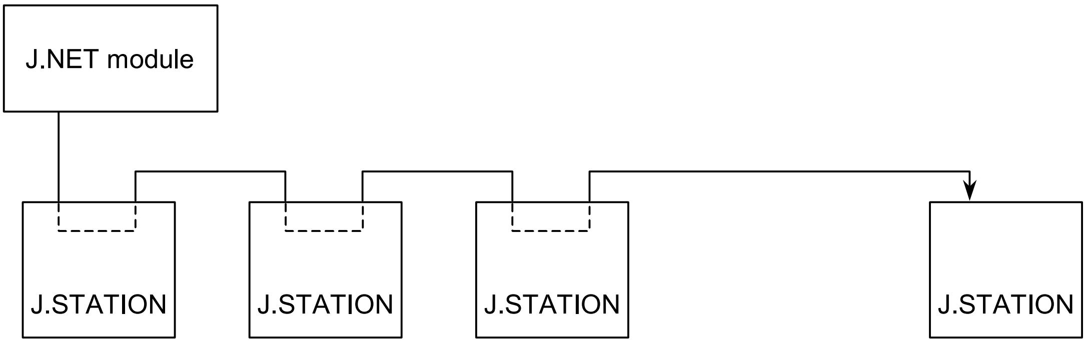
data 1	data 2
data 3	data 4
data 5	data 6
0	0
0	0

The value 0 is set.

4.8 Communication Time

The J.NET module communication time depends on how the J.NET system is configured.

- When the J.NET module is connected to J.STATIONs (I/O communication only and data not transmitted)

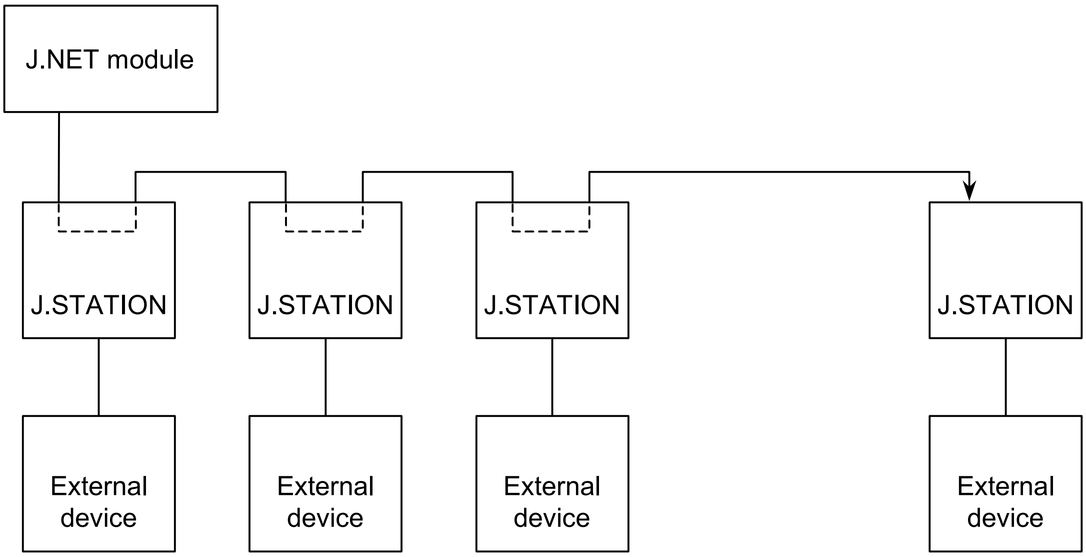


J.STATIONs are mounted on an I/O unit to constitute a remote I/O unit.
The J.NET module automatically inputs and outputs data to and from PI/O units (I/O service) in the set refresh cycle.
The time that the J.NET module and a J.STATION require for a single I/O communication (I/O service) depends on the number of PI/O points.

For N-point I/O unit	1.85+0.0031N [ms] (1 Mbps)
	2.35+0.0054N [ms] (0.5 Mbps)
	3.45+0.0092N [ms] (0.25 Mbps)
	4.85+0.0176N [ms] (0.125 Mbps)

Set a greater value than the total I/O communication time of all J.STATIONs as a refresh cycle.

- When the J.NET module is connected to J.STATIONs (data transmitted)



An external device (e.g., PC) can be connected to a J.STATION because the J.STATION has an RS-232C interface.

The time that the J.NET module and an external device require for a single communication is calculated from the following expression:

Communication time = A + B + C

A: J.NET module communication time (I/O service + message)

B: RS-232C interface communication time

C: External device processing time

The time that the J.NET module requires for a single communication (I/O service + message) depends on the number of transmission/reception bytes per communication. It is calculated by adding any of the following times to the I/O communication (I/O service) time explained on the previous page.

For N-byte transmission	0.014N [ms]	(1 Mbps)
For N-byte reception	0.025N [ms]	
For N-byte transmission	0.022N [ms]	(0.5 Mbps)
For N-byte reception	0.033N [ms]	
For N-byte transmission	0.038N [ms]	(0.25 Mbps)
For N-byte reception	0.049N [ms]	
For N-byte transmission	0.067N [ms]	(0.125 Mbps)
For N-byte reception	0.078N [ms]	

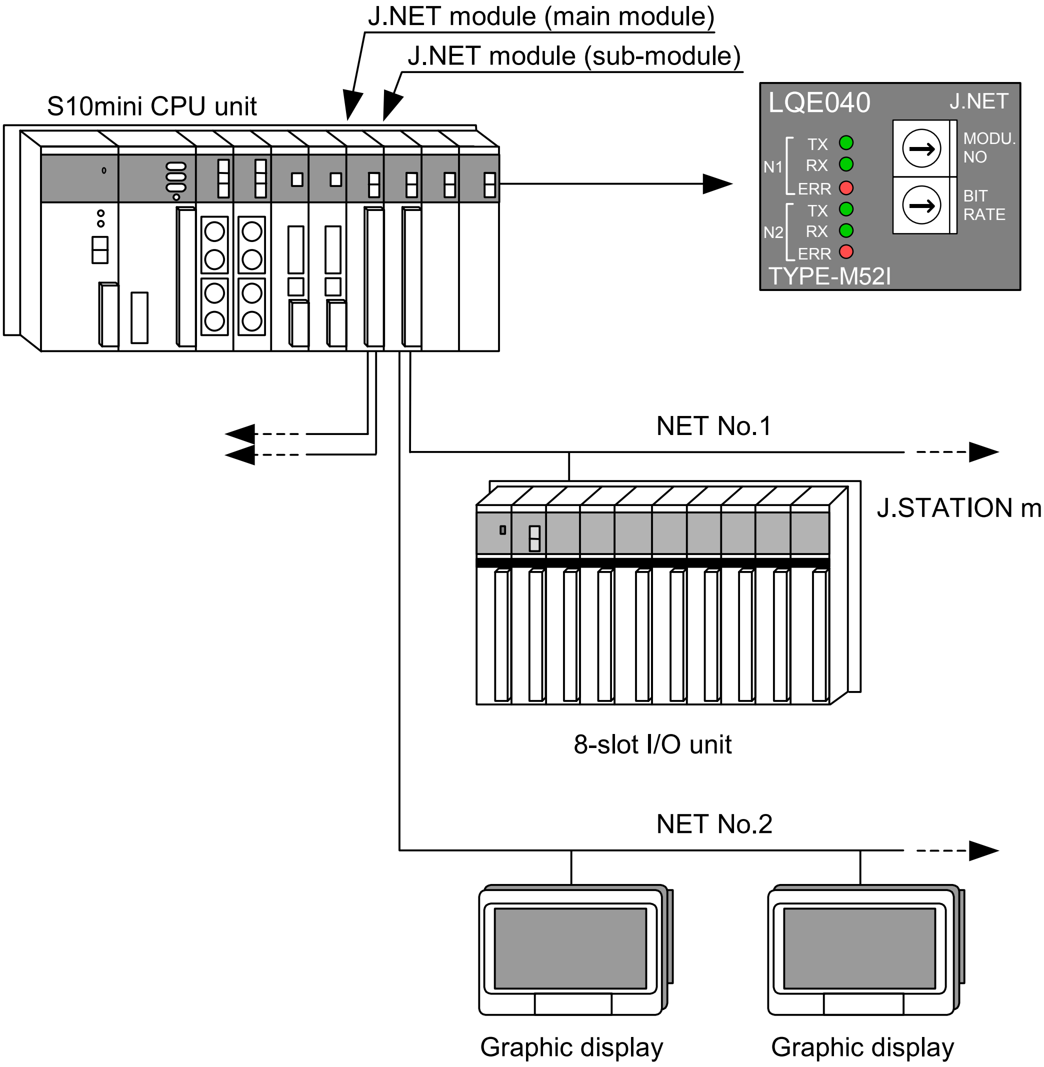
An external device communication delay does not affect the J.NET communication time because each J.STATION has a buffer.

5 OPERATION

5.1 About J.NET SUPPORT System

The J.NET SUPPORT system is a man-machine tool that sets communication information for J.NET modules and stations.

5.1.1 System configuration

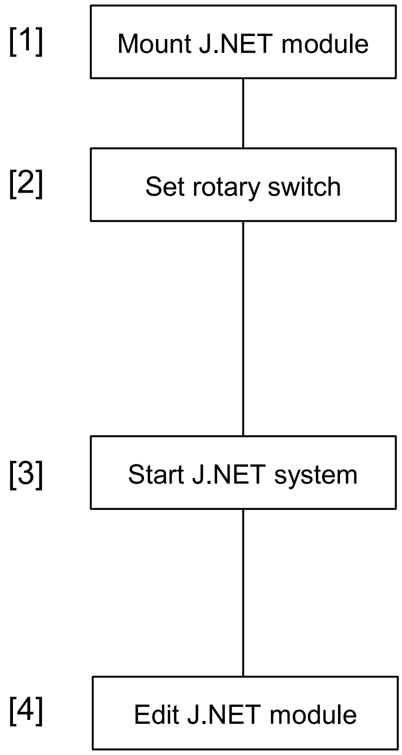


No.	Module
0	Main module
1	Sub-module

When using only one J.NET module, be sure to set it up as the main module. When using two J.NET modules, be sure to set up one module as the main module and the other as the sub-module.

5.2 System Startup

5.2.1 J.NET SUPPORT system startup procedure



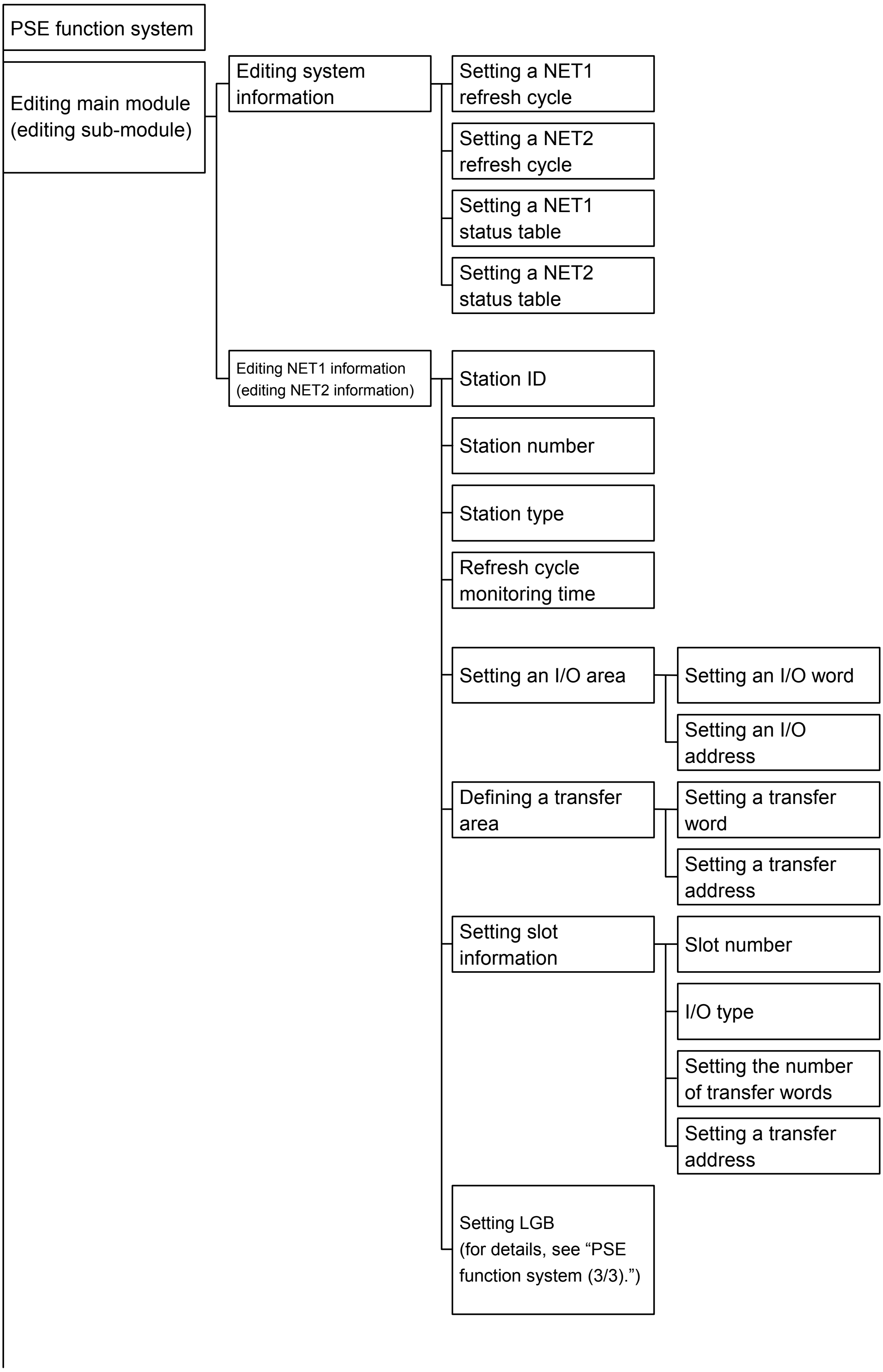
[1] Power down the CPU and then mount the J.NET module.

[2] (a) Set the MODU No. switch of the J.NET module to 0 (main module) or 1 (sub-module).
(b) Use the BIT RATE switch of the J.NET module to set the bit rate to 0 (1 Mbps), 1 (0.5 Mbps), 2 (0.25 Mbps), or 3 (0.125 Mbps).

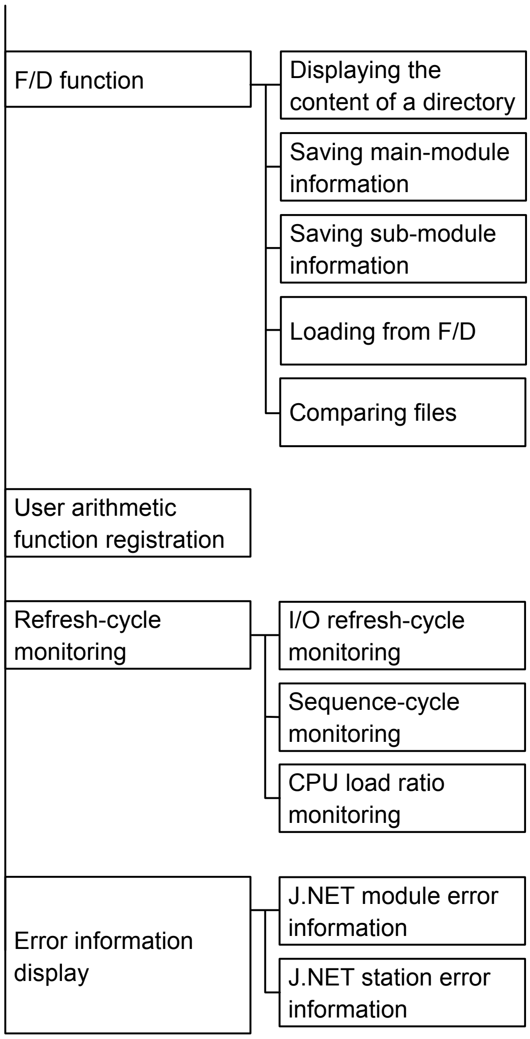
[3] Connect the CPU to the Windows personal computer, and then start up the J.NET system. (Refer to the SOFTWARE MANUAL OPTION J.NET For Windows® [manual number SAE-3-146]).

[4] Edit the J.NET module. (Refer to the SOFTWARE MANUAL OPTION J.NET For Windows® [manual number SAE-3-146]).

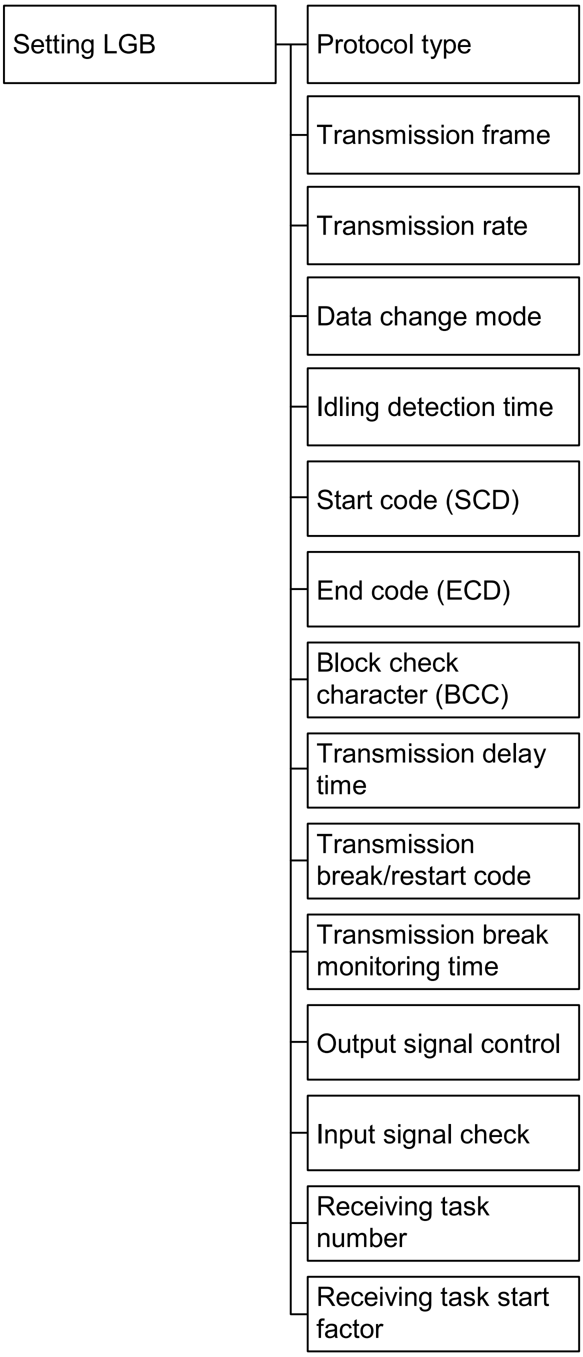
5.2.2 PSE function system (1/3)


PSE function system
Editing main module (editing sub-module)
Editing system information
Setting a NET1 refresh cycle
Setting a NET2 refresh cycle
Setting a NET1 status table
Setting a NET2 status table
Editing NET1 information (editing NET2 information)
Station ID
Station number
Station type
Refresh cycle monitoring time
Setting an I/O area
Setting an I/O word
Setting an I/O address
Defining a transfer area
Setting a transfer word
Setting a transfer address
Setting slot information
Slot number
I/O type
Setting the number of transfer words
Setting a transfer address
Setting LGB (for details, see "PSE function system (3/3).")

PSE function system (2/3)


F/D function
Displaying the content of a directory
Saving main-module information
Saving sub-module information
Loading from F/D
Comparing files
User arithmetic function registration
Refresh-cycle monitoring
I/O refresh-cycle monitoring
Sequence-cycle monitoring
CPU load ratio monitoring
Error information display
J.NET module error information
J.NET station error information

PSE function system (3/3)



5.3 Editing System Information

- Setting a NET1 (NET2) refresh cycle
  Set an I/O service refresh cycle. (If a sequence cycle is selected, the I/O service is refreshed in sync with the sequence cycle.)

Setting range	3 to 3000	Unit: ms

- Setting a NET1 (NET2) status table
  Set a NET1 or NET2 status table.

Setting range	X000 to XFFF Y000 to YFFF J000 to JFFF Q000 to QFFF G000 to GFFF R000 to RFFF E400 to EFFF M000 to MFFF

5.4 Editing NET1 (NET2) Information

5.4.1 Station ID

Select a station ID (serial station information No.).

Setting range	/01 to /1F

- Station number
  Set the station number to be assigned to the sub-station.

Setting range	/00 to /7F

- Station type
  Set the station type of the sub-station. (Select one of the station types listed in the table below according to the sub-station specifications.)

Station type	Specifications	Necessary operation			
		I/O area definition	Transfer area definition	Slot information setting	LGB setting
AUTO	I/O as well as unspecified I/O transfer and PUT/GET by polling	√	–	–	–
I/O	I/O as well as specified I/O transfer	–	–	√	–
I/O+DR/DW	I/O as well as specified I/O transfer and PUT/GET by polling	–	–	√	–
DR/DW	PUT/GET by polling only	–	–	–	–
J.STATION (EXTENDED)	Hitachi Private Specifications (setting of slot information and LGB)	–	–	√	√
J.STATION (STANDARD)	Hitachi Private Specifications (setting of the number of transfer words and transfer address only)	–	√	–	–

√: Necessary –: Not necessary

- Refresh cycle monitoring time
  Set the refresh cycle monitoring time of the sub-station. This refresh cycle must be at least five times longer than the NET1 (NET2) refresh cycle setting.

Setting range	0 to 65535

Unit: 10 ms

5.4.2 Setting an I/O area

- Setting the number of input and output words

Setting range	/00 to /80

- Setting input and output addresses

Setting range	FW000 to FWBFF XW000 to XWFF0 YW000 to YWFF0 JW000 to JWFF0 QW000 to QWFF0 GW000 to GWFF0 RW000 to RWFF0 EW400 to EWFF0 MW000 to MWFF0 /100000 to /4FFFFE (extended memory)

5.4.3 Defining a transfer area

- Setting the number of transfer words

Setting range	/01 to /40

- Setting input and output transfer addresses

  The input and output transfer areas used are always the following two areas: an XW area for input and a YW area for output.

  Set the input transfer address in the “XW□□0” format.　Enter a value from 00 to FF for □□.

  Set the output transfer address in the “YW□□0” format.　Enter a value from 00 to FF for □□.

Setting range	/00 to /FF

5.4.4 Setting slot information

- Slot No.

  Select the number of the slot into which the set module is to be inserted.

Setting range	/0 to /F

- I/O type

  Set the I/O module to be inserted into each slot.

Setting value	Remarks
Slot information deletion	
DI	
DO	
AI	
AO	
S10 AI (4 ch)	Only J.STATION (EXTENDED) is valid
S10 AO (4 ch)	
S10 PCT (Pulse counter)	

⚠ CAUTION

- The analog I/O (S10 AI/S10 AO) module and pulse counter (S10 PCT) addresses set on the J.NET side must not overlap the CPU addresses (EW400 to EWFF0) set on the analog counter screen. Failure to observe this rule may result in a malfunction.
- When using S10/2α series or S10mini series AI or AO, set the I/O type according to the following rules:

S10/2α series	S10mini series	Set value
When the LWA series is used:	When the MODE switch is set in "1" position:	AI or AO
When the PAF series or PAN series is used:	When the MODE switch is set in "2" position:	S10 AI (4-channel) S10 AO (4-channel)

- Setting the number of transfer words

Setting range	/01 to /80

- Setting transfer addresses

Setting range	FW000 to FWBFF XW000 to XWFF0 YW000 to YWFF0 JW000 to JWFF0 QW000 to QWFF0 GW000 to GWFF0 RW000 to RWFF0 EW400 to EWFF0 MW000 to MWFF0 /100000 to /4FFFFE (extended memory)

5.4.5 Setting LGB

(1) Protocol type (PROTOCOL TYPE)

PSE selection item	Display data	Remarks
NOT USE	NOT USE	Initial value
USE (RS-232C)	USE (RS-232C)	

(2) Transmission frame (DATA FRAME)

PSE selection item	Display data	Transmission frame	Remarks
ST+7DT+EP+2SP	ST+7DT+EP+2SP	ST 2^0 … 2^6 EP SP SP	
ST+7DT+OP+2SP	ST+7DT+OP+2SP	ST 2^0 … 2^6 OP SP SP	
ST+7DT+EP+1SP	ST+7DT+EP+1SP	ST 2^0 … 2^6 EP SP	
ST+7DT+OP+1SP	ST+7DT+OP+1SP	ST 2^0 … 2^6 OP SP	
ST+7DT+2SP	ST+7DT+2SP	ST 2^0 … 2^6 SP SP	
ST+7DT+1SP	ST+7DT+1SP	ST 2^0 … 2^6 SP	
ST+8DT+EP+2SP	ST+8DT+EP+2SP	ST 2^0 … 2^7 EP SP SP	
ST+8DT+OP+2SP	ST+8DT+OP+2SP	ST 2^0 … 2^7 OP SP SP	
ST+8DT+EP+1SP	ST+8DT+EP+1SP	ST 2^0 … 2^7 EP SP	
ST+8DT+OP+1SP	ST+8DT+OP+1SP	ST 2^0 … 2^7 OP SP	Initial value
ST+8DT+2SP	ST+8DT+2SP	ST 2^0 … 2^7 SP SP	
ST+8DT+1SP	ST+8DT+1SP	ST 2^0 … 2^7 SP	

Symbols in the above table have the following meanings:

ST: Start bit
DT: Data bit
EP: Even parity bit
OP: Odd parity bit
SP: Stop bit

(3) Transmission rate (BAUD RATE)

PSE selection item	Display data	Remarks
150 [BPS]	150 [BPS]	
300 [BPS]	300 [BPS]	
600 [BPS]	600 [BPS]	
1200 [BPS]	1200 [BPS]	
2400 [BPS]	2400 [BPS]	
4800 [BPS]	4800 [BPS]	
9600 [BPS]	9600 [BPS]	
19200 [BPS]	19200 [BPS]	Initial value

BPS: Bits per second

(4) Data change mode (DATA CHANGE MODE)

PSE selection item	Display data	Remarks
BINARY	BINARY	Text data is transferred as binary data as is. (Initial value)
ASCII	ASCII	Text data is converted to ASCII format and transferred.

(5) Idling detection time (IDLE TIME)

Setting value	Display data	Idling detection time	Remarks
1 to 32767	00001 to 32767[*10mSEC]	10 to 327670[ms]	Initial value (1=10 ms)

(6) Start code (SCD: START CODE)

PSE selection item	Display data	Remarks
NO START CODE	NO START CODE	Initial value
1 START CODE	1 START CODE CD1	
2 START CODE	2 START CODE CD1+CD2	
3 START CODE	3 START CODE CD1+CD2+CD3	
4 START CODE	4 START CODE CD1+CD2+CD3+CD4	

CD1 to CD4: Hexadecimal numbers representing 00H to FFH start codes.

(7) End code (ECD: END CODE)

PSE selection item	Display data	Remarks
NO END CODE	NO END CODE	Initial value
1 END CODE	1 END CODE CD1	
2 END CODE	2 END CODE CD1+CD2	
3 END CODE	3 END CODE CD1+CD2+CD3	
4 END CODE	4 END CODE CD1+CD2+CD3+CD4	

CD1 to CD4: Hexadecimal numbers representing 00H to FFH end codes.

(8) Block check character (BCC: BCC MODE)

PSE selection item	Display data	Remarks
NO BCC	NO BCC	Initial value
ODD PARITY	ODD PARITY	
EVEN PARITY	EVEN PARITY	

(9) Transmission delay time (SEND DELAY TIME)

Setting value	Display data	Remarks
0	NO DELAY	No data transmission delay (Initial value)
1 to 32767	00001 to 32767[*10mSEC]	10 to 327670[ms]

Restrictions:

Select a transmission delay time within one of the setting ranges listed below, depending on the selected baud rate.

Baud rate	Transmission delay time setting range
150[BPS]	13 to 32,767 [*10mSEC]
300[BPS]	7 to 32,767 [*10mSEC]
600[BPS]	4 to 32,767 [*10mSEC]
1200[BPS]	2 to 32,767 [*10mSEC]
2400[BPS]	1 to 32,767 [*10mSEC]
4800[BPS]	1 to 32,767 [*10mSEC]
9600[BPS]	1 to 32,767 [*10mSEC]
19200[BPS]	1 to 32,767 [*10mSEC]

(10) Transmission break/restart code (SEND BREAK/CONTINUE)

PSE selection item	Display data	Remarks
NO BREAK/CONT.	NO BREAK/CONTINUE	Initial value
1BR + 1CD	BR:CD1 CD:CD2	
1BR + 2CD	BR:CD1 CD:CD2+CD3	
2BR + 1CD	BR:CD1+CD2 CD:CD3	
2BR + 2CD	BR:CD1+CD2 CD:CD3+CD4	

BR: Break code, CD: Restart code

CD1 to CD4: Hexadecimal numbers representing 00H to FFH transmission break/restart codes.

(11) Transmission break monitoring time (SEND BREAK TIMEOUT)

Setting value	Display data	Remarks
0	NO TIMEOUT	Transmission break is not monitored. (Initial value)
1 to 32767	00001 to 32767[*10mSEC]	10 to 327670[ms]

(12) Output signal control (OUTPUT SIGNAL (RS, ER))

PSE selection item	Display data	Remarks
OPEN	Output signals are not controlled.	Initial value
CONTROL	The RS and ER signals are controlled.	

(13) Input signal check (INPUT SIGNAL (CS, DR, CD))

PSE selection item	Display data	Remarks
NO CHECK	NO CHECK	Initial value
CHECK	CHECK	

(14) Receiving task number (RECEIVE TASK No.)

Setting value	Display data	Remarks
0	Unused	Initial value
001 to 127	001 to 127 (task number)	

(15) Receiving task start factor (TASK FACT)

Setting value	Display data	Remarks
0	Unused	Initial value
01 to 16	01 to 16: (Start factor)	

• Operation

LGB editing operations are divided into selection type, setting type, and combination type according to the items to be edited.

- Selection type: The item to be edited is selected from the data menu.
- Setting type: A numeric value is set within the range shown in the data menu.
- Combination type: A setting pattern is selected from the data menu and a numeric value is set according to the pattern.

LGB MENU No.	Editing item	Operation type
0	Protocol type	Selection type
1	Transmission frame	Selection type
2	Transmission rate	Selection type
3	Data change mode	Selection type
4	Idling detection time	Setting type
5	Start code	Combination type
6	End code	Combination type
7	BCC	Selection type
8	Transmission delay time	Setting type
9	Transmission break/restart code	Combination type
A	Transmission break monitoring time	Setting type
B	Output signal control	Selection type
C	Input signal check	Selection type
D	Receiving task number	Setting type
E	Receiving task start factor	Setting type

⚠ **CAUTION**

- As an LGB editing item, no item is included that sets the number of transfer words.
- Post-transmission data is transmitted and received only as much as specified by JSND, JRCV (J_SND, J_RCV). For details, see Section 4.7, "Handlers."

5.5 User Arithmetic Function Registration

5.5.1 Outline of function

This function registers or deletes an arithmetic function in or from a user function edition table (UFET).

5.5.2 Arithmetic functions

J.NET SYSTEM provides the four arithmetic functions listed in the table below.

Name	Function
JSND	Data transmission (non-procedural)
JRCV	Data reception (non-procedural)
JCMD	Service request (reset, data write, broadcast)
JRSP	Service check (data write [READ])

6 MAINTENANCE

6.1 Maintenance and Check

6.1.1 Periodic check

Item	Description	Frequency
Unit cleaning	Turn off all the power supplies and then vacuum the interior of the J.NET module through the slits in its casing. Do not raise dust during cleaning.	Once/year
Mechanical check	Check J.NET module mounting screws, TB mounting screws, and communication cable mounting screws for looseness and damage. If a mounting screw is loose, tighten it. Replace damaged parts.	Once/year
Operation check	Check J.NET module operation with a test/maintenance program (T/M). (T/M is started automatically due to any power failure and recovery after the Module No. and BIT RATE switches have been set.)	Once/year

⚠ **CAUTION**

Be sure to use T/M offline. Failure to observe this rule may lead to a program malfunction.

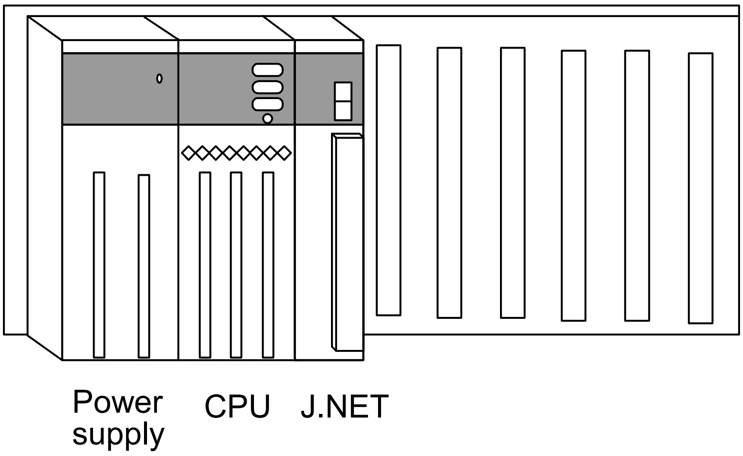
6.1.2 Test/maintenance program (T/M)

T/M is a program for J.NET module maintenance check and can be started automatically due to any power failure and recovery after the MODU No. and BIT RATE switches have been set.

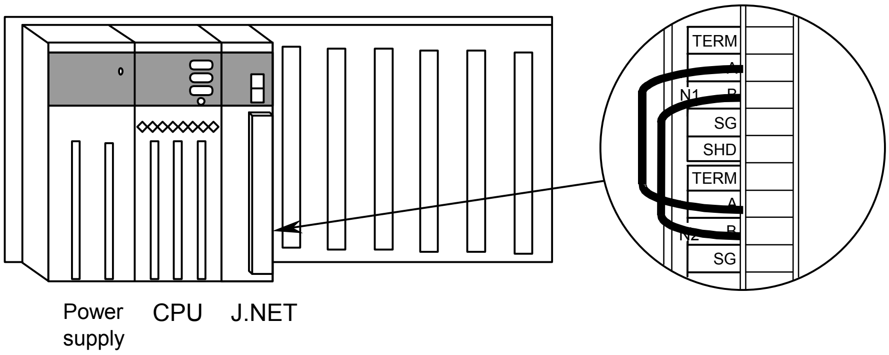
No.	MODU No.	BIT RATE	Description	Cabling
1	8	8	Internal loopback communication (main module)	Unnecessary
	9		Internal loopback communication (sub-module)	Unnecessary
2	8	9	Internal memory write/read/compare in J.NET module (main module)	Unnecessary
	9		Internal memory write/read/compare in J.NET module (sub-module)	Unnecessary
3	8	A	CPU memory function check (main module)	Unnecessary
	9		CPU memory function check (sub-module)	Unnecessary
4	8	B	Interrupt function check (main module)	Unnecessary
	9		Interrupt function check (sub-module)	Unnecessary
5	8	C	No. 2/3/4/6 combination check (main module)	Necessary
	9		No. 2/3/4/6 combination check (sub-module)	Necessary
6	8	D	External loopback communication (main module)	Necessary
	9		External loopback communication (sub-module)	Necessary
7	8	E	NET1 external loopback communication (main module)	Necessary
	9		NET1 external loopback communication (sub-module)	Necessary
8	8	F	NET1/NET2 external loopback communication (main module)	Necessary
	9		NET1/NET2 external loopback communication (sub-module)	Necessary

6.1.3 Hardware configuration for T/M operation

- T/M No.1 to T/M No. 4 (cabling unnecessary)

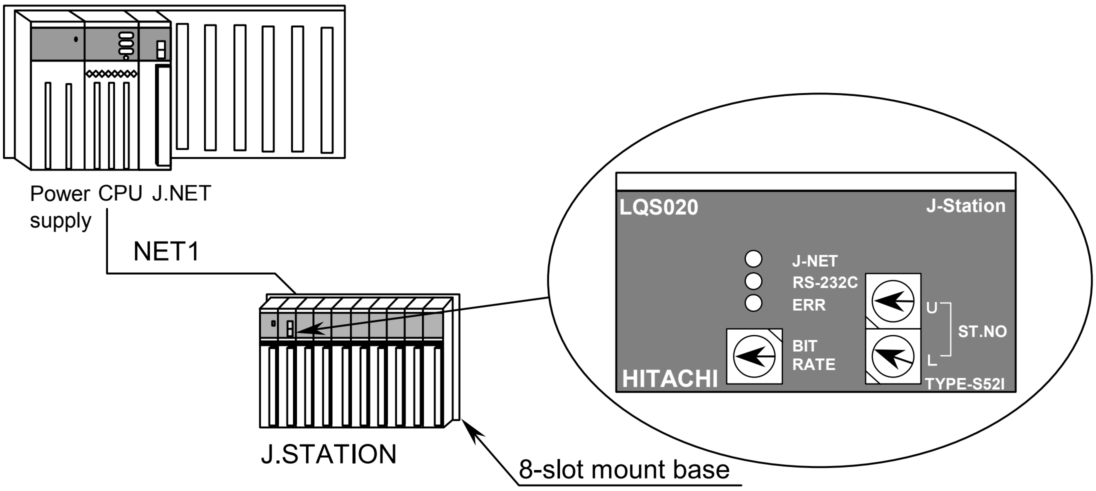


- T/M No.5 and T/M No.6 (cabling is performed as shown in the figure below)



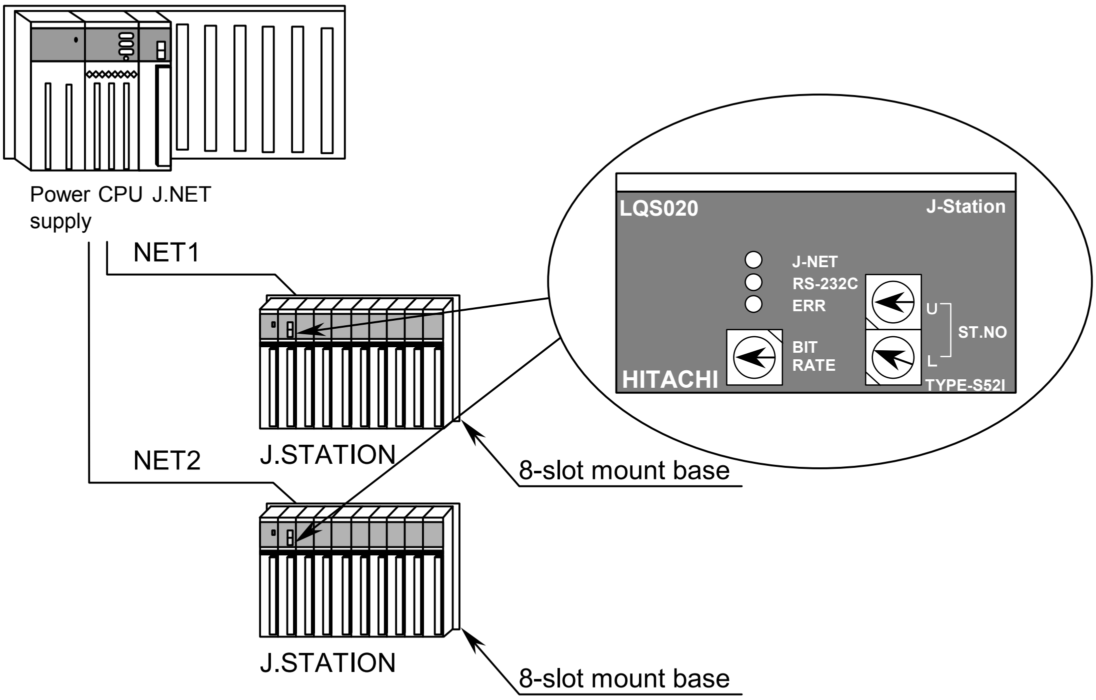
Short-circuit the A-terminals of N1 and N2, and the B-terminals of N1 and N2 as well, with jumpers.

- T/M No.7 (cabling is performed as shown in the figure below)



For J.STATION, set BIT RATE to 8 and ST No. to /01.

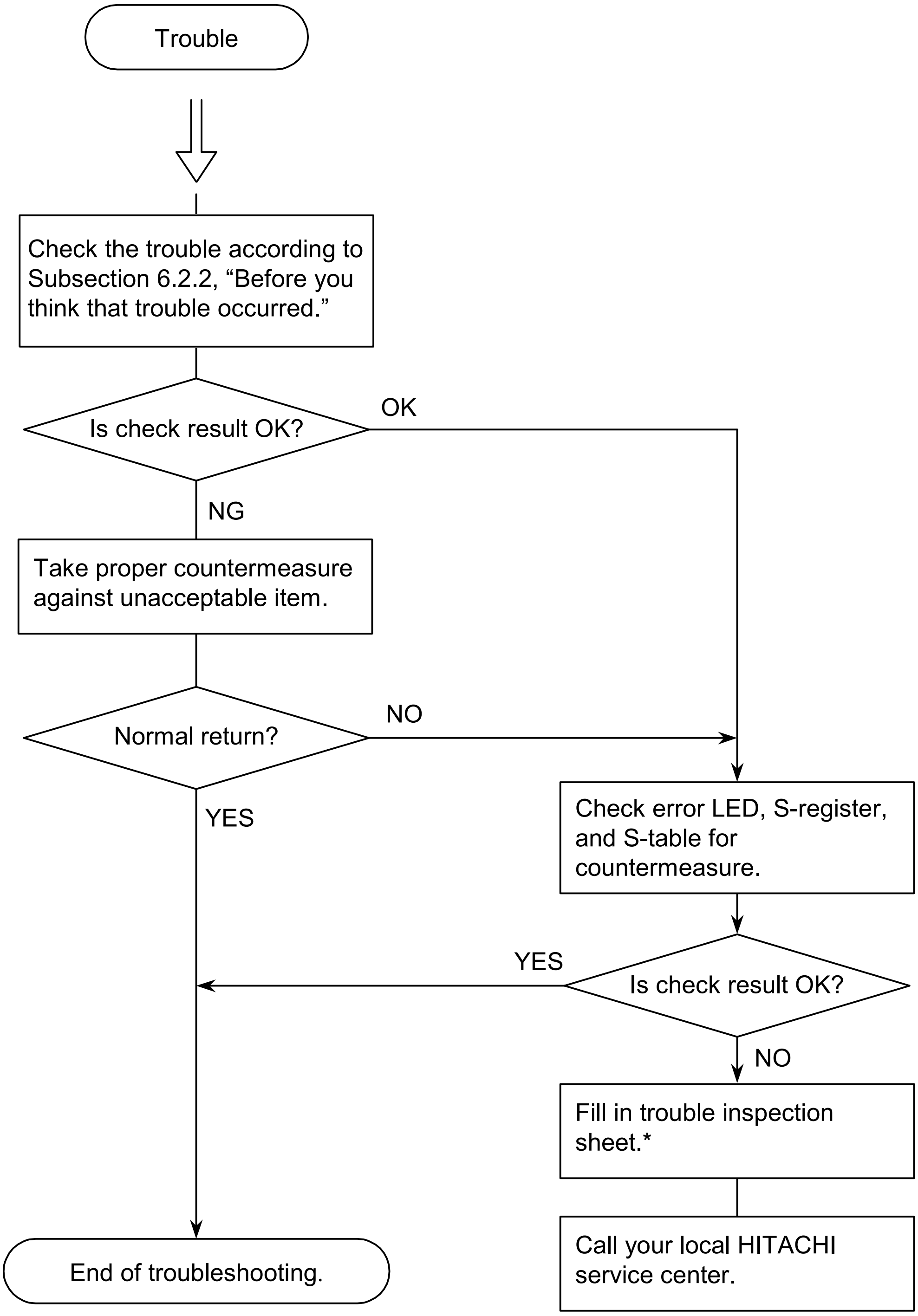
- T/M No.8 (cabling is performed as shown in the figure below)



For J.STATION, set BIT RATE to 8 and ST No. to /01.

6.2 Troubleshooting

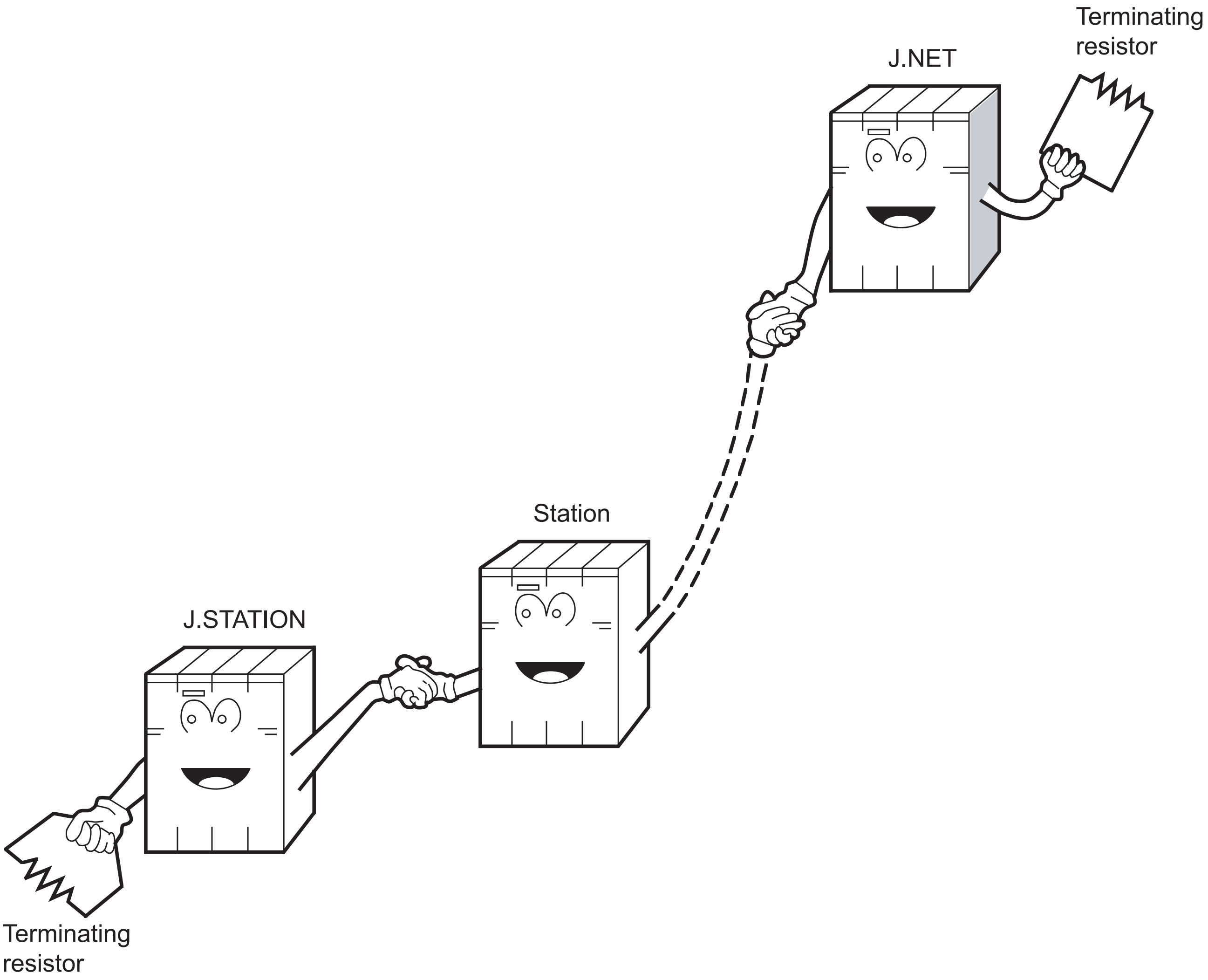
6.2.1 Procedure


Trouble
Check the trouble according to Subsection 6.2.2, “Before you think that trouble occurred.”
Is check result OK?
OK
NG
Take proper countermeasure against unacceptable item.
Normal return?
NO
YES
Check error LED, S-register, and S-table for countermeasure.
YES
Is check result OK?
NO
Fill in trouble inspection sheet.*
Call your local HITACHI service center.
End of troubleshooting.

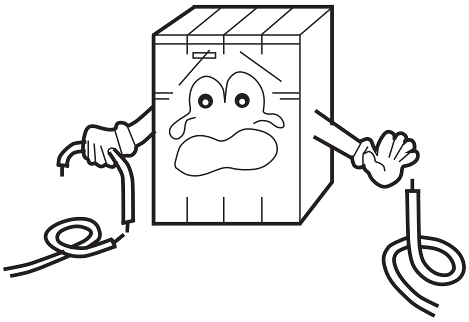
* See A.8, “Trouble Inspection Sheet.”

6.2.2 Before you think that trouble occurred

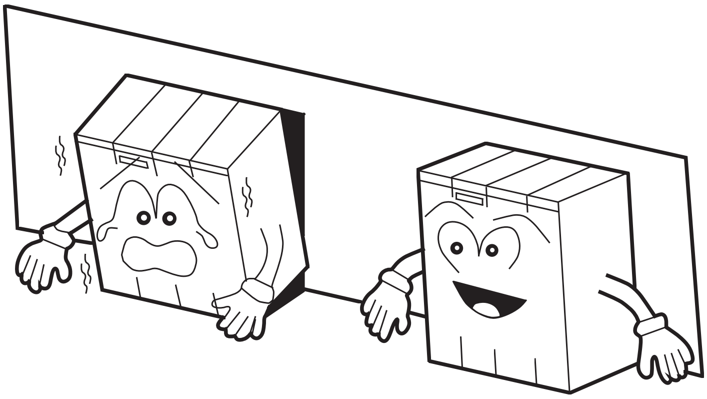
- Are the terminating resistors of the communication cable connected?
  - Terminating resistors (120 ohms) must always be connected to both ends of the communication cable line. (In J.NET and J.STATION, short the TERM terminals with a jumper and thereby connect to the 120-ohm internal resistor.)



- Is the cabling correct?
  - Check cables for disconnection or incorrect connection.

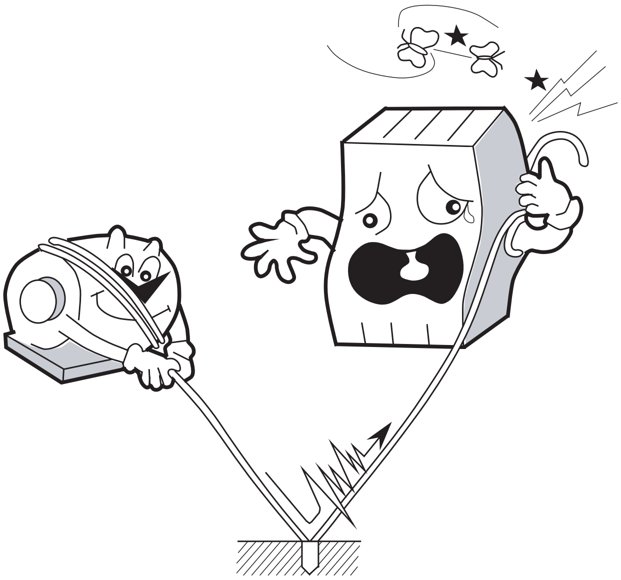


- Is the J.NET module mounted correctly?
  - Check the J.NET module mounting location and check J.NET mounting screws for looseness.

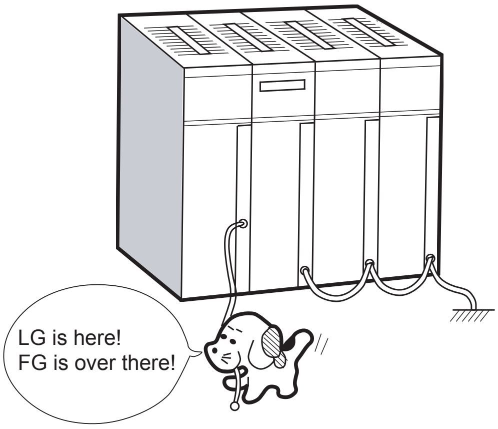


- Is grounding correct?
  - Do not ground the J.NET module in the same place where high-voltage equipment is grounded. They must be grounded in separate places.
  - Apply Class D grounding* or higher.

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



- Are LG and FG separate?
  - Be sure to separate the LG from the FG or vice versa because power noise enters the FG via the LG. Failure to observe this rule may result in an equipment malfunction.
  - Ground the LG to the power supply side.



6.3 Errors and Countermeasures

6.3.1 CPU LED display messages

As shown in the table below, CPU LED display messages depend on the main module and sub-module.

MDL	Message	Description	Countermeasure
Main module	JNTM @. @	The J.NET module (main module) was started normally.	This is not an error.
	EX92 PTY	A parity error was detected when the CPU read data from memory in the J.NET module (main module).	If this message does not disappear even after the CPU key switch has been reset once and then set to the original position again, replace the J.NET module.
	JNM □□□□	An error was detected in the J.NET module (main module) board.	See Subsection 6.3.2.
	JNMN ○○○○	An error was detected in the J.NET module (main module) network.	See Subsection 6.3.4.
	JNMS △△△△	An error was detected in the J.NET module (main module) station.	See Subsection 6.3.4.
Sub-module	JNTS @. @	The J.NET module (sub-module) was started normally.	This is not an error.
	EX93 PTY	A parity error was detected when the CPU read data from memory in the J.NET module (sub-module).	If this message does not disappear even after the CPU key switch has been reset once and then set to the original position again, replace the J.NET module.
	JNS □□□□	An error was detected in the J.NET module (sub-module) board.	See Subsection 6.3.2.
	JNSN ○○○○	An error was detected in the J.NET module (sub-module) network.	See Subsection 6.3.4.
	JNSS △△△△	An error was detected in the J.NET module (sub-module) station.	See Subsection 6.3.4.

- MDL: Module
- @. @: J.NET module version, revision
- □□□□: Any of the hardware error messages explained in Subsection 6.3.2, “Hardware errors”
- ○○○○: Any of the communication error codes explained in Subsection 6.3.4, “Communication errors”
- △△△△: Any of the communication error codes explained in Subsection 6.3.4, “Communication errors”

6.3.2 Hardware errors

When it detects a hardware error, the J.NET module displays an error message in the CPU LED. The module also turns on the error LED (N1, N2) and collects error freeze information.
The module then stops.

Message	Error	User response
BUS	Bus error	The J.NET module may be faulty. Replace it.
ADDR	Address error	
ILLG	Illegal instruction error	
ZERO	Division by zero error	
PRIV	Privilege violation	
WDT	WDT error	
FMAT	Format error	
SINT	Spurious interrupt	
EXCP	Unused interrupt	
PTY	Parity error	
MDSW	MODU. No. switch setting error	Check MODU. No. switch setting.
BRSW	BIT RATE switch setting error	Check BIT RATE switch setting.
ROM1	ROM1 sum error	The J.NET module may be faulty. Replace it.
RAM1	RAM1 compare error	
RAM2	RAM2 compare error	
ROM3	ROM3 sum error	
ROME	ROM3 erase error	
ROMW	ROM3 write error	
WOVR	ROM rewrite count exceeded	The ROM rewrite count exceeded 50,000. Replace the module.
PRME	Parameter error	Set the parameter again.

6.3.3 Handler-detected error codes

The errors detected by the handler started from a user program (e.g., C-mode program, ladder program) are not displayed in the CPU LED. When it detects an error, the handler sets the error flag and enters the error code in the S-table.

Type	Error code	Error	User response
Data transmission	F104	Transmission data length error	Review and correct the user program.
	F105	Station number error	Check SVPT registration and then set the station number again.
	F120	Module failure	Check the error LED of the J.NET module.
	F130	Start of transmission of other data during transmission of the current data	Review and correct the user program so that it does not start transmission of other data during transmission of the current data.
Data reception	F204	Transmission data length error	Review and correct the user program.
	F205	Station number error	Check SVPT registration and then set the station number again.
	F220	Module failure	Check the error LED of the J.NET module.
Command transmission	F304	Transmission data length error	Review and correct the user program.
	F305	Station number error	Check SVPT registration and then set the station number again.
	F320	Module failure	Check the error LED of the J.NET module.
	F330	Start of transmission of other data during transmission of the current data	Review and correct the user program so that it does not start transmission of another data during transmission of one data.
	F340	Service unsupported	Review the user program and correct the service code.
Response reception	F404	Transmission data length error	Review and correct the user program.
	F405	Station number error	Check SVPT registration and then set the station number again.
	F420	Module failure	Check the error LED of the J.NET module.
	F440	Service unsupported	Review the user program and correct the service code.

6.3.4 Communication errors

(1) Return code errors

When it detects an error on a communication line, the J.NET module stores the error information in the ALM field of the S-register, sets the NET status error flag, and writes the error code to the S-table.

The module also displays the error on the CPU LED.

Error code	Explanation	User response
7110	An undefined service was instructed.	• If this error recurs even after the CPU key switch has been reset once and then set to the original position again, restart the J.NET module. • If this error still recurs, replace the J.NET module.
7120	The data length is incorrect.	
7130	The packet configuration is incorrect.	
7061	The fetching of input data was not completed at the station.	• This condition is not an error condition. • Normal processing resumes upon completion of the input data fetch.
2010	An error was detected during CRC check.	• Check whether the network line is normal. • Check whether SVPT setting matches station setting. • If this error still recurs, replace the J.NET module.
2020	The station number is from 128 to 254 or the received station number is incorrect.	
2030	An undefined service was specified.	
2040	The I-frame length is greater than or equal to 137 bytes or the UI-frame length is greater than or equal to 134 bytes.	
2041	No I-frame exists in the I-response.	
2042	An I-frame exists in the monitoring frame.	
2050	Data link procedure error	
2060	A timeout was detected (no response was made from the slave station within the specified time).	• Power on the station again. • If this error still recurs, check whether the switches of the J.NET module and station are set correctly. • If this error still recurs even after the switches have been set correctly, replace the station.
2061	The error could not be recovered by retry.	
2070	No frame could be transmitted to the line or an error was detected during frame reception.	• Check the network line connection and terminating resistor connection. • Check whether SVPT setting matches station setting. • If this error recurs even after the CPU key switch has been reset once and then set to the original position again, restart the J.NET module. • If this error still persists, replace the J.NET module.
2080	Other errors	• If this error recurs even after the CPU key switch has been reset once and then set to the original position again, restart the J.NET module. • If this error still recurs, replace the J.NET module.

(2) Result and status errors

If an error occurs in the station connected to the J.NET module, the J.NET module stores the error information in the ALM field of the S-register, sets the NET status error flag, and writes the error code to the S-table.

The module also displays the error in CPU LED.

Error code	Explanation	User response
9001	The station is inactive.	• Power on the station again and then reset the CPU key switch. • If this error still recurs, replace the station.
9002	The station is in an abnormal state. (An error was detected in the station.)	
9003	The station is inactive and also in an abnormal state.	
8020	The initialization instruction was rejected.	• SVPT setting does not match station setting. Set SVPT again to match station setting. • If this error still recurs, replace the station.
8081	When the AUTO mode is specified, the number of registered transfer bytes does not match the response I/O size from the station.	
8082	When a slot is specified, the number of registered transfer bytes does not match the response I/O size from the station.	

(3) Polling errors

If an error occurs in the PUT/GET service request from the station connected to the J.NET module when the station can be polled, the J.NET module stores the error information in the ALM field of the S-register, sets the NET status error flag, and writes the error code to the S-table.

Error code	Explanation	User response
A020	Address data is insufficient or the associated symbol does not exist.	Review the PUT/GET service request from the station.
A022	The address field is numeric.	
A021	Address field count error	
A040	Odd address	

(4) J.STATION 232C errors

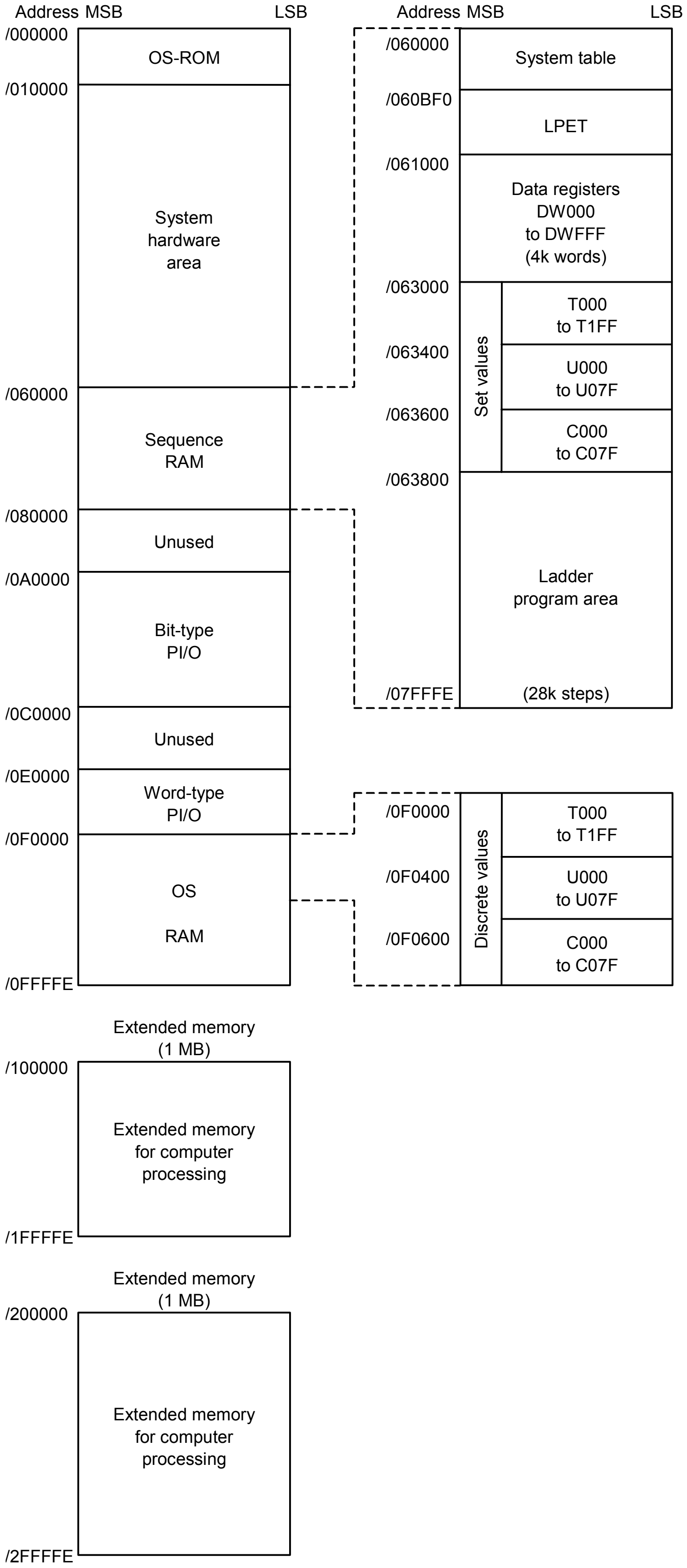
When it detects a J.STATION RS-232C transmission or reception error, the J.NET module stores the error information in the ALM field of the S-register, sets the NET status error flag, and writes the error code to the S-table.

However, the module does not display the error in CPU LED; check the error information with the error information display function stored in J.NET SYSTEM.

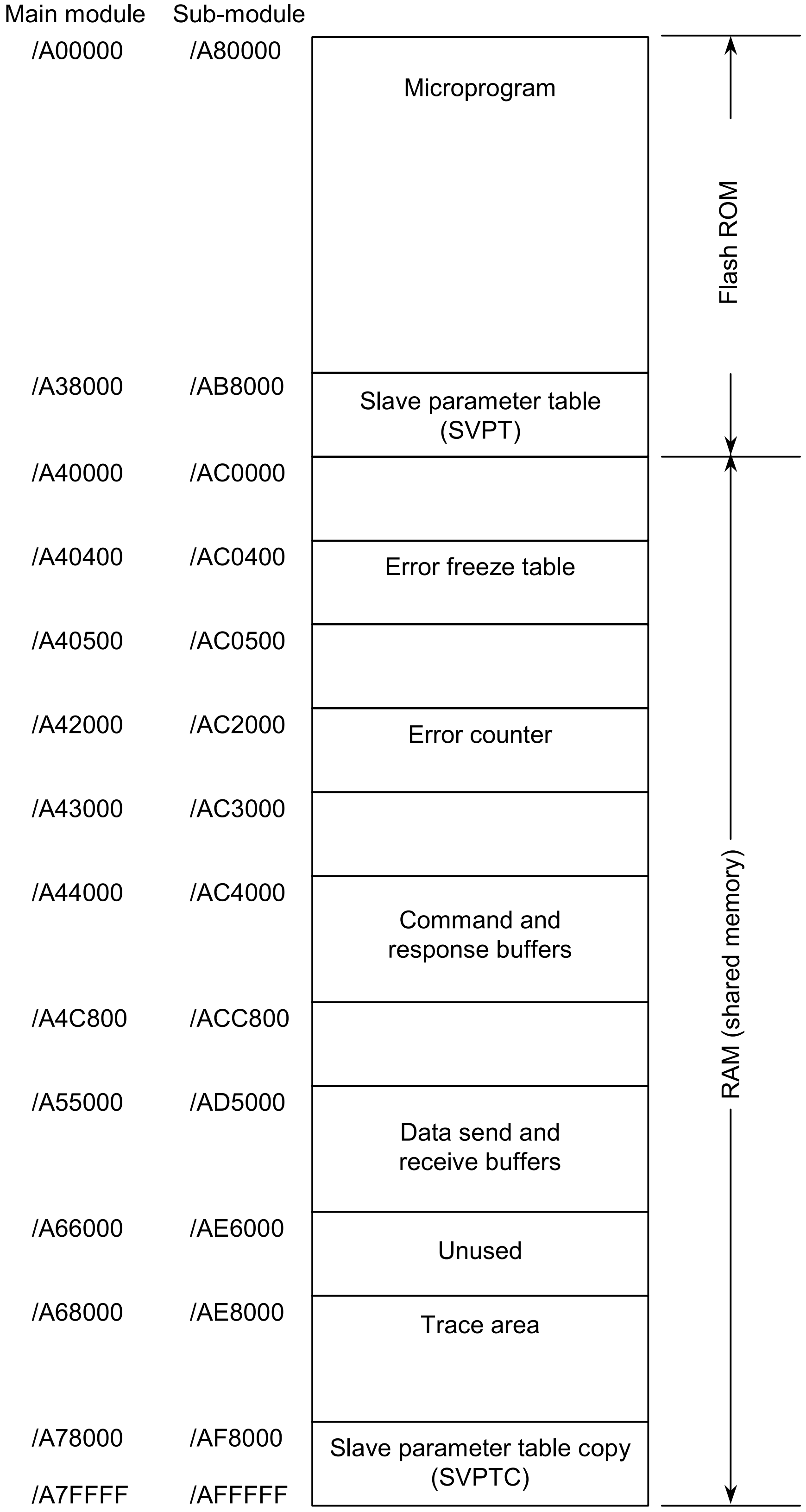
Type	Error code	Explanation
Data transmission	B081	Transmission disabled due to transmission of requested data This data cannot be transmitted, because the previously requested transmission data is currently being transmitted (or during suspension of data transmission).
	B082	Transmission disabled due to absence of transmission request This data cannot be transmitted, because CS input does not request transmission or DR (Data Set Ready) input is “not ready” in DR check specification.
	B083	Transmission request timeout Transmission was canceled according to the transmission suspension code during transmission and data was not transmitted according to the transmission restart code within the transmission suspension monitoring time.
	B084	Transmission request timeout Transmission was suspended because input CS became “no transmission request” during transmission, and transmission was not restarted, because input CS did not become “transmission request issued” within the transmission suspension monitoring time.
	B085	Dataset ready timeout Transmission was suspended because DR input became “not ready” in DR check specification during transmission, and transmission was not restarted, because DR input did not become “ready” within the transmission suspension monitoring time.
Data reception	C080	Parity error. A parity error was detected in received data.
	C081	Overrun error. An overrun error was detected in received data.
	C082	Framing error. A framing error was detected in received data.
	C083	Reception timeout. Some data could not be received within the specified reception monitoring time.
	C084	ASCII conversion error. With ASCII conversion specified, data other than 0 to 9 or A to F was received.
	C085	End code error. With ASCII conversion specified, data other than 0 to 9, A to F, or end code was received.
	C086	BCC error. The received BCC is incorrect.
	C087	Reception carrier detection timeout Reception was suspended because CD (Carrier Detect) input became “no carrier”, and reception was not restarted, because CD input did not become “carrier detected” within the reception monitoring time.
	C088	Reception data over. A total of 531 bytes or more were received.
	C089	Data was received when the receive buffer was full.

APPENDIX

A.1 CPU Memory Map



A.2 J.NET Module Memory Map



Main module	Sub-module	
/A00000	/A80000	Microprogram
/A38000	/AB8000	Slave parameter table (SVPT)
/A40000	/AC0000	
/A40400	/AC0400	Error freeze table
/A40500	/AC0500	
/A42000	/AC2000	Error counter
/A43000	/AC3000	
/A44000	/AC4000	Command and response buffers
/A4C800	/ACC800	
/A55000	/AD5000	Data send and receive buffers
/A66000	/AE6000	Unused
/A68000	/AE8000	Trace area
/A78000	/AF8000	Slave parameter table copy (SVPTC)
/A7FFFF	/AFFFFF	

A.3 Error Freeze

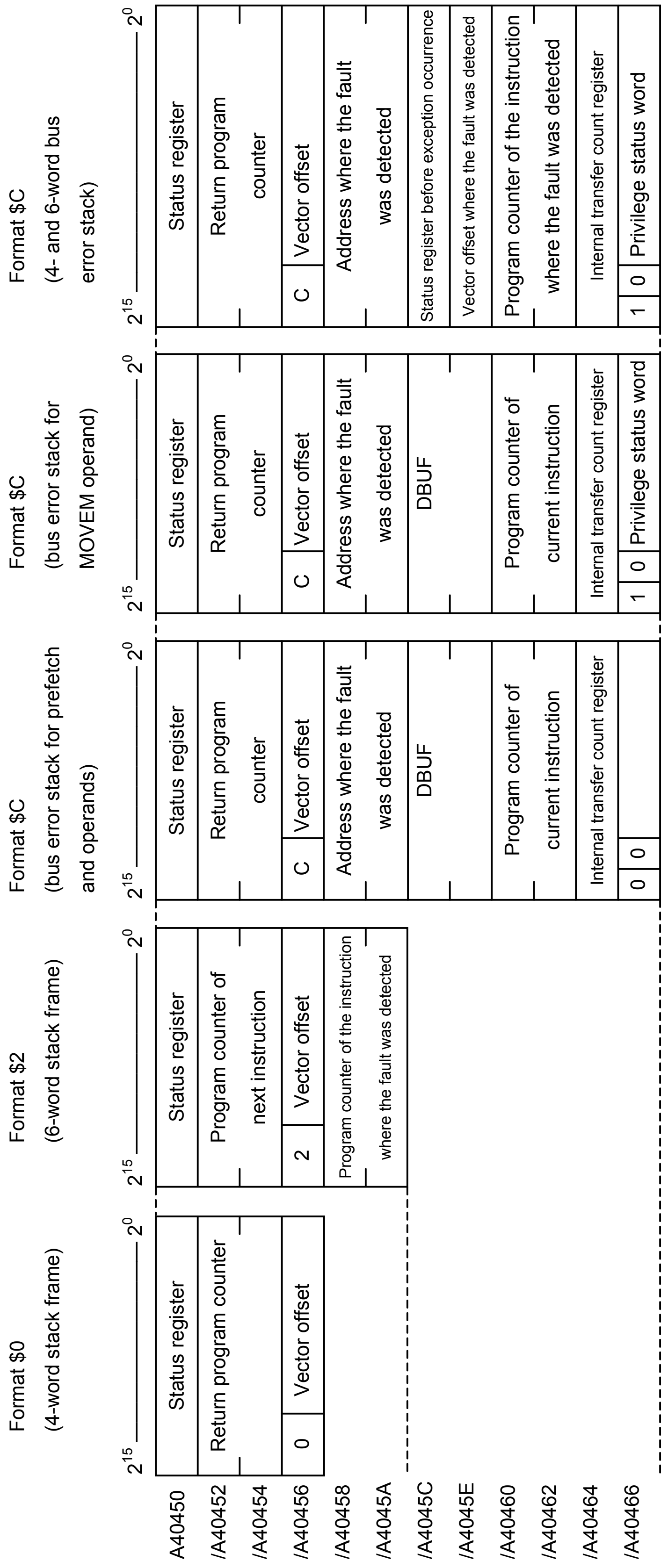
When it detects a hardware error, the J.NET module lights the error LED (NET1, NET2), stores error freeze information, and stops.

Main module	Sub-module	2^{31}—— 2^{16}	2^{15}—— 2^{0}
/A40400	/AC0400	Error code	–
/A40404	/AC0404	Time from reset (ms)	
		–	
/A40410	/AC0410	D0 register	
/A40414	/AC0414	D1 register	
/A40418	/AC0418	D2 register	
/A4041C	/AC041C	D3 register	
/A40420	/AC0420	D4 register	
/A40424	/AC0424	D5 register	
/A40428	/AC0428	D6 register	
/A4042C	/AC042C	D7 register	
/A40430	/AC0430	A0 register	
/A40434	/AC0434	A1 register	
/A40438	/AC0438	A2 register	
/A4043C	/AC043C	A3 register	
/A40440	/AC0440	A4 register	
/A40444	/AC0444	A5 register	
/A40448	/AC0448	A6 register	
/A4044C	/AC044C	A7 register	
/A40450	/AC0450	Stack frame (4 words, 6 words, bus error)	
/A404FC	/AC04FC		

Note: The stack frame is explained on the next page.

No.	Code	Error	Display
1	0010H	Bus error	BUS
2	0011H	Address error	ADDR
3	0012H	Illegal instruction error	ILLG
4	0013H	Division by zero error	ZERO
5	0014H	Privilege violation	PRIV
6	0015H	WDT error	WDT
7	0016H	Format error	FMAT
8	0017H	Spurious interrupt	SINT
9	0018H	Unsupported interrupt (e.g., CHK, TRAPV, L1010)	EXSP
10	0019H	Parity error	PTY
11	001AH	Power failure forecast	GR
12	0100H	MODU. No. switch setting error	MDSW
13	0101H	BIT RATE switch setting error	BRSW
14	0102H	ROM sum error	ROM1
15	0103H	RAM1 compare error	RAM1
16	0105H	RAM2 compare error	RAM2
17	0107H	DMA transfer error (J.NET module error)	
18	0108H		
19	0109H		
20	010AH		
21	010BH	ROM sum error	ROM3
22	010CH	ROM write error 2	
23	010DH		
24	010EH		
25	010FH		
26	0110H	Parameter rewrite count exceeded the limit	WOVR

Stack frames in the error freeze information table are explained below.



	Format $0 (4-word stack frame) 2^{15} — 2^0	Format $2 (6-word stack frame) 2^{15} — 2^0	Format $C (bus error stack for prefetch and operands) 2^{15} — 2^0	Format $C (bus error stack for MOVEM operand) 2^{15} — 2^0	Format $C (4- and 6-word bus error stack) 2^{15} — 2^0
A40450	Status register	Status register	Status register	Status register	Status register
/A40452	Return program counter	Program counter of next instruction	Return program counter	Return program counter	Return program counter
/A40454					
/A40456	0 | Vector offset	2 | Vector offset	C | Vector offset	C | Vector offset	C | Vector offset
/A40458		Program counter of the instruction where the fault was detected	Address where the fault was detected	Address where the fault was detected	Address where the fault was detected
/A4045A					
/A4045C			DBUF	DBUF	Status register before exception occurrence
/A4045E					Vector offset where the fault was detected
/A40460			Program counter of current instruction	Program counter of current instruction	Program counter of the instruction where the fault was detected
/A40462					
/A40464			Internal transfer count register	Internal transfer count register	Internal transfer count register
/A40466			0 | 0 |	1 | 0 | Privilege status word	1 | 0 | Privilege status word

A.4 Error Counter

The error counter counts the communication errors that occur between the J.NET module (master) and station (slave). This counter is initialized when it is reset.

[N1]	[N2]	
/A42000	/A42400	(At broadcast)
/A42020	/A42420	Station ID=01
/A42040	/A42440	Station ID=02
/A42060	/A42460	Station ID=03
/A42080	/A42480	Station ID=04
/A420A0	/A424A0	Station ID=05
/A420C0	/A424C0	Station ID=06
/A420E0	/A424E0	Station ID=07
/A42100	/A42500	Station ID=08
/A42120	/A42520	Station ID=09
/A42140	/A42540	Station ID=0A
/A42160	/A42560	Station ID=0B
/A42180	/A42580	Station ID=0C
/A421A0	/A425A0	Station ID=0D
/A421C0	/A425C0	Station ID=0E
/A421E0	/A425E0	Station ID=0F
/A42200	/A42600	Station ID=10
/A42220	/A42620	Station ID=11
/A42240	/A42640	Station ID=12
/A42260	/A42660	Station ID=13
/A42280	/A42680	Station ID=14
/A422A0	/A426A0	Station ID=15
/A422C0	/A426C0	Station ID=16
/A422E0	/A426E0	Station ID=17
/A42300	/A42700	Station ID=18
/A42320	/A42720	Station ID=19
/A42340	/A42740	Station ID=1A
/A42360	/A42760	Station ID=1B
/A42380	/A42780	Station ID=1C
/A423A0	/A427A0	Station ID=1D
/A423C0	/A427C0	Station ID=1E
/A423E0	/A427E0	Station ID=1F

	2^7 — 2^0	
+00	Transmitter underrun	(TXUN)
02	CTS dissipation	(TXCT)
04	Frame length violation	(RXLG)
06	Non-octet array frame	(RXNO)
08	Abort sequence	(RXAB)
0A	CRC error	(RXCR)
0C	Overrun	(RXOV)
0E	CD dissipation	(RXCD)
10	Timeout	(RXTO)
12 to +1E	Unused (14 bytes)	

A.5 Command and Response Buffers

- Command buffer

[N1]	[N2]	
/A44110	/A46310	Station ID=01
/A44220	/A46420	Station ID=02
/A44330	/A46530	Station ID=03
/A44440	/A46640	Station ID=04
/A44550	/A46750	Station ID=05
/A44660	/A46860	Station ID=06
/A44770	/A46970	Station ID=07
/A44880	/A46A80	Station ID=08
/A44990	/A46B90	Station ID=09
/A44AA0	/A46CA0	Station ID=0A
/A44BB0	/A46DB0	Station ID=0B
/A44CC0	/A46EC0	Station ID=0C
/A44DD0	/A46FD0	Station ID=0D
/A44EE0	/A470E0	Station ID=0E
/A44FF0	/A471F0	Station ID=0F
/A45100	/A47300	Station ID=10
/A45210	/A47410	Station ID=11
/A45320	/A47520	Station ID=12
/A45430	/A47630	Station ID=13
/A45540	/A47740	Station ID=14
/A45650	/A47850	Station ID=15
/A45760	/A47960	Station ID=16
/A45870	/A47A70	Station ID=17
/A45980	/A47B80	Station ID=18
/A45A90	/A47C90	Station ID=19
/A45BA0	/A47DA0	Station ID=1A
/A45CB0	/A47EB0	Station ID=1B
/A45DC0	/A47FC0	Station ID=1C
/A45ED0	/A480D0	Station ID=1D
/A45FE0	/A481E0	Station ID=1E
/A460F0	/A482F0	Station ID=1F

	2^7 — 2^0
+000	CFLAG
1	–
2	Return code (H)
3	(L)
4	Station No.
5	
6	
7	
8	
9	Service code
A	Data length (H)
B	(L)
C	Data (up to 250 bytes)
	≈
/105	
/106	Unused (10 bytes)
	≈
/10F	

- Response buffer

[N1]	[N2]	
/A48510	/A4A710	Station ID=01
/A48620	/A4A820	Station ID=02
/A48730	/A4A930	Station ID=03
/A48840	/A4AA40	Station ID=04
/A48950	/A4AB50	Station ID=05
/A48A60	/A4AC60	Station ID=06
/A48B70	/A4AD70	Station ID=07
/A48C80	/A4AE80	Station ID=08
/A48D90	/A4AF90	Station ID=09
/A48EA0	/A4B0A0	Station ID=0A
/A48FB0	/A4B1B0	Station ID=0B
/A490C0	/A4B2C0	Station ID=0C
/A491D0	/A4B3D0	Station ID=0D
/A492E0	/A4B4E0	Station ID=0E
/A493F0	/A4B5F0	Station ID=0F
/A49500	/A4B700	Station ID=10
/A49610	/A4B810	Station ID=11
/A49720	/A4B920	Station ID=12
/A49830	/A4BA30	Station ID=13
/A49940	/A4BB40	Station ID=14
/A49A50	/A4BC50	Station ID=15
/A49B60	/A4BD60	Station ID=16
/A49C70	/A4BE70	Station ID=17
/A49D80	/A4BF80	Station ID=18
/A49E90	/A4C090	Station ID=19
/A49FA0	/A4C1A0	Station ID=1A
/A4A0B0	/A4C2B0	Station ID=1B
/A4A1C0	/A4C3C0	Station ID=1C
/A4A2D0	/A4C4D0	Station ID=1D
/A4A3E0	/A4C5E0	Station ID=1E
/A4A4F0	/A4C6F0	Station ID=1F

	2^7 —— 2^0
+000	CFLAG
1	–
2	Return code (H)
3	(L)
4	Station No.
5	
6	
7	
8	
9	Service code
A	Data length (H)
B	(L)
C	Data (up to 250 bytes)
/105	
/106	Unused (10 bytes)
/10F	

A.6 Data Send and Receive Buffers

- Data send buffer

[N1]	[N2]	
/A55220	/A59620	Station ID=01
/A55440	/A59840	Station ID=02
/A55660	/A59A60	Station ID=03
/A55880	/A59C80	Station ID=04
/A55AA0	/A59EA0	Station ID=05
/A55CC0	/A5A0C0	Station ID=06
/A55EE0	/A5A2E0	Station ID=07
/A56100	/A5A500	Station ID=08
/A56320	/A5A720	Station ID=09
/A56540	/A5A940	Station ID=0A
/A56760	/A5AB60	Station ID=0B
/A56980	/A5AD80	Station ID=0C
/A56BA0	/A5AFA0	Station ID=0D
/A56DC0	/A5B1C0	Station ID=0E
/A56FE0	/A5B3E0	Station ID=0F
/A57200	/A5B600	Station ID=10
/A57420	/A5B820	Station ID=11
/A57640	/A5BA40	Station ID=12
/A57860	/A5BC60	Station ID=13
/A57A80	/A5BE80	Station ID=14
/A57CA0	/A5C0A0	Station ID=15
/A57EC0	/A5C2C0	Station ID=16
/A580E0	/A5C4E0	Station ID=17
/A58300	/A5C700	Station ID=18
/A58520	/A5C920	Station ID=19
/A58740	/A5CB40	Station ID=1A
/A58960	/A5CD60	Station ID=1B
/A58B80	/A5CF80	Station ID=1C
/A58DA0	/A5D1A0	Station ID=1D
/A58FC0	/A5D3C0	Station ID=1E
/A591E0	/A5D5E0	Station ID=1F

	2^7 — 2^0
+000	Transmission data length (H)
1	(L)
2	Control flag (H)
3	(L)
4	Error code (H)
5	(L)
6	Unused (4 bytes)
7	
8	
9	
A	Data (up to 512 bytes)
≈	
/209	
/20A	Unused (22 bytes)
≈	
/21F	

- Data receive buffer

[N1]	[N2]	
/A5DA20	/A61E20	Station ID=01
/A5DC40	/A62040	Station ID=02
/A5DE60	/A62260	Station ID=03
/A5E080	/A62480	Station ID=04
/A5E2A0	/A626A0	Station ID=05
/A5E4C0	/A628C0	Station ID=06
/A5E6E0	/A62AE0	Station ID=07
/A5E900	/A62D00	Station ID=08
/A5EB20	/A62F20	Station ID=09
/A5ED40	/A63140	Station ID=0A
/A5EF60	/A63360	Station ID=0B
/A5F180	/A63580	Station ID=0C
/A5F3A0	/A637A0	Station ID=0D
/A5F5C0	/A639C0	Station ID=0E
/A5F7E0	/A63BE0	Station ID=0F
/A5FA00	/A63E00	Station ID=10
/A5FC20	/A64020	Station ID=11
/A5FE40	/A64240	Station ID=12
/A60060	/A64460	Station ID=13
/A60280	/A64680	Station ID=14
/A604A0	/A648A0	Station ID=15
/A606C0	/A64AC0	Station ID=16
/A608E0	/A64CE0	Station ID=17
/A60B00	/A64F00	Station ID=18
/A60D20	/A65120	Station ID=19
/A60F40	/A65340	Station ID=1A
/A61160	/A65560	Station ID=1B
/A61380	/A65780	Station ID=1C
/A615A0	/A659A0	Station ID=1D
/A617C0	/A65BC0	Station ID=1E
/A619E0	/A65DE0	Station ID=1F

	2^7 ——	—— 2^0
+000	Transmission data length	(H)
1		(L)
2	Control flag	(H)
3		(L)
4	Error code	(H)
5		(L)
6		
7	Unused (4 bytes)	
8		
9		
A	Data	
	(up to 512 bytes)	
	≈	≈
/209		
/20A	Unused (22 bytes)	
	≈	≈
/21F		

A.7 Trace

The J.NET module collects trace data for each network (N1, N2).
Trace is started in the error stop mode (error occurrence, trace stop) when the CPU is reset or power is recovered. Trace data is recorded in transmission or reception units of each service.
The trace data structure is shown below.

Trace data structure

+00	Type
+02	Return code
+04	Unused
+06	
+08	20-byte data from beginning of
⋮	send or receive buffer in
+1A	physical layer
+1C	Timer counter value
+1E	(Lapse time from CPU reset)

- Type

1030	Normal initialization service transmission
1010	Normal I/O service transmission
2030	Normal initialization service reception
2010	Normal I/O service reception
3030	Initialization service transmission error
3010	I/O service transmission error
4030	Initialization service reception error
4010	I/O service reception error

- Return code

  See Subsection 6.3.4, “Communication errors.”

- 20-byte data from beginning of send or receive buffer in physical layer

08	A: Station No.
09	C: Control flag
0A	DL-SC: DL layer service code
0B	DL-len: DL layer length
0C	7L-hd: 7L layer header
0D	7L-sc: 7L layer service code
0E	7U-sc: 7U layer service code
0F	len(L): Low-order bytes of length
10	len(H): High-order bytes of length
11	data[0]: Data
⋮	⋮
1B	data[9]: Data

- Timer counter value (1-ms unit)

  The timer count value, however, is updated in 4-ms units.

Trace area

Main module	Sub-module		
/A68000	/AE8000	Trace pointer	N1
/A68002	/AE8002	Forced traced stop	
/A68004	/AE8004	Trace stop code	
/A68006	/AE8006	Trace mode	
/A68008 ⋮ /A6800E	/AE8008 ⋮ /AE800E	Unused	
/A68010 ⋮ /A6802E	/AE8010 ⋮ /AE802E	Trace data #0	
/A68030 ⋮ /A6DFCE	/AE8030 ⋮ /AEDFCE	⋮	
/A6DFD0 ⋮ /A6DFEE	/AEDFD0 ⋮ /AEDFEE	Trace data #2FE	
/A6DFF0 ⋮ /A6DFFE	/AEDFF0 ⋮ /AEDFFE	Unused	
/A6E000	/AEE000	Trace pointer	N2
/A6E002	/AEE002	Forced trace stop	
/A6E004	/AEE004	Trace stop code	
/A6E006	/AEE006	Trace mode	
/A6E008 ⋮ /A6E00E	/AEE008 ⋮ /AEE00E	Unused	
/A6E010 ⋮ /A6E02E	/AEE010 ⋮ /AEE02E	Trace data #0	
/A6E030 ⋮ /A73FCE	/AEE030 ⋮ /AF3FCE	⋮	
/A73FD0 ⋮ /A73FEE	/AF3FD0 ⋮ /AF3FEE	Trace data #2FE	
/A73FF0 ⋮ /A73FFE	/AF3FF0 ⋮ /AF3FFE	Unused	

- Trace pointer
  The trace pointer points to a value of from 0 to 2FE. The reference address of trace data from the trace pointer is calculated by the expression “reference address = /A68010 + (trace pointer * /20)”. /AE8010 of the sub-module becomes the reference address.

- Forced trace stop
  0: Forced trace stop
  Other values: Stop release

- Trace stop code
  Set a trace data type as the trace stop code.

- Trace mode
  0: Trace stop
  1: Endless trace
  2: Stop on error occurrence
  (The trace mode when an error occurs is 0.)

- Trace data
  The trace data area is of ring structure. Data next to #2FE is #0.

A.8 Trouble Inspection Sheet

◆ Trouble inspection sheet

Your company name			Person in charge	
Data and time of occurrence				(year / month / day / hour / minute)
Where to make contact	Address			
	Telephone			
	FAX			
	E-mail			
Model of defective module			CPU model	
OS Ver. Rev.		Program name:		Ver. Rev.
Support program		Program name:		Ver. Rev.
Symptom of defect				
Connection load	Type			
	Model			
	Wiring state			
System configuration and switch setting				
Space for correspondence				

SUPPLEMENTARY

Supplementary: Replacing or Adding on the Module

- What you should get in preparation

① Personal computer (with Hitachi's S10 J.NET System installed in it)

② RS-232C cable (or 10BASE-T cable if the communication module used is an ET.NET module)

③ New or add-on J.NET module (LQE040)

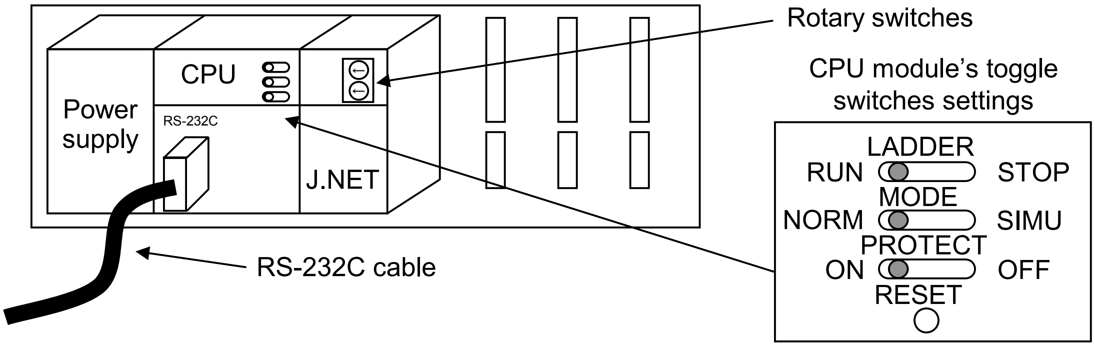
④ Copies of the parameter values for the module to be replaced. (These copies are prepared for use in cases where the parameters are not accessible for some reason.)

⑤ The above-mentioned ET.NET module is an optional module and, if it is mounted in place, may be selected as the type of communication module to be used. For more information, refer to Section 1.2, "Mounting Optional Modules," and Section 3.1, "Names and Functions of Each Part," in the S10mini HARDWARE MANUAL, OPTION ET.NET (manual number SME-1-103).

- Replacement procedure

① Write down, on a piece of paper, the current settings of the rotary switches that are, as shown below, accessible at the front side of the J.NET module to be replaced.

② Write down also the current settings of three switches, labeled LADDER (toggle switch), MODE (toggle switch), and PROTECT (toggle switch), respectively, that are, as shown below, accessible at the front side of the CPU module.



③ Connect the personal computer and the CPU module together with the RS-232C cable.

④ Start the S10 J.NET System and save the set values of all the existing parameters by using its F/D function. (If the existing parameters are not accessible for some reason, use the copies of their set values [item ④] that were obtained in preparation.)

⑤ Set the CPU module's LADDER switch in STOP position and turn off the power supply of the controller unit.

⑥ Remove the connecting cables from the J.NET module to be replaced.

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

⑧ Turn on the power supply of the controller unit and send to the new J.NET module the set parameter values that you saved in Step ④ using the F/D function.

⑨ By using the F/D function, compare the set parameter values before and after you sent. This comparison may reveal a discrepancy for the following memory areas:
• /A3BFFE for the main J.NET module mounted
• /ABBFFE for the J.NET sub-module mounted
However, if no discrepancies are found for any other area, the set parameter values (system information) for the new J.NET module may be considered as being identical between the saved file and memory.

⑩ Reset the CPU module by pressing the RESET switch at its front.

⑪ Turn off the power supply of the controller unit.

⑫ Remove the RS-232C cable from both the personal computer and CPU module, which were connected together in Step ③.

⑬ Connect to the new J.NET module the connecting cables that you removed in Step ⑥.

⑭ Set the CPU module's LADDER, MODE, and PROTECT switches in the same way as you wrote down in Step ②.

⑮ Turn on the power supply of the controller unit and check that the new J.NET module is running normally.

● Add-on procedure

① Write down, on a piece of paper, the current settings of three switches, labeled LADDER (toggle switch), MODE (toggle switch), and PROTECT (toggle switch), respectively, that are accessible at the front side of the CPU module, the one that is installed in the controller unit in which you are adding on a J.NET module.

② Ensure that your application system has been shut down. Then, set the CPU module's LADDER switch in STOP position and turn off the power supply of the controller unit.

③ Mount the add-on J.NET module in place according to the instructions given under "1.2 Mounting Optional Modules."

④ Set the add-on J.NET module's rotary switches in such a way that a new module No. setting, which must be a sub-module No. setting, will not duplicate with the current rotary switch settings of the existing main J.NET module.

⑤ Connect the personal computer and the CPU module together with the RS-232C cable. Then, turn on the power supply of the controller unit and set parameters for the add-on J.NET module by using the S10 J.NET System.

⑥ Reset the CPU module by pressing the RESET switch at its front.

⑦ Turn off the power supply of the controller unit and connect the connecting cables to the add-on J.NET module.

⑧ Set the CPU module's LADDER, MODE, and PROTECT switches in the same way as you wrote down in Step ①.

⑨ Remove the RS-232C cable from both the personal computer and CPU module, which were connected together in Step ⑤.

⑩ Turn on the power supply of the controller unit and check that the add-on J.NET module is running normally.