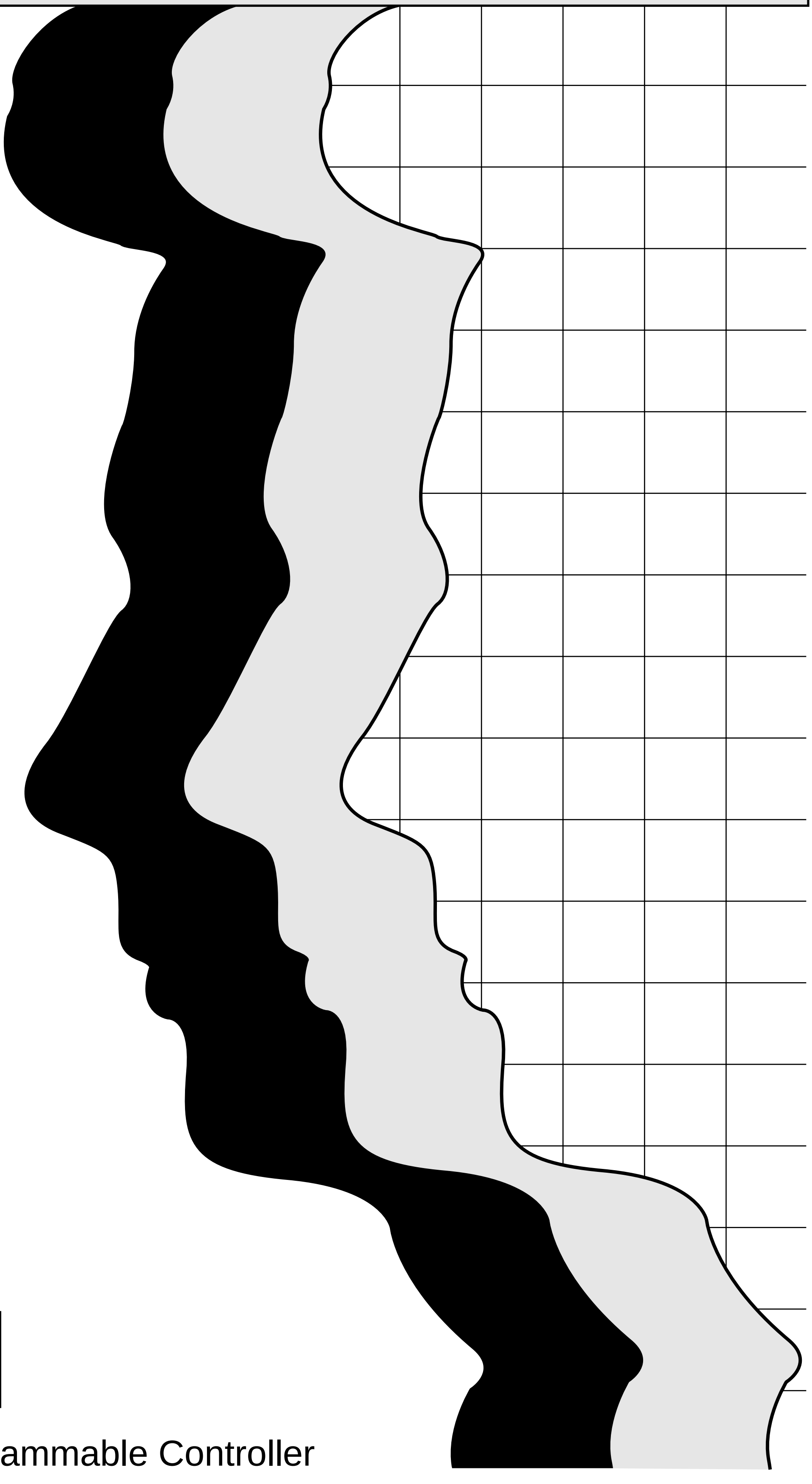
MITSUBISHI

For IBM-PC/AT Computers Model A70BDE-J71LP23GE
MELSECNET/10 Interface Board

User's Manual



MELSEC

Mitsubishi Programmable Controller

● SAFETY PRECAUTIONS ●

(Please be sure to read these precautions before use.)

When using the MELSEC-QnA/A series sequencer, familiarize yourself with the contents of the manual provided with each product and the related manuals referred to in the manuals provided, and take every care to ensure safety and adhere to correct operating procedures.
In this manual, there are two levels of safety precautions: "danger" and "caution."

◇! Danger	This is indicated when product mishandling can lead to a dangerous situation in which death or serious injury may result.
⚠ Caution	This is indicated when product mishandling can lead to a dangerous situation in which moderate damage or minor injury may result or in which only physical damage may occur.

Depending on the circumstances, even a situation in which the user is given warning by the ⚠Caution indication can lead to serious consequences. Under whichever heading the cautionary note falls under, the warning given is important and the precautions described must be adhered to.

Ensure that the manual provided with the product is kept in a safe place for future reference and that it is handed over to the end user.

◇! Danger

[Safety Precautions for Design]

- Install a safety circuit outside the sequencer so that the entire system will operate on the side of safety when the personal computer has failed.
  Accidents may be caused by erroneous outputs and incorrect operation.
  (1) Configure all circuits in the following categories outside the sequencer: emergency stop circuits, protection circuits, interlock circuits for forward/reverse rotation and other reciprocal operations, interlock circuits for positioning upper/lower limits and other mechanisms for safeguarding the machine from damage.
  (2) When a data link malfunction cuts off the station where the board (A70BDE-J71LP23GE) is installed from the data link, the output data from the station where the board is installed which was saved in the other stations will remain unchanged from immediately before the occurrence of the data link malfunction, and the data will be held until the data link of the station where the board is installed is restored (returned).
  Provide a mechanism by which the stations connected to the data link system will monitor the data link status, and take the necessary action in the event of data link malfunctions.
- If a coaxial cable is disconnected, this may destabilize the line, and a data link communication error may occur in multiple stations. Make sure to create an interlock circuit in the sequence program so that the system will operate safely even if the above error occurs. Failure to do so may result in a serous accident due to faulty output or malfunctions.
- Start up the data link system in the following sequence.
  (1) Start up the lower system (third tier) and then start up the higher system (second tier). Alternatively, start up the lower and higher systems simultaneously.
  (2) In the same tier, start up the sub-stations (normal or remote stations where the boards are installed), and then start up the control station.

⚠ Caution

[Safety Precautions for Design]

- Do not bundle the control and communication cables together with the main circuit lines or power lines, and do not even run them in the vicinity of such.
  As a general guideline, maintain a distance of at least 100 mm between them.
  Otherwise, noise may result causing malfunctions.

⚠ Caution

[Safety Precautions for Installation]

- Use the board in an environment where all the general specifications given in the manual are met.
  Use in an environment which does not comply with the general specifications can cause electric shocks, fire, malfunctioning and damage or deterioration to the product.
- Install the board properly in the ISA bus slot of the target unit.
  Improper installation of the board can cause malfunctioning and failures or it can cause the board to fall out of the unit in which it was installed.
- Connect the communication cables properly to the connectors on the board. After installation, check whether the cables have lifted free from the connectors.
  Defective contact can cause problems in input and output.
- When installing the board, take care to protect yourself from injury by the mounted components, surrounding parts, etc.
- Before handling the interface board, touch a grounded metal object to discharge the static electricity from the human body.
  Failure to do so may cause failure or malfunction of the interface board.

◇! Danger

[Safety Precautions for Wiring]

- Before installing the board and connecting the wires, be absolutely sure to shut down at the external sources all the power supplies for all the phases concerned.
  Failure to do so can cause electric shocks or damage the product.
- After having installed the board and connected the wires, be absolutely sure when turning on the power or performing operations to attach the cover of the unit in which the board has been installed.
  Failure to attach the cover may lead to electric shocks.

⚠ Caution

[Safety Precautions for Wiring]

- Tighten up the screws anchoring the board using the specified torque.
  Looseness of these screws can cause short-circuiting, fires and malfunctioning.
- Take care to keep chips, scraps of wire and other foreign matter away from the board.
  Failure to do so can cause fire, failures or malfunctioning.
- Special skills and tools are required to connect the communication cables and connector plugs. For the connector plugs, specially designed products are required.
  To purchase the plugs, contact your nearest Mitsubishi Electric Service Center.
  Imperfect connections can cause short-circuiting, fires and malfunctioning.

Danger

[Safety Precautions at Startup and Maintenance]

- Do not connect or disconnect the communication cables while the power is still on.
  This can cause malfunctioning.
- Switch off the power before tightening up the screws which anchor the board.
  Tightening up the screws while the power is on can cause electric shocks.

⚠ Caution

[Safety Precautions at Startup and Maintenance]

- Familiarize yourself thoroughly with the manual and make sure that it is safe to proceed before attempting to go ahead with program changes during operation, forced output and such operations as RUN, STOP and PAUSE.
  Errors in operation can cause damage to the machine and result in accidents.
- Do not disassemble or remodel the board.
  This can cause failures, malfunctioning injury or fire.
- Switch off the power before installing or removing the board.
  Installing or removing the board while the power is still on can cause the board to fail or malfunction.
- The microprocessor contained on the interface board heats up during operation. Do not touch it directly when replacing the interface board or on other occasions.
- Before handling the interface board, touch a grounded metal object to discharge the static electricity from the human body.
  Failure to do so may cause failure or malfunction of the interface board.

⚠ Caution

[Safety Precautions for Disposal]

- Dispose of this product as an industrial waste product.

REVISIONS

* The manual number is given on the bottom left of the back cover.

Print Date	* Manual Number	Revision
Dec., 1996	IB(NA)-66749-A	First edition
May, 1997	IB(NA)-66749-B	Addition WindowsNT4.0 Workstation,Windows95
May, 2004	IB(NA)-66749-C	Correction SAFETY PRECAUTIONS, Section 4.1, Section 4.2, Subsection 6.4.1, Subsection 16.2.1 Addition Section 3.2, Subsection 4.1.8, Subsection 5.3.1, Subsection 5.3.2, Subsection 6.5.1, Subsection 6.5.2, Subsection 6.5.3, Subsection 6.5.4, Subsection 6.5.5, Subsection 6.5.6, Chapter 8, WARRANTY
Jun., 2007	IB(NA)-66749-D	Correction Subsection 9.4.5

This manual does not imply guarantee or implementation right for industrial ownership or implementation of other rights. Mitsubishi Electric Corporation is not responsible for industrial ownership problems caused by use of the contents of this manual.

INTRODUCTION

Thank you for purchasing this MELSEC series general-purpose sequencer made by Mitsubishi Electric.
Please read through this manual carefully and familiarize yourself with the functions and performance of this MELSEC series so that you will be able to operate it correctly.

Outline of Contents

1 GENERAL DESCRIPTION

This chapter describes the features of the model A70BDE-J71LP23GE MELSECNET/10 interface board, and gives details on the parts packed with the board.

2 SYSTEM CONFIGURATION

This chapter describes how the system may be configured when IBM-PC/AT computers are incorporated into a data link system using the model A70BDE-J71LP23GE MELSECNET/10 interface board, and it provides some cautionary notes.
It must be read before designing a system.

3 SPECIFICATIONS

This chapter details the specifications of the model A70BDE-J71LP23GE MELSECNET/10 interface board, and it gives an outline of the functions of the interface board's drivers.
It must be read before designing a system.

4 NAMES OF THE PARTS AND THEIR SETTINGS

This chapter describes the names of the parts on the model A70BDE-J71LP23GE MELSECNET/10 interface board and the switch settings available.
It must be read before installing the interface board in an IBM-PC/AT computer.

5 DRIVER INSTALLATION, DELETION, SHUTDOWN

This chapter outlines the methods used to install and delete the model A70BDE-J71LP23GE MELSECNET/10 interface board's drivers for DOS and Windows NT.
It must be read before installing or deleting a driver.

6 HOW TO USE THE UTILITIES

This chapter describes how to use the utility functions of the model A70BDE-J71LP23GE MELSECNET/10 interface board's drivers.
It must be read before starting up the model A70BDE-J71LP23GE MELSECNET/10 interface board after the drivers have been installed and before checking the data link statuses in the network system.

7 FUNCTIONS

This chapter describes the functions which are used with the model A70BDE-J71LP23GE MELSECNET/10 interface board to prepare the user programs which are run on the IBM-PC/AT computer.
It must be read before accessing the sequencer using a user program.

8 ERROR CODES

This chapter describes the error codes which are returned by the execution of the functions for the model A70BDE-J71LP23GE MELSECNET/10 interface board when user programs run on an IBM-PC/AT computer are executed.
It must be read before accessing the sequencer using a user program.

9 TROUBLESHOOTING

This chapter gives the troubleshooting flow to be referred to when trouble has occurred and when the operation of the model A70BDE-J71LP23GE MELSECNET/10 interface board is to be checked.
Read it when checking out the operations of the interface board or when trouble has occurred.

NOTES

o Mitsubishi Electric offers no guarantees for any software products sold on the market which are compatible with DOS or Windows NT and which are introduced by Mitsubishi Electric.

o The software copyrights of this product belong to Mitsubishi Electric.

o It is strictly forbidden to quote or copy this manual by any means either in part or in its entirety without permission from Mitsubishi Electric.

o Every effort is made to reflect software and hardware revisions in the contents of this manual but sometimes it is impossible to synchronize the printing of the manual with the latest revisions.

o This product's software programs are designed in such a way that one package is to be purchased per computer in principle.

o This product (and its manual) can be used only after the user has agreed to the provisions of the License Agreement.

o Mitsubishi Electric will not be held liable for whatever effects have resulted from the operation of this product (and its manual).

o The specifications of this product and the contents of this manual are subject to change without notice.

IMPORTANT

(1) Design the system in such a way that the safety circuit and circuit for protection from sequencer trouble are placed outside the sequencer.

(2) Mounted on the PCBs of the interface board are electronic components which are susceptible to static. Therefore, observe the following precautions when handling the PCBs.

    (a) Ground your body and the work bench.

    (b) Do not directly touch the product's conductive parts or electrical components.

CONTENTS

1 GENERAL DESCRIPTION 1- 1~1- 2

1.1 Features ··· 1- 1
1.2 Accessories ··· 1- 2

2 SYSTEM CONFIGURATION 2- 1~2- 4

2.1 System Configuration ··· 2- 1
2.2 Overall Configuration ··· 2- 2
2.2.1 Two-Tier System ··· 2- 2
2.2.2 Multi-Tier System ··· 2- 3
2.3 Checkpoints For System Configuration ··· 2- 4

3 SPECIFICATIONS 3- 1~3- 3

3.1 General Specifications ··· 3- 1
3.2 Performance Specifications ··· 3- 2
3.3 Function List ··· 3- 3

4 NAMES OF THE PARTS AND THEIR SETTINGS 4- 1~4-11

4.1 Installation ··· 4- 1
4.1.1 Precautions when handling ··· 4- 1
4.1.2 Installation environment ··· 4- 1
4.2 Names of Parts ··· 4- 2
4.3 Parts Settings ··· 4- 4
4.3.1 I/O Access Setting ··· 4- 4
4.3.2 I/O Address Setting ··· 4- 5
4.3.3 Memory Address Access Range Setting ··· 4- 6
4.3.4 Memory Address Setting ··· 4- 8
4.3.5 Interrupt Signal Setting ··· 4- 9
4.3.6 Network Condition Setting ··· 4-10
4.3.7 Station Number Setting ··· 4-10
4.3.8 Group Number Setting ··· 4-11
4.3.9 Network Number Setting ··· 4-11

5 DRIVER INSTALLATION, DELETION 5- 1~5-11

5.1 Installation/Deletion of DOS Driver ··· 5- 1
5.1.1 Installation of DOS Driver ··· 5- 1
5.1.2 Directories Configuration After Installation ··· 5- 3
5.1.3 Deletion of DOS Driver ··· 5- 4
5.2 Installation/Shutdown of Windows NT Driver ··· 5- 5
5.2.1 Installation of Windows NT Driver ··· 5- 5
5.2.2 Directories Configuration After Installation ··· 5- 7
5.2.3 Deletion of Windows NT Driver ··· 5- 8
5.3 Installation/Deletion of Windows 95 Driver ··· 5- 10
5.3.1 Installation of Windows 95 Driver ··· 5- 10
5.3.2 Deletion of Windows 95 Driver ··· 5- 11

6 HOW TO USE THE UTILITIES 6- 1~6-38

6.1 How To Use DOS MELSECNET/10 Utility ············6- 1
6.1.1 Utility Startup and Exit ············6- 1
6.1.2 Function List ············6- 2
6.1.3 Board Information ············6- 3
6.1.4 Network Settings ············6- 4
6.1.5 Network Monitor ············6- 5
6.1.6 Network Diagnosis ············6- 7
6.1.7 Device Monitor ············6-11
6.1.8 Information ············6-13
6.2 How To Use DOS Board Diagnosis Utility ············6-14
6.2.1 Utility Startup and Exit ············6-14
6.2.2 Function List ············6-14
6.2.3 Board Self-Diagnosis ············6-15
6.3 How To Use Windows NT Utilities ············6-16
6.3.1 Utility Startup and Exit ············6-16
6.3.2 Function List ············6-17
6.3.3 Information ············6-18
6.3.4 Board information ············6-19
6.3.5 Network Monitor (Own Station Information Display) ············6-21
6.3.6 Network Monitor (Station's Communication Status) ············6-22
6.3.7 Network Monitor (Station's Link Status) ············6-23
6.3.8 Device Batch Monitor ············6-24
6.3.9 Device 16 Points Registration Monitor ············6-26
6.3.10 Version Read ············6-28
6.4 How To Use Windows NT 3.51/NT 4.0/95 Board Diagnosis Utility ············6-29
6.4.1 Utility Startup and Exit ············6-29
6.4.2 Function List ············6-30
6.4.3 Board Information ············6-30
6.4.4 2-Port Diagnosis ············6-31
6.5 Error Viewer ············6-32
6.5.1 How to Start ············6-32
6.5.2 Functions ············6-32
6.5.3 Screen ············6-33
6.5.4 Log ············6-34
6.5.5 View ············6-36
6.5.6 Help ············6-38

7 FUNCTIONS 7- 1~7-36

7.1 Function List ············7- 1
7.2 Restrictions on Driver Usage ············7- 2
7.2.1 Common Restrictions ············7- 2
7.2.2 Restrictions on DOS Driver Usage ············7- 2
7.2.3 Restrictions on Windows NT 3.51 Driver Usage ············7- 2
7.2.4 Restrictions on Windows NT 4.0 Driver Usage ············7- 2

7.3 Common Data of Functions ························7- 3
7.4 Detailed Description of Functions ························7- 6
7.4.1 mdClose (communication circuit close) ························7- 6
7.4.2 mdControl (remote RUN/STOP/PAUSE) ························7- 7
7.4.3 mdDevRst (bit device reset) ························7- 8
7.4.4 mdDevSet (bit device set) ························7- 9
7.4.5 mdOpen (communication circuit open) ························7-10
7.4.6 mdRandR (device random read) ························7-11
7.4.7 mdRandW (device random write) ························7-14
7.4.8 mdReceive (all devices read) ························7-16
7.4.9 mdSend (all devices write) ························7-18
7.4.10 mdTypeRead (sequencer CPU type name read) ························7-20
7.5 Sample Programs ························7-22
7.5.1 Visual Basic Sample Program ························7-22
7.5.2 Visual C++ Sample Program ························7-24

8 ERROR CODES 8- 1~8- 7

9 TROUBLESHOOTING 9- 1~9-25

9.1 Troubleshooting Tips ························9- 1
9.2 Cause Location Method By Trouble Case ························9- 2
9.3 Case 1 ························9- 3
9.3.1 Driver Startup Error Message List ························9- 6
9.3.2 Message List For Error Events Occurring At Driver Startup ························9- 7
9.3.3 Self-Diagnosis Test ························9- 9
9.3.4 Self-Loopback Test ························9- 9
9.4 Case 2 ························9-11
9.4.1 Check Procedure Flow When RUN LED is OFF ························9-12
9.4.2 Check Procedure Flow When SD/RD LED Fails to Light ························9-13
9.4.3 Check Procedure Flow When F.E./R.E. LED Lights ························9-14
9.4.4 Check Procedure Flow When Entire System Data Link Is Disabled ························9-15
9.4.5 Check Procedure Flow When Specific Station Data Link Is Disabled ························9-17
9.5 Case 3 ························9-19
9.6 Information Required When Making Inquiries ························9-25

APPENDIX APP- 1

Appendix 1 Outline Drawings ························APP- 1
Appendix 1.1 A70BDE-J71LP23GE ························APP- 1

MEMO

1 GENERAL DESCRIPTION

This manual describes the specifications, handling and monitoring methods for the network system of the model A70BDE-J71LP23GE MELSECNET/10 interface board which is used to configure the MELSECNET/10 network system through its installation in IBM-PC/AT computers as an optional board.

The interface board is compatible with the network system below.

- Compatible with normal stations of MELSECNET/10

1.1 Features

The interface board has the following features.

(1) IBM-PC/AT computers can be incorporated into the MELSECNET/10 network system. When the interface board is installed in such a computer, it can used as a normal station.

(2) The test and monitor information relating to the data links is displayed on a CRT screen. Operation is made easier since the test and monitoring conditions of the data links can be displayed on the IBM PC-AT computer's CRT screen.

(3) Drivers which support various operating systems are provided. Drivers for both DOS and Windows NT applications are available to make it easy to construct the system which best suits the user's environment.

(4) Functions which support user programming are provided. By means of functions which support Visual C++ and Visual Basic, sequencer CPU remote control and device read/write are enabled so that user applications can be prepared with ease.

(5) Communication with N:N is possible using the transient transmission functions. A sequencers serving as a normal station can be accessed from an IBM-PC/AT computer serving as a normal station to make it possible to send/receive data, read/write devices, etc.

1.2 Accessories

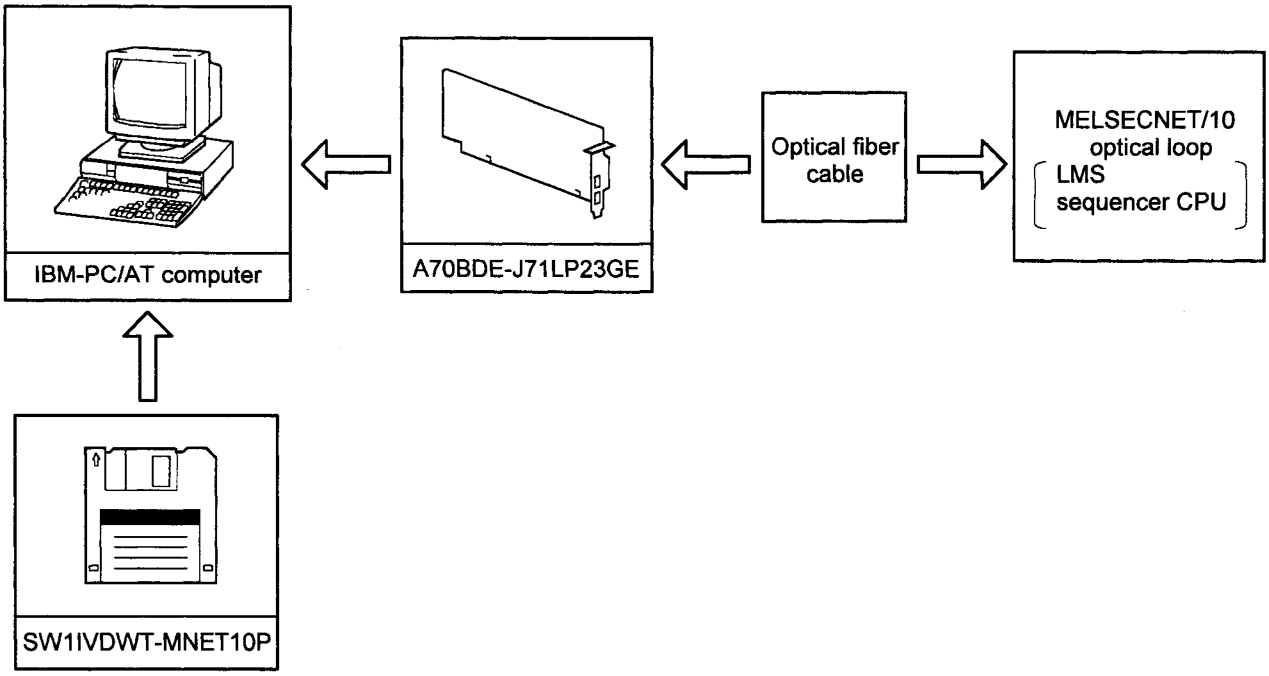
After opening the box in which the product was packed, check that all the parts listed in the table below are present and accounted for in the box.

Name of part	Quantity
Model A70BDE-J71LP23GE MELSECNET/10 interface board	1
Model SW1IVDWT-MNET10P MELSECNET/10 software package	1
User's manual for model A70BDE-J71LP23GE MELSECNET/10 interface board installed in IBM-PC/AT computer (this manual)	1

2 SYSTEM CONFIGURATION

2.1 System Configuration

Shown below is the system configuration when the interface board is installed in an IBM-PC/AT computer for operation.



REMARKS

Only optical or coaxial cable may be used in each MELSECNET/10.

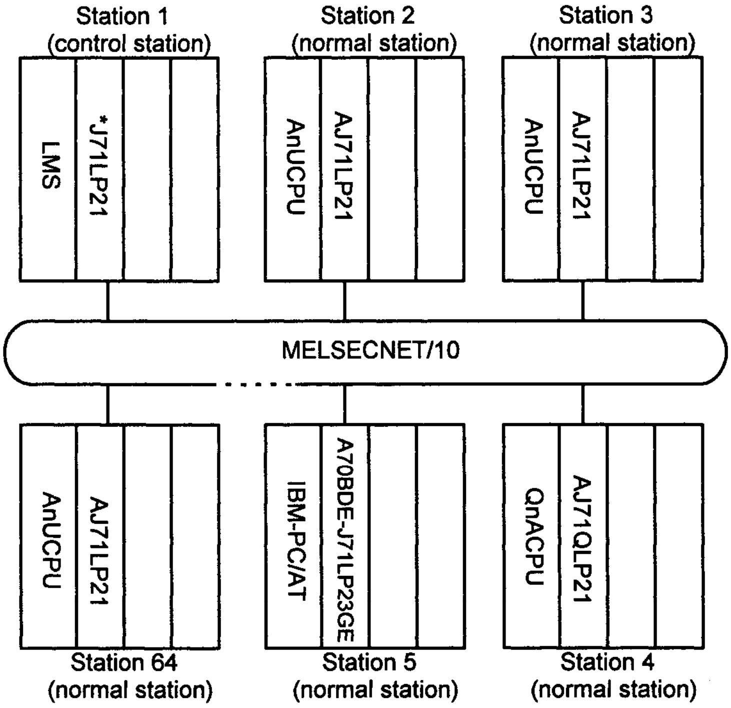
2.2 Overall Configuration

How the system is configured when the interface board is installed in an IBM-PC/AT computer for use in a MELSECNET/10 2-tier or multi-tier system is described below.
For details on configurations using combinations of the MELSECNET/10, refer to the MELSECNET/10 Network System (PC to PC network) Reference Manual.

2.2.1 Two-Tier System

A "2-tier system" is a single system in which the network unit control and normal stations are connected by optical fiber cables.

(1) Optical loop system
A total of 64 stations consisting of one control station and 63 normal stations can be connected.
The interface board can be provided only as a normal station.
In the example of the system shown below, station 1 is provided as the control station.



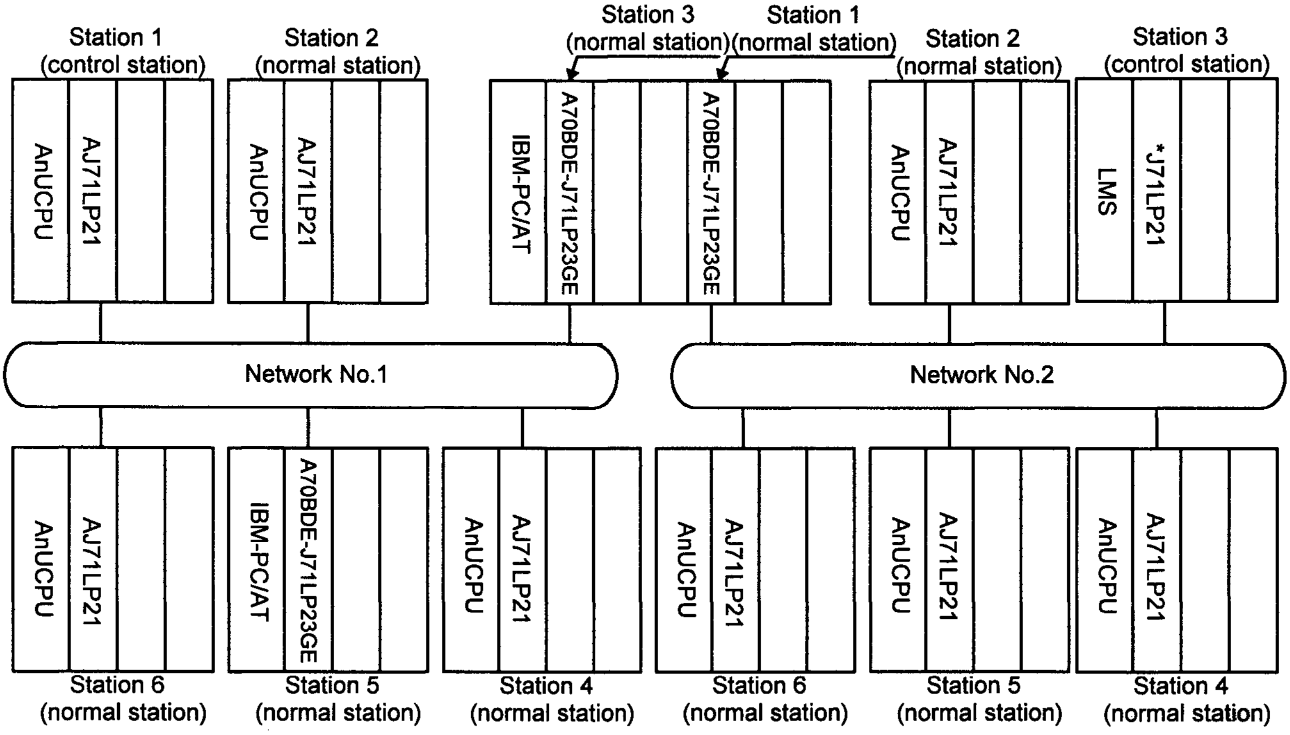
* This indicates an arbitrary character string.

The "control station" sets the data link parameters and controls the whole network; a "normal station" receives the data link parameters from the control station and establishes the data link in accordance with the parameter settings.

2.2.2 Multi-Tier System

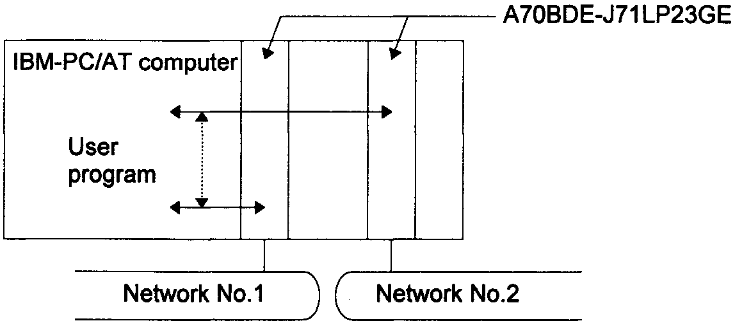
A "multi-tier system" is a system in which a multiple number of networks reside.
Two or more interface boards are installed in the IBM-PC/AT computer for connection between networks.

In the system configuration shown below, station 3 in network No.1 and station 1 in network No.2 are installed in the same IBM-PC/AT computer to make the connection between the networks.



* This indicates an arbitrary character string.

When the LMS is used for the station which connects several networks, the function for transfer between data links and the routing function are not supported by the IBM-PC/AT computer.
When the function for transfer between data links is required, exchange the data between the networks via a user program, as shown below.



For details on the function for transfer between data links and the routing function, refer to the MELSECNET/10 Network System (PC to PC network) Reference Manual.

2.3 Checkpoints For System Configuration

The interface boards can be installed in IBM-PC/AT computers and used as normal stations to construct a MELSECNET/10 network system.
Some checkpoints to be heeded when configuring a system are given below.

(1) An optical fiber cable and coaxial cable cannot be used together for the interface board within a loop. Therefore, select an interface board with the specifications which match the network system.

(2) The number of interface boards which can be installed in an IBM-PC/AT computer are as follows:
- 4 boards for DOS
- 4 boards for Windows NT

3 SPECIFICATIONS

This chapter describes the performance specifications and functions of the interface board.

3.1 General Specifications

(1) The general specifications of the interface board are provided below.

Item	Specification				
Usage environment temperature	0 to 55°C				
Storage environment temperature	-20 to 75°C				
Usage environment humidity	10 to 90% RH, No. condensation formation				
Storage environment humidity	10 to 90% RH, No. condensation formation				
		Frequency	Acceleration	Amplitude	Sweep cycles
Anti-vibration	In compliance with JIS C 0911	10 to 55 Hz	---	0.075	10 cycles (1 octave/minute) (See *)
		55 to 150 Hz	9.8 m/s^2	---	
Anti-shock	In compliance with JIS C 0912 (10 g, 3 times each in 3 directions)				
Noise resistance	1500 Vp-p noise voltage, 1μs noise width, 25 to 60 Hz noise frequency (generated by noise simulator)				
Usage environment	No corrosive gases, no significant dust				
Cooling method	Self-cooling				

* "1 octave" denotes the range from the initial frequency to a frequency which is twice or one-half of that initial frequency.
Examples: 10 Hz -> 20 Hz, 20 Hz -> 40 Hz, 40 Hz -> 20 Hz or 20 Hz -> 10 Hz
All these changes in frequency are known as one octave.

(2) The general specifications applying after the interface board has been installed comply with the specifications for the IBM-PC/AT computer main unit.

3.2 Performance Specifications

The performance specifications of the interface board are listed in the table below.

Item		Specification
Maximum number of link points per network	X/Y	8192 points
	B	8192 points
	W	8192 points
Maximum number of link points per link		W * 2+(B+Y)/8 <=2000 bytes
Communication speed		10 MBPS
Communication method		Token ring system
Synchronous method/modulation method		Frame synchronization
Transmission code		NRZI (Non Return to Zero Inverted)
Transmission path format		Duplex
Transmission path		Optical (GI)
Transmission format		Conforms to HDLC (frame method)
Maximum number of networks		239
Maximum number of groups		9
Number of stations connected in inter-PC network		64 stations (1 control station, 63 normal stations)
Overall loop distance		30 km Distance between stations: 2 km
Error control system		CRC ($X^{16} + X^{12} + X^5 + 1$) and retry based on time over
Transient transmission function		N:N communication (data send/receive, data read/write, etc.)
RAS functions		System down prevention by transfer to control station, error detection by special relays and registers, network monitoring, diagnostic functions, etc.
5V DC Internal current Consumption		0.55 A
Weight		0.19 kg

When installed in an IBM-PC/AT computer, the MELSECNET/10 does not have functions for transfer between loops and routing.

3.3 Function List

The interface board provides the following functions.
For further details on the functions relating to data communication as well as on the special relays and special registers, refer to the MELSECNET/10 Network System (PC to PC network) Reference Manual.

Function	Description
Data communication functions	(1) Cyclic transmission function allowing communication via MELSECNET/10 for input (X), output (Y), link relay (B) and link register (W). ・ This function is independent for each interface board, and it supports 8,000 points in each device. ・ It supports 2,000 bytes as the number of link points per station. (2) Transient transmission function allowing communication by N:N. ・ Communication enabled even when cyclic transmission is not being performed. ・ Designation of maximum number of transient times during scanning of one link enabled.
Loopback function	Using the duplex loop format for the optical fiber cable, normal operation is performed only for the operable stations by removing the faulty stations from the system when the cable has broken or when normal stations are disconnected.
Duplex transmission function	Enables transmission speed to be doubled by using each transmission path independently with duplex loop format for optical fiber cable.
Automatic return function	Even when a station has been disconnected by a fault, this automatically returns the station when the area affected by the fault is restored to normal operation. It is executed by a board information setting.
Test function	By means of the test mode settings, this conducts a test and checks the hardware and loop lines.
Loop monitor function	By means of the loop monitor settings, this enables the own station and other stations to be monitored and their operating statuses to be checked.
Self-diagnostic function	(1) This displays error messages by means of error codes. (2) This stores the fault detection data in the special relays and special registers.

4 NAMES OF THE PARTS AND THEIR SETTINGS

4.1 Installation

This section explains precautions when handling the interface board and the installation environment.

4.1.1 Precautions when handling

The following are precautions to be noted when handling the interface board.

DANGER

- While energizing, do not touch the connector.
  Doing so may result in electric shock or cause malfunctioning.

CAUTION

- Fasten the interface board securely using the installation screws and tighten the installation screws securely within the specified torque range.
  If the screws are loose, this may cause malfunctioning.
  If the screws are tightened too much, this could cause damage to the screws or unit, leading to malfunctioning.
- Do not directly touch the conductive section of the interface board.
  Doing so could result in malfunctioning or breakdown of the interface board.
- Before handling the interface board, touch a grounded metal object to discharge the static electricity from the human body.
  Failure to do so may cause failure or malfunction of the interface board.
- Handle the interface board in a location where there is no static electricity.
  Static electricity could result in failure or malfunctioning.
- The interface board is packed in a bag for preventing static electricity.
  Always place the interface board in this bag when storing or transporting.
  Otherwise, failure or malfunctioning may result.
- Take care that foreign objects such as chips or wiring debris do not get into the PC.
  This could result in fire, breakdowns or malfunctioning.
- Do not dismantle or rebuild the interface board.
  This will result in failure, malfunctioning, injury or fire.
- Always turn off all external power before installing or removing the interface board.
  If power is not turned off, there is a risk of electric shock or damage to the product.
- When disposing of the product, handle it as an industrial waste.
- Do not drop the interface board or subject it to strong impact.
  This will result in failure or malfunctioning of the board.

See the instruction manual provided with the IBM PC/AT compatible PC for the clamping torque of the interface board mounting screws.

4.1.2 Installation environment

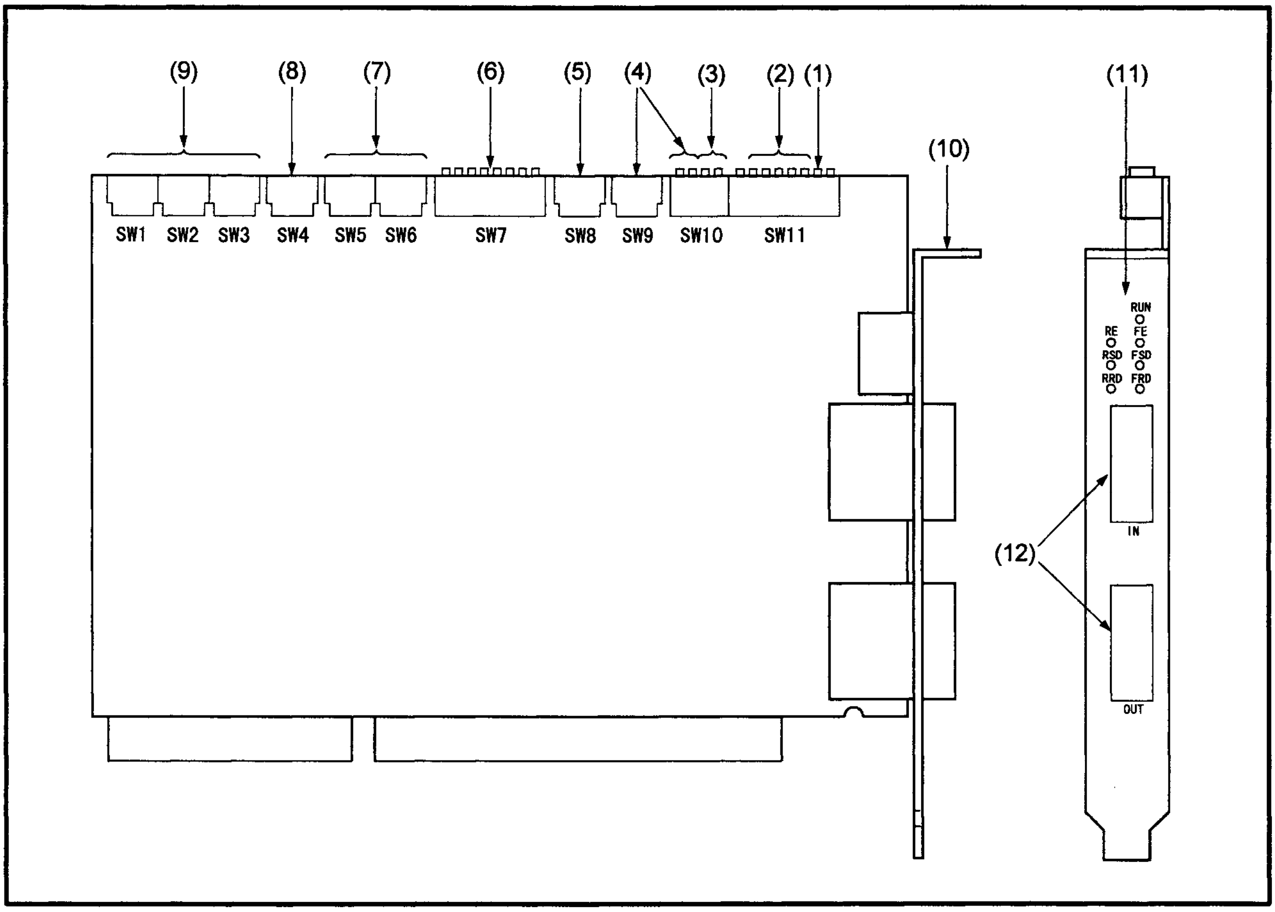
See the instruction manual accompanying the PC module regarding installation of the PC module in which the interface board is mounted.

CAUTION

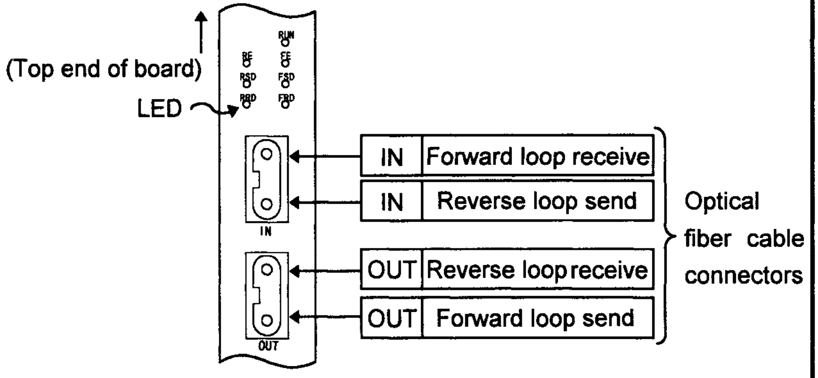
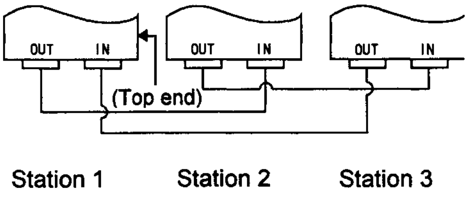
- Always ground the PC module using grounding type D (Class 3 grounding).
  Otherwise, there is the risk of malfunctioning.

4.2 Names of Parts

The names of the parts on the interface board are shown below.



No.	Part	Switch No.	Description
(1)	I/O access setting switch	SW11-2	This sets the number of bits accessed in the shared memory.
(2)	I/O address setting switches	SW11-3 to 7	These set the I/O addresses occupied by the board.
(3)	Memory access range setting switches	SW10-1, 2	These set the range accessed in the shared memory.
(4)	Memory address setting switches	SW10-3, 4 and SW9	These set the addresses of the shared memory occupied by the board.
(5)	Interrupt signal setting	SW8	Using a rotary switch, this sets the number by which the operating system identifies the interface board.
(6)	Network condition setting switch	SW7	This sets the operating status of the network.
(7)	Station number setting switches	SW5, 6	Using two rotary switches, these set the station numbers from 01 to 64.
(8)	Group number setting switch	SW4	Using a rotary switch, this sets the group number.
(9)	Network number setting switches	SW1, 2, 3	Using three rotary switches, these set the network numbers from 1 to 239.

No.	Part	Description
(10)	Mounting fixture	This is used to secure the interface board to the IBM-PC/AT computer.
(11)	LEDs for displaying operating status	These indicate the operating status, errors, etc.
(12)	Connectors for optical fiber cables	(a) The cable connectors are shown below.  (Top end of board) LED RUN RF FF RSD FSD RRD FRD IN OUT IN Forward loop receive IN Reverse loop send OUT Reverse loop receive OUT Forward loop send Optical fiber cable connectors (b) The cables are connected as shown below.  OUT IN (Top end) Station 1 Station 2 Station 3 IN: Connected to OUT of previous station OUT: Connected to IN of next

4.3 Parts Settings

This section describes how the setting switches on the interface board are set.
Be absolutely sure to turn off the power supplied to the IBM-PC/AT computer before setting the switches.
The setting items are as follows:

(1) I/O access setting switch	(Refer to Item 4.3.1)
(2) I/O address setting switches	(Refer to Item 4.3.2)
(3) Memory address access range setting switches	(Refer to Item 4.3.3)
(4) Memory address setting switches	(Refer to Item 4.3.4)
(5) Interrupt signal setting switch	(Refer to Item 4.3.5)
(6) Network condition setting switch	(Refer to Item 4.3.6)
(7) Station number setting switches	(Refer to Item 4.3.7)
(8) Group number setting switch	(Refer to Item 4.3.8)
(9) Network number setting switches	(Refer to Item 4.3.9)

4.3.1 I/O Access Setting

This switch selects the number of bits which the IBM-PC/AT computer accessed in the shared memory of the board.

SW11-2	Occupied I/O addresses
OFF	Not used (cannot be set)
ON	16bits accessed (factory setting)

Keep this switch at ON for use.

4.3.2 I/O Address Setting

These switches set the I/O addresses occupied by the board.
Switches other than those listed below cannot be set.

Switch setting					Occupied I/O addresses
SW11-3	SW11-4	SW11-5	SW11-6	SW11-7	
OFF	OFF	OFF	ON	OFF	0100h to 012Fh
OFF	OFF	OFF	ON	ON	0180h to 01AFh
OFF	OFF	ON	OFF	OFF	0200h to 022Fh
OFF	OFF	ON	OFF	ON	0280h to 02AFh
OFF	OFF	ON	ON	OFF	0300h to 032Fh (factory setting)
OFF	OFF	ON	ON	ON	0380h to 03AFh
OFF	ON	OFF	ON	OFF	1100h to 112Fh
OFF	ON	OFF	ON	ON	1180h to 11AFh
OFF	ON	ON	OFF	OFF	1200h to 122Fh
OFF	ON	ON	OFF	ON	1280h to 12AFh
OFF	ON	ON	ON	OFF	1300h to 132Fh
OFF	ON	ON	ON	ON	1380h to 13AFh
ON	OFF	OFF	ON	OFF	2100h to 212Fh
ON	OFF	OFF	ON	ON	2180h to 21AFh
ON	OFF	ON	OFF	OFF	2200h to 222Fh
ON	OFF	ON	OFF	ON	2280h to 22AFh
ON	OFF	ON	ON	OFF	2300h to 232Fh
ON	OFF	ON	ON	ON	2380h to 23AFh
ON	ON	OFF	ON	OFF	3100h to 312Fh
ON	ON	OFF	ON	ON	3180h to 31AFh
ON	ON	ON	OFF	OFF	3200h to 322Fh
ON	ON	ON	OFF	ON	3280h to 32AFh
ON	ON	ON	ON	OFF	3300h to 332Fh
ON	ON	ON	ON	ON	3380h to 33AFh

4.3.3 Memory Address Access Range Setting

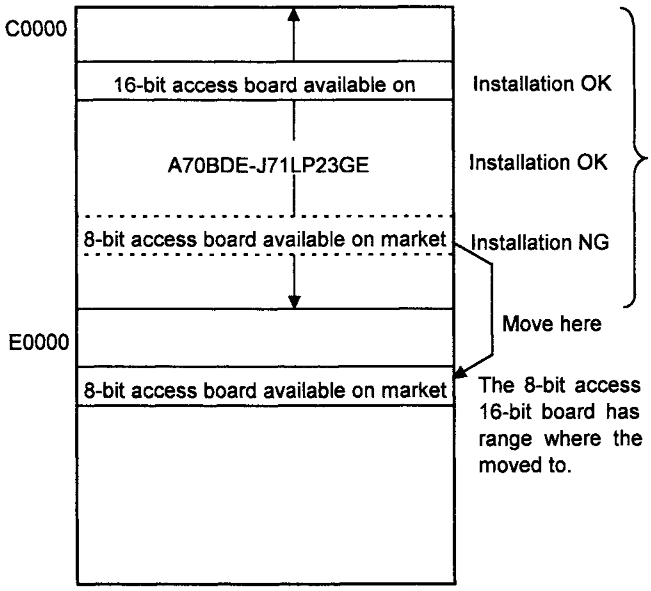
These switches set the memory range in which the IBM-PC/AT computer accesses the shared memory of the interface board.

Switch setting		Description
SW10-1	SW10-2	
OFF	---	8 bits accessed (8KB)
ON	OFF	16 bits accessed (128KB) (factory setting)
ON	ON	Not used (cannot be set)

[NOTE]

Depending on the type of board, proper operation may not be possible with the following installation. If this is the case, change the installation location.

(1) When the switches have been set to 16 bits (128KB)

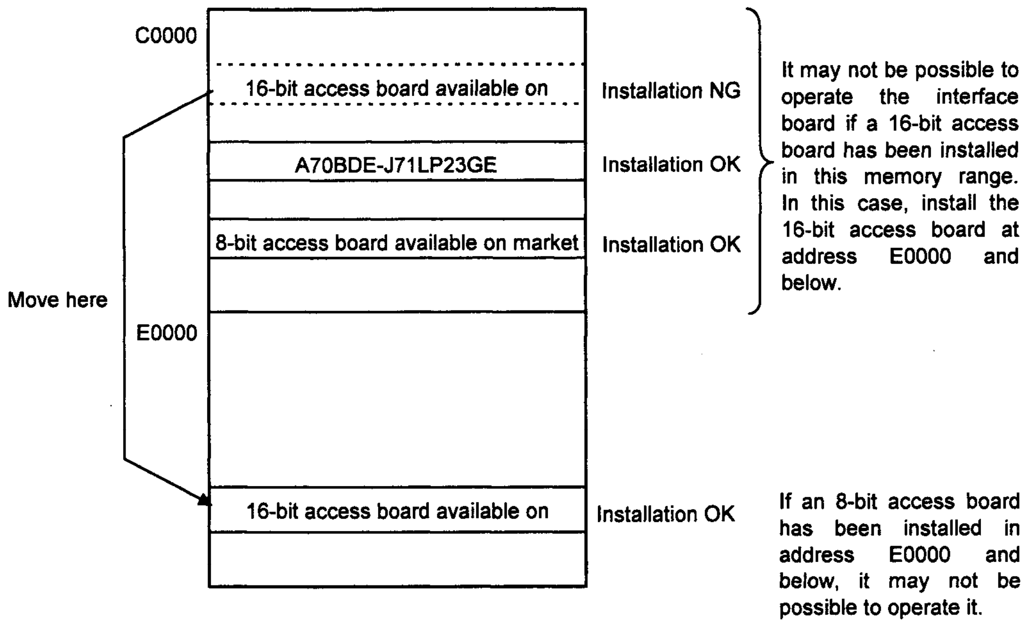

C0000
E0000
16-bit access board available on
A70BDE-J71LP23GE
8-bit access board available on market
8-bit access board available on market
Installation OK
Installation OK
Installation NG
Move here

An 8-bit access board cannot be installed in this memory range. Install the 8-bit access board at address E0000 and below.

The 8-bit access board cannot be installed if a 16-bit board has been installed in the memory range where the 8-bit board was going to be moved to.

The same applies when the A70BDE-J71LP23GE has been installed in an area starting with address E0000.

(2) When the switches have been set to 8 bits


C0000
16-bit access board available on
A70BDE-J71LP23GE
8-bit access board available on market
Move here
E0000
16-bit access board available on
Installation NG
Installation OK
Installation OK
Installation OK

It may not be possible to operate the interface board if a 16-bit access board has been installed in this memory range. In this case, install the 16-bit access board at address E0000 and below.

If an 8-bit access board has been installed in address E0000 and below, it may not be possible to operate it.

The same applies when the A70BDE-J71LP23GE has been installed in an area starting with address E0000.

4.3.4 Memory Address Setting

This section describes how to set the addresses of the shared memory to be occupied by the interface board.

(1) The shared memory area settings are shown below.

SW10-3	SW10-4	
OFF	OFF	C
OFF	ON	D
ON	OFF	E
ON	ON	F

Rotary switch SW9 number

Memory addresses: * * 000h to * *FFFh
Default: C8000h to C8FFFh

* This indicates an arbitrary character string.

(2) At the factory, SW10-3 and SW10-4 are set to OFF, and rotary switch SW9 is set to "8."

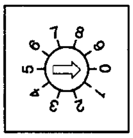
(3) Set the switches in such a way that their settings will not duplicate the settings of the shared memory areas of other optional boards.

(4) Normally, the interface board cannot be installed in addresses C0000 to C7FFF because this is where the video BIOS is or in addresses F0000 to FFFFF because this is where the ROM BIOS is.

4.3.5 Interrupt Signal Setting

The interface board interrupt signal setting is provided as a number for identifying from which optional board the operating system was accessed when the operating system is accessed from an interface board.

(1) The numbers available to the switch for setting the interrupt signals and whether setting is possible or not are shown below.

SW8	Switch No.	Setting enabled/disabled	IRQ No.
	0	Enabled (disabled when serial port 2 is used)	3
	1	Enabled (disabled when serial port 1 is used)	4
	2	Enabled (disabled when parallel port 2 is used)	5
	3	Enabled (disabled when parallel port 1 is used)	7
	4	Enabled	10
	5	Enabled	11
	6	Enabled (disabled when PS/2 mouse is used)	12
	7	Enabled	15
	8	(Setting prohibited)	--
	9	(Setting prohibited)	--

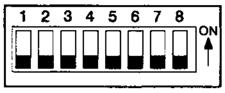
(2) Set the switch in such a way that its setting will not duplicate the interrupt signal settings of other optional boards.

(3) At the factory, the switch is set to "4."

4.3.6 Network Condition Setting

The network condition setting switches are used to set the operating condition of the MELSECNET/10.

(1) How to set the network condition setting switches on the interface board is shown below.

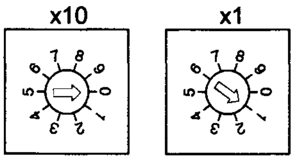
SW7	Setting	
	SW	Description
	0	Not used (always OFF)
	1	
	2	
	3	
	4	
	5	
	6	
	7	
	8	

(2) At the factory, all the switches are set to "OFF."

4.3.7 Station Number Setting

The interface board station number refers to the number used to set the station number of the own station when it is connected to MELSECNET/10.

(1) How to set the station number setting switches on the interface board is shown below.

SW5, SW6	Setting Description
	(1) x10 switch: For setting the station number 10's digit. (2) x1 switch: For setting the station number 1's digit. (3) A number from 1 to 64 is used as the station number.

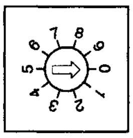
(2) At the factory, the switches are set to station number "01."

(3) For the cautionary notes on setting the station number when the interface board is to be connected to MELSECNET/10, refer to the MELSECNET/10 Network System (PC to PC network) Reference Manual.

4.3.8 Group Number Setting

The interface board group number is the number used for setting the group number of the own station when the interface board is to be connected to MELSECNET/10.

(1) How to set the interface board group number is shown below.

SW4	Setting Description
	A number from 1 to 9 is used. The "0" setting denotes that no group is designated.

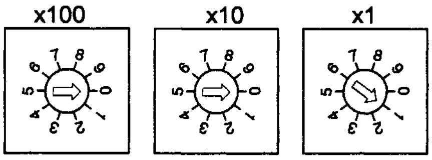
(2) At the factory, the switch is set to "0."

(3) For the cautionary notes on setting the group number when the interface board is to be connected to MELSECNET/10, refer to the MELSECNET/10 Network System (PC to PC network) Reference Manual.

4.3.9 Network Number Setting

The interface board network number refers to the number used to set the network number of the own station when it is to be connected to MELSECNET/10.

(1) How to set the network number setting switches on the interface board is shown below.

SW1, SW2, SW3	Setting Description
	(1) x100 switch: For setting the network number 100's digit. (2) x10 switch: For setting the network number 10's digit. (3) x1 switch: For setting the network number 1's digit. (4) A number from 1 to 239 is used as the network number.

(2) At the factory, the switches are set to network number "001."

(3) For the cautionary notes on setting the network number when the interface board is to be connected to MELSECNET/10, refer to the MELSECNET/10 Network System (PC to PC network) Reference Manual.

5 DRIVER INSTALLATION, DELETION

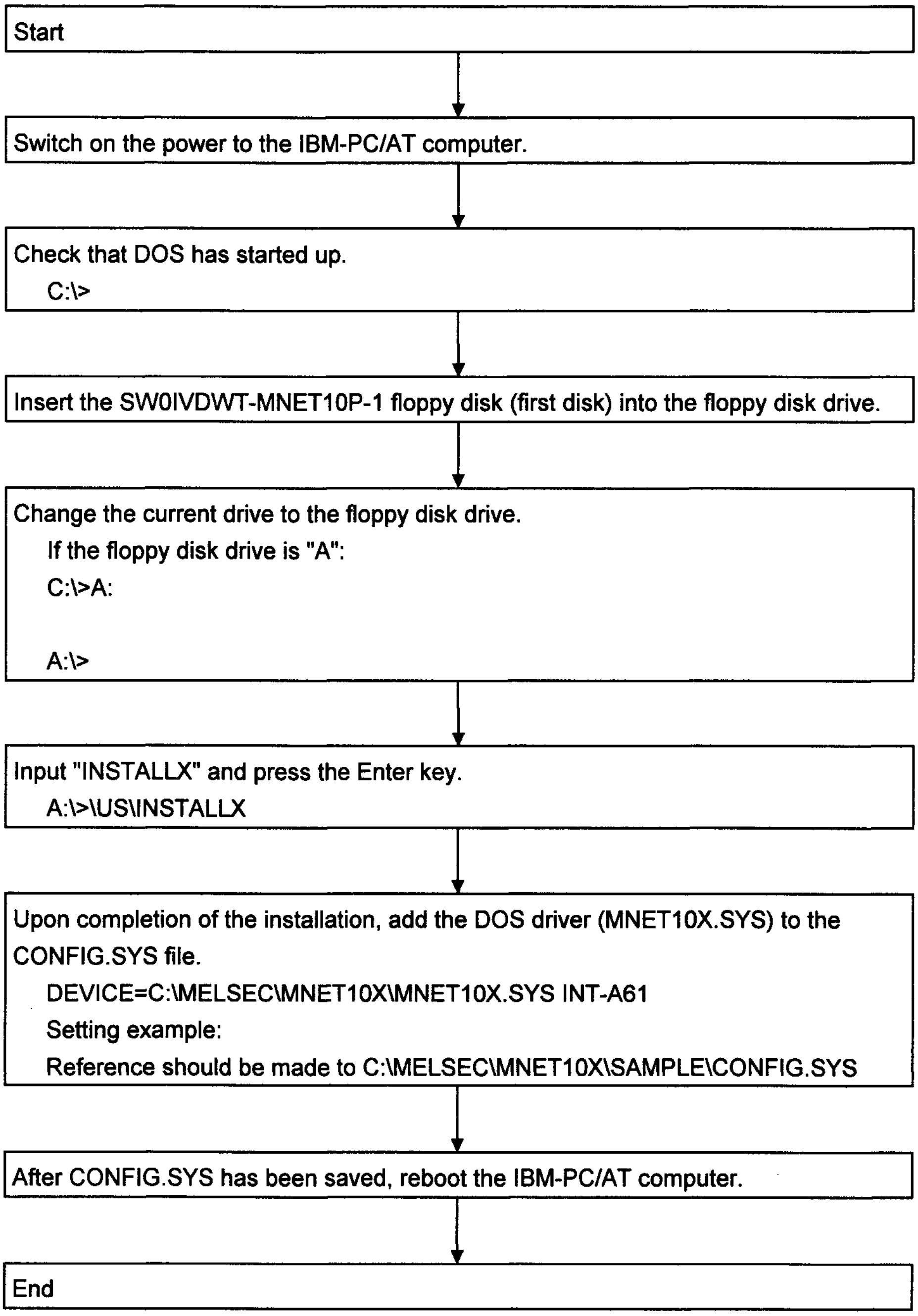
This chapter describes the installation and deletion of the drivers for operating the board for each operating system.

5.1 Installation/Deletion of DOS Driver

This section describes how to install and delete the driver for DOS.

5.1.1 Installation of DOS Driver

Follow the procedure below to install the DOS driver.



Start

Switch on the power to the IBM-PC/AT computer.

Check that DOS has started up.

```
C:\>
```

Insert the SW0IVDWT-MNET10P-1 floppy disk (first disk) into the floppy disk drive.

Change the current drive to the floppy disk drive.
If the floppy disk drive is "A":

```
C:\>A:

A:\>
```

Input "INSTALLX" and press the Enter key.

```
A:\>\US\INSTALLX
```

Upon completion of the installation, add the DOS driver (MNET10X.SYS) to the CONFIG.SYS file.

```
DEVICE=C:\MELSEC\MNET10X\MNET10X.SYS INT-A61
```

Setting example:
Reference should be made to C:\MELSEC\MNET10X\SAMPLE\CONFIG.SYS

After CONFIG.SYS has been saved, reboot the IBM-PC/AT computer.

End

The driver will automatically recognize the board settings.

<CONFIG.SYS description example>

```
DEVICE=C:\MELSEC\MNET10X\MNET10X.SYS  INT-A61
                                      ( See (1) )
```

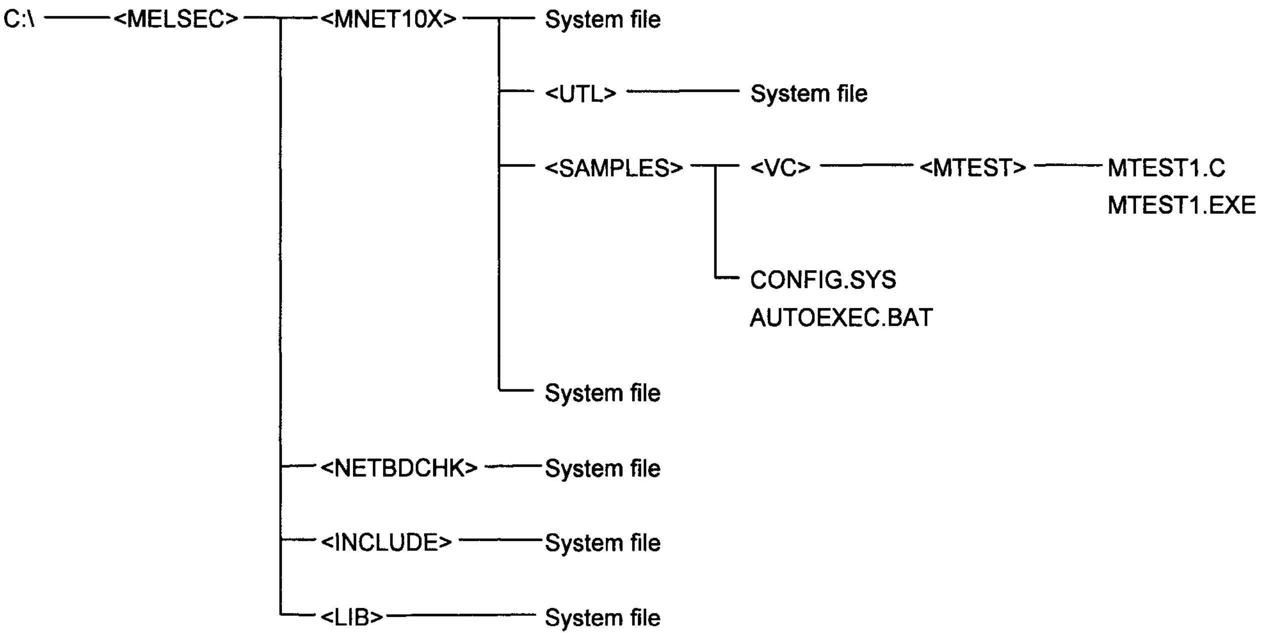
(1) Assignment of software program interrupt number
->Assign a number in the 60 to FF (hexadecimal notation) range.   INT-Ann

CAUTION

Ensure that there is no duplication of the settings with other drivers and software packages.

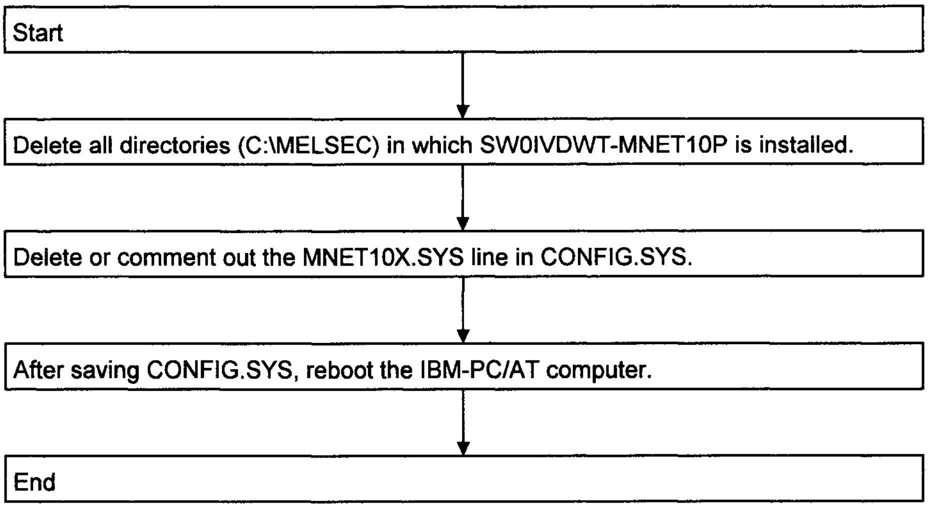
5.1.2 Directories Configuration After Installation

The Directories configuration after the DOS driver has been installed is shown below.



5.1.3 Deletion of DOS Driver

Follow the procedure below to delete the DOS driver.



Start

Delete all directories (C:\MELSEC) in which SW0IVDWT-MNET10P is installed.

Delete or comment out the MNET10X.SYS line in CONFIG.SYS.

After saving CONFIG.SYS, reboot the IBM-PC/AT computer.

End

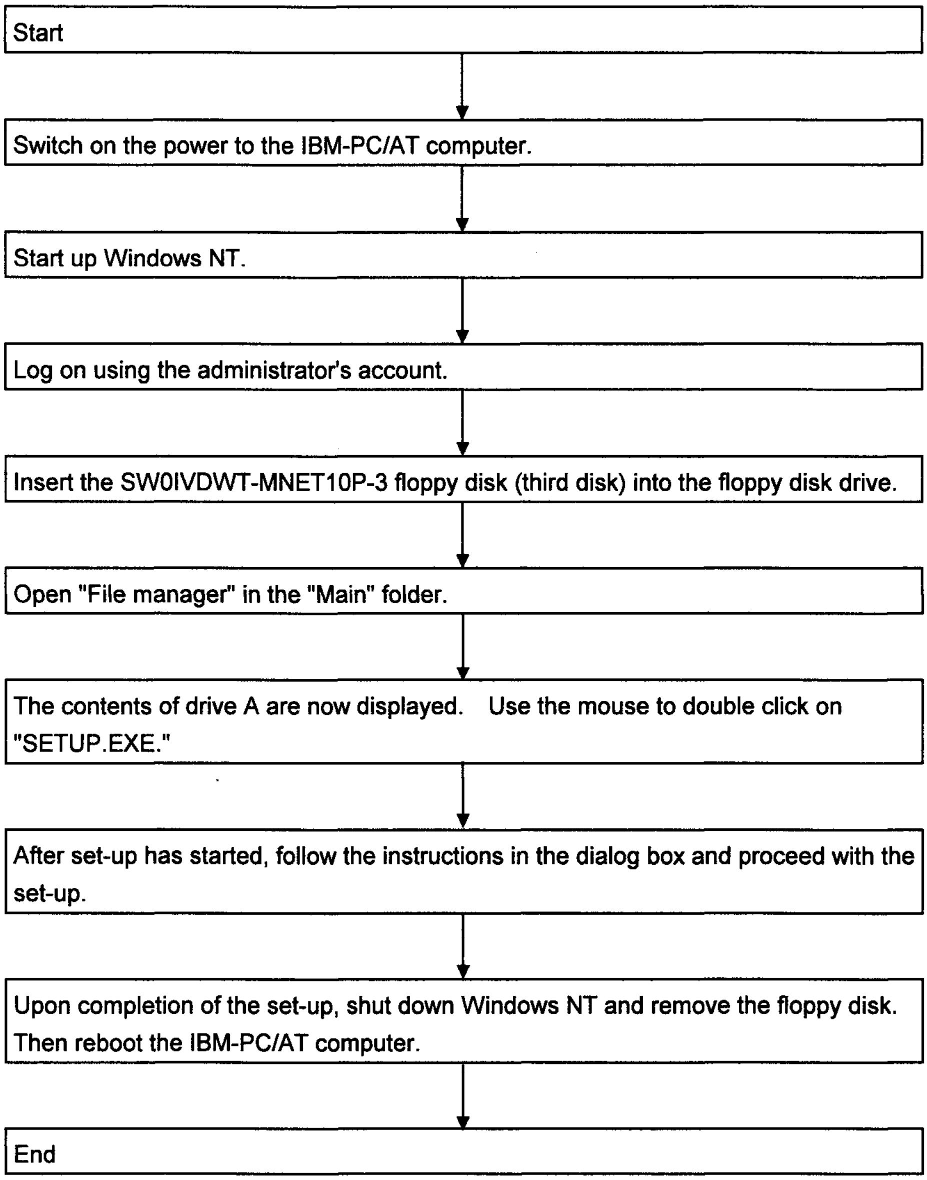
5.2 Installation/Deletion of Windows NT Driver

This section describes how to install and deletion the driver for Windows NT.

5.2.1 Installation of Windows NT Driver

Follow the procedure below to install the Windows NT driver.

(1) for Windows NT 3.51



Start

Switch on the power to the IBM-PC/AT computer.

Start up Windows NT.

Log on using the administrator's account.

Insert the SW0IVDWT-MNET10P-3 floppy disk (third disk) into the floppy disk drive.

Open "File manager" in the "Main" folder.

The contents of drive A are now displayed. Use the mouse to double click on "SETUP.EXE."

After set-up has started, follow the instructions in the dialog box and proceed with the set-up.

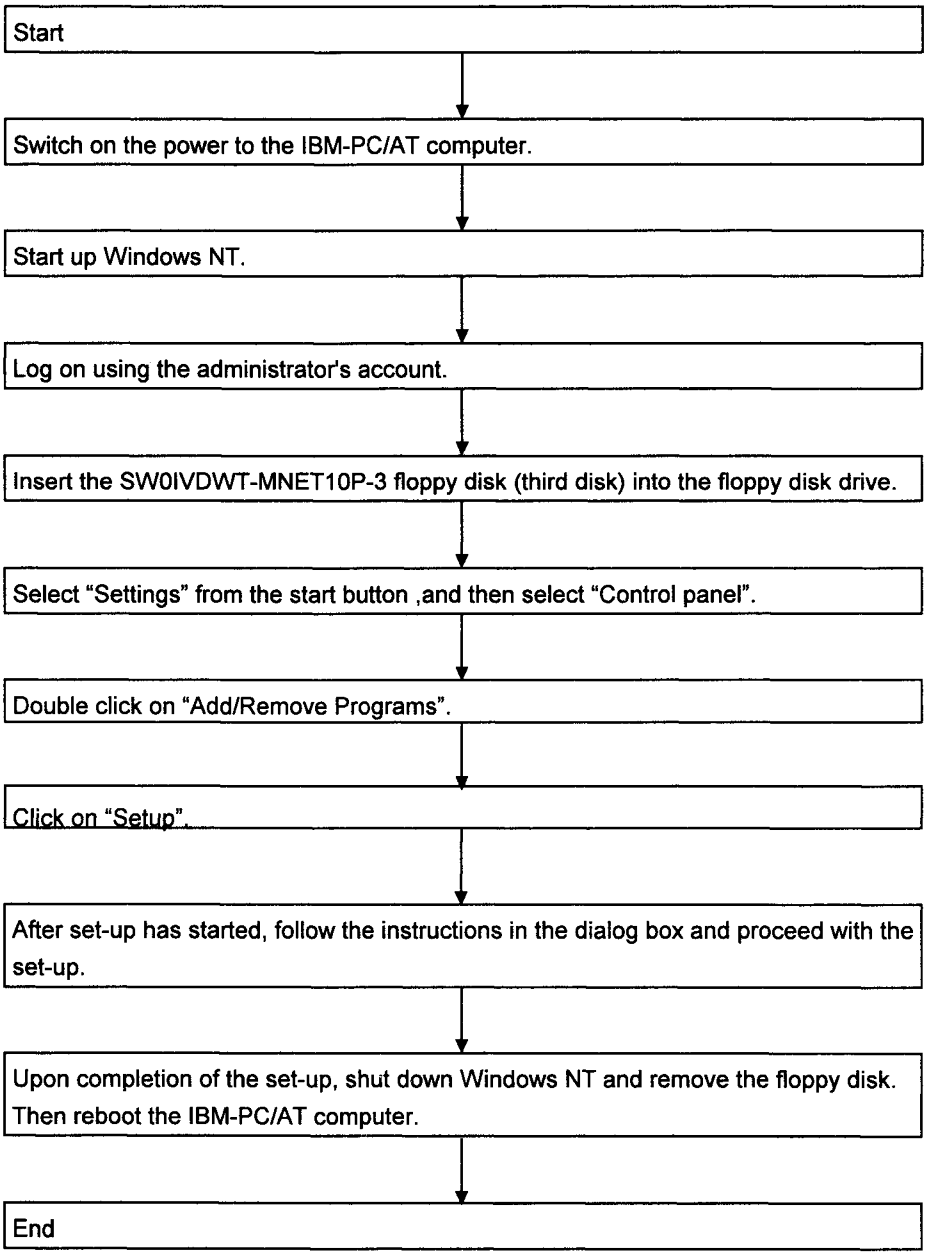
Upon completion of the set-up, shut down Windows NT and remove the floppy disk. Then reboot the IBM-PC/AT computer.

End

CHECKPOINTS

To re-install the driver, reboot the system and ensure that no other application has been started up before proceeding with re-installation. If the system is not rebooted first, a DLL copy error will occur at some point during re-installation.

(2) for Windows NT 4.0



Start

Switch on the power to the IBM-PC/AT computer.

Start up Windows NT.

Log on using the administrator's account.

Insert the SW0IVDWT-MNET10P-3 floppy disk (third disk) into the floppy disk drive.

Select “Settings” from the start button ,and then select “Control panel”.

Double click on “Add/Remove Programs”.

Click on “Setup”.

After set-up has started, follow the instructions in the dialog box and proceed with the set-up.

Upon completion of the set-up, shut down Windows NT and remove the floppy disk. Then reboot the IBM-PC/AT computer.

End

CHECKPOINTS

To re-install the driver, reboot the system and ensure that no other application has been started up before proceeding with re-installation. If the system is not rebooted first, a DLL copy error will occur at some point during re-installation.

5.2.2 Directories Configuration After Installation

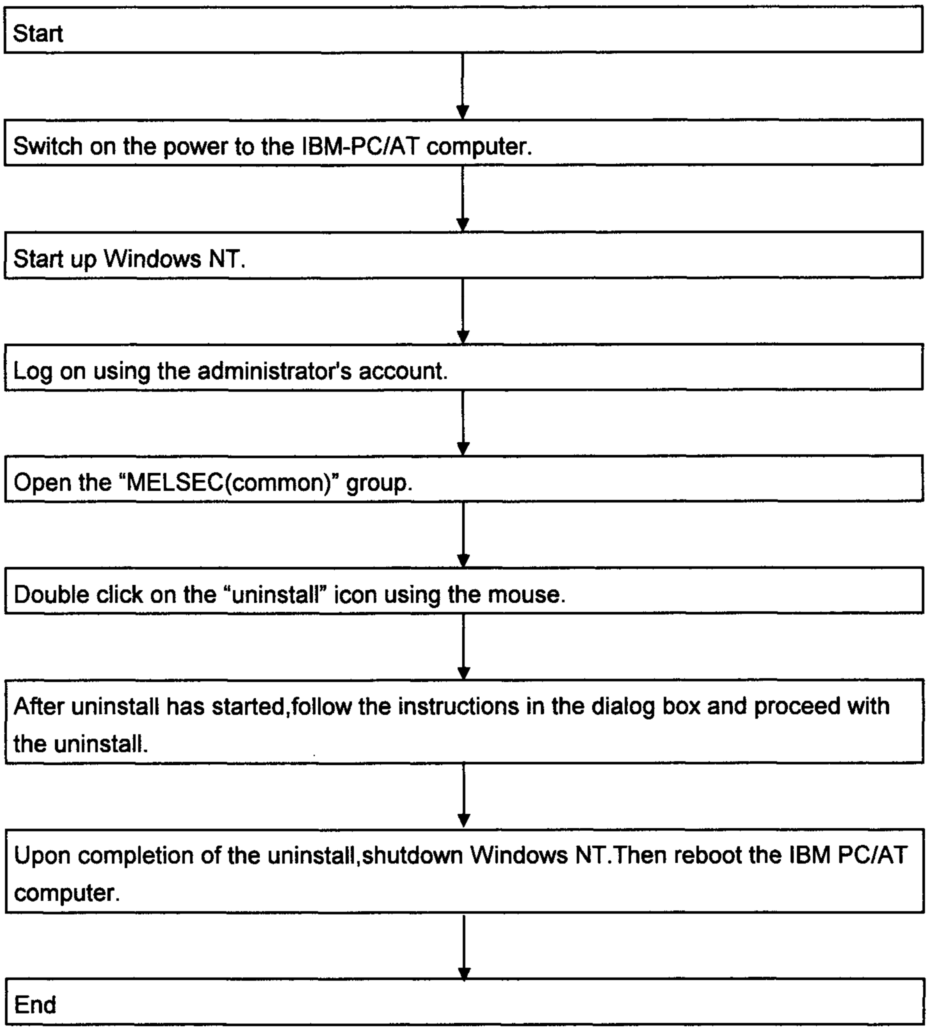
The Directories configuration after the Windows NT driver has been installed is shown below.

```
C:\ ──┬──<MELSEC>──┬──<MNET10W>──┬── System file
      │            │             │
      │            │             ├── <UTL> ─────── System file
      │            │             │
      │            │             └── <SAMPLES> ─┬─ <VB> ──┬──<MTEST>───MTEST.MAK
      │            │                            │         │            MTEST.FRM
      │            │                            │         │            MTEST.FRX
      │            │                            │         │
      │            │                            │         └──<DEMO>────DEMO.FRM
      │            │                            │                      DEMOPRO.MAK
      │            │                            │
      │            │                            └─ <VC>────<MTEST>─────NETSMP1.C
      │            │                                                   NETSMP1.EXE
      │            │
      │            ├──<NETBDCHK>───System file
      │            │
      │            ├──<LIB>────────System file
      │            │
      │            └──<INCLUDE>────System file
      │
      └──<WINNT35>──┬──<SYSTEM32> ─┬─System file
                    │              │
                    │              └─<Drivers> ────────── System file
                    │
                    └──<SYSTEM>─────System file
```

5.2.3 Deletion of Windows NT Driver

The steps for deleting the Windows NT driver are outlined below.

(1) for Windows NT 3.51



Start

Switch on the power to the IBM-PC/AT computer.

Start up Windows NT.

Log on using the administrator's account.

Open the “MELSEC(common)” group.

Double click on the “uninstall” icon using the mouse.

After uninstall has started,follow the instructions in the dialog box and proceed with the uninstall.

Upon completion of the uninstall,shutdown Windows NT.Then reboot the IBM PC/AT computer.

End

(2) for Windows NT 4.0

Start

Switch on the power to the IBM-PC/AT computer.

Start up Windows NT.

Log on using the administrator's account.

Select “Settings” from the start button,and then select “Control panel”.

Double click on “Add/Remove Programs”.

Select “SWnIVDWT-MNET10P” from the install/uninstall.

Click on “Add/Remove”.

After uninstall has started,follow the instructions in the dialog box and proceed with the uninstall.

Upon completion of the uninstall,shutdown Windows NT.Then reboot the IBM PC/AT computer.

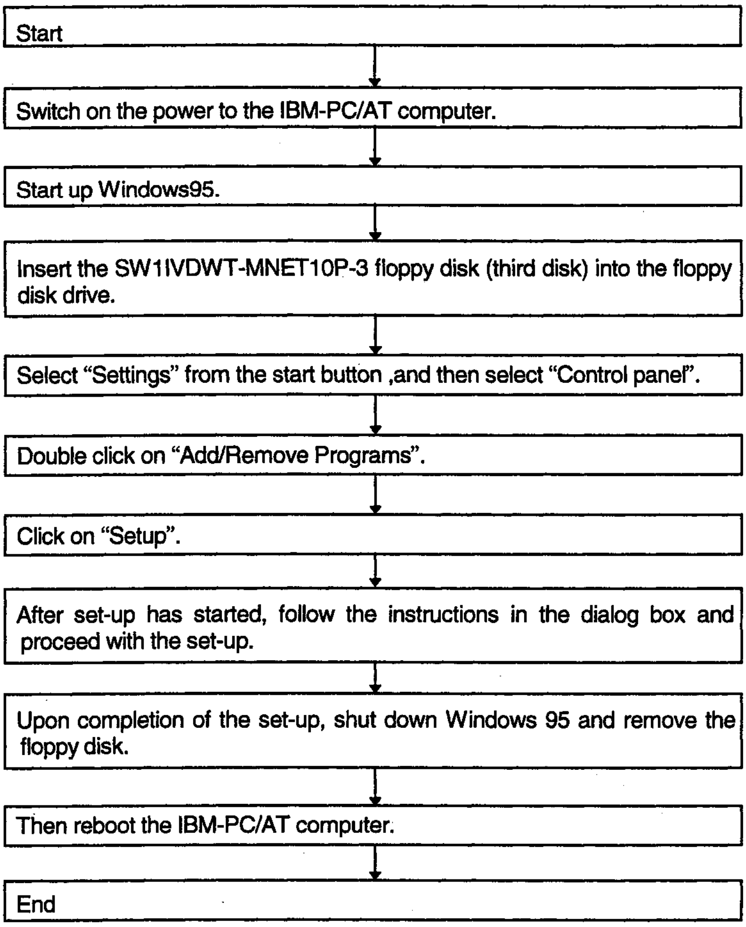
End

5.3 Installation/Deletion of Windows 95 Driver

This section describes how to install and deletion the driver for Windows95.

5.3.1 Installation of Windows 95 Driver

Follow the procedure below to install the Windows95 driver.



Start

Switch on the power to the IBM-PC/AT computer.

Start up Windows95.

Insert the SW1IVDWT-MNET10P-3 floppy disk (third disk) into the floppy disk drive.

Select “Settings” from the start button ,and then select “Control panel”.

Double click on “Add/Remove Programs”.

Click on “Setup”.

After set-up has started, follow the instructions in the dialog box and proceed with the set-up.

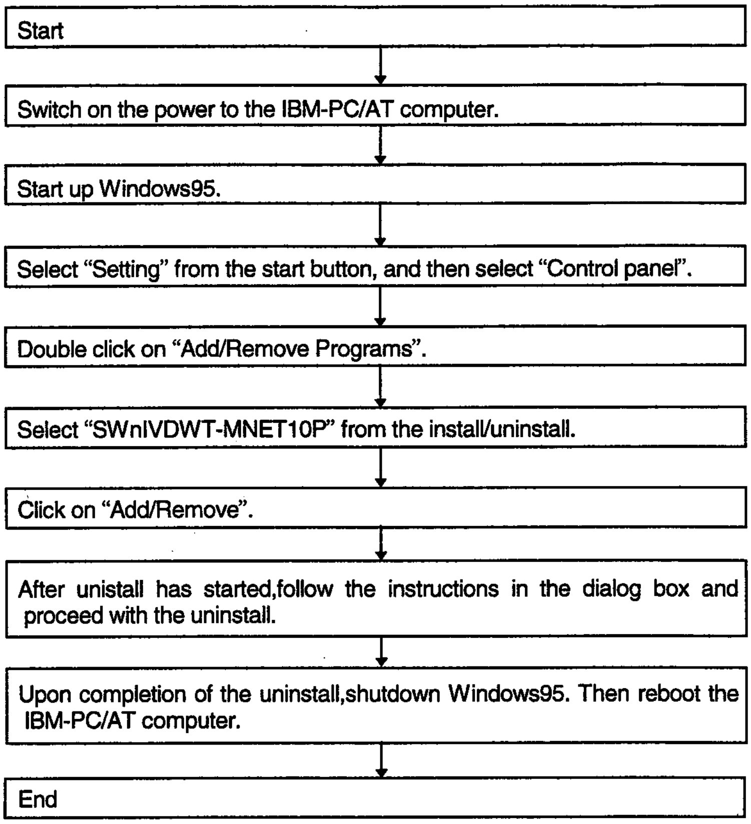
Upon completion of the set-up, shut down Windows 95 and remove the floppy disk.

Then reboot the IBM-PC/AT computer.

End

5.3.2 Deletion of Windows 95 Driver

The steps for deleting the Windows95 driver are outlined below.



Start

Switch on the power to the IBM-PC/AT computer.

Start up Windows95.

Select “Setting” from the start button, and then select “Control panel”.

Double click on “Add/Remove Programs”.

Select “SWnIVDWT-MNET10P” from the install/uninstall.

Click on “Add/Remove”.

After unistall has started,follow the instructions in the dialog box and proceed with the uninstall.

Upon completion of the uninstall,shutdown Windows95. Then reboot the IBM-PC/AT computer.

End

6 HOW TO USE THE UTILITIES

By installing SW0IVDWT-MNET10P, it is possible to use one of the following utilities depending on the operating system used.

- DOS MELSECNET/10 utility
- DOS board diagnosis utility
- Windows NT MELSECNET/10 utility
- Windows NT board diagnosis utility

6.1 How To Use DOS MELSECNET/10 Utility

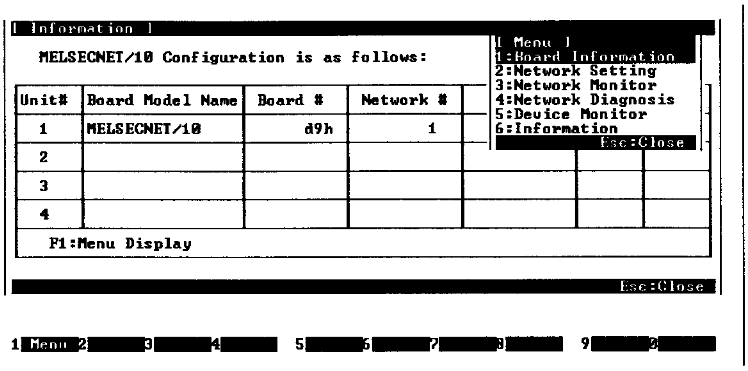
This section describes how the DOS MELSECNET/10 utility is used.

6.1.1 Utility Startup and Exit

Follow the procedures below to start up and exit from the MELSECNET/10 utility.

<Procedure for starting up utility>

1. Change the current directory to \MELSECNET\MNET10X\UTL.
2. Start MNET10PR.EXE from the MS-DOS command prompt.
3. The MELSECNET/10 utility starts up, and a menu screen now appears.



[Information]

MELSECNET/10 Configuration is as follows:

Unit#	Board Model Name	Board #	Network #
1	MELSECNET/10	d9h	1
2			
3			
4			

F1:Menu Display

<Procedure for exiting from utility>

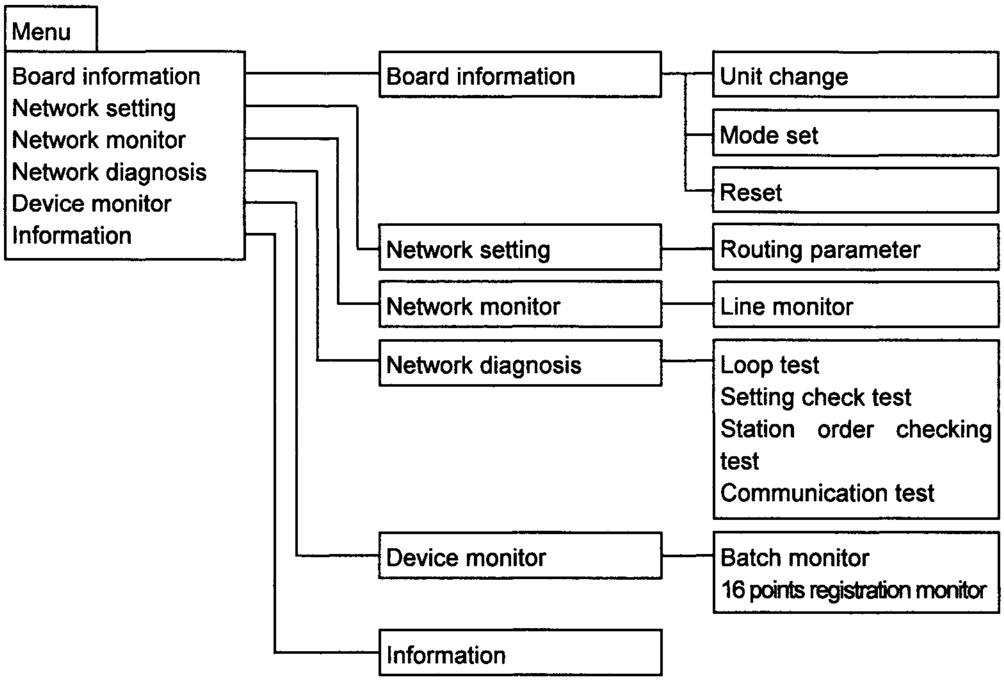
1. Press the ESC key until the utility screen is cleared.
2. The MELSECNET/10 utility is exited and the DOS prompt appears.

6.1.2 Function List

The functions of the MELSECNET/10 utility are listed below.

Function	Description	Reference Item
Information	Displays the setting statuses of the board. · Network No., station No., group No., board No., channel	Item 6.1.8
Board information	Displays the statuses of the MELSECNET/10 interface board. · Board LED displays · Network No., group No., station No. · Mode status Sets the mode and resets the board.	Item 6.1.3
Network setting	Sets the routing parameters.	Item 6.1.4
Network monitor	Displays the own station communication status, link scan time, settings, and error data. Displays the information on each station. · Communication status, link status	Item 6.1.5
Network diagnosis	Conducts network diagnosis. · Loop test · Setting check test · Station order checking test · Communication test	Item 6.1.6
Device monitor	Monitors the own and other station devices. Writes data into the own and other station devices. · Changes, continuous changes, set, reset	Item 6.1.7

Shown below is a diagram of the menus available with the MELSECNET/10 utility.



6.1.3 Board Information

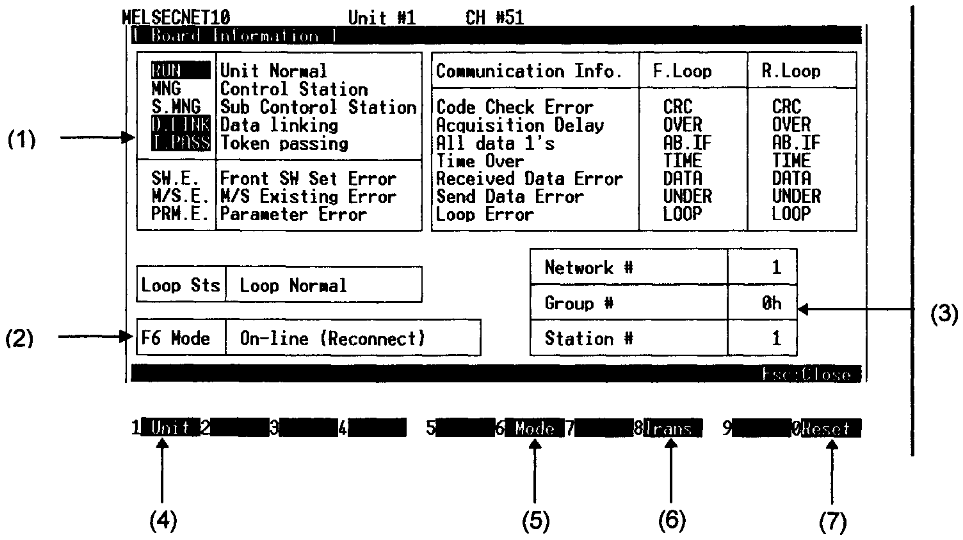
This screen has the following functions.

- Board setting status display
- Mode setting

[How to display the board information screen]

1. Start up utility.
2. Display board information screen.
3. Select Menu followed by the Board information command.

[How to read the screen]



MELSECNET10 Unit #1 CH #51

[Board Information]

RUN	Unit Normal
MNG	Control Station
S.MNG	Sub Contorol Station
D.LINK	Data linking
T.PASS	Token passing
SW.E.	Front SW Set Error
M/S.E.	M/S Existing Error
PRM.E.	Parameter Error

Communication Info.	F.Loop	R.Loop
Code Check Error	CRC	CRC
Acquisition Delay	OVER	OVER
All data 1's	AB.IF	AB.IF
Time Over	TIME	TIME
Received Data Error	DATA	DATA
Send Data Error	UNDER	UNDER
Loop Error	LOOP	LOOP

Loop Sts	Loop Normal
F6 Mode	On-line (Reconnect)

Network #	1
Group #	0h
Station #	1

(1) The board and loop statuses appear in this area .

(2) The mode status set in the board appears in this area.

(3) The network, group and station numbers set in the board appear in this area.

(4) This is used to select the unit (interface board) whose board information is to be displayed.

(5) This is used to change the mode setting.

(6) Transfer the routing parameters to the board.

(7) This is used to reset the board whose board information is now displayed.

CAUTION

When re-testing after an error has occurred, first remove the cause of the trouble, then reset the error display using board reset, and finally conduct the self-loopback test again.

6.1.4 Network Settings

This screen has the following functions.

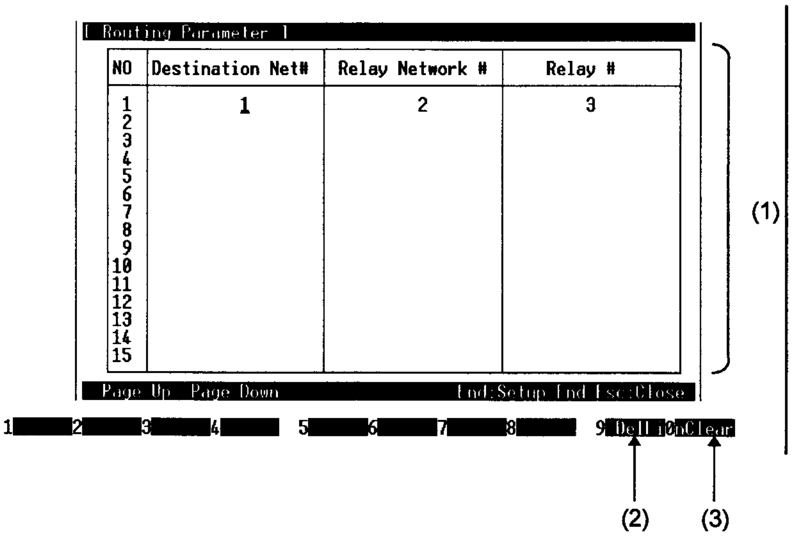
- Routing parameter setting

[How to display the routing parameter setting screen]

1. Start up utility.
2. Display routing parameter setting screen.
3. Select Menu followed by the Network setting command.

[How to read the screen]

Routing parameters



[Routing Parameter]

NO	Destination Net#	Relay Network #	Relay #
1	1	2	3
2			
3			
4			
5			
6			
7			
8			
9			
10			
11			
12			
13			
14			
15			

(1) Routing parameter setting
The destination network numbers, relay network numbers and relay station numbers are set on this screen. For details, refer to the MELSECNET/10 Network System (PC to PC network) Reference Manual.

(2) This is used to delete the routing parameter on the line concerned.

(3) This is used to clear all the routing parameters.

6.1.5 Network Monitor

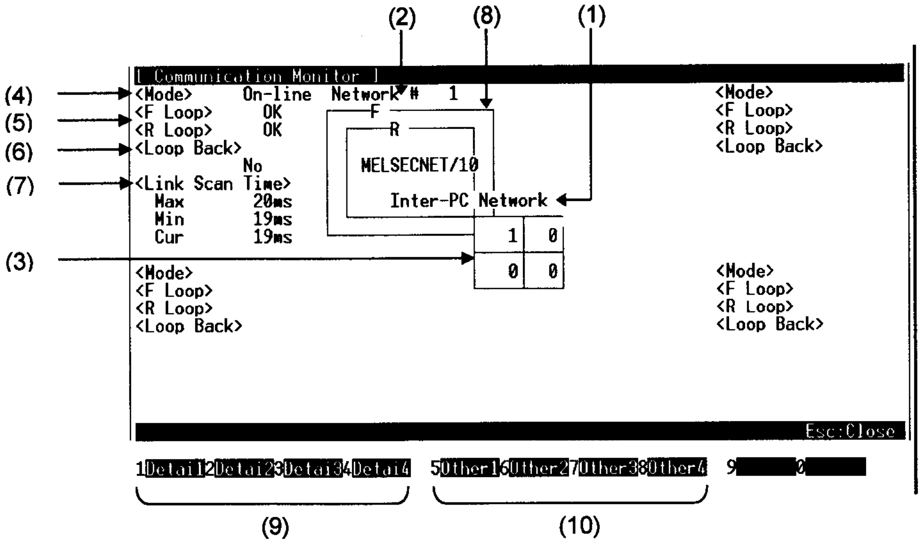
This screen has the following functions.

- Own station operating mode display
- Own station communication status display

[How to display the network monitor screen]

1. Start up utility.
2. Display network monitor screen.
3. Select Menu followed by the Network monitor command.

[How to read the screen]



```
[ Communication Monitor ]
<Mode>       On-line  Network #  1                     <Mode>
<F Loop>      OK        F                              <F Loop>
<R Loop>      OK         R                             <R Loop>
<Loop Back>                                            <Loop Back>
              No        MELSECNET/10
<Link Scan Time>
  Max        20ms        Inter-PC Network
  Min        19ms
  Cur        19ms                     1   0
                                      0   0
<Mode>                                                 <Mode>
<F Loop>                                               <F Loop>
<R Loop>                                               <R Loop>
<Loop Back>                                            <Loop Back>

                                                     Esc:Close
1Detail12Detail23Detail34Detail4 5Other16Other27Other38Other4 9      0
```

(1) Loop status display
The current loop status is displayed here.

(2) Network type display
The inter-PC network is displayed here.

(3) Network No. display

(4) Station No. display
The number of the own station is displayed here.

(5) Operating mode display
(a) Either "On-line" or "Off-line" is displayed as the operating mode of the own station.
(b) The highlighted area indicates the current status.

(6) Loop line status displays
(a) Whether the forward (F) loop line and reverse (R) loop line of the own station are functioning properly is displayed.
(b) If the line is functioning properly, "OK" is displayed; otherwise, "NG" is displayed.

(7) Loopback implementation status display

(a) Whether loopback is implemented by the own station or not is displayed.

(b) The highlighted area indicates the current status.

(8) Link scan time display

The link scan times between the control station and all nomal stations are displayed here.

- Max ······· Indicates the maximum link scan time.
- Min ········ Indicates the minimum link scan time.
- Cur ········ Indicates the current link scan time.

(9) The detailed data is displayed here.

F1 (Detai 1)····· This displays the detailed data of the first interface board.

F2 (Detai 2)····· This displays the detailed data of the second interface board.

F3 (Detai 3)····· This displays the detailed data of the third interface board.

F4 (Detai 4)····· This displays the detailed data of the fourth interface board.

(10) The monitoring conditions of the other stations in each network are displayed here.

F5 (Other 1) ···· This displays the monitoring conditions of the other stations connected to the first interface board.

F6 (Other 2) ···· This displays the monitoring conditions of the other stations connected to the second interface board.

F7 (Other 3) ···· This displays the monitoring conditions of the other stations connected to the third interface board.

F8 (Other 4) ···· This displays the monitoring conditions of the other stations connected to the fourth interface board.

6.1.6 Network Diagnosis

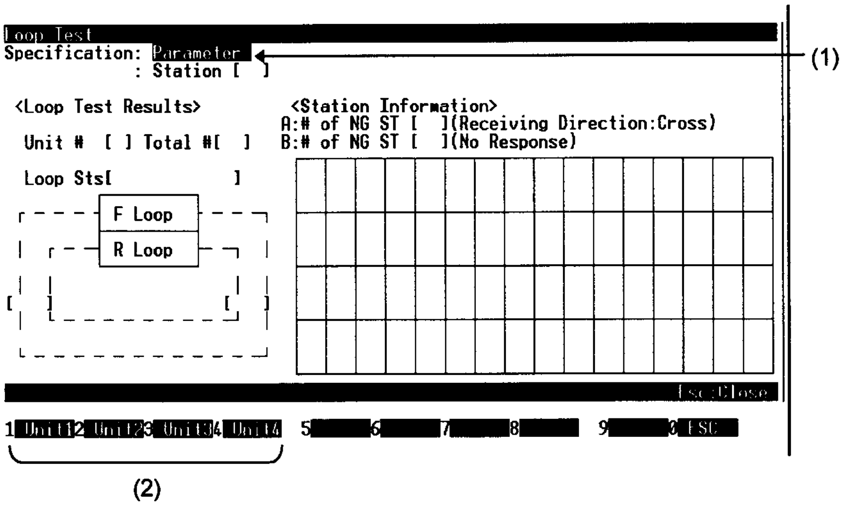
This screen has the following functions.

- Loop test
- Setting check test
- Station order checking test
- Communication test

[How to display the loop test screen]

1. Start up utility.
2. Display loop test screen.
3. Select Menu followed by the Network diagnosis and Loop test commands.

[How to read the screen]



```
Loop Test
Specification: Parameter
             : Station [  ]

 <Loop Test Results>          <Station Information>
                             A:# of NG ST [  ](Receiving Direction:Cross)
  Unit #  [ ] Total #[  ]    B:# of NG ST [  ](No Response)

  Loop Sts[             ]
```

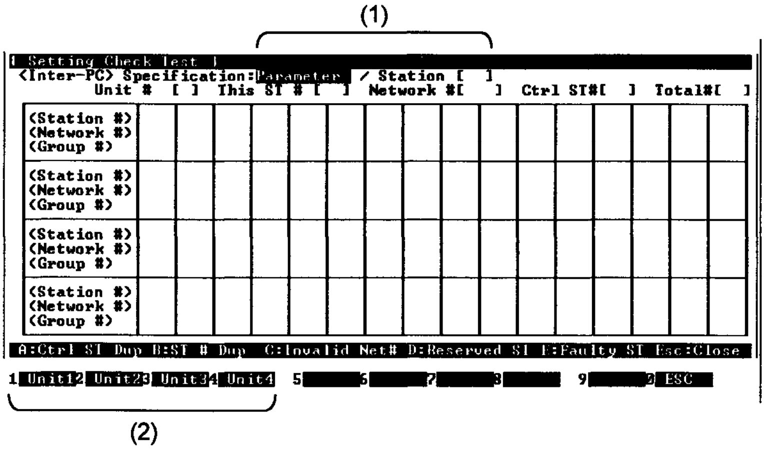
(1) Test implementation assignments
The number of stations for which the loop test is to be implemented is assigned here.
- Parameter: When a parameter has been assigned, the test is conducted on all the stations up to the total number of link stations set in the parameter of the control station.
- Station: When station numbers have been assigned, the test is conducted on the assigned number of stations.

(2) Assignment of interface board to be tested
F1 (Unit 1)······Test is conducted on the network connected to the first interface board.
F2 (Unit 2)······Test is conducted on the network connected to the second interface board.
F3 (Unit 3)······Test is conducted on the network connected to the third interface board.
F4 (Unit 4)······Test is conducted on the network connected to the fourth interface board.

[How to display the setting check test screen]

1. Start up utility.
2. Display setting check test screen.
3. Select Menu followed by the Network diagnosis and Setting check test commands.

[How to read the screen]



```
[ Setting Check Test ]
 <Inter-PC> Specification:Parameter / Station [  ]
          Unit #  [ ]  This ST # [  ]  Network #[    ]  Ctrl ST#[  ]  Total#[  ]
 <Station #>
 <Network #>
 <Group #>
 <Station #>
 <Network #>
 <Group #>
 <Station #>
 <Network #>
 <Group #>
 <Station #>
 <Network #>
 <Group #>
 A:Ctrl ST Dup B:ST # Dup  C:Invalid Net# D:Reserved ST E:Faulty ST Esc:Close
1 Unit1 2 Unit2 3 Unit3 4 Unit4  5      6      7      8      9      0 ESC
```

(1) Test implementation assignments
The number of stations for which the setting check test is to be implemented is assigned here.

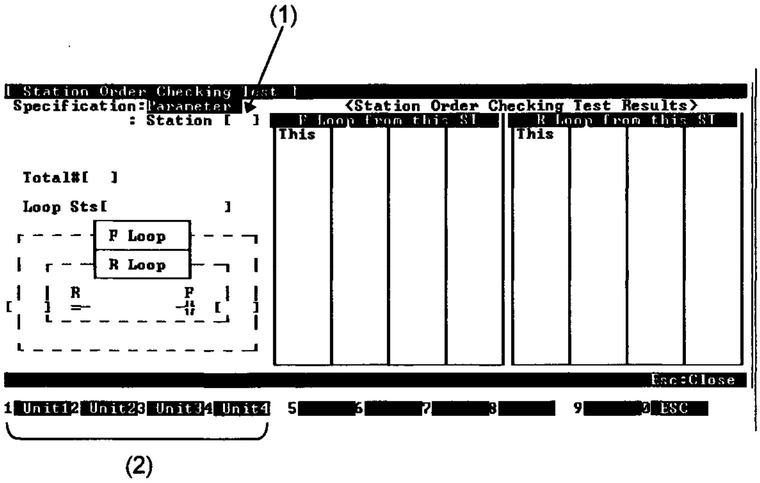
- Parameter: When a parameter has been assigned, the test is conducted on all the stations up to the total number of link stations set in the parameter of the control station.
- Station: When station numbers have been assigned, the test is conducted on the assigned number of stations.

(2) Assignment of interface board to be tested
F1 (Unit 1) · · · · · · Test is conducted on the network connected to the first interface board.
F2 (Unit 2) · · · · · · Test is conducted on the network connected to the second interface board.
F3 (Unit 3) · · · · · · Test is conducted on the network connected to the third interface board.
F4 (Unit 4) · · · · · · Test is conducted on the network connected to the fourth interface board.

[How to display the station order checking test screen]

1. Start up utility.
2. Display station order checking test screen.
3. Select Menu followed by the Network diagnosis and Station order checking test commands.

[How to read the screen]



```
[ Station Order Checking Test ]
 Specification:Parameter          <Station Order Checking Test Results>
              : Station [  ]   F Loop from this ST      R Loop from this ST
                               This                     This
  Total#[  ]

  Loop Sts[          ]
                                                                  Esc:Close
1 Unit1 2 Unit2 3 Unit3 4 Unit4  5   6   7   8   9   0 ESC
```

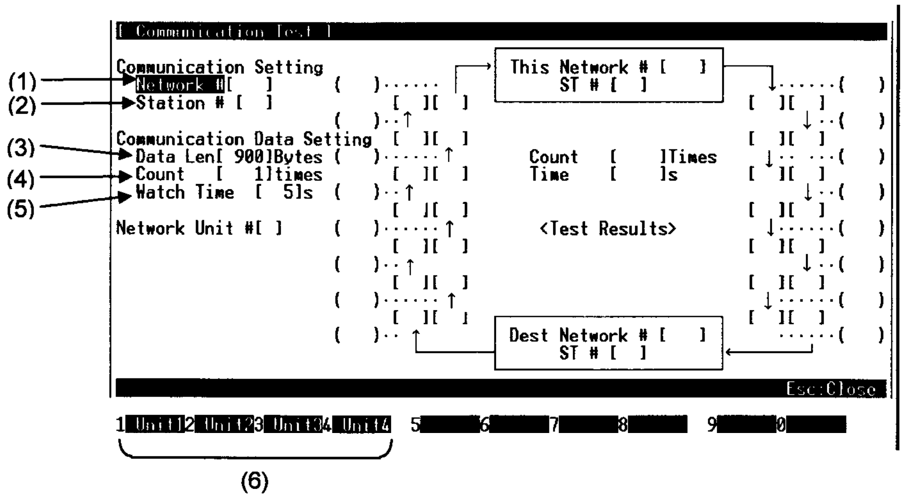
(1) Test implementation assignments
The number of stations for which the station order checking test is to be implemented is assigned here.
- Parameter: When a parameter has been assigned, the test is conducted on all the stations up to the total number of link stations set in the parameter of the control station.
- Station: When station numbers have been designated, the test is conducted on the designated number of stations.

(2) Assignment of interface board to be tested
F1 (Unit 1)······Test is conducted on the network connected to the first interface board.
F2 (Unit 2)······Test is conducted on the network connected to the second interface board.
F3 (Unit 3)······Test is conducted on the network connected to the third interface board.
F4 (Unit 4)······Test is conducted on the network connected to the fourth interface board.

[How to display the communication test screen]

1. Start up utility.
2. Display communication test screen.
3. Select Menu followed by the Network diagnosis and Communication test commands.

[How to read the screen]



```
Communication Setting
 Network #[    ]
 Station # [  ]
Communication Data Setting
 Data Len[ 900]Bytes
 Count    [   1]times
 Watch Time  [  5]s
Network Unit #[ ]

Count    [      ]Times
Time     [      ]s
```

(1) Network number
The network number of the station assigned as the communication destination is set here.

(2) Station number
The station number of the station assigned as the communication destination is set here.

(3) Data length
The data length (in bytes) ranging from 1 to 900 bytes of the test message is set here.
Default: 100 bytes

(4) Communication count
A communication count ranging from 1 to 100 times for the test message is set here.
Default: 1 time

(5) Communication monitor (watch) time
The length of time during which the response message is monitored, ranging from 1 to 100 seconds, is set here.
Default: 5 sec.

(6) Assignment of interface board to be tested
F1 (Unit 1)······ Test is conducted on the network connected to the first interface board.
F2 (Unit 2)······ Test is conducted on the network connected to the second interface board.
F3 (Unit 3)······ Test is conducted on the network connected to the third interface board.
F4 (Unit 4)······ Test is conducted on the network connected to the fourth interface board.

6.1.7 Device Monitor

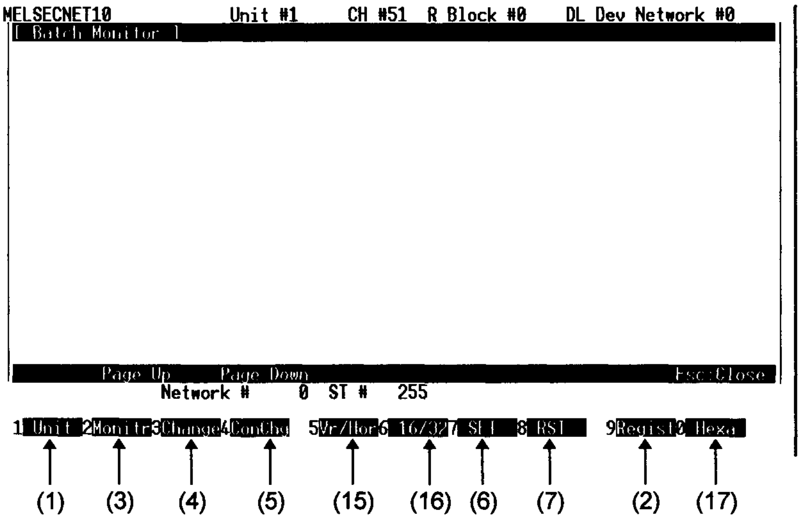
This screen has the following functions.

- Device setting information display
- Device entry
- Device data change
- Continuous device data change
- Bit device set
- Bit device reset
- Display switch

[How to display the device monitor screen]

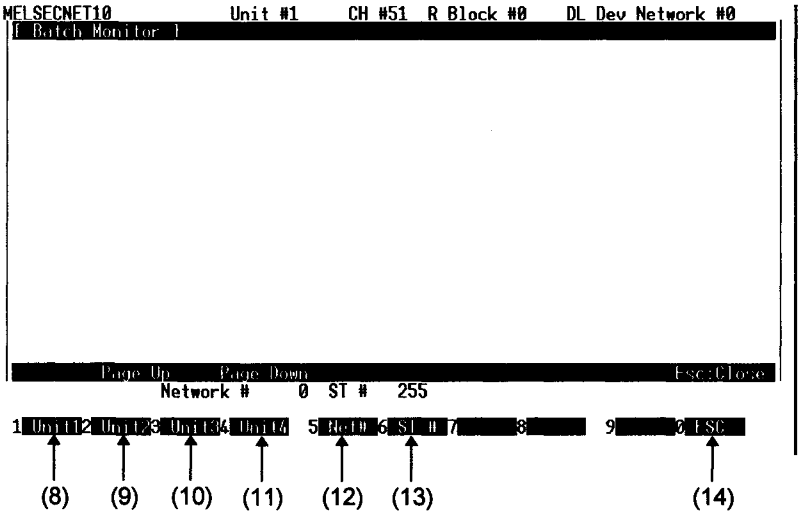
1. Start up utility.
2. Display menu screen.
3. Select Menu followed by the Device monitor command.

[How to read the screen]



```
MELSECNET10            Unit #1      CH #51   R Block #0     DL Dev Network #0
[ Batch Monitor ]

         Page Up      Page Down                                      Esc:Close
                    Network #     0  ST #   255
```



```
MELSECNET10            Unit #1      CH #51   R Block #0     DL Dev Network #0
[ Batch Monitor ]

         Page Up      Page Down                                      Esc:Close
                    Network #     0  ST #   255
```

(1) Monitor destination unit select
This selects the unit at the monitor destination.

(2) This is used to display the name of the device to be monitored, the name of the head device, and the column in which the names are assigned.

(3) This is used to start the monitoring.

(4) This is used to display the number of the word device whose value is to be changed and the column in which the setting is assigned.

(5) This is used to continuously change the data from the assigned number of words for the assigned number of points to the same value.
By pressing the F4 key, the number of the head word device for which the value is to be changed continuously, and the column in which the number of setting points is to be assigned are displayed.

(6) This is used to display the column in which the bit device to be set is assigned.

(7) This is used to display the column in which the bit device to be reset is assigned.

(8), (9), (10), (11) These are used to select the corresponding interface board.

(12) This is used to set the network number.

(13) This is used to set the station number

(14) This is used to return operation to the first screen on which the utility was started up.

(15) This is used to switch between a vertical display and horizontal display. (Bit device only)

(16) This is used to switch between a 16-bit display and 32-bit display. (Word device only)

(17) This is used to switch between a decimal, hexadecimal, binary and octal display. (Word device only)

6.1.8 Information

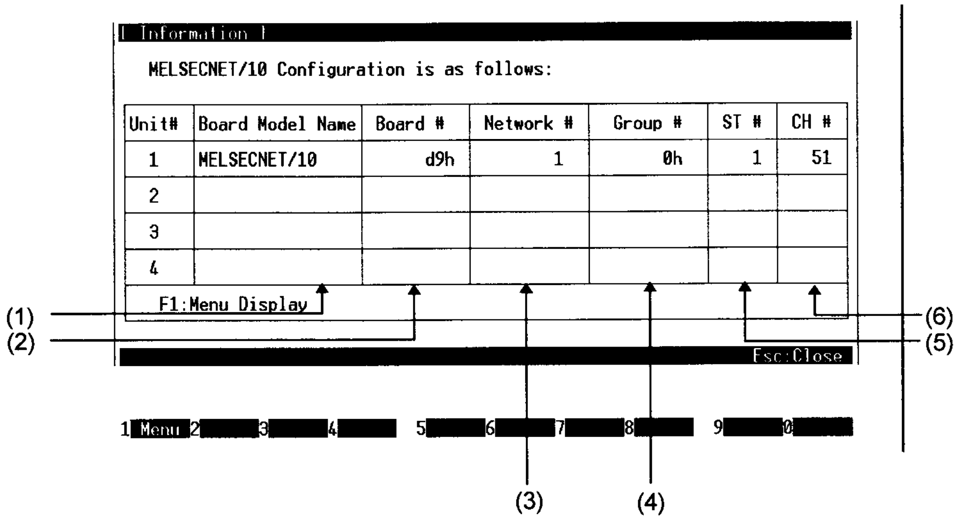
This screen has the following functions.

- Display of rotary switches and DIP switches which have been set on the boards

[How to display the board information screen]

1. Start up utility.
2. Display board information screen.

[How to read the screen]



[Information]

MELSECNET/10 Configuration is as follows:

Unit#	Board Model Name	Board #	Network #	Group #	ST #	CH #
1	MELSECNET/10	d9h	1	0h	1	51
2						
3						
4						

F1:Menu Display

Esc:Close

(1) The model names of the boards are displayed here.

(2) The board numbers are displayed here.

(3) The network numbers are displayed here.

(4) The group numbers are displayed here.

(5) The station numbers are displayed here.

(6) The channel numbers are displayed here.

CHECKPOINTS

The current network status is displayed as follows:

- When the host (current board) is specified... Network No.:0, Station No.: 255
- When other station is specified..... Network No.: *1, Station No.: *1

*1: "Network No." and "Station No." preset in the network setting are displayed.

6.2 How To Use DOS Board Diagnosis Utility

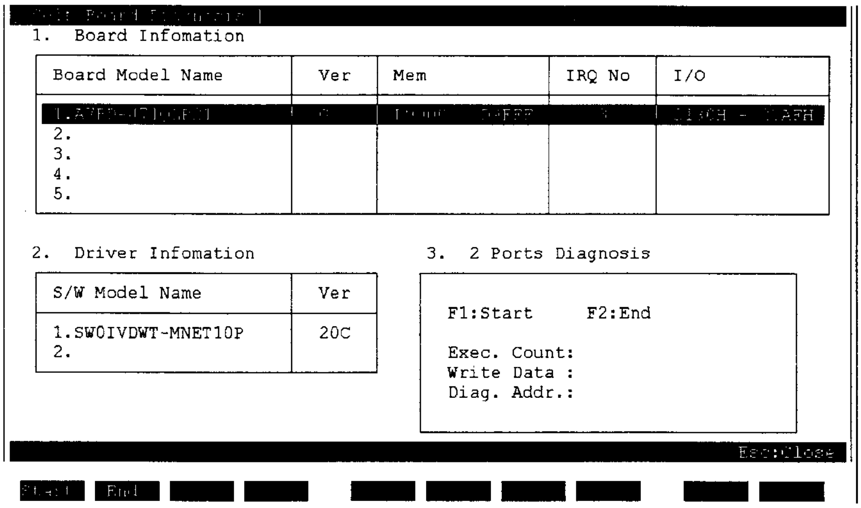
This section describes how the DOS board diagnosis utility is used.

6.2.1 Utility Startup and Exit

Follow the procedures below to start up and exit from the board diagnosis utility.

<Procedure for starting up utility>

1. Change the current directory to \MELSEC\NETBDCHK.
2. Start NETBDCHK.EXE from the MS-DOS command prompt.
3. The board diagnosis utility starts up, and the board self-diagnosis screen now appears.



```
[illegible] ]
1.  Board Infomation

Board Model Name      Ver    Mem                IRQ No    I/O
1.[illegible]         [illegible]  [illegible]  [illegible]  [illegible]H - [illegible]AFH
2.
3.
4.
5.

2.  Driver Infomation                 3.  2 Ports Diagnosis

S/W Model Name        Ver              F1:Start      F2:End
1.SW0IVDWT-MNET10P    20C
2.                                     Exec. Count:
                                       Write Data :
                                       Diag. Addr.:

                                                          Esc:Close
```

<Procedure for exiting from utility>

1. Press the ESC key until the utility screen is cleared.
2. The board diagnosis utility is exited and the DOS prompt appears.

6.2.2 Function List

The functions of the board diagnosis utility are listed below.

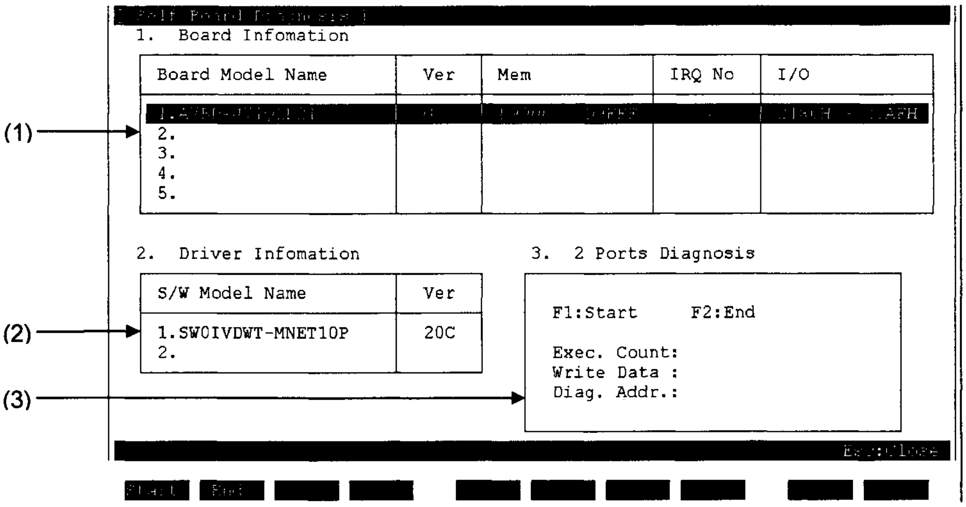
Function	Description	Reference Item
Board information	Displays the board model names, versions, 2-port memories used, etc.	Item 6.2.3
Driver information	Displays the software program name and version.	
2-port diagnosis	Diagnoses the 2-port memory of the board.	

6.2.3 Board Self-Diagnosis

This screen has the following functions.

- Board information
- Driver information
- 2-port memory diagnosis

[How to read the screen]



[Self Board Diagnosis]

1. Board Infomation

Board Model Name	Ver	Mem	IRQ No	I/O
1.[illegible]	[illegible]	[illegible]	[illegible]	[illegible]H - [illegible]AFH
2.				
3.				
4.				
5.				

2. Driver Infomation

S/W Model Name	Ver
1.SW0IVDWT-MNET10P	20C
2.	

3. 2 Ports Diagnosis

```
F1:Start      F2:End

Exec. Count:
Write Data :
Diag. Addr.:
```

Esc:Close

F1:Start F2:End

1. The board information is displayed in this area.

Board model name:	The model names of the installed boards are displayed here.
Ver:	The hardware versions of the installed boards are displayed here.
Mem:	The 2-port memories used are displayed here.
IRQ No:	The interrupt numbers which have been set are displayed here.
I/O:	The numbers of the memory address setting switches which have been set are displayed here.

2. The driver information is displayed in this area.

S/W Model Name:	The model names of the installed drivers are displayed here.
Ver:	The software versions of the installed drivers are displayed here.

3. The 2-port memory diagnosis is conducted here.
   Diagnosis is started when the F1 key is pressed.
   Diagnosis is ended when the F2 key is pressed.
   * After the 2-port memory diagnosis has been conducted, an illegal value may sometimes appear for a board model name or version.

6.3 How To Use Windows NT Utilities

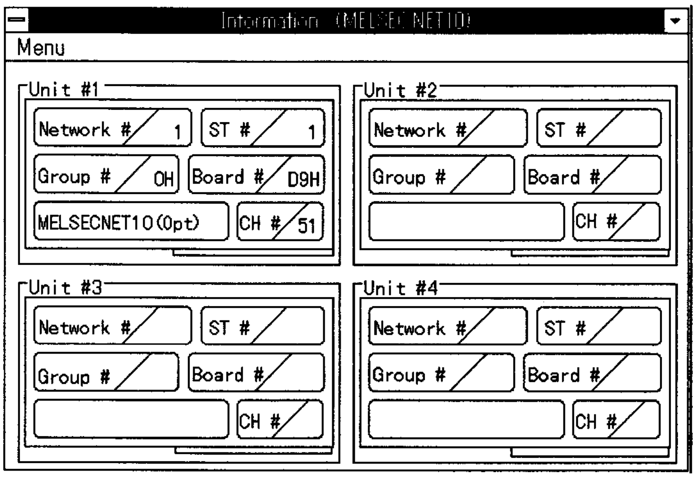
This section describes how the Windows NT MELSECNET/10 utility is used.

6.3.1 Utility Startup and Exit

Follow the procedures below to start up and exit from the MELSECNET/10 utility.

<Procedure for starting up utility>

1. Open the "MELSECNET/10 Utility".
   (1) Windows NT 3.51
      (a) Open the "MELSEC (common)" group.
      (b) Double click on the "MELSECNET/10 Utility" icon using the mouse.
   (2) WindowsNT 4.0
      (a) Select "Programs" from the start button ,and then select "MELSECNET/10 Utility".
2. The MELSECNET/10 utility starts up, and the information screen now appears.



Information (MELSEC NET10)

Unit #1
Network # 1
ST # 1
Group # 0H
Board # D9H
MELSECNET10(Opt)
CH # 51

<Procedure for exiting from utility>

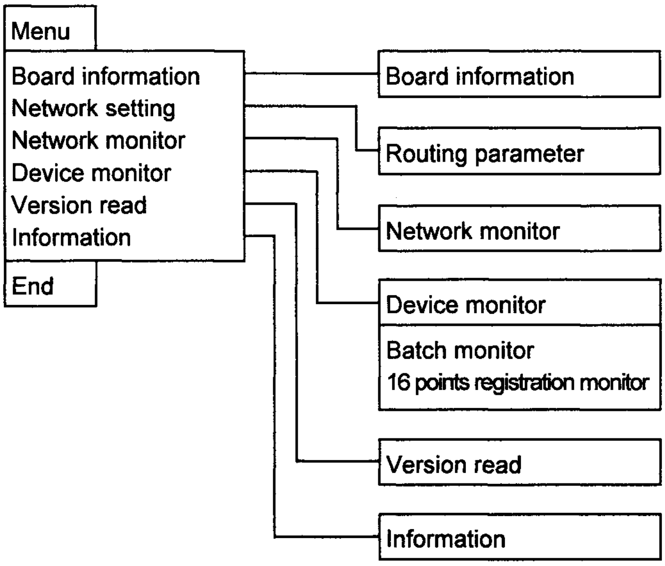
1. Select the Menu and End commands.
2. The MELSECNET/10 utility is exited.

6.3.2 Function List

All the functions of the MELSECNET/10 utility are listed below.

Function	Description	Reference Item
Information	Displays the setting statuses of the boards connected. · Station No., network No., group No., board No., channel, board model name	Item 6.3.3
Board information	Displays the statuses of the MELSECNET/10 interface board. · Board LED displays · Board No., channel, station No. · Mode status Sets the mode and resets the board.	Item 6.3.4
Network monitor (own station information display)	Displays the own station communication status, settings, and error data.	Item 6.3.5
Network monitor (station's communication status)	Displays the information of each station. · Communication status	Item 6.3.6
Network monitor (station's kink status)	Displays the information of each station. · Link status	Item 6.3.7
Batch device monitor	Monitors the own and other station devices. Writes data into the own and other station devices. · Changes, continuous changes, set, reset	Item 6.3.8
Device 16 points registration monitor	Monitors the own and other station devices. Writes data into the own and other station devices. · Changes, continuous changes, set, reset	Item 6.3.9
Version read	Displays the version information of the boards and packages.	Item 6.3.10

Shown below is a diagram of the menus available with the MELSECNET/10 utility.



6.3.3 Information

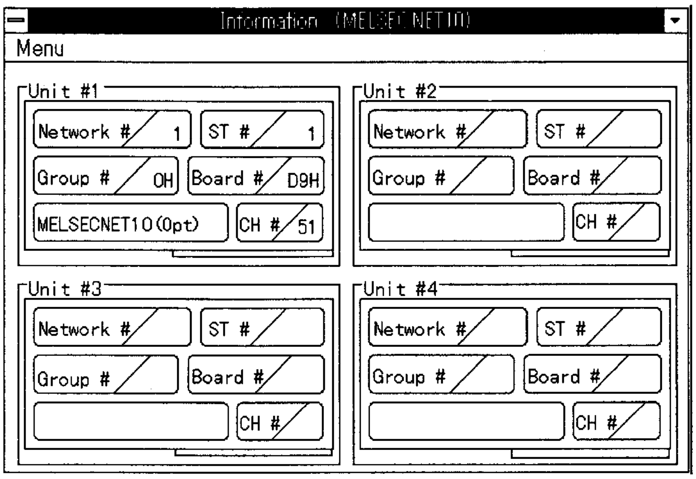
This screen has the following functions.

- Board setting status display

[How to display the information screen]

1. Start up utility.
2. Display information screen.

[How to read the screen]



Information (MELSEC NET10)

Menu

Unit #1
Network # 1
ST # 1
Group # 0H
Board # D9H
MELSECNET10(Opt)
CH # 51

Unit #2
Network #
ST #
Group #
Board #
CH #

Unit #3
Network #
ST #
Group #
Board #
CH #

Unit #4
Network #
ST #
Group #
Board #
CH #

1. The setting statuses for as many boards as those connected are displayed on this screen.

[Commands]

Menu: For moving to another screen.

6.3.4 Board information

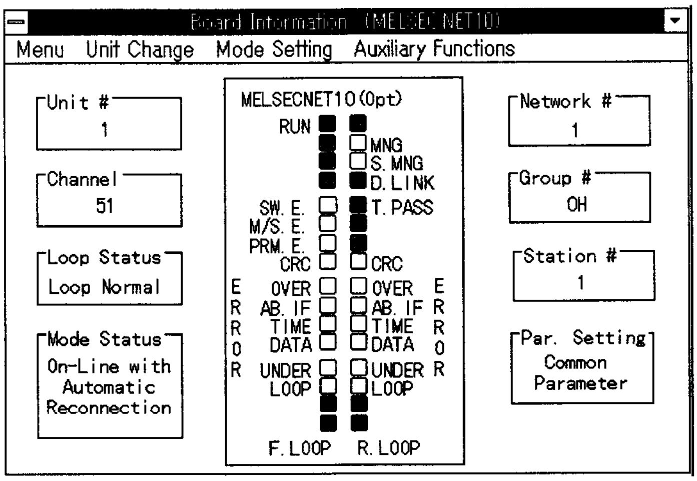
This screen has the following functions.

- Network status check
- Board mode setting
- Board resetting

[How to display the board information screen]

1. Start up utility.
2. Display information screen.
3. Select Menu followed by the Board information command.

[How to read the screen]



Unit #: 1

Channel: 51

Loop Status: Loop Normal

Mode Status: On-Line with Automatic Reconnection

Network #: 1

Group #: 0H

Station #: 1

Par. Setting: Common Parameter

1. The unit number, channels, loop status, mode status, network number, group number, station number and parameter settings are displayed on this screen.
2. The current statuses of the unit are displayed by LED images. To find out the significance of a particular LED, a description can be displayed in Japanese using the auxiliary functions and term explanation commands.

[Commands]

Menu: For moving to another screen.

Unit change: For displaying the information of other units.

Mode setting: For setting the board mode. When the test mode has been selected, the test results can be ascertained from the screen display.
OK: ·············· The LED image display flashes.
NG: ·············· The LED image display stops flashing.

Auxiliary functions: Board reset: ······ For resetting and initializing the board.
Term explanation: · For displaying the description of an LED image in Japanese.

6.3.5 Network Monitor (Own Station Information Display)

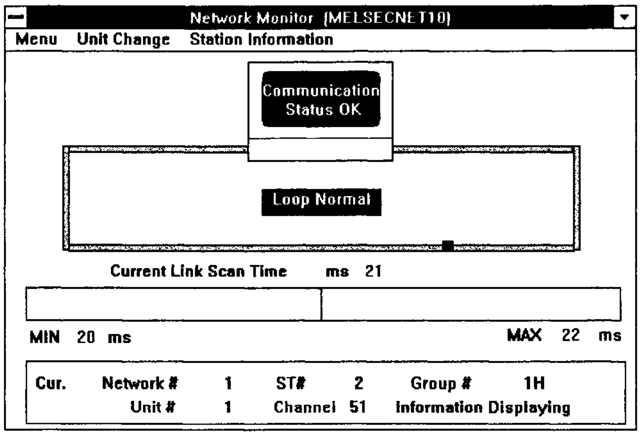
This screen has the following functions.

- Own station communication status display
- Network status check
- Details of board settings

[How to display the network monitor (own station information display) screen]

1. Start up utility.
2. Display information screen.
3. Select Menu followed by the Network monitor command.

[How to read the screen]



Network Monitor (MELSECNET10)
Menu Unit Change Station Information
Communication Status OK
Loop Normal

Current Link Scan Time ms 21

MIN 20 ms MAX 22 ms

Cur. Network # 1 ST# 2 Group # 1H
Unit # 1 Channel 51 Information Displaying

1. The communication status of the own station is displayed on this screen.
2. Details of the board settings are displayed on this screen.

[Commands]

Menu:	For moving to another screen.
Unit change:	For displaying the information of other units.
Station information:	Station's communication status: · For switching to the screen displaying the communication status of each station.
	Station's link status:············ For switching to the screen displaying the link status of each station.

6.3.6 Network Monitor (Station's Communication Status)

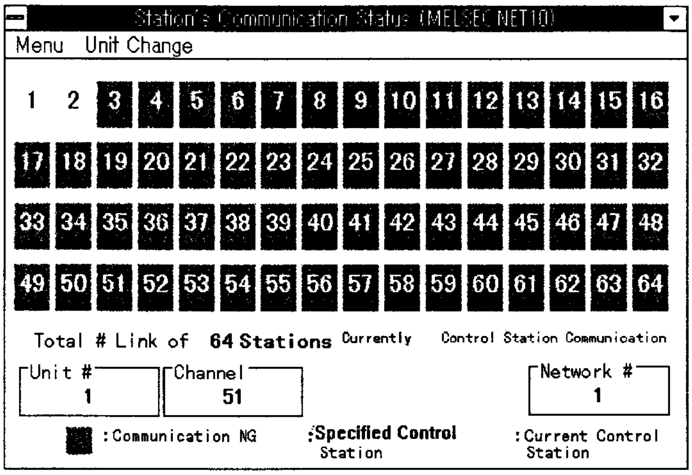
This screen has the following functions.

- Display of communication status of each station

[How to display the network monitor (station's communication status) screen]

1. Start up utility.
2. Display board information screen.
3. Select Menu followed by the Network monitor, Station information, and Station communication status commands.

[How to read the screen]



Station's Communication Status (MELSEC NET10)

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32
33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48
49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64

Total # Link of 64 Stations Currently Control Station Communication

Unit # 1
Channel 51
Network # 1

▪ : Communication NG
:Specified Control Station
: Current Control Station

1. Information of each station is displayed on this screen.
2. The total number of stations is displayed on this screen.
3. The unit number, channel and network number are displayed.

[Commands]

Menu: For moving to another screen.

Unit change: For displaying the information of other units.

6.3.7 Network Monitor (Station's Link Status)

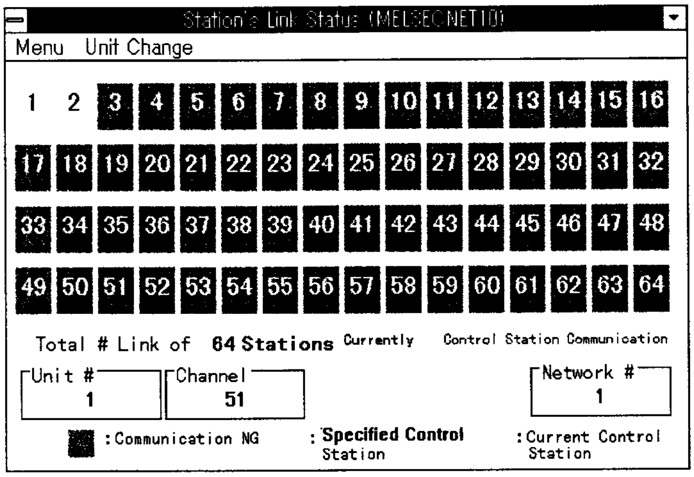
This screen has the following functions.

- Display of link status of each station

[How to display the network monitor (station's link status) screen]

1. Start up utility.
2. Display information screen.
3. Select Menu followed by the Network monitor, Station information, and Station link status commands.

[How to read the screen]



1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32
33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48
49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64

Total # Link of 64 Stations Currently Control Station Communication

Unit # 1

Channel 51

Network # 1

■ : Communication NG : Specified Control Station : Current Control Station

1. Information of each station is displayed on this screen.
2. The total number of stations is displayed on this screen.
3. The unit number, channel and network number are displayed.

[Commands]

Menu: For moving to another screen.

Unit change: For displaying the information of other units.

6.3.8 Device Batch Monitor

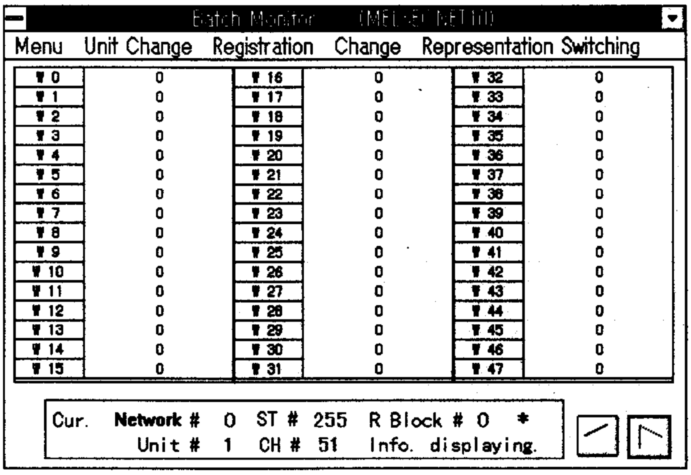
This screen has the following functions.

- Device setting contents display
- Board setting information display
- Device entry
- Monitor station entry
- Device data change
- Device data continuous change
- Bit device setting
- Bit device resetting
- Display switching

[How to display the device batch monitor screen]

1. Start up utility.
2. Display information screen.
3. Select Menu followed by the Device monitor and batch monitor commands.

[How to read the screen]



▼ 0	0	▼ 16	0	▼ 32	0
▼ 1	0	▼ 17	0	▼ 33	0
▼ 2	0	▼ 18	0	▼ 34	0
▼ 3	0	▼ 19	0	▼ 35	0
▼ 4	0	▼ 20	0	▼ 36	0
▼ 5	0	▼ 21	0	▼ 37	0
▼ 6	0	▼ 22	0	▼ 38	0
▼ 7	0	▼ 23	0	▼ 39	0
▼ 8	0	▼ 24	0	▼ 40	0
▼ 9	0	▼ 25	0	▼ 41	0
▼ 10	0	▼ 26	0	▼ 42	0
▼ 11	0	▼ 27	0	▼ 43	0
▼ 12	0	▼ 28	0	▼ 44	0
▼ 13	0	▼ 29	0	▼ 45	0
▼ 14	0	▼ 30	0	▼ 46	0
▼ 15	0	▼ 31	0	▼ 47	0

Cur. **Network #** 0 ST # 255 R Block # 0 *
Unit # 1 CH # 51 Info. displaying.

1. The setting information of the devices currently being monitored is displayed.
2. The information of the boards currently being monitored and the station numbers at monitor destination are displayed.

[Commands]

Menu: For moving to another screen.

Unit change: For displaying the information of other units.

Registration:

- Device registration: ········ For registering the device names and head device numbers.
- Monitor station registration: · For registering the number of the station being monitored and its network number.

Change:

- Data change: ············ For changing the data of a device and saving it in the device.
- Data continuous change: ··· For continuously changing the data of a device and saving it in the device.
- Bit device set:············· For setting a bit device and saving it in the device.
- Bit device reset: ··········· For resetting a bit device and saving it in the device.

Representation switching

- Decimal display: ·········· For switching the display format to decimal notation.
- Hexadecimal display: ······ For switching the display format to hexadecimal notation.

6.3.9 Device 16 Points Registration Monitor

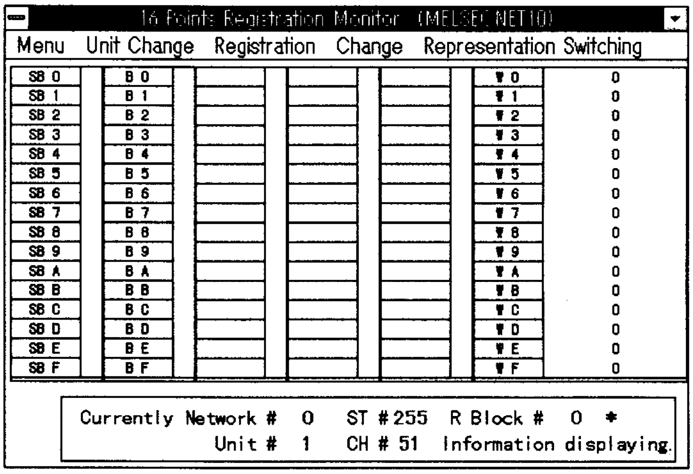
This screen has the following functions.

- Device monitor setting information display
- Board setting information display
- Device entry
- Monitor station entry
- Device data change
- Device data continuous change
- Bit device setting
- Bit device resetting
- Display switching

[How to display the 16 points registration monitor screen]

1. Start up utility.
2. Display information screen.
3. Select Menu followed by the Device monitor and 16 points registration monitor commands.

[How to read the screen]



SB 0		B 0								W 0	0
SB 1		B 1								W 1	0
SB 2		B 2								W 2	0
SB 3		B 3								W 3	0
SB 4		B 4								W 4	0
SB 5		B 5								W 5	0
SB 6		B 6								W 6	0
SB 7		B 7								W 7	0
SB 8		B 8								W 8	0
SB 9		B 9								W 9	0
SB A		B A								W A	0
SB B		B B								W B	0
SB C		B C								W C	0
SB D		B D								W D	0
SB E		B E								W E	0
SB F		B F								W F	0

```
Currently Network #  0   ST #255  R Block #  0  *
             Unit #  1   CH # 51  Information displaying.
```

1. The setting information of the devices currently being monitored is displayed. Word devices are displayed in the column on the right; bit devices are displayed in the 5 columns on the left.
2. The information of the boards currently being monitored and the station numbers at the monitor destination are displayed.

[Commands]

Menu: For moving to another screen.

Unit change: For displaying the information of other units.

Registration:

Device registration: ········ For registering the device names and head device number.

Monitor station registration: · For registering the number of the station being monitored and its network number.

Change:

Data change: ············· For changing the data of a device and saving it in the device.

Data continuous change: ··· For continuously changing the data of a device and saving it in the device.

Bit device set: ············ For setting a bit device and saving it in the device.

Bit device reset: ··········· For resetting a bit device and saving it in the device.

Representation switching

Decimal display: ·········· For switching the display format to decimal notation.

Hexadecimal display: ······ For switching the display format to hexadecimal notation.

6.3.10 Version Read

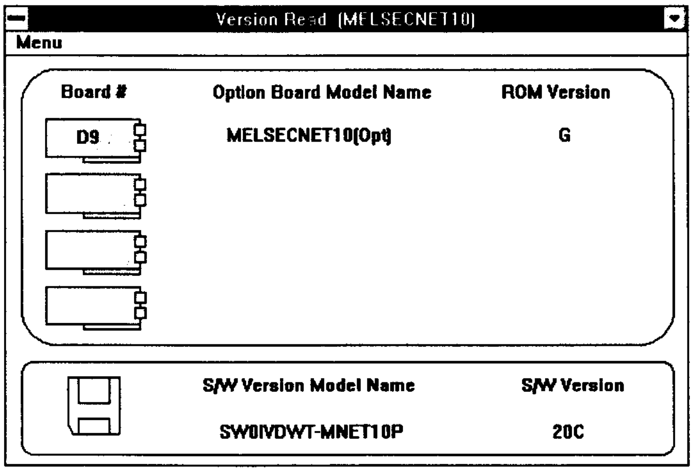
This screen has the following functions.

- Board version information display
- Package version information display

[How to display the version readout screen]

1. Start up utility.
2. Display information screen.
3. Select Menu followed by the Version read command.

[How to read the screen]



Board #	Option Board Model Name	ROM Version
D9	MELSECNET10(Opt)	G

S/W Version Model Name	S/W Version
SW0IVDWT-MNET10P	20C

1. The board number, option board model name and ROM version are displayed.
2. The software version model name and software version are displayed.

[Commands]

Menu: For moving to another screen.

6.4 How To Use Windows NT 3.51/NT 4.0/95 Board Diagnosis Utility

This section describes how to use the Windows NT3.51/NT4.0/95 board diagnosis utility.

6.4.1 Utility Startup and Exit

Follow the procedures below to start up and exit from the board diagnosis utility.

<Procedure for starting up utility>

1. Shut down the MELSECNET/10 driver.
   (1) Windows NT 3.51
      (a) Double click on the "Device" icon of "Control Panel" in the "Main" group.
      (b) Select "MELSECNET/10" from Device.
      (c) Click on the [Stop] button.
   (2) Windows NT 4.0
      (a) Select "Settings" from start button ,and then select "Control panel".
      (b) Double click on "Device" .
      (c) Select "MELSECNET/10" from Device.
      (d) Click on the [Stop] button.
   (3) Windows 95
      (a) The system.ini file under C:*windows is opened in the editor.
      (b) Comments on the description of the MELSECNET/10 screwdriver under "386Enh".
      (c) The file is saved.

2. Start the MELSECNET/10 driver.
   (1) Windows NT 3.51/4.0
   Select the board check device, and click on the [start] button.
   (2) Windows 95
   Reboot the IBM-PC/AT computer.

3. Start the board diagnosis utility.
   (1) Windows NT 3.51
      (a) Open the "MELSECNET(common)" group.
      (b) Double click on the "Board Diagnosis Utility" icon.
      (c) The board diagnosis utility is started up, and the board information screen appears.
   (2) Windows NT4.0/95
      (a) Select "Programs" from start button ,and then select "Board Diagnosis Utility" icon.
      (b) The board diagnosis utility is started up , and the board information screen appears.

<Procedure for exiting from utility>

1. Press the End key.
2. The board diagnosis utility is exited.

6.4.2 Function List

The functions of the board diagnosis utility are listed below.

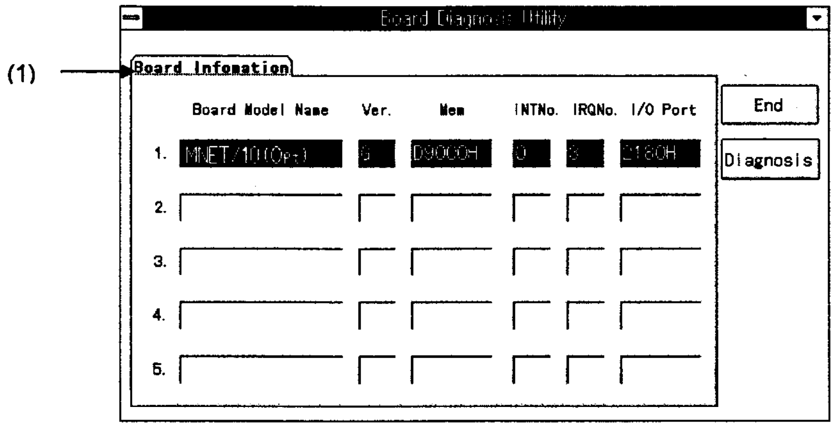
Function	Description	Reference Item
Board information	Displays the board model names, versions, 2-port memories used, etc.	Item 6.4.3
2-port diagnosis	Diagnoses the 2-port memory of the board.	Item 6.4.4

6.4.3 Board Information

This screen has the following functions.

- Board information

[How to read the screen]



Board Diagnosis Utility

Board Infomation

	Board Model Name	Ver.	Mem	INTNo.	IRQNo.	I/O Port
1.	MNET/10(Opt)	6	D9000H	0	3	2180H
2.						
3.						
4.						
5.						

(1) The board information is displayed in this area.

Board model name:	The model names of the installed boards are displayed here.
Ver:	The hardware versions of the installed boards are displayed here.
Mem:	The 2-port memories used are displayed here.
INT No:	The interrupt numbers which have been set are displayed here.
IRQ No:	The IRQ numbers used are displayed here.
I/O port:	The I/O addresses which have been set are displayed here.

6.4.4 2-Port Diagnosis

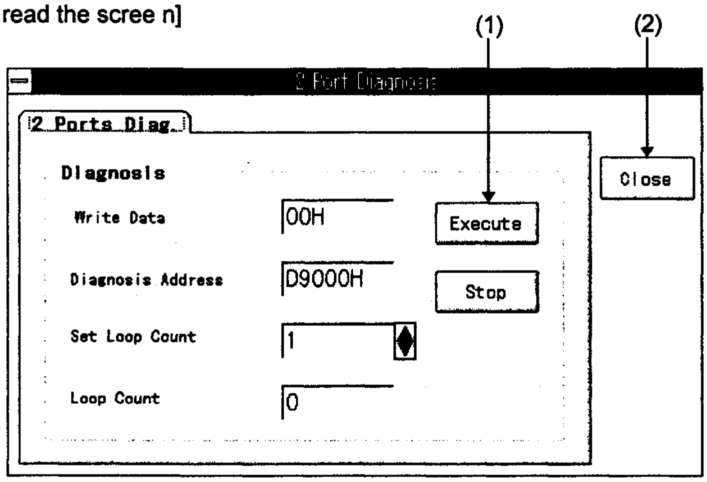
This screen has the following functions.

- 2-port diagnosis

[How to display the 2-port diagnosis screen]

1. Start up utility.
2. Display board information screen.
3. Select the [2-port diagnosis] button.

[How to read the screee n]



Write Data 00H

Diagnosis Address D9000H

Set Loop Count 1

Loop Count 0

(1) Two-port diagnosis is conducted for the memory by pressing the Execute button. If an error has occurred, an error message appears.

(2) Operation is returned to the board information screen by pressing the Close button.

* Upon completion of the 2-port diagnosis, illegal values may appear for the board model names and versions.

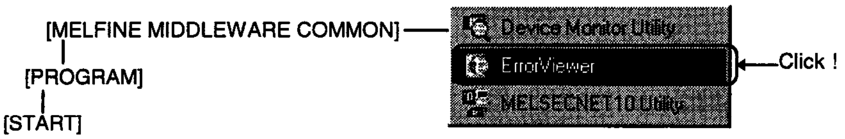
6.5 Error Viewer

This chapter explains how to use of set Error Viewer.

6.5.1 How to Start

The following explains how to start Error Viewer.

Click in sequence of [Start]-[Program]-[MELFINE MIDDLEWARE COMMON]-[Error Viewer].



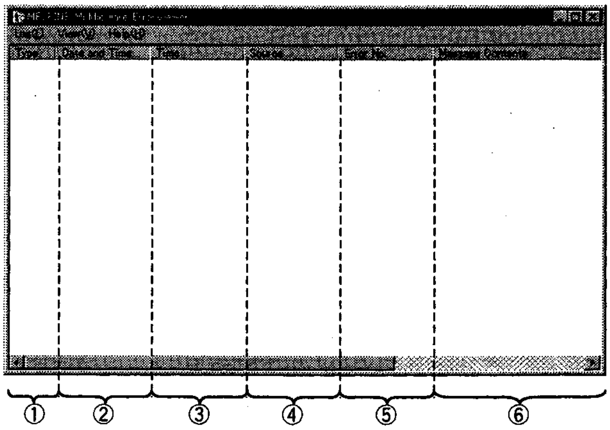
6.5.2 Functions

The table below list the functions of Error Viewer.

		Explanations	Reference Section
Log	Type of Error Register Source	Indicates an error log group displayed.	Section 2.4
	Open the Selected File	Opens log files saved so far.	
	Save with a Name	Saves an error log information currently chosen in a file specified.	
	Delete	Deletes an error log specified.	
	Log Setting	Sets a control method for a log exceeding a register size.	
	Exit	Terminates Error Viewer.	
Display	All Errors	Displays information on all error logs.	Section 2.5
	Specific Error	Set a condition on a error to be displayed.	
	Search	Finds an error log according to a source name and an error number.	
	Details	Displays details of determination of an error registered.	
	Renew	Renews a screen to the latest information.	
Help	Help	Displays Help screen of Error Viewer.	Section 2.6
	Version	Displays information on version of Error Viewer.	

6.5.3 Screen

This section explains the screen of Error Viewer.



① Type
Indicates error types with the marks as shown below.

ℹ · · · · · Normal message (Implies a notification of an incidence resulting from a normal operation.)

⚠ · · · · · Caution message (Gives a caution, however there occurred no errors.)

✖ · · · · · Error message (Gives an explanation of an error occurred to each module. Double-click it and see the detailed explanations of this error so as to immediately overcome the error for the line to which this message has been given.)

② Date and Time
Indicates a date on which an error occurred.

③ Time
Indicates a time at which an error occurred.

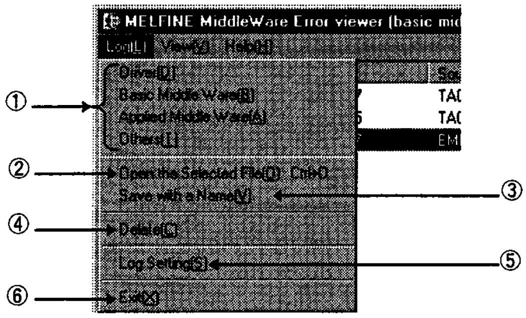
④ Source
Indicates the source of an error.

⑤ Error No
Indicates the number of an error occurred. For details of error number, see Error List Manual.

⑥ Message Contents
Indicates explanations of error.

6.5.4 Log

The following explains the log menu.



① Selection of Type of Error Register Source
Chooses a type of error register source displayed in Error Viewer.

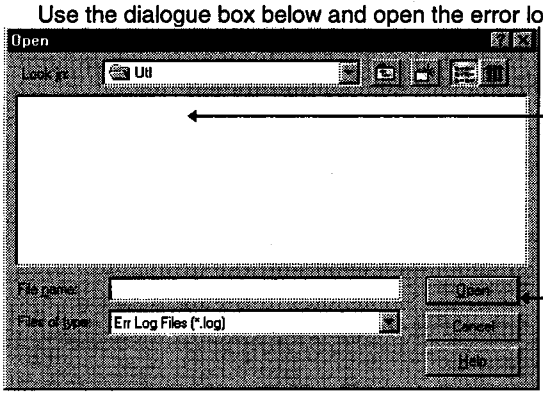
Driver ·············· Displays a message from a driver for a shared memory device, etc.

Basic Middle Ware ·· Displays a message from basic middleware such as shared memory data server or tag management process.

Applied Middle Ware Displays a message from applied middleware such as task control.

Others ············· Displays a message from an application package.

② Open the Selected File
Use the dialogue box below and open the error log file (*ELF).



Look in: Util

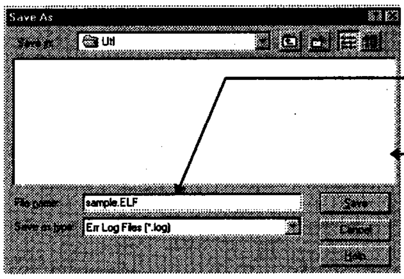
File name:

Files of type: Err Log Files (*.log)

Choose an error log file to be opened.

Open an error log file chosen.

③ Save with a Name
Error log information on an error register source (driver, basic middleware, etc.) currently chosen is saved in a file specified in the dialogue box below.



Save in: Util

File name: sample.ELF

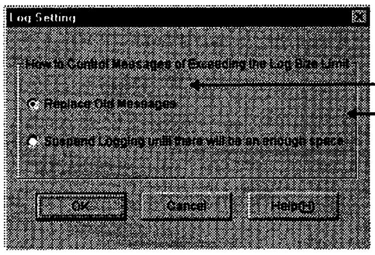
Save as type: Err Log Files (*.log)

Sets a file which saves error log information.

Saves error log information with a file name specified.

④ Delete
Deletes error log information on an error register source (driver, basic middleware, etc.) currently displayed. To delete, follow the instructions in the dialogue box.

⑤ Log Setting
Chooses a method to transact in the case the number of error logs exceeds the maximum register number.



How to Control Messages of Exceeding the Log Size Limit

● Replace Old Messages

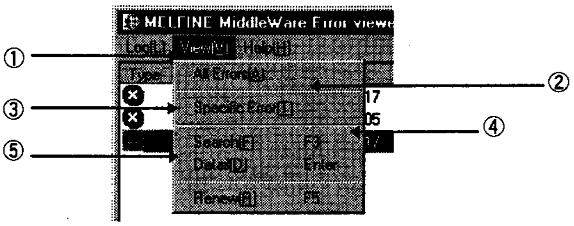
○ Suspend Logging until there will be an enough space

Superscribes information in order of old ones.
Not registered, unless a vacancy gets available by deletion, etc.

⑥ Exit
This screen terminates Error Viewer.

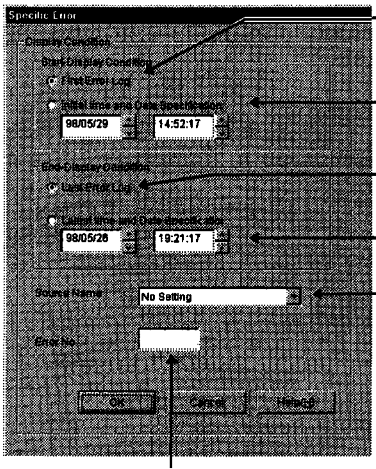
6.5.5 View

The following explains the details of the display menu.



① All Error
Displays every error which occurred to each type of error register source.

② Specific Error
Sets an error displayed on the screen, based on the condition as specified in the dialogue box below.



98/05/29 14:52:17

98/05/26 19:21:17

Source Name: No Setting

Displays errors from that of the first error log.

Displays errors from that of the date specified.

Displays errors up to that of the last error log.

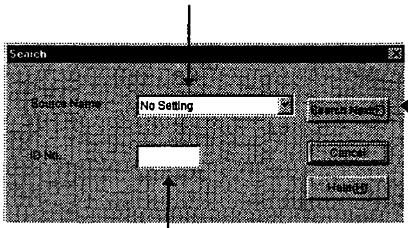
Displays errors up to that of the date specified.

Displays only errors whose source names are specified.

Displays only errors specified with error numbers.

③ Search

With the dialogue box below, finds source names and error information on error code, referring to the details of an error log currently displayed. (F3 key allows the same operation.)



Source Name: No Setting

ID No.:

Sets a source name to be found.

Finds next error information.

Enters an error code to be found.

④ Details

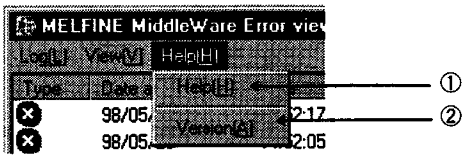
Displays detailed information about an error log currently chosen. (ENTER key allows the same operation, after items displayed are chosen.)

⑤ Renew

Renews information currently displayed.

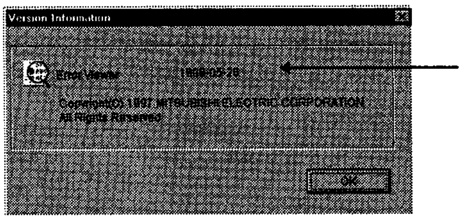
6.5.6 Help

The following explains the details of the display menu.



① Help
Displays Help screen for Error Viewer.

② Version
Displays information on a version of Error Viewer.


Version Information

Error Viewer 1998-05-20

Copyright(C) 1997 MITSUBISHI ELECTRIC CORPORATION
All Rights Reserved

Indicates a version of Error Viewer.

7 FUNCTIONS

7.1 Function List

The functions described in this chapter are listed below.

Function	Outline
mdClose	Communication function close
mdControl	Remote RUN/STOP/PAUSE
mdDevRst	Bit device reset
mdDevSet	Bit device set
mdOpen	Communication line open
mdRandR	Device random read
mdRandW	Device random write
mdReceive (RECV function)	Data receive
mdSend (SEND function)	Data send
mdTypeRead	Sequencer CPU model name read

7.2 Restrictions on Driver Usage

The restrictions which apply when each driver for each operating system is used are described in the following.

NOTE

Check out the restrictions before using the drivers.

7.2.1 Common Restrictions

The following common restriction applies when any of the drivers is used.

(1) Operation is not guaranteed if the interface board's common memory area overlaps the video RAM area or system ROM area.

7.2.2 Restrictions on DOS Driver Usage

The following restrictions apply when the DOS driver is used.

(1) The QnA-compatible SEND and RECV functions are not supported.

(2) Operation is not guaranteed in a UMB, high memory and task swapping environment.

(3) Only Visual C++ version 1.51 is supported as the development language.

7.2.3 Restrictions on Windows NT 3.51 Driver Usage

The following restrictions apply when the Windows NT driver is used.

(1) Only Visual Basic version 4.0 (32 bits) and Visual C++ version 4.0 are supported as the development languages.

(2) The driver does not support multi-processors.

7.2.4 Restrictions on Windows NT 4.0 Driver Usage

The following restrictions apply when the Windows NT driver is used.

(1) Only Visual Basic version 4.0 (32 bits) and Visual C++ version 4.2 are supported as the development languages.

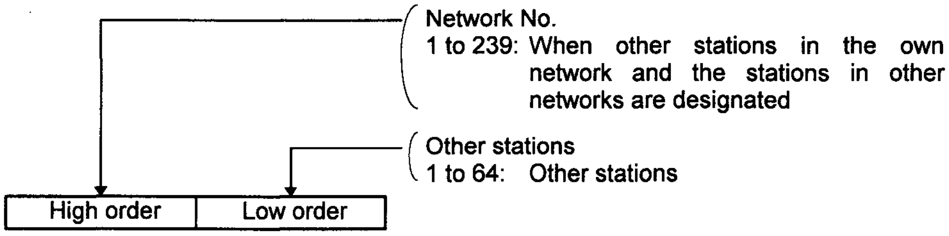
(2) The driver does not support multi-processors.

7.3 Common Data of Functions

This section describes the details common to the functions supported by the library.

1 Channel No. 51: MNET10 (1st board) 52: MNET10 (2nd board)
53: MNET10 (3rd board) 54: MNET10 (4th board)

2 Station numbers Oxff: Own station
Other stations as below



Network No.
1 to 239: When other stations in the own network and the stations in other networks are designated

Other stations
1 to 64: Other stations

3 Device type
Either the code number or device name may be assigned as the device type in the communication functions.

(a) When other stations are accessed

Device type		Device		A1(N)	A0J2H
					A1S(-S1)
					A1SJ
					A2C
					A2(N)
					A2(N)-S1
					A2S(-S1)
						A2A(-S1)
						A2U(-S1)
						A2AS(-S1)
							A3(N)
							A3V
							A3A
							A3U
							A73
								A4U

When assigned with code	When assigned with device name							
1	DevX	X	Batch	o	o	o	o	o
			Random					
2	DevY	Y	Batch	o	o	o	o	o
			Random					
3	DevL	L	Batch	o	o	o	o	o
			Random					
4	DevM	M	Batch	o	o	o	o	o
			Random					
5	DevSM	Special M	Batch	o	o	o	o	o
			Random					
		SB (link special B for MNET/10)	Batch	o	o	o	o	o
			Random					
6	DevF	F	Batch	o	o	o	o	o
			Random					
7	DevTT	T (contact)	Batch	o	o	o	o	o
			Random					
8	DevTC	T (coil)	Batch	o	o	o	o	o
			Random					

Device type		Device		A1(N)	A0J2H
					A1S(-S1)
					A1SJ
					A2C
					A2(N)
					A2(N)-S1
					A2S(-S1)
						A2A(-S1)
						A2U(-S1)
						A2AS(-S1)
							A3(N)
							A3V
							A3A
							A3U
							A73
								A4U

When assigned with code	When assigned with device name							
9	DevCT	C (contact)	Batch	o	o	o	o	o
			Random					
10	DevCC	C (coil)	Batch	o	o	o	o	o
			Random					
11	DevTN	T (current value)	Batch	o	o	o	o	o
			Random					
12	DevON	C (current value)	Batch	o	o	o	o	o
			Random					
13	DevD	D	Batch	o	o	o	o	o
			Random					
14	DevSD	Special D	Batch	o	o	o	o	o
			Random					
		SW (link special W for MNET10)	Batch	o	o	o	o	o
			Random					
15	DevTM	T (setting main)	Batch	o	o	o	o	o
			Random	x	x	x	x	x
16	DevTS	T (setting sub 1)	Batch	x	x	x	o	o
			Random				x	x
16002	DevTS2	T2 (setting sub 2)	Batch	x	x	x	x	o
			Random					x
16003	DevTS3	T3 (setting sub 3)	Batch	x	x	x	x	o
			Random					x
17	DevCM	C (setting main)	Batch	o	o	o	o	o
			Random	x	x	x	x	x
18	DevCS	C (setting sub 1)	Batch	x	x	x	o	o
			Random				x	x
18002	DevCS2	C (setting sub 2)	Batch	x	x	x	x	o
			Random					x
18003	DevCS3	C (setting sub 3)	Batch	x	x	x	x	o
			Random					x
19	DevA	A	Batch	o	o	o	o	o
			Random					
20	DevZ	Z	Batch	o	o	o	o	o
			Random					
21	DevV	V	Batch	o	o	o	o	o
			Random					
22	DevR	R	Batch	x	o	o	o	o
			Random					
22001-22256	DevER1-DevER256	Expansion R	Batch	x	o	o	o	o
			Random					
23	DevB	B	Batch	o	o	o	o	o
			Random					
24	DevW	W	Batch	o	o	o	o	o
			Random					

(b) When the own board is accessed

Device type		Device		A70BDE-J71LP23GE
When assigned with code	When assigned with device name			
1	DevX	X	Batch	o
			Random	
2	DevY	Y	Batch	o
			Random	
5	DevSM	SB (link special B for MNET/10)	Batch	o
			Random	
14	DevSW	SW (link special W for MNET/10)	Batch	o
			Random	
23	DevB	B	Batch	o
			Random	
24	DevW	W	Batch	o
			Random	
100	DevUSER	User data	Batch	x
			Random	

4 Device No. What is within the CPU device range is effective.
A multiple of 8 is assigned when accessing the ACPU bit device; a multiple of
16 is assigned when accessing the QnACPU bit device.

5 Access size This varies as shown below depending on the conditions.

(1) With device batch read/write
Device range: Inside total device range

(2) With device random read/write
Device range: 40 random read points
24 random write points
When a word device is assigned or when an A1(N), A2(N) or A3(N) CPU input (X) device is assigned, half of the number of points shown above is the maximum.
Random read: Number of input (X) points x 2 <= number of random read points
Random write: Number of input (X) points x 2 <= number of random write points

6 Timeout The timeout period for all the functions is as shown below.
Timeout period: 15 sec.

7.4 Detailed Description of Functions

Details on the functions supported by the library are given below in alphabetical order of function name.

7.4.1 mdClose (communication circuit close)

Function

Closes the communication line.

Format

VB:	ret%=mdClose(path&)		
	Integer ret	Return value	OUT
	Long path	Path pointer of opened line	IN
VC++:	ret=mdClose(path);		
	short ret;	Return value	OUT
	long path;	Path pointer of opened line	IN

Explanation

This function closes the communication line as befitting each channel.

Return value

"0" when processing is terminated normally.

Other than "0" when processing is terminated abnormally.

-1: Path error   Designated path is invalid.

Related functions

mdOpen()

Examples

VB: Refer to Item 7.5.1.

VC++: Refer to Item 7.5.2.

CHECKPOINTS

- When an error is returned from the communication driver, the error code is returned as is. (For details, refer to Chapter 8.)
- All the channels are closed when "0" is assigned for the path.

7.4.2 mdControl (remote RUN/STOP/PAUSE)

Function

Executes remote run, stop and pause.

Format

VB:	ret%=mdControl(path&,stno%,buf%)		
	Integer ret	Return value	OUT
	Long path	Channel path	IN
	Integer stno	Station number (Refer to the common data of functions in Item 7.3.)	IN
	Integer buf	Designated code	IN
VC++:	ret=mdControl(path, stno,buf);		
	short ret;	Return value	OUT
	long path;	Channel path	IN
	short stno;	Station number (Refer to the common data of functions in Item 7.3.)	IN
	short buf;	Designated code	IN

Explanation

- This executes remote run, stop and pause for the sequencer CPU.
- It indicates the designated code.

Instruction	Designated code
Remote run	0
Remote stop	1
Remote pause	2

Return value

"0" when processing is terminated normally.

Other than "0" when processing is terminated abnormally.

-1:	Path error	Designated path is invalid.
-4:	CPU error	Invalid station has been designated.
-16:	Station number error	Station number is outside the allowable range.
-17:	All station designation error	Designating all stations is invalid.
-18:	Remote designation error	A code other than the designated code was designated.

Examples

VB: Refer to Item 7.5.1.

VC++: Refer to Item 7.5.2.

CHECKPOINTS

- When an error is returned from the communication driver, the error code is returned as is. (For details, refer to Chapter 8.)

7.4.3 mdDevRst (bit device reset)

Function

Resets the bit device.

Format

VB: ret%=mdDevRst(path&,stno%,devtyp%,devno%)

Integer ret	Return value	OUT
Long path	Channel path	IN
Integer stno	Station number (Refer to the common data of functions in Item 7.3.)	IN
Integer devtyp	Device type	IN
Integer devno	Designated device No.	IN

VC++: ret=mdDevRst(path,stno,devtyp,devno);

short ret;	Return value	OUT
long path;	Channel path	IN
short stno;	Station number (Refer to the common data of functions in Item 7.3.)	IN
short devtyp;	Device type	IN
short devno;	Designated device No.	IN

Explanation

- This resets (turns OFF) the designated bit device.
- It checks the argument.

Return value

"0" when processing is terminated normally.

Other than "0" when processing is terminated abnormally.

-1:	Path error	Designated path is invalid.
-2:	Device No. error	Designated device number is outside the allowable range.
-3:	Device type error	Designated device type is invalid.
-4:	CPU error	Invalid station has been designated.
-16:	Station number error	Station number is outside the allowable range.
-17:	All station designation error	Designating all stations is invalid.

Examples

VB: Refer to Item 7.5.1.

VC++: Refer to Item 7.5.2.

CHECKPOINTS

- When an error is returned from the communication driver, the error code is returned as is. (For details, refer to Chapter 8.)
- Assign 0xff (255) as the station number when accessing the own station. If its actual number is assigned, an error (-4) will result.

7.4.4 mdDevSet (bit device set)

Function

Sets the bit device.

Format

VB:	ret%=mdDevSet(path&,stno%,devtyp%,devno%)		
	Integer ret	Return value	OUT
	Long path	Channel path	IN
	Integer stno	Station number	IN
		(Refer to the common data of functions in Item 7.3.)	
	Integer devtyp	Device type	IN
	Integer devno	Designated device No.	IN
VC++:	ret=mdDevSet(path,stno,devtyp,devno);		
	short ret;	Return value	OUT
	long path;	Channel path	IN
	short stno;	Station number	IN
		(Refer to the common data of functions in Item 7.3.)	
	short devtyp;	Device type	IN
	short devno;	Designated device No.	IN

Explanation

- This sets (turns ON) the designated bit device.
- It checks the argument.

Return value

"0" when processing is terminated normally.

Other than "0" when processing is terminated abnormally.

-1:	Path error	Designated path is invalid.
-2:	Device No. error	Designated device number is outside the allowable range.
-3:	Device type error	Designated device type is invalid.
-4:	CPU error	Invalid station has been designated.
-16:	Station number error	Station number is outside the allowable range.
-17:	All station designation error	Designating all stations is invalid.

Related functions

mdOpen(), mdClose(), mdDevRst()

Examples

VB: Refer to Item 7.5.1.

VC++: Refer to Item 7.5.2.

CHECKPOINTS

- When an error is returned from the communication driver, the error code is returned as is. (For details, refer to Chapter 8.)
- Assign 0xff (255) as the station number when accessing the own station. If its actual number is assigned, an error (-4) will result.

7.4.5 mdOpen (communication circuit open)

Function

Opens the communication line.

Format

VB: ret%=mdOpen(chan%,mode%,path&)

Integer ret	Return value	OUT
Integer chan	Communication line channel No.	IN
Integer mode	Dummy	IN
Long path	Path pointer of opened line	OUT

VC++: ret=mdOpen(chan,mode,path);

short ret;	Return value	OUT
short chan;	Communication line channel No.	IN
short mode;	Dummy	IN
long path;	Path pointer of opened line	OUT

Explanation

- This initializes and opens the channels.

Chan	51: MNET10 (1st board)	52: MNET10 (2nd board)
	53: MNET10 (3rd board)	54: MNET10 (4th board)

- It checks the argument range and, if an error is found, it returns the error.
- When an error is returned from the communication driver, the error code is returned as is.

Return value

"0" when processing is terminated normally.

Other than "0" when processing is terminated abnormally.

-8: Channel No. error   Designated channel No. is invalid.

Related functions

mdClose()

CHECKPOINTS

- When an error is returned from the communication driver, the error code is returned as is. (For details, refer to Chapter 8.)

7.4.6 mdRandR (device random read)

Function

Performs device random read.

Format

VB: ret%=mdRandR(path&,stno%,dev(0)%,buf(0)%,bufsize%)

Integer ret	Return value	OUT
Long path	Channel path	IN
Integer stno	Station number (Refer to the common data of functions in Item 7.3.)	IN
Integer dev(n)	Randomly designated device	IN
Any buf(n)	Read data (single precision integer array)	OUT
Integer bufsize	No. of bytes in read data storage area	IN

VC++: ret=mdRandR(path,stno,dev,buf,bufsize);

short ret;	Return value	OUT
long path;	Channel path	IN
short stno;	Station number (Refer to the common data of functions in Item 7.3.)	IN
short dev[ ];	Randomly designated device	IN
Short buf[ ];	Read data (single precision integer array)	OUT
Short bufsize;	No. of bytes in read data storage area	IN

Explanation

- This reads data from the devices designated at random.

  Randomly designated device dev[ ]

dev[0]	Number of blocks	
dev[1]	Device type	Block 1
dev[2]	Head device No.	
dev[3]	Number of points	
dev[4]	Device type	Block 2
⋮		

- The designation methods are described using the examples provided below.

  M100 to M115 ············ All bits are OFF.

  D10 to D13 ··············· 10 is stored in D10, 200 in D11, 300 in D12 and 400 in D13.

  M0 to M13 ················ All bits are ON.

  T10 current value·········· Current value of T10 is 1 sec. (10).

  * It is assumed that the current statuses listed above apply to the target devices.

  Random designated device

  dev[0]=4; ············· 4 designation ranges (M100-M115, D10-D13, M0-M13 and T10 setting)

  dev[1]=DevM;
  dev[2]=100; ····· M100 and above

  dev[3]=16; ············ 16 points (M100-M115)

  dev[4]=DevD;
  dev[5]=10; ····· D10 and above

  dev[6]=4; ············· 4 points (D10-D13)

dev[7]=DevM;
dev[8]=0;
	M0 and above

dev[9]=14;	14 points (M0-M13)

dev[10]=DevTN;
dev[11]=10;
	T10 timer current value and above

dev[12]=1;	1 point (T10)
Read data	The read data is stored in the buf[0] and subsequent array variables.
buf[0]=0;	All the M100-M113 bits are OFF.
buf[1]=10;	Current value of D10
buf[2]=200;	Current value of D11
buf[3]=300;	Current value of D12
buf[4]=400;	Current value of D13
buf[5]=0x3fff;	All the M0-M13 bits are ON.
buf[6]=10;	Current value of T10 is 10 (1 sec.)
Read data storage	This designates the number of bits for array variable buf[ ] in which the read data is stored.
Number of bytes in area	"14" is input in this example. buf[0]-buf[6]7 x 2 = 14 bytes

Return value

"0" when processing is terminated normally.

Other than "0" when processing is terminated abnormally.

-1:	Path error	Designated path is invalid.
-2:	Device No. error	Designated device number is outside the allowable range.
-3:	Device type error	Designated device type is invalid.
-4:	CPU error	Invalid station has been designated.
-5:	Size error	· The "device number + size" has exceeded the allowable device range. · Access has been made with an odd number of bytes. · The "device number + size" has exceeded the allowable range for the same block. · The number of read points has exceeded the restriction.
-6:	Block count error	The block count in dev[0] cannot be handled.
-12:	Block error	The designated block is invalid.
-15:	Read area length error	Read area size based on array variable containing read data is too small.
-16:	Station number error	Station number is outside the allowable range.

Examples

VB: Refer to Item 7.5.1.

VC++: Refer to Item 7.5.2.

CHECKPOINTS

- When an error is returned from the communication driver, the error code is returned as is. (For details, refer to Chapter 8.)
- Assign 0xff (255) as the station number when accessing the own station. If its actual number is assigned, an error (-4) will result.
- Random reading is disabled for the following devices:
  1. T.C. (setting)
  2. Buffer memory
- Refer to item "5" in the description of the functions provided in Item 7.3 for the number of points which can be read at random.

7.4.7 mdRandW (device random write)

Function

Performs device random write.

Format

VB: ret%=mdRandW(path&,stno%,dev(0)%,buf(0)%,bufsize%)

Integer ret	Return value	OUT
Long path	Channel path	IN
Integer stno	Station number (Refer to the common data of functions in Item 7.3.)	IN
Integer dev(n)	Randomly designated device	IN
Any buf(n)	Write data	IN
Integer bufsize	Dummy	IN

VC++: ret=mdRandW(path,stno,dev,buf,bufsize);

short ret;	Return value	OUT
long path;	Channel path	IN
short stno;	Station number (Refer to the common data of functions in Item 7.3.)	IN
short dev[ ];	Randomly designated device	IN
Short buf[ ];	Write data	IN
Short bufsize;	Dummy	IN

Explanation

- This writes data into the devices designated at random.

Randomly designated device dev[ ]

dev[0] ················ Number of blocks
dev[1] ················ Device type
dev[2] ················ Head device No. Block 1
dev[3] ················ Number of points
dev[4] ················ Device type Block 2
⋮

- The designation method is described using the example provided below.

M100 to M115 ············ All bits are OFF.
D10 to D13 ··············· 10 is stored in D10, 200 in D11, 300 in D12 and 400 in D13.

Randomly designated devices

dev[0]=2; ············· 2 designation ranges (M100-M115, D10-D13)
dev[1]=DevM;
dev[2]=100; ····· M100 and above
dev[3]=16; ············ 16 points (M100-M115)
dev[4]=DevD;
dev[5]=10; ····· D10 and above
dev[6]=4; ············· 4 points (D10-D13)

Write data

```
buf[0]=0; ·············· All the bits are OFF.
buf[1]=10; ············· Stores 10 in D10.
buf[2]=200; ············ Stores 200 in D11.
buf[3]=300; ············ Stores 300 in D12.
buf[4]=400; ············ Stores 400 in D13.
```

Return value

"0" when processing is terminated normally.

Other than "0" when processing is terminated abnormally.

-1:	Path error	Designated path is invalid.
-2:	Device No. error	Designated device number is outside allowable range.
-3:	Device type error	Designated device type is invalid.
-4:	CPU error	Invalid station has been designated.
-5:	Size error	· The "device number + size" has exceeded the allowable device range. · Access has been made with an odd number of bytes. · The "device number + size" has exceeded the allowable range for the same block. · The number of write points has exceeded the restriction.
-6:	Block count error	The block count in dev[0] cannot be handled.
-12:	Block error	The designated block is invalid.
-13:	Write protect error	The block to be accessed overlaps a memory protect area.
-16:	Station number error	Station number is outside allowable range.

Examples

VB: Refer to Item 7.5.1.

VC++: Refer to Item 7.5.2.

CHECKPOINTS

- When an error is returned from the communication driver, the error code is returned as is. (For details, refer to Chapter 8.)
- Assign 0xff (255) as the station number when accessing the own station. If its actual number is assigned, an error (-4) will result.
- Random writing is disabled for the following devices:
  1. T.C. (setting)
  2. Buffer memory
- Refer to item "5" in the description of the functions provided in Item 7.3 for the number of points which can be written at random.
- Bear in mind that writing data in a block (extension file register) to which an extension comment has been allocated will cause the extension comment information to be deleted.
- Bear in mind that writing data in a block (extension file register) which overlaps the sub2or sub3 program setting area will cause the sub2 or sub3 program to be deleted.
- Bear in mind that if an error occurs in the course of writing at random in B or W of the own station, there will be blocks with the data written properly and blocks in which an error occurs mixed together.

7.4.8 mdReceive (all devices read)

Function

All devices read

Format

VB:	ret%=mdReceive(path&,stno%,devtyp%,devno%,size%,data(0)%)		
	Integer ret	Return value	OUT
	long path	Channel path	IN
	Integer stno	Station number (Refer to the common data of functions in Item 7.3.)	IN
	Integer devtyp	Device type	IN
	Integer devno	Head device No.	IN
	Integer size	Read byte size	IN
	Any data(n)	Read data (single-precision integer array)	OUT
VC++:	ret=mdReceive(path,stno,devtyp,devno,size,data)		
	short ret;	Return value	OUT
	long path;	Channel path	IN
	short stno;	Station number (Refer to the common data of functions in Item 7.3.)	IN
	short devtyp;	Device type	IN
	short devno;	Head device No.	IN
	short *size;	Read byte size	IN
	short data[ ];	Read data (single-precision integer array)	OUT

Explanation

- This reads the data into the designated device.
- It checks the argument and checks whether the address and size, which have been calculated from the argument, are within the device memory range.
- When the designated read size exceeds the device range, the read enable size is returned for "size."

Return value

"0" when processing is terminated normally.

Other than "0" when processing is terminated abnormally.

-1:	Path error	Designated path is invalid.
-2:	Device No. error	Designated device number is outside allowable range.
-3:	Device type error	Designated device type is invalid.
-4:	CPU error	Invalid station has been designated.
-5:	Size error	· The "device number + size" has exceeded the allowable device range. · Access has been made with an odd number of bytes. · The "device number + size" has exceeded the allowable range for the same block.
-11:	Insufficient buffer area	Read area size based on array variable containing read data is too small.
-12:	Block error	The designated block is invalid.
-14:	Memory cassette error	The memory cassette has not been installed or the wrong memory cassette has been installed in the accessed CPU.
-16:	Station number error	Station number is outside the allowable range.
-17:	All station designation error	All station designation is not possible.

Related functions

mdOpen(), mdClose()

Examples

VB: Refer to Item 7.5.1.

VC++: Refer to Item 7.5.2.

CHECKPOINTS

- When an error is returned from the communication driver, the error code is returned as is. (For details, refer to Chapter 8.)
- Use 0xff (255) for designating the station number to access the own station. An error (-4) will result if the actual station number is used.
- To access the bit device, designate the device number using a multiple of 8.

7.4.9 mdSend (all devices write)

Function

All devices write

Format

VB: ret%=mdSend(path&,stno%,devtyp%,devno%,size%,data(0)%)

Integer ret	Return value	OUT
long path	Channel path	IN
Integer stno	Station number (Refer to the common data of functions in Item 7.3.)	IN
Integer devtyp	Device type	IN
Integer devno	Head device No.	IN
Integer size	Write byte size	IN
Any data(n)	Write data	IN

VC++: ret=mdSend(path,stno,devtyp,devno,size,data)

short ret;	Return value	OUT
long path;	Channel path	IN
short stno;	Station number (Refer to the common data of functions in Item 7.3.)	IN
short devtyp;	Device type	IN
short devno;	Head device No.	IN
short *size;	Write byte size	IN
short data[ ];	Write data	IN

Explanation

- This writes the data into the designated device.
- It checks the argument and checks whether the address and size, which have been calculated from the argument, are within the device memory range.
- When the designated write size exceeds the write enable size, the write enable size is returned for "size."

Return value

"0" when processing is terminated normally.

Other than "0" when processing is terminated abnormally.

-1:	Path error	Designated path is invalid.
-2:	Device No. error	Designated device number is outside the allowable range.
-3:	Device type error	Designated device type is invalid.
-4:	CPU error	Invalid station has been designated.
-5:	Size error	· The "device number + size" has exceeded the allowable device range. · Access has been made with an odd number of bytes. · The "device number + size" has exceeded the allowable range for the same block.
-12:	Block error	The block number of the designated extension file register is invalid.
-13:	Write protect error	The block of the extension file register to be accessed overlaps a write protect area in the memory cassette.
-14:	Memory cassette error	The memory cassette has not been installed or the wrong memory cassette has been installed in the accessed CPU.
-16:	Station number error	Station number is outside the allowable range.
-17:	All station designation error	All station designation is not possible.

Examples

VB: Refer to Item 7.5.1.

VC++: Refer to Item 7.5.2.

CHECKPOINTS

- When an error is returned from the communication driver, the error code is returned as is. (For details, refer to Chapter 8.)
- Use 0xff (255) for designating the station number to access the own station. An error (-4) will result if the actual station number is used.
- To access the bit device, designate the device number using a multiple of 8.
- Bear in mind that writing data in a block to which an extension comment has been allocated will cause the extension comment information to be deleted.
- Bear in mind that writing data in a block which overlaps the sub2 or sub3 program setting area will cause the sub2 or sub3 program information to be deleted.

7.4.10 mdTypeRead (sequencer CPU type name read)

Function

Reads the sequencer CPU type name.

Format

```
VB:    ret%=mdTypeRead(path&,stno%,buf%)
       Integer ret         Return value                                         OUT
       Long path           Channel path                                         IN
       Integer stno        Station number                                       IN
                           (Refer to the common data of functions in Item 7.3.)
       Integer buf         Type name code                                       OUT
VC++:  ret=mdTypeRead(path,stno,buf)
       short ret;          Return value                                         OUT
       long path;          Channel path                                         IN
       short stno;         Station number                                       IN
                           (Refer to the common data of functions in Item 7.3.)
       short buf;          Type name code                                       OUT
```

Explanation

- This reads the type name of the sequencer CPU of the designated station.

Sequencer CPU type name	Type name code
A0J2	A0H
A1N	A1H
A2N, A2S	A2H
A3N	A3H
AJ72	ABH
A2U, A2AS	82H
A2US, A2AS-S1	83H
A3U	84H
A4U	85H
A2A, Q2A, Q2AS, Q2ASH	92H
A2A-S1, Q2A-S1, Q2AS-S1, Q2ASH-S1	93H
A3A, Q3A, Q4A	94H
A0J2H, A1S	98H
A2C	9AH
Interface board	90H

Return value

"0" when processing is terminated normally.

Other than "0" when processing is terminated abnormally.

-1:	Path error	Designated path is invalid.
-4:	CPU error	Invalid station has been designated.
-16:	Station number error	Station number is outside the allowable range.

Examples

VB: Refer to Item 7.5.1.

VC++: Refer to Item 7.5.2.

CHECKPOINTS

- When an error is returned from the communication driver, the error code is returned as is. (For details, refer to Chapter 8.)

7.5 Sample Programs

This section contains two sample programs; one was prepared using Visual Basic and the other using Visual C++.

7.5.1 Visual Basic Sample Program

Presented below is a sample program (filename: DEMOPRO.MAK) using Visual Basic.

NOTE

Only Windows NT can be used to run this Visual Basic sample program.

```
' *******************************************************************
                    Type 2 Function Test Program
' *******************************************************************


' Type 2 function library
Declare Function mdOpen Lib "MDFUNC32.DLL" (ByVal Chan As Integer,ByVal Mode As Integer,Path As Long)As
Integer
Declare Function mdClose Lib "MDFUNC32.DLL" (ByVal Path As Long)As Integer
Declare Function mdSend Lib "MDFUNC32.DLL" (ByVal Path As Long,ByVal Stno As Integer,ByVal Devtyp As
Integer,ByVal Devno As Integer,Size As Integer,Buf As Any)As Integer
Declare Function mdReceive Lib "MDFUNC32.DLL" (ByVal Path As Long,ByVal Stno As Integer,ByVal Devtyp As
Integer, Byval Devno As Integer, Size As Integer,Buf As Any)As Integer
Declare Function mdDevSet Lib "MDFUNC32.DLL" (ByVal Path As Long,ByVal Stno As Integer,ByVal Devtyp As
Integer,ByVal Devno As Integer)As Integer
Declare Function msDevRst Lib "MDFUNC32.DLL" (ByVal Path As Long,ByVal Stno As Integer,ByVal Devtyp As
Integer,ByVal Devno As Integer)As Integer
Declare Function msRandW Lib "MDFUNC32.DLL" (ByVal Path As Long,ByVal Stno As Integer,dev As Integer,Buf As
Any ByVal bufsiz As Integer)As Integer
Declare Function mdRandR Lib "MDFUNC32.DLL" (ByVal Path As Long,ByVal Stno As Integer,dev As Integer,Buf As
Any,ByVal bufsiz As Integer)As Integer
Declare Function mdControl Lib "MDFUNC32.DLL" (ByVal Path As Long,ByVal Stno As Integer,ByVal Buf As
Integer)As Integer
Declare Function mdTypeRead Lib "MDFUNC32.DLL" (ByVal Path As Long,ByVal Stno As Integer,Buf As Integer)As
Integer

' Common variable declaration
Dim Path As Long

Sub Command1 Click ()
        ' Declare the local variables.
        Dim Stno As Integer             'Station number to be accessed
        Dim Devtyp As Integer           'Device name to be read
        Dim Devno As Integer            'Device number to be read
        Dim Size As Integer             'Number of bytes to be read
        ReDim Buf(128) As Integer       'Buffer in which read data is stored
        ' Execute the device read function.
        Stno = &H101                    'Station No.1 in network No.2 is accessed.
        Devtyp = 13                     'Device "D"
        Devno = 0                       'Device number
        Size = 2                        'Number of bytes
        Ret = mdReceive(Path,Stno,Devtyp,Devno,Size,Buf(0))

        ' Display the device number which has been read.
        RdDevno.Caption = Devno
        If Ret = 0 Then
                ' Display the data which has been read.
                RdDat.Caption = Buf(0)
        Else
                ' Clear the screen.
                RdDat.Caption = " "
        End If
End Sub

Sub Command2_Click ()
        ' Declare the local variable.
        Dim Ret As Integer              'Return value

' Execute the channel close function.
        Ret = mdClose(Path)
        If (Ret < > 0) Then
                MsgBox "Channel close error"
        End If
        End
End Sub

Sub Form_Load ()
        ' Declare the local variables.
        Dim Chan As Integer             'Channel number to be opened
        Dim Mode AS Integer             'Open mode
        Dim Ret As Integer              'Return value

        ' Execute the channel open function.
        Chan = 21
        Mode = -1
        Ret = mdOpen(Chan,Mode,Path)
        If (Ret < > 0) And (Ret < > 66) Then
                MsgBox "Channel open error"
        End
        End If
End Sub
```

7.5.2 Visual C++ Sample Program

Presented below is a sample program (filename: MTEST1.C) using Visual C++.

```
/*******************************************************************
 *                                                                 *
 *  Test program for communication type 2 function                *
 *                                                                 *
*******************************************************************/
/******************************
 *      Include            *
******************************/
#include            <stdio.h>
#include            <stdlib.h>
#include            <malloc.h>
#include            <memory.h>
#include            <string.h>
#include            "mdfunc2,h"

/**********************************************
 *           Common data               *
**********************************************/
long path;                     /*     Channel open path        */

static char          *main_menu [ ] = {
        "\n ******************************************************************************************\n",
        "**************** Test program for communication type 2 function ****************\n",
        "******************************************************************************************\n\n",
        "  [menu]            1 : Open                         2 : Close                         3 : Write\n",
        "          4 : Read                         5 : -(RESERVE)                 6 : -(RESERVE)\n",
        "          7 : Set                          8 : Reset                      9 : Random write\n",
        "         10 : Random read                 11: Sequencer control\n        12: Sequencer type name read\n",
        "          0 : End\n\n",
        "  menu No = ",
        "end"
};

/************************************************
 *            List of functions            *
************************************************/
void     main(void);
short    menu();
void     open_test(void);
void     close_test(void);
void     send_test(void);
void     recv_test(void);
void     rndsend_test(void);
void    rndrecv_test(void);
void    set_test(void);
void    rst_test(void);
void    control_test(void);
void    typeread_test(void);
void    dump(unsigned char *,short);
void    input(char *,short *);

/*******************************************
 *              Main Test                  *
 *******************************************/
void    main()
{
        path = OL;
        while(l){
                switch(menu(main_menu)){              /* Test item selection */
                        case 1:                       /* Open function test selection */
                                open_test();
                                break;
                        case 2:                       /* Close function test selection */
                                close_test();
                                break;
                        case 3:                       /* Device write function test selection */
                                send_test();
                                break;
                        case 4:                       /* Device read function test selection */
                                recv_test();
                                break;
                        case 5:
                                break;
                        case 6:
                                break;
                        case 7:                       /* Bit device set function test selection */
                                set_test();
                                break;
                        case 8:                       /* Bit device reset function test selection */
                                rst_test();
                                break;
                        case 9:                       /* Random write function test selection */
                                rndsend_test();
                                break;
                        case 10:                      /* Random read function test selection */
                                rndrecv_test();
                                break;
                        case11:                             /* Sequencer control function test selection */
                                control_test();
                                break;
                        case12:                             /* Sequencer type name read function test selection */
                                typeread_test();
                                break;
                        case 0:                             /* Test exit selection */
                                exit(0);
                                break;
                        default:                            /* Invalid test No. selection */
                                break;
                }
        }
}
/*****************************************************
 *              Test item selection                  *
 *****************************************************/
short   menu(menu_dat)
char    *menu_dat [ ] ;
{
        short i,ret;

        for(i = 0;stromp(menu_dat[i],"end");i++)
                printf("%s",menu_dat[i]);
        }
        input("%d" , &ret);
        return(ret);
}

/**********************************************
 *              Open function test            *
 **********************************************/
static  char*open_menu [ ] = {
        "\n+++++ MDOPEN TEST +++++\n",
        "  [chanel]     21 : MNET2\n",
        "               51 : MNET10-1        52 : MNET10-2        53 : MNET10-3        54 : MNET10-4\n",
        "  [mode]       -1 : No mode set\n",
        "  chanel = ",
        "end"
};
void    open_test()
{
        short chan,result,mode;
        chan = menu(open_menu);
        printf("  operation mode  =  ");
        input("%d",&mode);
        result = mdOpen(chan,mode,&path);
        printf("\n - result = %d(%xH)\n", result, result);
        printf(" - path = %lx\n",path);
}

/*****************************************************
 *            Close function test              *
 *****************************************************/
static    char*close_menu [ ] = {
        "\n+++++ MDCLOSE TEST +++++\n",
        "After inputting with number keys, use <Enter> key to execute\n",
        "end"
};
void      close_test()
{
        short       result;

        menu(close_menu);
        resulut = mdClose(path);
        printf("\n - result = %d(%xH)\n",result,result);
}

/***********************************************************
 *            Device write function test              *
 ***********************************************************/
static char*send_menu [ ] = {
        "\n+++++ MDSEND TEST +++++\n",
        "[Station number]  MNET2              : 255(Own station),0(Control station)\n",
        "                  MNET10 channel     : 1~64(Other stations),255(Own station),\n",
        "[Network No]                         : 1~239(MNET10 own station:0, other than MNET10:0)\n",
        "[Device type]                        : 1~28,30,35,\n",
        "                                       16002~3,18002~3,22001~22256,\n",
        "                                       1001~1255,2001~2255,\n",
        "                                       23001~23255,24001~24255,\n",
        "                                       25001~25255,28001~28255,29001~29255\n",
        "Station No. = ",
        "end"
};
static char *dat_menu [ ] = {
        " \n   Write data type selection \n",
        "  0:  All data 0x00\n",
        "  1:  All data 0xFF\n",
        "  2:  0,1,2,3, .... Size '\n",
        "  3:  Data preparation \n",
        " Write data type = ",
        "end"
};
```

```
void    send_test()
{
    short   stno,devtyp,devno,size,result;
    short   i;
    unsigned char   *dat,c;

    stno = menu(send_menu);
    printf(" Network No. = ");
    input("%d",&i);
    stno |= ((i<< 8) & 0xFF00);
    printf(" Device type   = ");
    input("%d",&devtyp);
    printf(" Device No. (or mail address) = ");
    input ("%d ", &devno);
    printf(" Write size = ");
    input("%d",&size);
    if((dat = malloc(size)) != NULL){
        switch(menu(dat_menu)){
            case 0:
                memset(dat,'\0',size);
                break;
            case 1:
                memset(dat,(unsigned char)0xff,size);
                break;
            case 2:
                c = 0;
                for(i = 0;i<size; i++) {
                    dat[i] = c;
                    if(c = 255){
                        c = 0;
                    }else{
                        C++;
                    }
                }
                break;
            case 3:
                for(i = 0;i<size; i++){
                    printf("data[%d] = ",i);
                    input("%d",(short *)&dat[i]);
                }
                break;
    }
        result = mdSend(path,stno,devtyp,devno,&size,dat);
        printf("\n -size   = %d\n",size);
        printf(" -result = %d(%xH)\n", result, result);
        free(dat);
        }else{
                printf("\n!!!!!!!!! Data memory cannot be acquired!!\n");
        }
}

/*************************************************************
 *            Device read function test                 *
*************************************************************/
static  char *recv_menu [ ] = {
        "\n+++++ MDRECEIVE TEST +++++\n",
        " [Station No.]       MNET              : 255(Own station),0(Control station)\n",
        "                     MNET10 channel    : 1~64(Other stations),255(Own station),\n",
        " [Network No.]                         : 1~239(MNET10 own station:0, other than MNET10:0)\n"
        " [Device type]                         : 1~28,30,35,\n",
        "                                         16002~3,18002~3,22001~22256,\n",
        "                                         1001~1255,2001~2255,\n",
        "                                         23001~23255,24001~24255,\n",
        "                                         25001~25255,28001~28255,29001~29255\n",
        "Station No. = ",
        "end"
};
static  char *dati_menu [ ] = {
        "\n Read buffer initial status selection\n",
        " 0: All data 0x00\n",
        " 1: All data 0xFF\n",
        " 2: 0,1,2,3,.... size \n",
        " Read buffer initial status = ",
        "end"
};
void    recv_test()
{
        short        stno,devtyp,devno,size,result;
        short        i;
        unsigned char        *dat,c;

        stno = menu(recv_menu);
        printf("   Network No.   = ");
        input("%d",&i);
        stno |=((i<<8)&0xFF00);
        printf("   Device type   = ");
        input("%d",&devtyp);
        printf("   Device No   (or mail address) = ");
        input("%d",&devno);
        printf("   Read size   = ");
        input("%d",&size);
        if((dat= malloc(size+3)) != NULL) {
                switch(menu(dati_menu)){
                    case 0:
                            memset(dat,'\0',size+3);
                            break;
                    case 1:
                            memset(dat,(unsigned char)0xff,size+3);
                            break;
                    case 2:
                            c=0;
                            for(i=0;i<size+3;i++){
                                    dat[i] = c;
                                    if(c = 255){
                                            c = 0;
                                    }else{
                                            c++;
                                    }
                            }
                            break;
            }
            result = mdReceive(path,stno,devtyp,devno,&size,dat);
            printf("\n -size = %d\n",size);
            printf(" -result = %d(%xH)\n", result, result);
            if(result = 0){
            dump(dat,size+3);
            }
            free(dat);
    }else{
            printf("\n!!!!!!!!! Data memory cannot be acquired!!\n");
    }
}

/**************************************************************
 *          Random write function test                        *
**************************************************************/
static char *rndsend_menu [ ] = {
    "\n+++++ MDRANDW TEST +++++\n",
    "[Station No.]      MNET2             : 255(Own station),0(Control station) \n",
    "                   MNET10 channel    : 1~64(Other stations),255(Own station),\n",
    "[Network No.]                        : 1~239(MNET10 own station:0, other than MNET10:0)\n",
    "[Device type]                        : 1~14,19~25,28,30,35,\n",
    "                                       22001~22256,\n",
    "                                       1001~1255,2001~2255,\n",
    "                                       23001~23255,24001~24255,\n",
    "                                       25001~25255,28001~28255,29001~29255\n",
    "Station No. =",
    "end"
};
```

```
void rndsend_test()
{
        short       stno,size,result;
        short       dev[1000];
        short       i;
        short       ret;
        unsigned char       *dat,c;

        size    =   0;
        stno = menu(rndsend_menu);
        printf(" Network No. = ");
        input("%d",&i);
        stno |= ((i<<8)&0xFF00);
        printf(" Block count =");
        input("%d",&dev[0]);
        for(i = 0;i<dev[0];i++){
                printf(" Device type = ");
                input("%d",&dev[i*3+1]);
                printf(" Device No. = ");
                input("%d",&dev[i*3+2]);
                printf(" Write point count = ");
                input("%d",&dev[i*3+3]);

                if((dev[i*3+1]<11) ||                                           //DeviceX˜CC
                        (dev[i*3+1] = 23) ||                                    //DeviceB
                        ((dev[i*3+1] >= 25)&&(dev[i*3+1] <= 27)) ||             //DeviceSB,STT,STC
                        ((dev[i*3+1] >= 1001)&&(dev[i*3+1] <= 1255)) ||         //DeviceLXI˜LX255
                        ((dev[i*3+1] >= 2001)&&(dev[i*3+1] <= 2255)) ||         //DeviceLYI˜LY255
                        ((dev[i*3+1] >= 23001)&&(dev[i*3+1] <= 23255)) ||       //DeviceLB1˜LB255
                        ((dev[i*3+1] >= 25001)&&(dev[i*3+1] <= 25255))) {       //DeviceSB1˜SB255
                                                                    /*          Bit device          */
                        size+= ((dev[i*3+3]-1)/16*2+2);
                }else{                                              /*          Word device         */
                        size+= dev[i*3+3]*2;
                }
}
        printf("Write byte count        =%d OK in bytes?(y/n)",size);
        input("%c",&ret);
        if(ret='n'){
                printf("size=");
                input("%d",&size);
        }
        if((dat=malloc(size))!=NULL){
                switch(menu(dat_menu)){
                        case 0:
                                memset(dat,'\0',size);
                                break;
                        case 1:
                                memset(dat,(unsigned char)0xff,size);
                                break;
                        case 2:
                                c = 0;
                                for(i = 0;i<size; i++){
                                        dat[i] = c;
                                        if(c = 255){
                                                c = 0;
                                        }else{
                                                c++;
                                        }
                                }
                                break;
                        case 3:
                                for(i = 0;i<size; i++){
                                        printf("data[%d] = ",i);
                                        input("%d",(short*)&dat[i]);
                                }
                                break;
                }
                result = mdRandW(path,stno,dev,(short*)dat,size);
                printf("\n -size = %d\n",size);
                printf(" -result = %d(%xH)\n",result,result);
                free(dat);
        }else{
                printf("\n!!!!!!!!! Data memory cannot be acquired!!\n");
        }
}

/***************************************************************
 *              Random read function test               *
***************************************************************/
static char             *rndrecv_menu [ ] = {
        "\n+++++ MDRANDR TEST +++++\n",
        "[Station No.]       MNET2              : 255(Own station),0(Control station)\n",
        "                    MNET10 channel     : 1~64(Other stations),255(Own station),\n",
        "[Network No.]                          : 1~239(MNET10 own station:0, other than MNET10:0)\n",
        "[Device type]                          : 1~14,19~25,28,30,35,\n",
        "                                         22001~22256,\n",
        "                                         1001~1255,2001~2255,\n",
        "                                         23001~23255,24001~24255,\n",
        "                                         25001~25255,28001~28255,29001~29255\n",
        "Station No. = ",
        "end"
};
```

```
void    rndrecv_test()
{
        short       stno,result;
        short       i,size;
        short       dev[1000];
        unsigned char         *dat,c;
        short       ret;

        size  =     0;
        stno  =     menu(rndrecv_menu);
        printf(" Network No. = ");
        input("%d",&i);
        stno | =    ((i<<8)&0xFF00);
        printf(" Block count = ");
        input("%d",&dev[0]);
        for(i = 0;i<dev[0];i++){
                    printf(" Device type = ");
                    input("%d",&dev[i*3+1]);
                    printf(" Device No. = ");
                    input("%d",&dev[i*3+2]);
                    printf(" Read point count = ");
                    input("%d",&dev[i*3+3]);
                    if((dev[i*3+1]<11) ||                                        //DeviceX~CC
                              (dev[i*3+1] = 23) ||                               //DeviceB
                              ((dev[i*3+1] >= 25)&&(dev[i*3+1] <= 27)) ||        //DeviceSB,STT,STC
                              ((dev[i*3+1] >= 1001)&&(dev[i*3+1] <= 1255)) ||    //DeviceLX1~LX255
                              ((dev[i*3+1] >= 2001)&&(dev[i*3+1] <= 2255)) ||    //DeviceLY1~LY255
                              ((dev[i*3+1] >= 23001)&&(dev[i*3+1] <= 23255)) ||  // DeviceLB1~LB255
                              ((dev[i*3+1] >= 25001)&&(dev[i*3+l] <= 25255))){   // DeviceSB1~SB255
                                                                        /*       Bit device          */
                              size += ((dev[i*3+3]-1)/16*2+2);
                    }else{                                              /*       Word device         */
                              size += dev[i*3+3]*2;
                    }
}
printf("Read byte count          = %d OK in bytes?(y/n)",size);
input("%c",&ret);
if(ret='n'){
                    printf("size   = ");
                    input("%d" &size);
        }
        if((dat=malloc(size+3))!= NULL){
                    switch(menu(dati_menu)){
                              case 0:
                                        memset(dat,'\0',size+3);
                                        break;
```

```
                        case 1:
                                memset(dat,(unsigned char)0xff,size+3);
                                break;
                        case 2:
                                c=0;
                                for(i = 0;i<size+3;i++){
                                        dat[i] = c;
                                        if(c = 255){
                                                c = 0;
                                        }else{
                                                c++;
                                        }
                                }
                                break;
                }
                result=mdrandr(path,stno,dev,(short*)dat,size);
                printf("\n -size =%d\n",size);
                printf(" -result =%d (%xH)\n",result,result);
                if(result = 0){
                        dump(dat,size+3);
                }
                free(dat);
        }else{
                printf("\n!!!!!!!!! Data memory cannot be acquired!!\n");
        }
}

/***************************************************************
 *              Bit device set function test                   *
 ***************************************************************/
static char*set_menu [ ] = {
        "\n+++++ MDDEVSET TEST +++++\n",
        "[Station No.]     MNET2             : 255(Own station),0(Control station)\n",
        "                  MNET10 channel    : 1~64(Other stations),255(Own station),\n",
        "[Network No.]                       : 1~239(MNET10 own station:0, other than MNET10:0)\n",
        "[Device type]                       : 1~10,23,25,30,\n",
        "                                      1001~1255,2001~2255,\n",
        "                                      23001~23255,25001~25255\n",
        "Station No. = ",
        "end"
};
void    set_test()
{
        short stno,devtyp,devno,result;
        short i;
        stno=menu(set_menu);
        printf(" Network No. = ");
        input("%d",&i);
        stno l= ((i<<8)&0xFF00);
        printf(" Device type = ");
        input ("%d", &devtyp);
        printf (" Device No. = " );
        input ("%d " &devno);
        result = mdDevSet(path,stno,devtyp,devno);
        printf("\n -result = %d(%xH)\n",result,result);
}

/*****************************************************************
 *            Bit device reset function test              *
*****************************************************************/
static char*rst_menu [ ] = {
        "\n+++++ MDDEVRST TEST +++++\n",
        "[Station No.]      MNET2              : 255(Own station),0(Control station)\n",
        "                   MNET10 channel     : 1~64(Other stations),255(Own station),\n",
        "[Network No.]                         : 1~239(MNET10 own station:0, other than MNET10:0)\n"
        "[Device type]                         : 1~10,23,25,30,\n",
        "                                        1001~1255,2001~2255,\n",
        "                                        23001~23255,25001~25255\n",
        "Station No. =",
        "end"
};
void    rst_test()
{
        short       stno,devtyp,devno,result;
        short       i;
        stno   =    menu(rst_menu);
        printf(" Network No. = ");
        input("%d",&i);
        stno   l=   ((i<<8)&0xFF00);
        printf(" Device type = ");
        input("%d",&devtyp);
        printf(" Device No. = ");
        input("%d",&devno);
        result = mdDevRst(path,stno,devtyp,devno);
        printf("\n -result = %d(%xH)\n",result,result);
}
/*****************************************************************
 *            Sequencer control function test             *
*****************************************************************/
static char*cntrl_menu [ ] = {
        "\n+++++ MDCONTROL TEST +++++\n",
        "[Station No.]      MNET2              : 0(Control station)\n",
        "                   MNET10 channel     : 1~64(Other stations) \n",
        "[Network No.]                         : 1~239(MNET10 own station:0, other than MNET10:0)\n",
        "[Designated code]                     : 0(RUN) 1(STOP) 2(PAUSE)\n",
        "Station No. = ",
        "end"
};
void control_test()
{
        short      stno,i,code,result;

        stno = menu(contrl_menu);
        printf(" Network No. = ");
        input("%d",&i);
        stno |= ((i<<8)&0xFF00);
        printf(" Designated code = ");
        input("%d",&code);
        result = mdControl(path,stno,code);
        printf("\n -result = %d(%xH)\n",result,result);
}

/*****************************************************************************
 *           Sequencer type name read function test               *
 *****************************************************************************/
static char *typrd_menu [ ] = {
        "\n+++++ MDCONTROL TEST +++++\n",
        "[Station No.]       MNET2              : 255(Own station),0(Control station)\n",
        "                    MNET10 channel     : 1~64(Other stations),255(Own station),\n",
        "[Network No.]                          : 1~239(MNET10 own station:0, other than MNET10:0)\n",
        "Station No. = ",
        "end"
};
void    typeread_test()
{
        short stno,i,code,result;
        stno = menu(typrd_menu);
        printf(" Network No. = ");
        input("%d",&i);
        stno |= ((i<<8)&0xFF00);
        code=0;
        result = mdTypeRead(path,stno,&code);
        printf("\n -code    = %xH\n",code);
        printf(" -result = %d(%xH)\n",result,result);
}
void    input(usng,arg)
char    *usng;
short   *arg;
{
        char buf[80];

        scanf("%s",buf);
        sscanf(buf,usng,arg);
}
```

8 ERROR CODES

The error codes returned when functions are executed are returned as is in the form of return values.

The table below gives a description of the error corresponding to each error code and the remedial action to be taken.

Return value (HEX)	Description of error	Remedy
0	Normal termination	
1	Driver has not started up. The same interrupt number or I/O address is used by another board.	Correct the error occurring at driver startup. Check the board settings.
2	Board response error. Timeout has occurred while waiting for response concerning remedial action.	Review the board installation status.
65 (41)	Channel error. An unregistered channel number has been set.	Set the correct channel number.
66 (42)	Already opened error. The designated channel has already been opened.	Set to OPEN one time only.
67 (43)	Already closed error. The designated channel has already been closed.	Set to CLOSE one time only.
68 (44)	Path error. A path other than the one opened for the line has been set.	Set the path to the number used when it was opened.
71 (47)	No receive data error. (When RECV was requested)	Wait until the data is received.
72 (48)	Mode setting wait. The mode has not been set.	Set the mode.
73 (49)	Mode error. A request was made to another station even though a setting other than online was used for the mode.	Set the mode to online or withdraw the request.
	Interrupt number error. The same interrupt number is used by another board.	Check the board setting.
	I/O address error. The same I/O address is used by another board.	
77 (4D)	Memory error Enough memory could not be secured.	End all other application programs that are currently running. Check if the system is operating normally. Restart the system. Increase the minimum working set area of the PC. *1
78 (4E)	Mode setting wait timeout error. The mode was set but there was no response from the board.	Check the board setting.
85 (55)	Network channel number error. (When RECV was requested)	Review the channel number designated when RECV is requested.
90 (5A)	Default parameter error.	Reset the default parameter.
100 (64)	Own station now being accessed or now requesting SEND. A request was issued to access the own station board while it was being accessed.	Issue the request again.

Return value (HEX)	Description of error	Remedy
101 (65)	Routing parameter error. The routing parameter has not been set.	Set the correct routing parameter.
102 (66) 103 (67)	System error. Error has occurred in the system.	Re-boot the system. The memory has been used up and the system memory was insufficient.
128 (80)	Read count error. A read byte count outside the allowable range was set.	Set the read count to a value inside the range.
129 (81)	Device name error. A non-existent device name was set when the interface board device was accessed.	Set the correct device name.
130 (82)	Device No. error. A non-existent device number was set when the interface board device was accessed. The device number was not a multiple of 8 for the ACPU or of 16 for the QnACPU when the bit device was accessed.	Set the correct device number.
131 (83)	Error in number of device points. A number of points exceeding the allowable device range was set when the interface board device was accessed. The device number was not a multiple of 8 for the ACPU or of 16 for the QnACPU when the bit device was accessed.	Set the correct number of device points.
132 (84)	Write count error. A write byte count outside the allowable range was set.	Set the write count to a value inside the range.
135 (87)	Remote run/stop/pause designation error. A setting other than 0, 1 or 2 was designated with remote run/stop/pause.	Reset the setting to 0, 1 or 2.
136 (88)	Random write designation error. A setting other than 0, 1 or 2 was designated with random write.	Reset the setting to 0, 1 or 2.
137 (89)	Processing canceled.	Re-issue the processing request upon completion of the previous processing.
210 (D2)	Request denied while CPU running. A sequence program shift processing request was issued while the CPU was running.	Stop the CPU and then issue the request.
212 (D4)	Processing canceled. A new processing request was issued while processing was already underway.	Re-issue the request.

Return value (HEX)	Description of error	Remedy
215 (D7)	Receive length error. The receive length data or byte length exceeded the allowable range.	Check the cables.
	Request data buffer length exceeded. The data length of the request data exceeded the data area.	Reduce the size of the request data.
1280 (500)	Access to own board memory denied. Memory is mixed up with another 16-bit board.	Check the settings of the own board switches, and move the memory addresses to an area unaffected by other boards. If 8-bit memory access has been set, change it to 16 bits.
1281 (501)	I/O port access denied.	Check the I/O port address settings. Conduct the self-loopback test on the board, and check out the hardware.
16384 to 20479 (4000 to 4FFF)	Errors detected by the access target CPU.	Refer to the user's manual of the access target CPU module.
16433 (4031)	Designated device No. outside range.	Check the designated device number.
16448 (4040)	No such unit.	Do not issue a request causing an error for a designated special unit.
16449 (4041)	Number of device points outside range.	Check the head address and number of access points, and proceed with access within the existing range.
16450 (4041)	Number of device points outside range.	Check the head address and number of access points, and proceed with access within the existing range.
16451 (4043)	Trouble in unit concerned.	Check whether the designated unit is operating correctly.
28672 to 32767 (7000 to 7FFF)	Errors detected by intelligent function modules such as the serial communication module.	Refer to the user's manual of the access target intelligent function module.
-1 (FFFF)	Path error. Designated path is invalid.	Use the pointer of the path returned by mdOpen.
-2 (FFFE)	Device No. error. Designated device number was outside the allowable range.	Check the head device number of the designated device.
-3 (FFFD)	Device type error. Designated device type is invalid.	Check whether the device type given on the device type list is used.
-4 (FFFC)	CPU error. An invalid station was been designated.	Check the statuses of the communication stations. Check the numbers of the designated stations.

Return value (HEX)	Description of error	Remedy
-5 (FFFB)	Size error. The "device number + size" exceeded allowable device range. Access was made with an odd-numbered device. The "device number + size" exceeded allowable range of the same block.	Check the designated device size. Check the "device number + size." When the A1(N), A2(N), A3(N) CPU input (X) has been designated, the size must be: number of input (X) points x 2 <= 40 points or number of input (X) points x 24 points. (Refer to Item 7.3)
-6 (FFFA)	Block count error. The block count designated by dev[0] for device random read/write was outside the allowable range.	Check the number of blocks designated by dev[0].
-8 (FFF8)	Channel No. error. Channel number designated by mdOpen is invalid.	Check the designated channel number.
-11 (FFF5)	Insufficient buffer area. The read area size based on array variable containing read data is too small.	Check the read size and the size of the read data storage destination.
-12 (FFF4)	Block error. The block number of the designated extension file register is invalid.	Check the block number (device type) of the designated extension file register.
-13 (FFF3)	Write protect error. The block number of the designated extension file register overlaps a write protect area in the memory cassette.	Check the block number (device type) of the designated extension file register. Check the write protect switch on the memory cassette at the access destination.
-14 (FFF2)	Memory cassette error. The memory cassette has not been installed in the accessed CPU. Alternatively, the wrong memory cassette has been installed.	Check the memory cassette at the access destination.
-15 (FFF1)	Read area length error. The read area size based on array variable containing read data is too small.	Check the read size and the size of the read data storage destination.
-16 (FFF0)	Station number error. The designated station number is outside the allowable range.	Check the designated station number.
-17 (FFEF)	All station error.	An attempt has been made to use a function for which designating all stations is not possible.
-18 (FFEE)	Remote instruction error. A code other than one which can be designated has been designated.	Check the designated code.

Return value (HEX)	Description of error	Remedy
-100 (FF9C) -101 (FF9B)	Internal error.	Re-opening has been repeated 3 or more times without closing.
-102 to -108 (FF9A to FF94)	Pointer error.	Review the pointers of the communication function arguments.
-4096 to -257 (F000 to FEFF)	Errors detected in the MELSECNET/H, MELSECNET/10 network system.	MELSECNET/10 network system Refer to the MELSECNET/H, MELSECNET/10 network system reference manual.
-16384 to -12289 (C000 to CFFF)	Errors detected by the Ethernet interface module.	Refer to the user's manual of the Ethernet interface module.
-20480 to -16385 (B000 to BFFF)	Errors detected in the CC-Link system.	Refer to the CC-Link system master/local module user's manual. * Refer to the QJ61BT11N User's Manual for whether the cyclic data can be transmitted to/from stations compatible with CC-Link Ver.2.
-26336 (9920)	A request for routing to another loop was issued to the interface board.	Change the routing request designation to AnUCPU or QnACPU.
-26334 (9922)	Board reset error. The board was reset by a task using the same channel of another station while that station was being accessed.	Access the other station again.
-25056 (9E20)	Request destination interface board cannot be processed.	Do not use any functions other than mdTypeRead for the interface boards of other stations.

*1: Procedures and sample program for increasing the minimum working set area of the PC
The following provides measures for increasing the minimum working set area of the PC when an error of error code 77 occurs due to MD function execution, and its sample program.
The PC board driver runs using the minimum working set area in the memory area reserved in the application program. Some application program may use a large area of the minimum working set area. In such a case, when the minimum working set area for the PC board driver cannot be reserved, an error code 77 is returned.
If this situation occurs, increase the minimum working set area in the application program before executing the MD function. (See the following sample program.)
The minimum working set area of 200KB is reserved at startup of the personal computer.

Sample program

This section gives a processing overview for setting a greater size to the minimum working set and provides sample programs.

(a) Processing overview of sample program

1) Obtain the application program ID by the GetCurrentProcessID function.
2) Using the ID obtained in step 1), obtain the application program handle by the OpenProcess function.
3) The current minimum and maximum working set sizes can be obtained by executing the GetProcessWorkingSetSize function.
4) Set a size greater than the minimum working set obtained in step 3) and execute the SetProcessWorkingSetSize.
5) Release the application program handle by the CloseHandle function.

(b) Sample program: When setting by VB
(Example of min. working set size 1MB and max. working set size 3MB)

```
Dim id As Long          ‘Application program ID variable
Dim ph As Long          ‘Application program handle variable
Dim wkmin As Long       ‘Minimum working set variable
Dim wkmax As Long       ‘Maximum working set variable

‘Obtain the application program ID
id = GetCurrentProcessID()
‘Open the application program handle
‘PROCESS_SET_QUOTA = 256,PROCESS_QUERY_INFORMATION = 1024
ph = OpenProcess(256 + 1024,False,id)
‘Obtain the maximum working set size and minimum working set size of the application program
bret = GetProcessWorkingSetSize(ph,wkmin,wkmax)
‘Set the minimum working set size to 1MB
wkmin = 1 * 1024 * 1024
‘Set the maximum working set size to 3MB
wkmax = 3 * 1024 * 1024
‘Change the maximum working set size and minimum working set size of the application program
bret = SetProcessWorkingSetSize(ph,wkmin,wkmax)
‘Close the application program handle
bret = CloseHandle(ph)
```

The set sizes shown here are reference sizes. Adjust the sizes in accordance with your system.

(c) Sample program: When setting by VC
(Example of min. working set size 1MB and max. working set size 3MB)

```
#define ERROR   -1
short ChangeWorkingSetSize()
{
     DWORD      dwProcessId;                   /*Application program ID variable*/
     HANDLE     hProcess;                      /*Application program handle variable*/
     DWORD      dwMinimumWorkingSetSize;       /*Minimum working set variable*/
     DWORD      dwMaximumWorkingSetSize;       /*Maximum working set variable*/

     /*Obtain the application program ID*/
     dwProcessId = GetCurrentProcessId();

     /*Open the application program handle*/
     hProcess =
     OpenProcess(PROCESS_SET_QUOTA+PROCESS_QUERY_INFORMATION,FALSE,dwProcessId);
     if(hProcess == NULL){
                /*Error end*/
                return(ERROR);
     }

     /*Obtain the maximum working set size and minimum working set size of the application program */
     if(GetProcessWorkingSetSize(hProcess,&dwMinimumWorkingSetSize,&dwMaximumWorkingSetSize)==0){
                /*Error end*/
                CloseHandle(hProcess);
                return(ERROR);
     }

     /*Set the minimum working set size to 1MB*/
     dwMinimumWorkingSetSize = 1 * 1024 * 1024;
     /*Set the maximum working set size to 3MB*/
     dwMaximumWorkingSetSize = 3 * 1024 * 1024;

     /*Change the maximum working set size and minimum working set size of the application program */
     if(SetProcessWorkingSetSize(hProcess,dwMinimumWorkingSetSize,dwMaximumWorkingSetSize)==0){
                /*Error end*/
                CloseHandle(hProcess);
                return(ERROR);
     }

     /*Close the application program handle*/
     CloseHandle(hProcess);

     /*Normal return*/
     return(0);
```

The set sizes shown here are reference sizes. Adjust the sizes according to your system.

9 TROUBLESHOOTING

This chapter describes the various procedures for pinpointing the causes of various kinds of trouble which may occur during the construction of the system, and the suggested remedial action for rectifying the trouble.

9.1 Troubleshooting Tips

To start up the system quickly, there obviously should be as few incidences of trouble as possible but it is also important if trouble does occur that its cause be located as quickly as possible.
Three basic points are to be borne in mind when proceeding with troubleshooting.

(1) Visual checks
Check out the following points
(a) Movements of the external equipment (stop, operating modes)
(b) Power supply on/off status
(c) Wiring connection status (connecting cables, etc.)
(d) LED display statuses (power display LED, etc.)
After checking out (a) to (d), connect the personal computer to the external equipment, and check the operation of the user program.

(2) Trouble checks
Observe how the trouble changes by carrying out the following operations.
(a) Change the input mode. Can the new changes be read by running the test program?
(b) Repeatedly turn the output on and off. Does the status of the external equipment change correctly?

(3) Narrowing down the possible cause of the trouble
On the basis of what has been ascertained in (1) and (2), figure out the general area where the trouble has occurred.
(a) Personal computer side or external equipment side?
(b) Personal computer itself or board?
(c) Connecting cables?
(d) User program?

9.2 Cause Location Method By Trouble Case

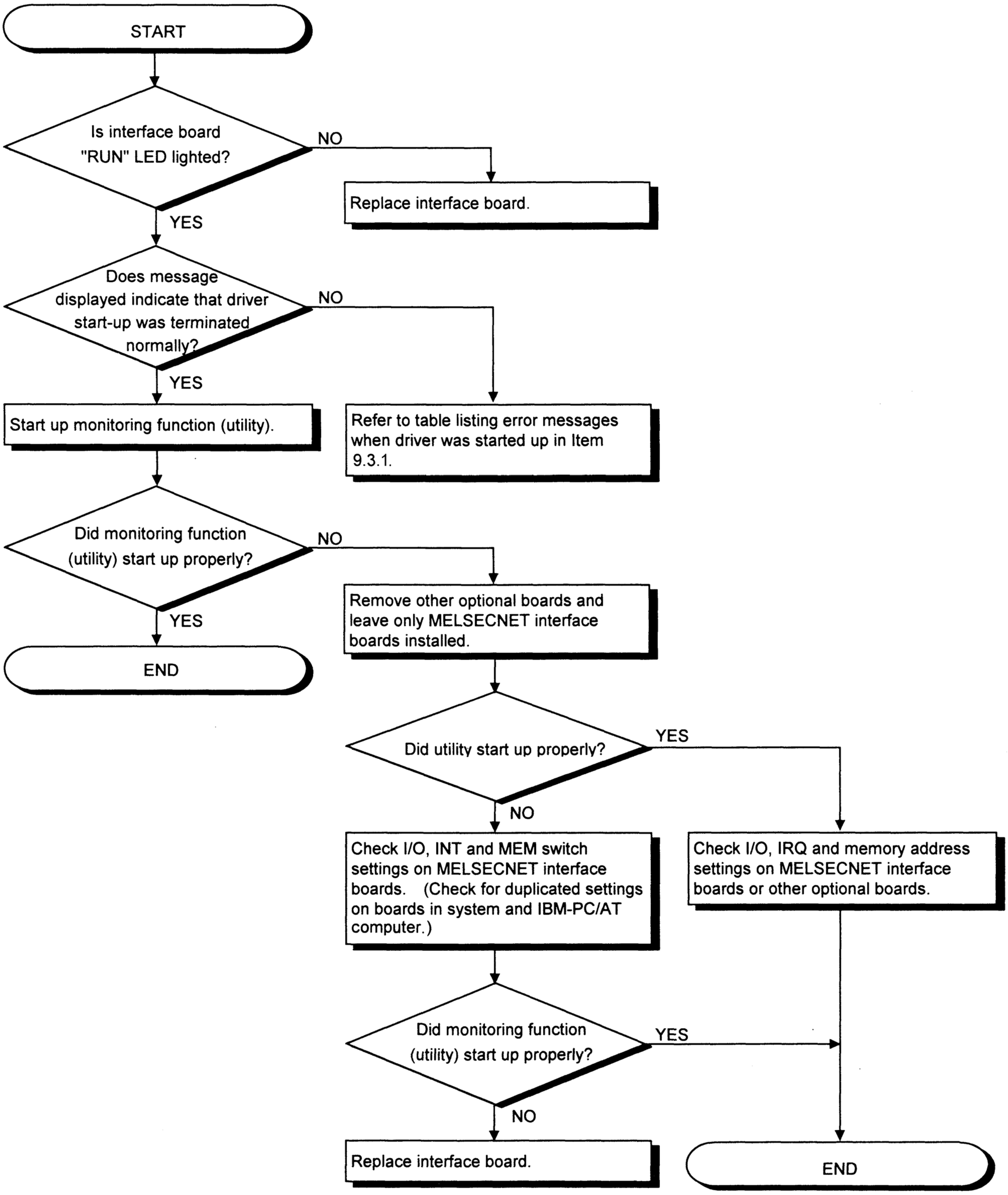
When trouble has occurred, use the table presented below to find where the troubleshooting method for pinpointing the cause of the trouble is given.

No.	Trouble	Method of pinpointing cause
1	Failure to operate properly when interface board was started up	Refer to the flow in Case 1 presented in Item 9.3.
2	Data link failure after connection of interface board, control station and other normal stations	Refer to the flow in Case 2 presented in Item 9.4.
3	Trouble in data link (a) A value differing from that planned has been input to a specific link device (B, W, X or Y). (b) A device has been accessed using the communication functions in the user program but data could not be written or read. (c) Communication is disabled from time to time while the user program is running. (d) System resetting or system hang-up has occurred while the user program was running in DOS/Windows 3.1. (e) System down (blue screen) or system resetting has occurred while the user program was running in Windows NT.	Refer to the Q&A in Case 3 presented in Item 9.5.

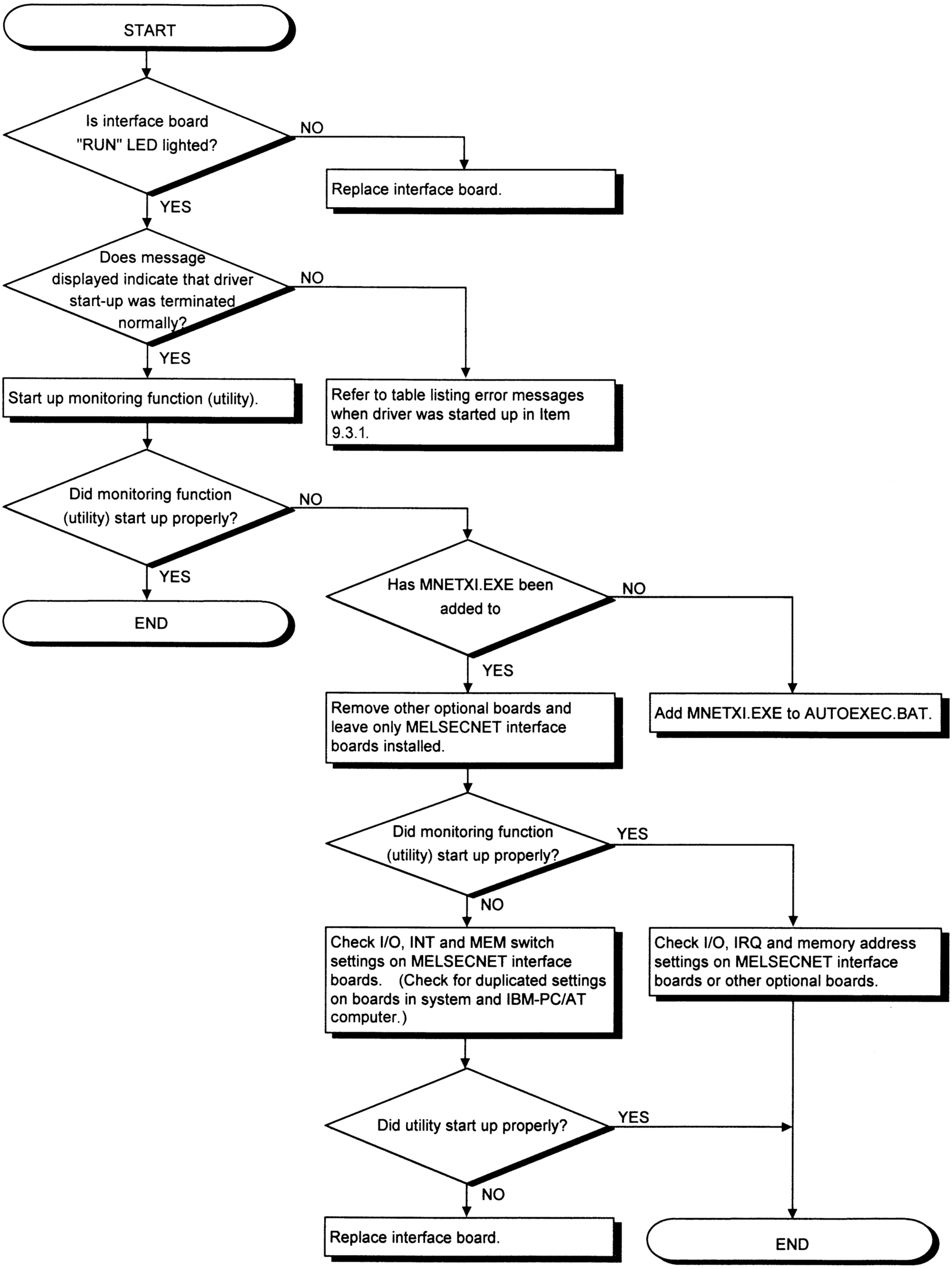
9.3 Case 1

Illustrated below is the procedure for checking the IBM-PC/AT computer unit when proper operation has failed after the interface board was started up.

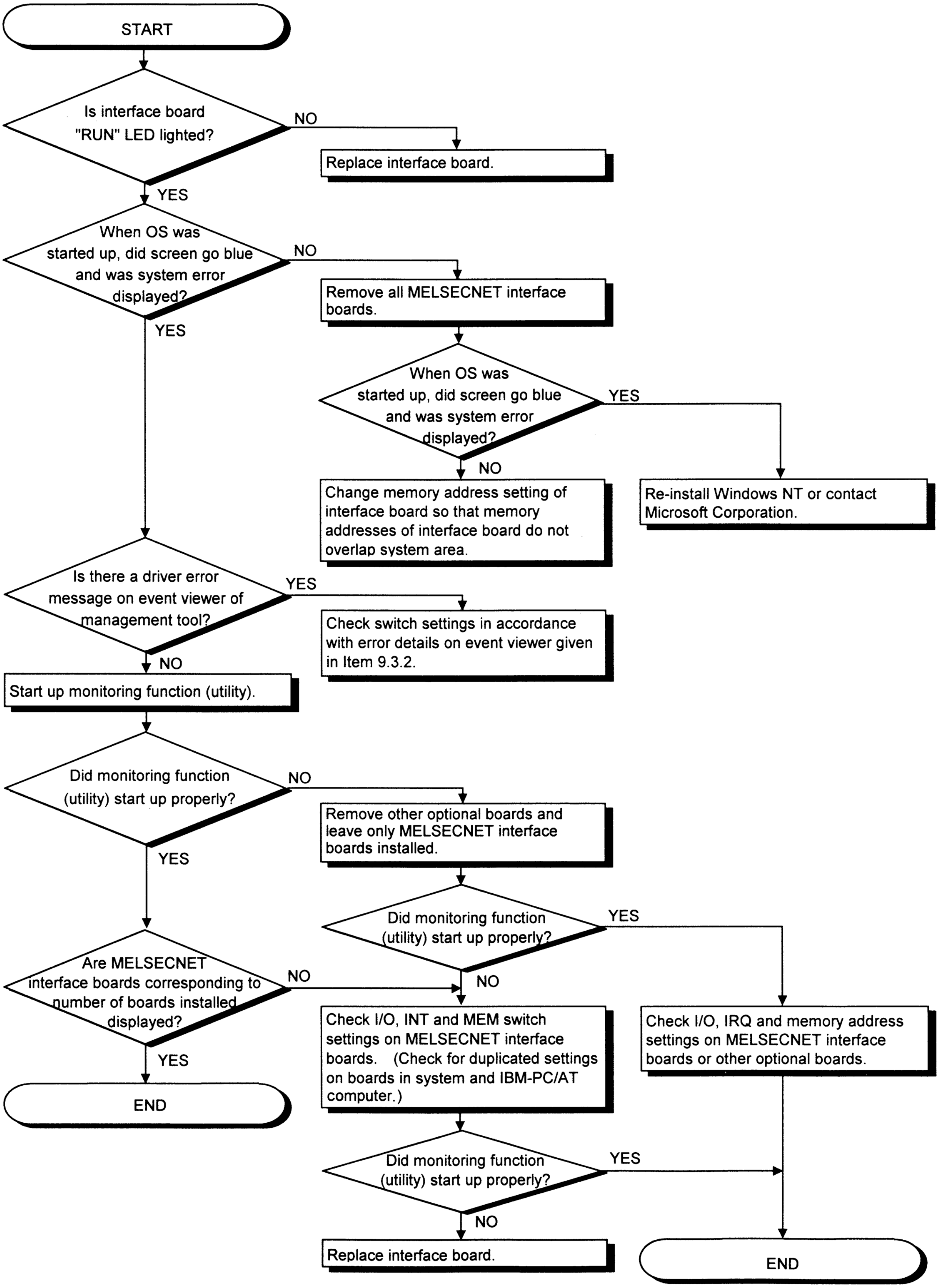
(1) With DOS


START
Is interface board "RUN" LED lighted?
NO
Replace interface board.
YES
Does message displayed indicate that driver start-up was terminated normally?
NO
YES
Start up monitoring function (utility).
Refer to table listing error messages when driver was started up in Item 9.3.1.
Did monitoring function (utility) start up properly?
NO
YES
END
Remove other optional boards and leave only MELSECNET interface boards installed.
Did utility start up properly?
YES
NO
Check I/O, INT and MEM switch settings on MELSECNET interface boards. (Check for duplicated settings on boards in system and IBM-PC/AT computer.)
Check I/O, IRQ and memory address settings on MELSECNET interface boards or other optional boards.
Did monitoring function (utility) start up properly?
YES
NO
Replace interface board.
END

(2) With Windows 3.1


START
Is interface board "RUN" LED lighted?
NO
Replace interface board.
YES
Does message displayed indicate that driver start-up was terminated normally?
NO
YES
Start up monitoring function (utility).
Refer to table listing error messages when driver was started up in Item 9.3.1.
Did monitoring function (utility) start up properly?
NO
YES
END
Has MNETXI.EXE been added to
NO
YES
Remove other optional boards and leave only MELSECNET interface boards installed.
Add MNETXI.EXE to AUTOEXEC.BAT.
Did monitoring function (utility) start up properly?
YES
NO
Check I/O, INT and MEM switch settings on MELSECNET interface boards. (Check for duplicated settings on boards in system and IBM-PC/AT computer.)
Check I/O, IRQ and memory address settings on MELSECNET interface boards or other optional boards.
Did utility start up properly?
YES
NO
Replace interface board.
END

(3) With Windows NT


START
Is interface board "RUN" LED lighted?
NO
Replace interface board.
YES
When OS was started up, did screen go blue and was system error displayed?
NO
Remove all MELSECNET interface boards.
When OS was started up, did screen go blue and was system error displayed?
YES
Re-install Windows NT or contact Microsoft Corporation.
NO
Change memory address setting of interface board so that memory addresses of interface board do not overlap system area.
YES
Is there a driver error message on event viewer of management tool?
YES
Check switch settings in accordance with error details on event viewer given in Item 9.3.2.
NO
Start up monitoring function (utility).
Did monitoring function (utility) start up properly?
NO
Remove other optional boards and leave only MELSECNET interface boards installed.
Did monitoring function (utility) start up properly?
YES
Check I/O, IRQ and memory address settings on MELSECNET interface boards or other optional boards.
NO
Check I/O, INT and MEM switch settings on MELSECNET interface boards. (Check for duplicated settings on boards in system and IBM-PC/AT computer.)
Did monitoring function (utility) start up properly?
YES
NO
Replace interface board.
END
YES
Are MELSECNET interface boards corresponding to number of boards installed displayed?
NO
YES
END

9.3.1 Driver Startup Error Message List

(MS-DOS, Windows 3.1)

No.	Status	Description	Startup result
--	Message	MELSEC DRIVER MNET10X.SYS Ver.00A	Successful
	Error description	None	
1	Message	ERROR 0001 IN MELSEC DRIVER MNET10X.SYS INT-A PARAMETER ERROR	Failed
	Error description	Characters in argument (1) are not "INT-A."	
	Remedial action	Change characters in config.sys argument (1) to "INT-A".	
2	Message	ERROR 0002 IN MELSEC DRIVER MNET10X.SYS INT-A NUMBER ERROR	Failed
	Error description	Number in argument (1) is outside the 0x60 to 0xFF range.	
	Remedial action	Change number in config.sys argument (1) to number within the 0x60 to 0xFF range.	
7	Message	ERROR 0007 IN MELSEC DRIVER MNET10X.SYS BOARD NOT FOUND	Failed
	Error description	Board cannot be found. [Causes] (1) Board is not installed. (2) Same 2-port memory address is used for other boards.	
	Remedial action	Refer to Item 6.2 on how to use the DOS board diagnosis utility, and self-diagnose the board.	
8	Message	ERROR 0008 IN MELSEC DRIVER MNET10X.SYS NO RESPONSE FROM BOARD	Failed
	Error description	Communication with the board was not possible during start-up. [Causes] (1) Board has not been installed properly. (2) The board itself is damaged.	
	Remedial action	Re-examine the board installation status, and re-start. If the error persists, replace the interface board.	
9	Message	ERROR 0009 IN MELSEC DRIVER MNET10X.SYS STATION NO ERROR	Failed
	Error description	Station number switch on the board is set to a number outside the allowable range.	
	Remedial action	Set the station number switch on the board to a number from 1 to 64.	
15	Message	ERROR 0015 IN MELSEC DRIVER MNET10X.SYS NO 16BIT ACCESS ERROR	Failed
	Error description	I/O port 16-bit cannot be accessed.	
	Remedial action	Replace the interface board.	
16	Message	ERROR 0016 IN MELSEC DRIVER MNET10X.SYS OTHER BOARD USE IN MEMORY AREA ERROR	Failed
	Error description	Common memory overlaps memory of other boards. (Mode setting error)	
	Remedial action	Adjust the MEM switch to change the common memory address to an address which is not used by other boards.	

9.3.2 Message List For Error Events Occurring At Driver Startup

Given below is a list of the errors which are displayed on the event viewer.
On the next page is an example of the event error message displayed when the same address has been set by the common memory address.

Event ID 0-FFFFh	Error description	Remedial action
256(100h)	Driver could not be executed since an error occurred at driver start-up.	Re-install the driver package. If the error persists, re-install Windows NT and then proceed.
257(101h)	The board is not installed or there is an overlap in the I/O address with other hardware.	Change the I/O address settings on the interface board.
258(102h)	No response from the hardware.	Replace the interface board.
259(103h)	More boards than the maximum number which can be installed have been detected.	Remove the boards beyond the maximum number.
260(104h)	The board is in contention with other device due to an error in the MEM, INT, station number, DIP switch setting, etc.	Check for settings which are duplicated with other optional boards. Check whether the same settings are used for two or more MELSECNET interface boards using the board diagnosis utility. (Refer to Item 6.4 on how to use the Windows NT board diagnosis utility.)
261(105h)	Failure in device object preparation.	Increase the amount of system memory.
262(106h)	Failure in device name link.	Start again from installation of operating system by re-installing Windows NT.
263(107h)	Failure in providing 2-port memory.	Change the memory address setting.
264(108h)	Failure in interrupt registration.	Change the setting of the INT number.
265(109h)	An error has occurred during I/O reading.	Replace the interface board.
266(10Ah)	An error has occurred during I/O writing.	
267(10Bh)	An error has occurred during memory transfer.	Start again from installation of operating system by re-installing Windows NT.
268(10Ch)	An error has occurred in receive processing.	---
269(10Dh)	An error has occurred in send processing.	---

Event ID 0-FFFFh	Error description	Remedial action
270(10Eh)	An error has occurred in interrupt processing.	Start again from installation of operating system by re-installing Windows NT.
271(10Fh)	An error has occurred in unload processing.	
272(110h)	An error has occurred in start I/O processing.	
273(111h)	An error has occurred in a critical session.	
274(112h)	An error has occurred in I/O timer processing.	
275(113h)	An error has occurred in I/O completion processing.	
276(114h)	An error has occurred in DPC processing.	
277(115h)	An error has occurred when the registry database was opened.	
278(116h)	An error has occurred when data was read from the registry database.	
279(117h)	An error has occurred when data was written into the registry database.	Increase the amount of system memory and disk space.
280(118h)	A request which cannot be processed has been received from another station.	---
281(119h)	Retry send has occurred with send processing.	---

9.3.3 Self-Diagnosis Test

(1) The self-diagnosis test is designed to check the hardware aspects of the interface board and the optical fiber cable for breaks. One of the following 3 items can be selected by switching the mode using the utility.

Mode setting	Description
Test between control stations	The lines between two stations are checked in this mode. For this check, the station with the lower number is set as the control station and the other station as the slave station.
Test between slave stations	
Self-loopback test	This checks the hardware inclusive of the send/receive circuitry of the transmission system provided on the interface board.

(2) For details on tests other than the self-loopback test, refer to the MELSECNET/10 Network System Reference Manual.

9.3.4 Self-Loopback Test

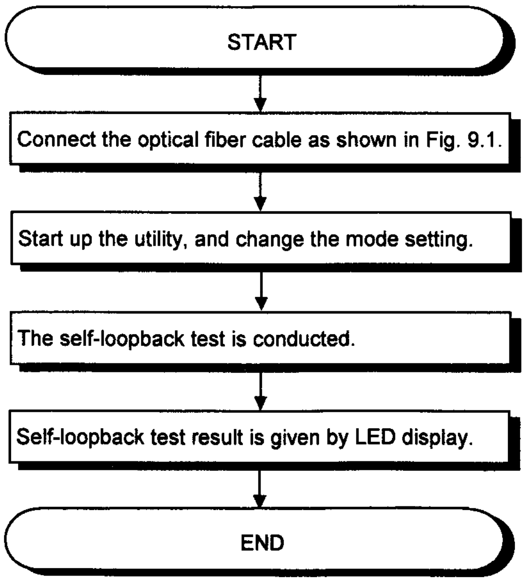
(1) Self-loopback test

(a) This test is conducted to check the interface unit's hardware inclusive of the send/receive circuitry (forward loop and reverse loop) of the transmission system provided on the interface board.

(b) It gives an OK/NG decision as to whether the data sent from the forward/reverse loop transmission end can be received by the forward/reverse loop reception end within the allotted time.

(2) Test method

The self-loopback test procedure is outlined below.


START
Connect the optical fiber cable as shown in Fig. 9.1.
Start up the utility, and change the mode setting.
The self-loopback test is conducted.
Self-loopback test result is given by LED display.
END

···· Select the self-loopback test.

···· The self-loopback test is conducted several moments after selection.

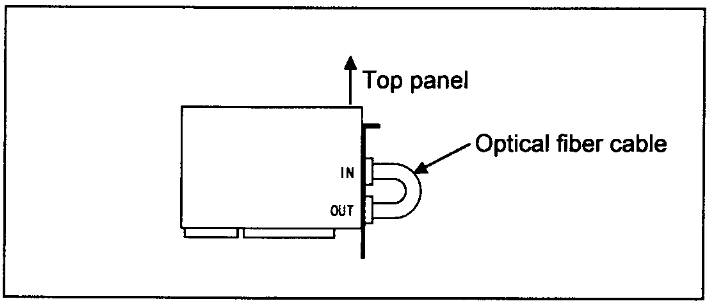


Fig. 9.1 Self-loopback test

(3) Decision on test results

The test results are indicated by the LEDs on the interface board and on the board information screen.

When OK ···· The "CRC," "OVER," "AB.IF," "TIME," "DATA" and "UNDER" signal name LEDs flash one after another.

When NG ···· The signal name LED whose error corresponds to the error information lights.

(a) When the "TIME" LED lights:

- The forward loop cable has a break.
- The forward loop transmission and reception ends are not connected by cable.
- The forward and reverse loops at the transmission end and the forward and reverse loops at the receptions end have been connected.

(b) When the "DATA" LED lights:

- The reverse loop cable has a break.
- The reverse loop transmission and reception ends are not connected by cable.

(c) When the "CRC," "OVER" or "AB.IF" LED on F.LOOP and R.LOOP side flash:

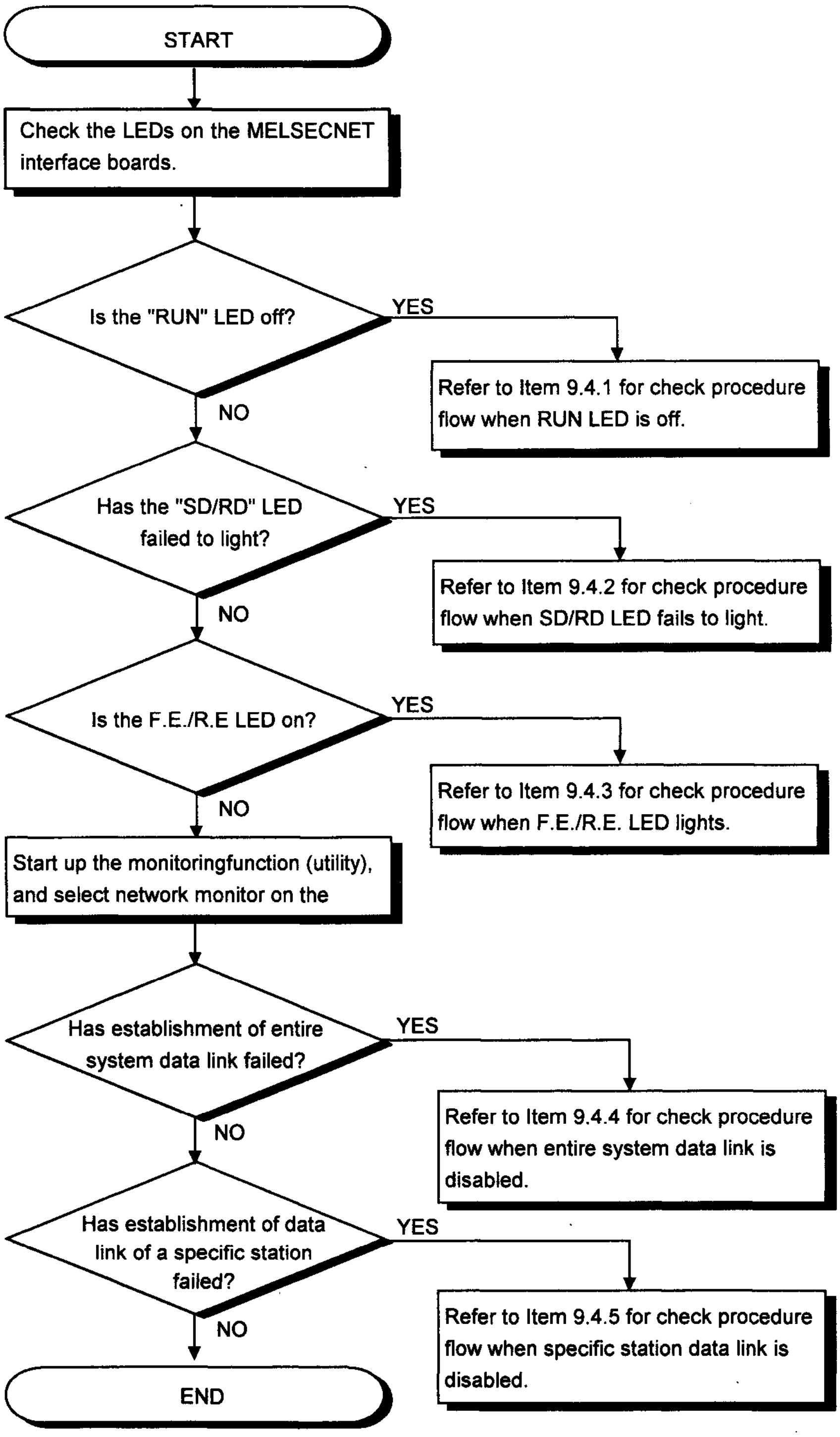
- Defective cable

(d) When an error LED other than (a), (b) and (c) lights:

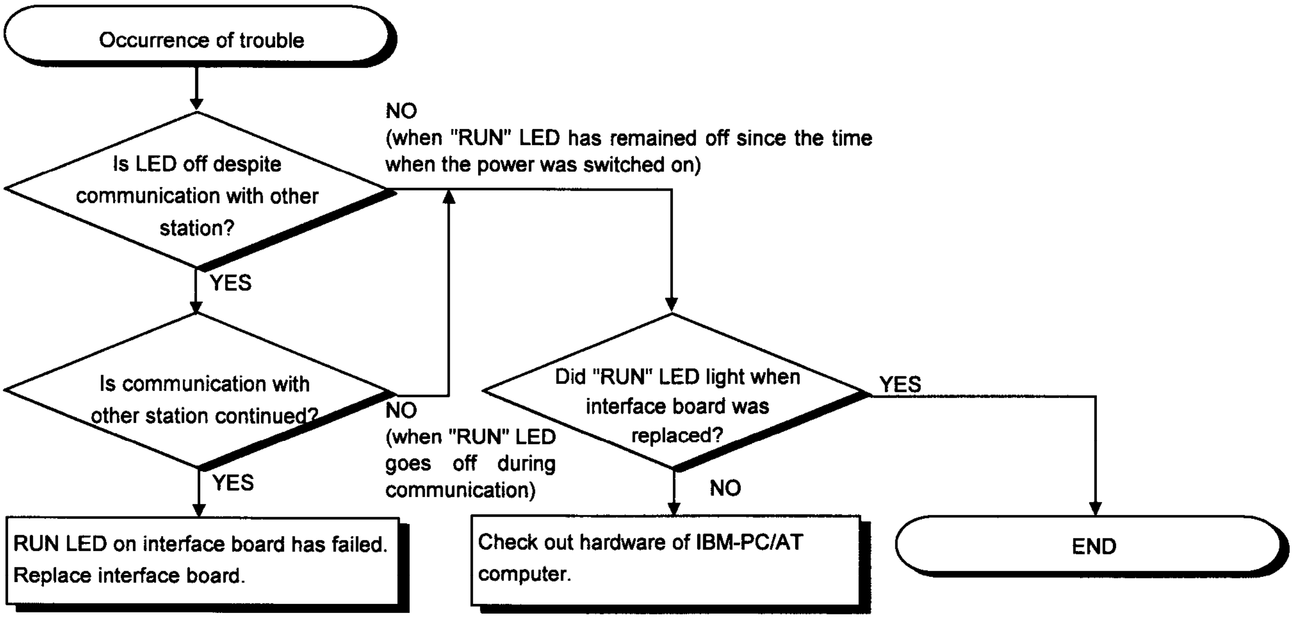
- Trouble in hardware
- A cable was disconnected while the test was in progress.
- A cable broke while the test was in progress.

9.4 Case 2

Illustrated below is the procedure for checking when the data link was not established although the interface board, control station and normal stations were connected.

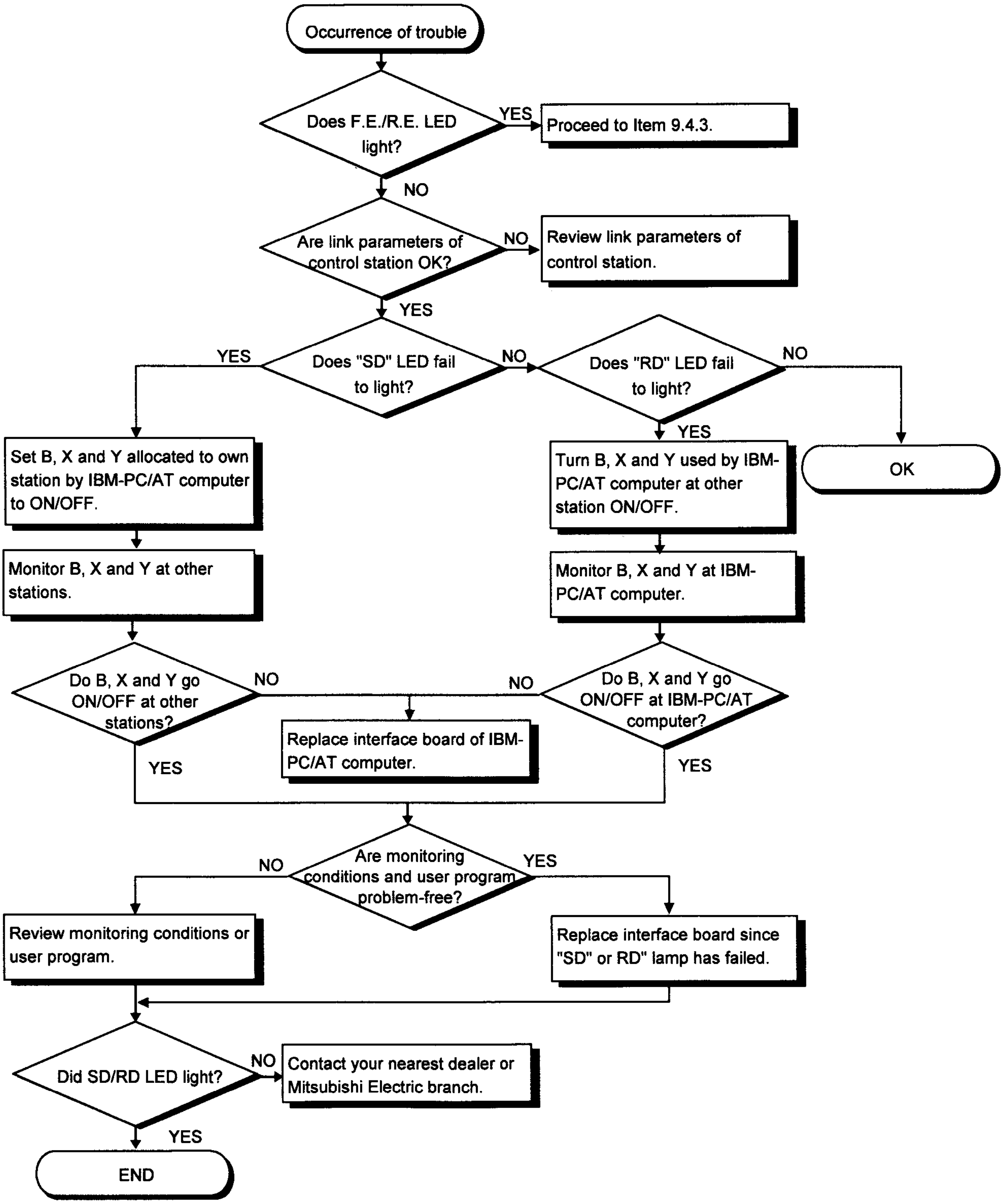

START
Check the LEDs on the MELSECNET interface boards.
Is the "RUN" LED off?
YES
Refer to Item 9.4.1 for check procedure flow when RUN LED is off.
NO
Has the "SD/RD" LED failed to light?
YES
Refer to Item 9.4.2 for check procedure flow when SD/RD LED fails to light.
NO
Is the F.E./R.E LED on?
YES
Refer to Item 9.4.3 for check procedure flow when F.E./R.E. LED lights.
NO
Start up the monitoringfunction (utility), and select network monitor on the
Has establishment of entire system data link failed?
YES
Refer to Item 9.4.4 for check procedure flow when entire system data link is disabled.
NO
Has establishment of data link of a specific station failed?
YES
Refer to Item 9.4.5 for check procedure flow when specific station data link is disabled.
NO
END

9.4.1 Check Procedure Flow When RUN LED is OFF

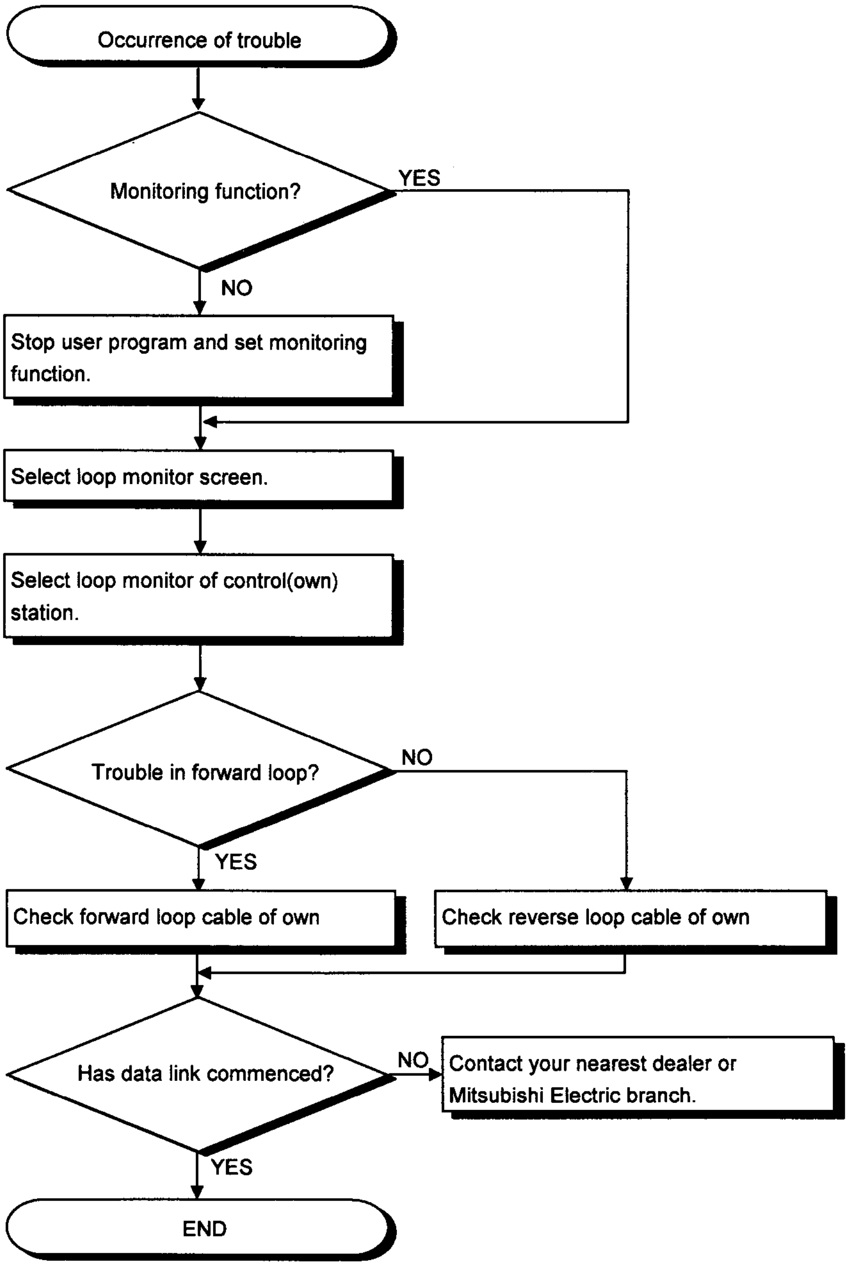

Occurrence of trouble
Is LED off despite communication with other station?
NO
(when "RUN" LED has remained off since the time when the power was switched on)
YES
Is communication with other station continued?
NO
(when "RUN" LED goes off during communication)
YES
Did "RUN" LED light when interface board was replaced?
YES
NO
RUN LED on interface board has failed. Replace interface board.
Check out hardware of IBM-PC/AT computer.
END

9.4.2 Check Procedure Flow When SD/RD LED Fails to Light

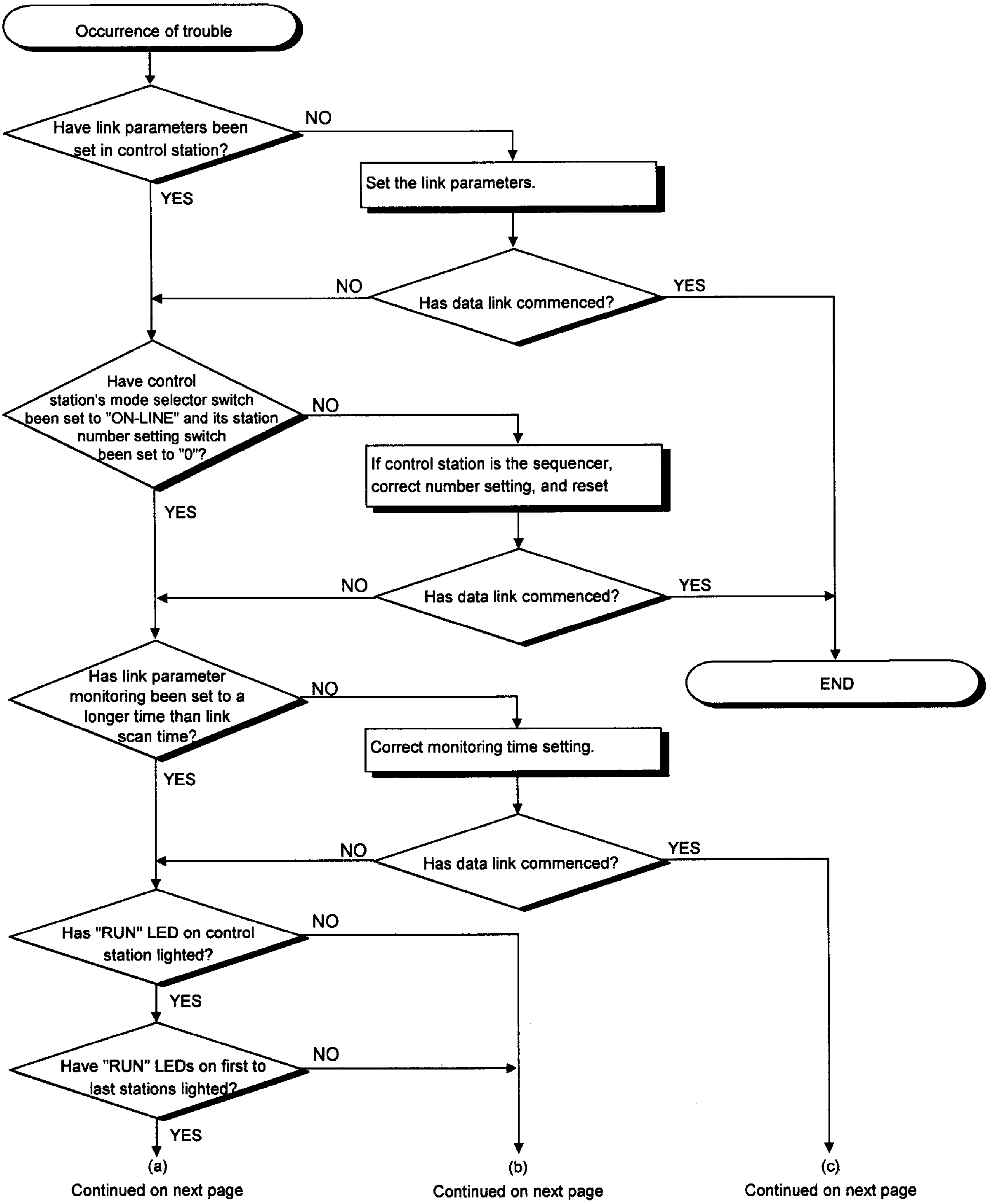
(1) "Sending data" or "Receiving data" appears on the board information screen.
This function is the same as the interface board's "SD" and "RD" LED display.

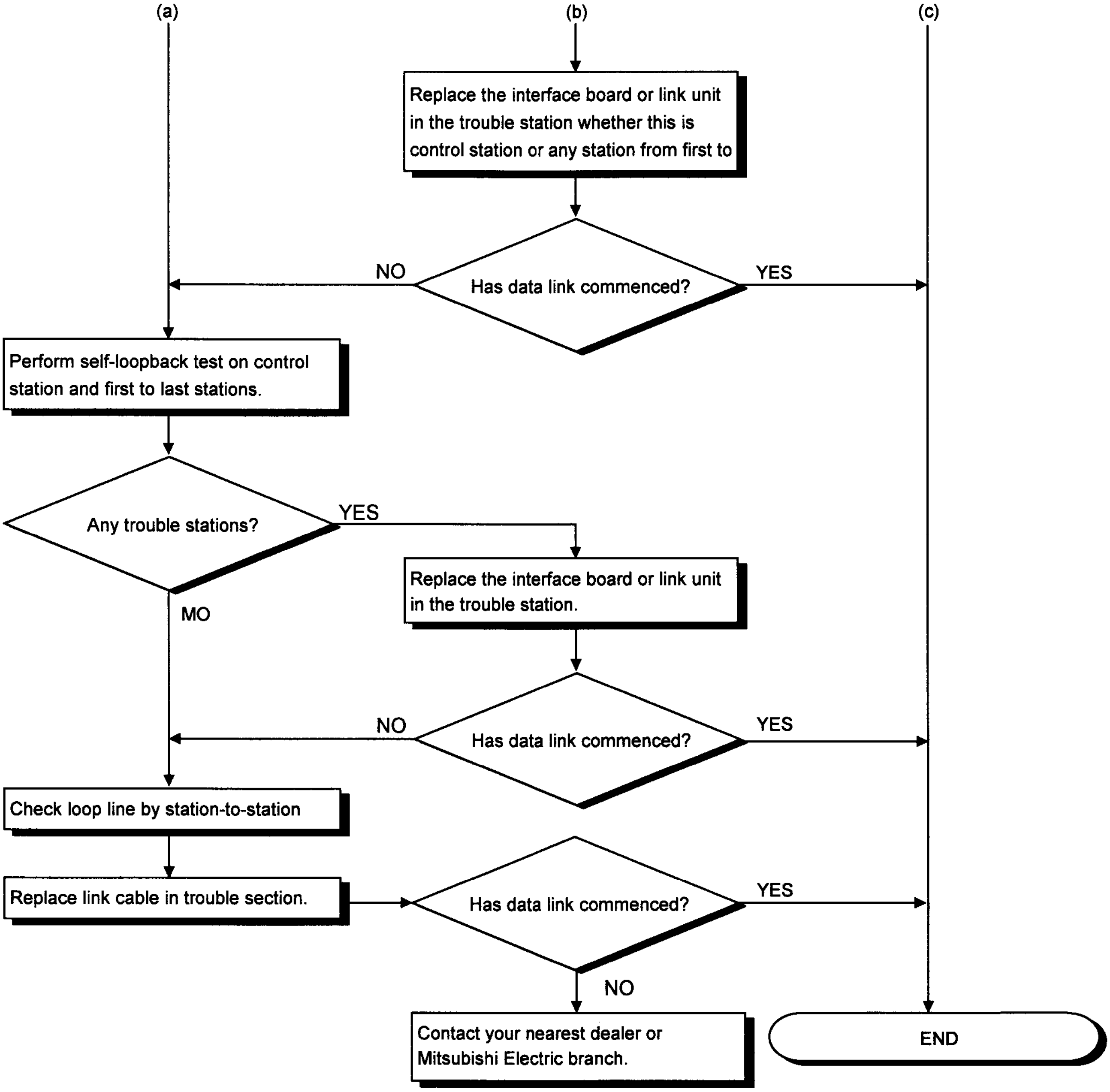

Occurrence of trouble
Does F.E./R.E. LED light?
YES
Proceed to Item 9.4.3.
NO
Are link parameters of control station OK?
NO
Review link parameters of control station.
YES
YES
Does "SD" LED fail to light?
NO
Does "RD" LED fail to light?
NO
Set B, X and Y allocated to own station by IBM-PC/AT computer to ON/OFF.
YES
Turn B, X and Y used by IBM-PC/AT computer at other station ON/OFF.
OK
Monitor B, X and Y at other stations.
Monitor B, X and Y at IBM-PC/AT computer.
Do B, X and Y go ON/OFF at other stations?
NO
NO
Do B, X and Y go ON/OFF at IBM-PC/AT computer?
Replace interface board of IBM-PC/AT computer.
YES
YES
NO
Are monitoring conditions and user program problem-free?
YES
Review monitoring conditions or user program.
Replace interface board since "SD" or RD" lamp has failed.
Did SD/RD LED light?
NO
Contact your nearest dealer or Mitsubishi Electric branch.
YES
END

9.4.3 Check Procedure Flow When F.E./R.E. LED Lights

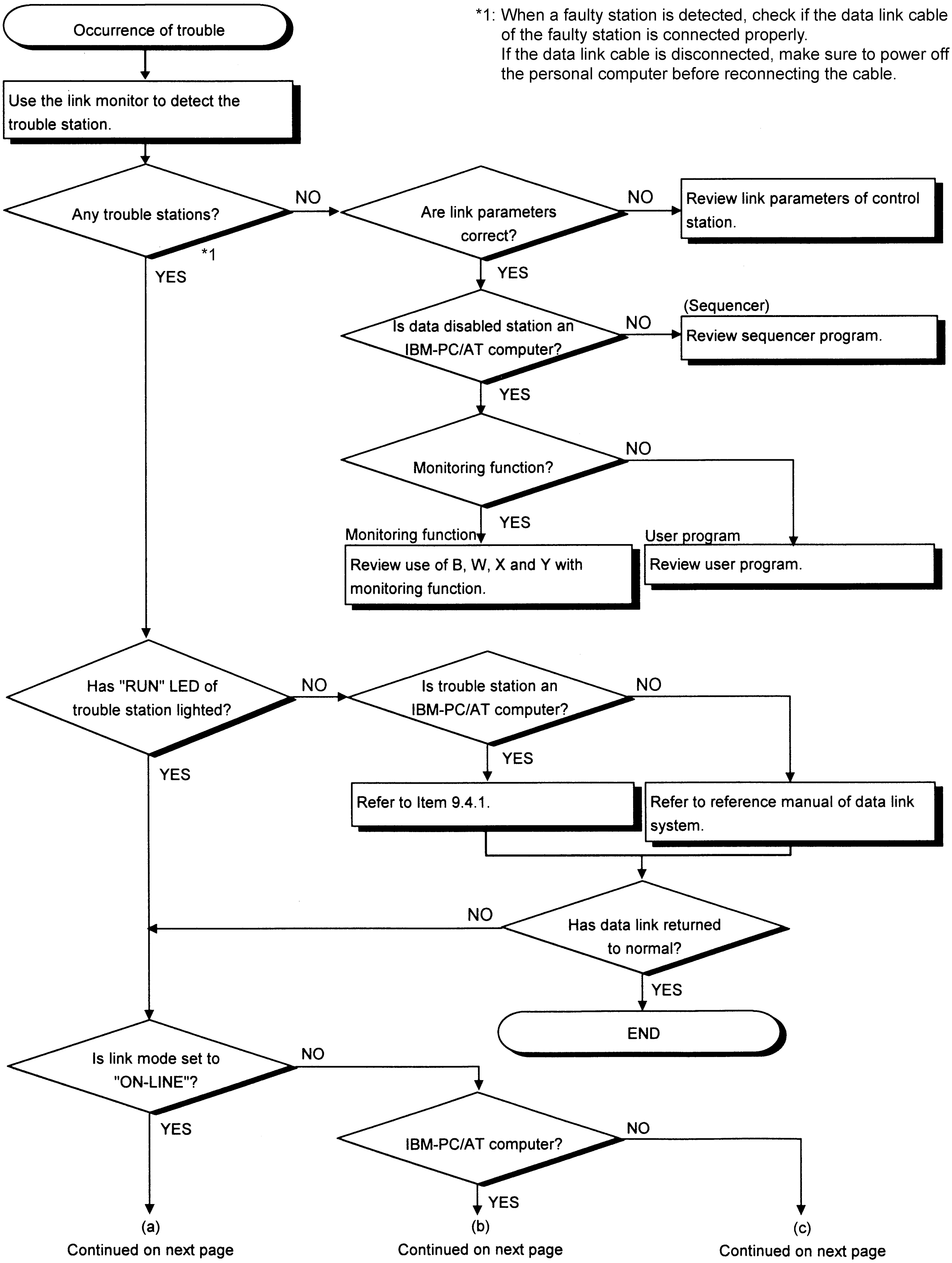

Occurrence of trouble
Monitoring function?
YES
NO
Stop user program and set monitoring function.
Select loop monitor screen.
Select loop monitor of control(own) station.
Trouble in forward loop?
NO
YES
Check forward loop cable of own
Check reverse loop cable of own
Has data link commenced?
NO
Contact your nearest dealer or Mitsubishi Electric branch.
YES
END

9.4.4 Check Procedure Flow When Entire System Data Link Is Disabled

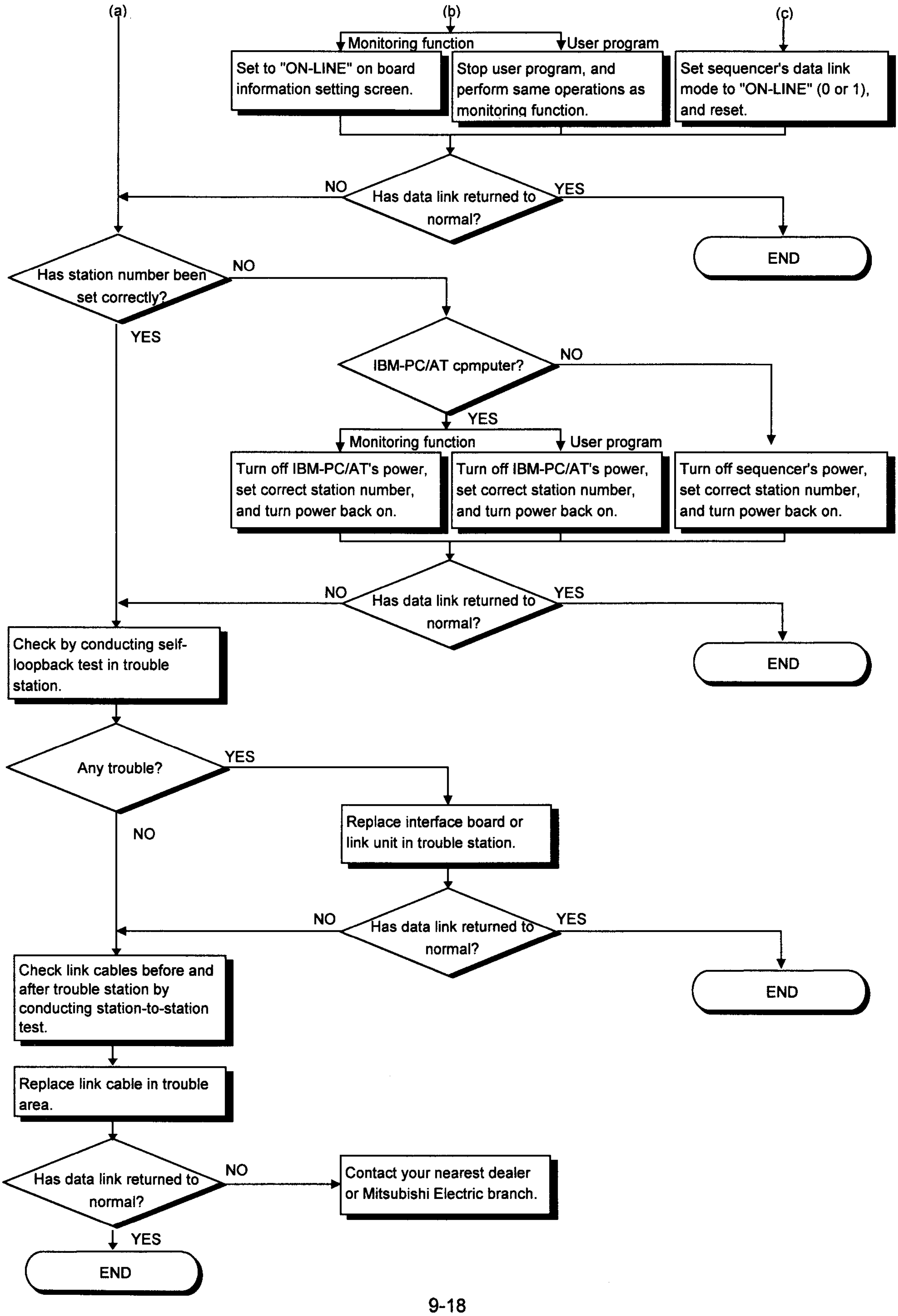

Occurrence of trouble
Have link parameters been set in control station?
NO
Set the link parameters.
YES
NO
Has data link commenced?
YES
Have control station's mode selector switch been set to "ON-LINE" and its station number setting switch been set to "0"?
NO
If control station is the sequencer, correct number setting, and reset
YES
NO
Has data link commenced?
YES
END
Has link parameter monitoring been set to a longer time than link scan time?
NO
Correct monitoring time setting.
YES
NO
Has data link commenced?
YES
Has "RUN" LED on control station lighted?
NO
YES
Have "RUN" LEDs on first to last stations lighted?
NO
YES
(a)
Continued on next page
(b)
Continued on next page
(c)
Continued on next page


(a)
(b)
(c)
Replace the interface board or link unit in the trouble station whether this is control station or any station from first to
Has data link commenced?
NO
YES
Perform self-loopback test on control station and first to last stations.
Any trouble stations?
YES
MO
Replace the interface board or link unit in the trouble station.
Has data link commenced?
NO
YES
Check loop line by station-to-station
Replace link cable in trouble section.
Has data link commenced?
YES
NO
Contact your nearest dealer or Mitsubishi Electric branch.
END

9.4.5 Check Procedure Flow When Specific Station Data Link Is Disabled


Occurrence of trouble
Use the link monitor to detect the trouble station.
Any trouble stations?
*1
YES
NO
Are link parameters correct?
NO
Review link parameters of control station.
YES
Is data disabled station an IBM-PC/AT computer?
NO
(Sequencer)
Review sequencer program.
YES
Monitoring function?
NO
YES
Monitoring function
Review use of B, W, X and Y with monitoring function.
User program
Review user program.
Has "RUN" LED of trouble station lighted?
NO
Is trouble station an IBM-PC/AT computer?
NO
YES
Refer to Item 9.4.1.
Refer to reference manual of data link system.
YES
Has data link returned to normal?
NO
YES
END
Is link mode set to "ON-LINE"?
NO
YES
IBM-PC/AT computer?
NO
YES
(a)
Continued on next page
(b)
Continued on next page
(c)
Continued on next page

*1: When a faulty station is detected, check if the data link cable of the faulty station is connected properly.
If the data link cable is disconnected, make sure to power off the personal computer before reconnecting the cable.


(a)
(b)
(c)
Monitoring function
User program
Set to "ON-LINE" on board information setting screen.
Stop user program, and perform same operations as monitoring function.
Set sequencer's data link mode to "ON-LINE" (0 or 1), and reset.
Has data link returned to normal?
NO
YES
END
Has station number been set correctly?
NO
YES
IBM-PC/AT cpmputer?
NO
YES
Monitoring function
User program
Turn off IBM-PC/AT's power, set correct station number, and turn power back on.
Turn off IBM-PC/AT's power, set correct station number, and turn power back on.
Turn off sequencer's power, set correct station number, and turn power back on.
Has data link returned to normal?
NO
YES
END
Check by conducting self-loopback test in trouble station.
Any trouble?
YES
NO
Replace interface board or link unit in trouble station.
Has data link returned to normal?
NO
YES
END
Check link cables before and after trouble station by conducting station-to-station test.
Replace link cable in trouble area.
Has data link returned to normal?
NO
YES
Contact your nearest dealer or Mitsubishi Electric branch.
END

9.5 Case 3

Questions about the occurrence of trouble while a data link is established and answers to those questions are provided below.

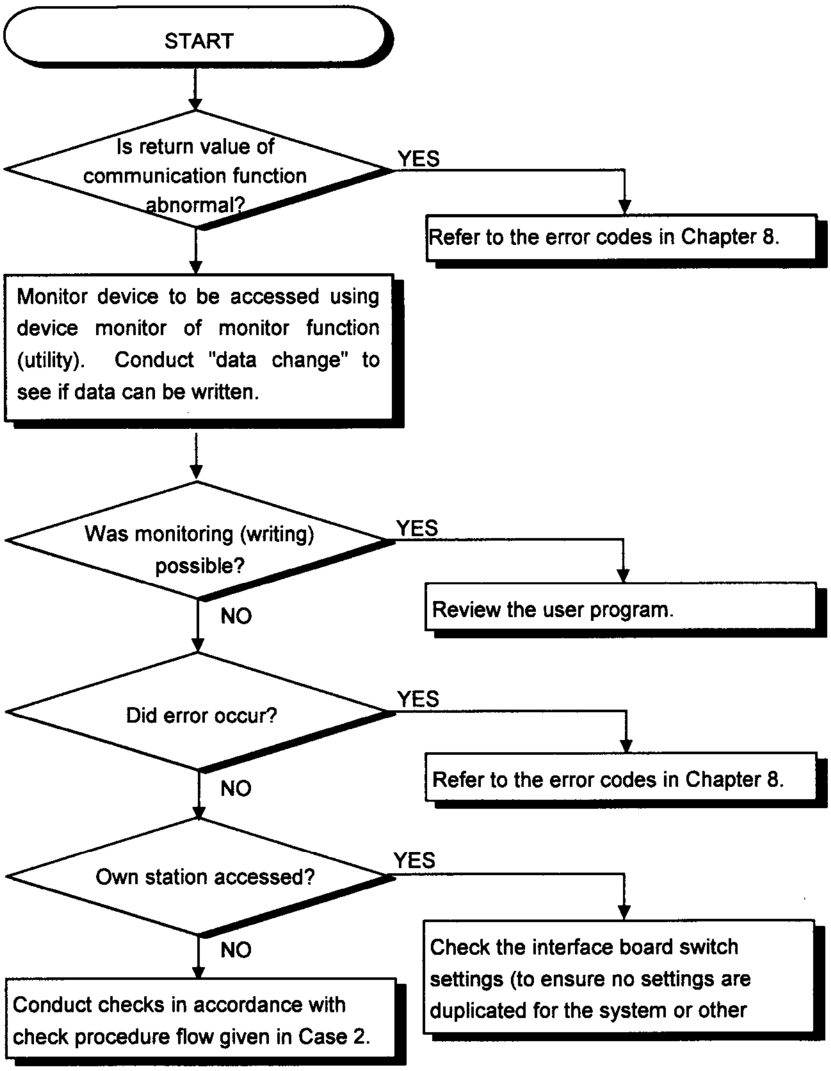
Q1: Why is a value differing from that planned has been input to a specific link device (B, W, X or Y)?

A1: Check out the following.

1) Check whether there are any stations with the link trouble using network monitor of the monitor function.
2) At the control station, check the allocation range for the link parameters.
3) In the sequencer, check the range of the devices used by the sequence program.
4) In the user program, check the argument data of the communication functions involved in accessing specific link devices.

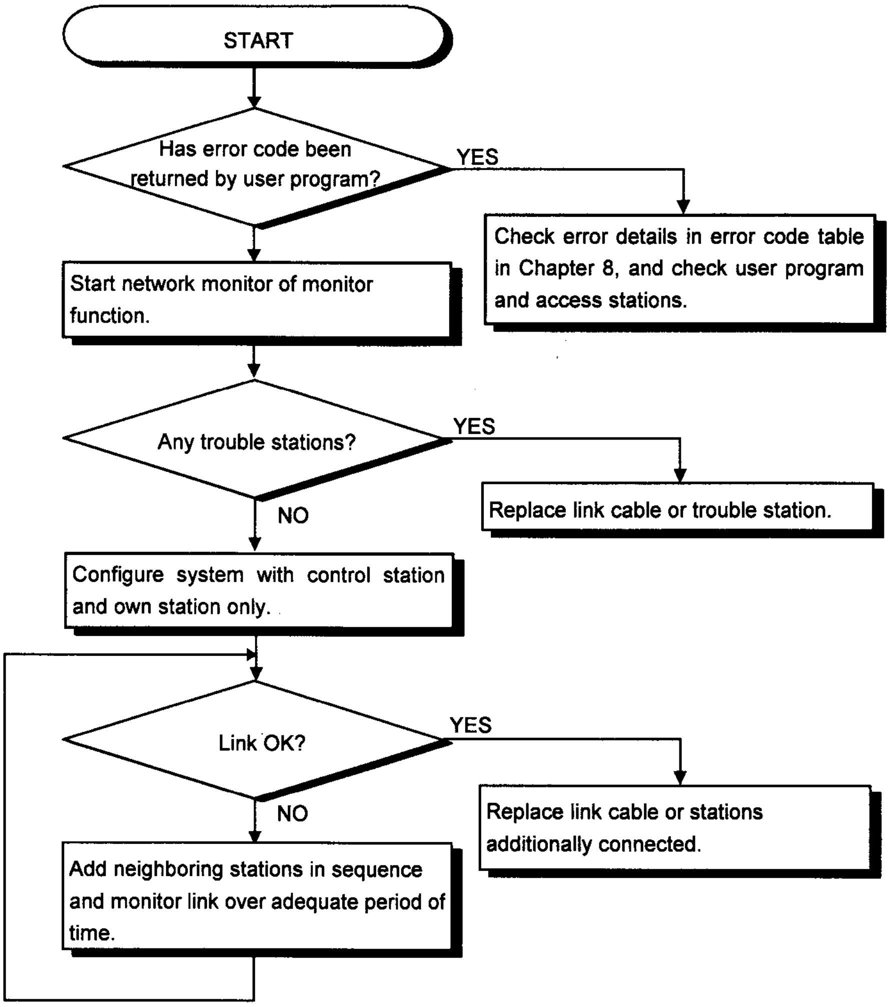
Q2: Why is it not possible to write or read data even although the communication functions were used in the user program for device access?

A2: Conduct checks in accordance with the check procedure flow given below.


START
Is return value of communication function abnormal?
YES
Refer to the error codes in Chapter 8.
Monitor device to be accessed using device monitor of monitor function (utility). Conduct "data change" to see if data can be written.
Was monitoring (writing) possible?
YES
Review the user program.
NO
Did error occur?
YES
Refer to the error codes in Chapter 8.
NO
Own station accessed?
YES
Check the interface board switch settings (to ensure no settings are duplicated for the system or other
NO
Conduct checks in accordance with check procedure flow given in Case 2.

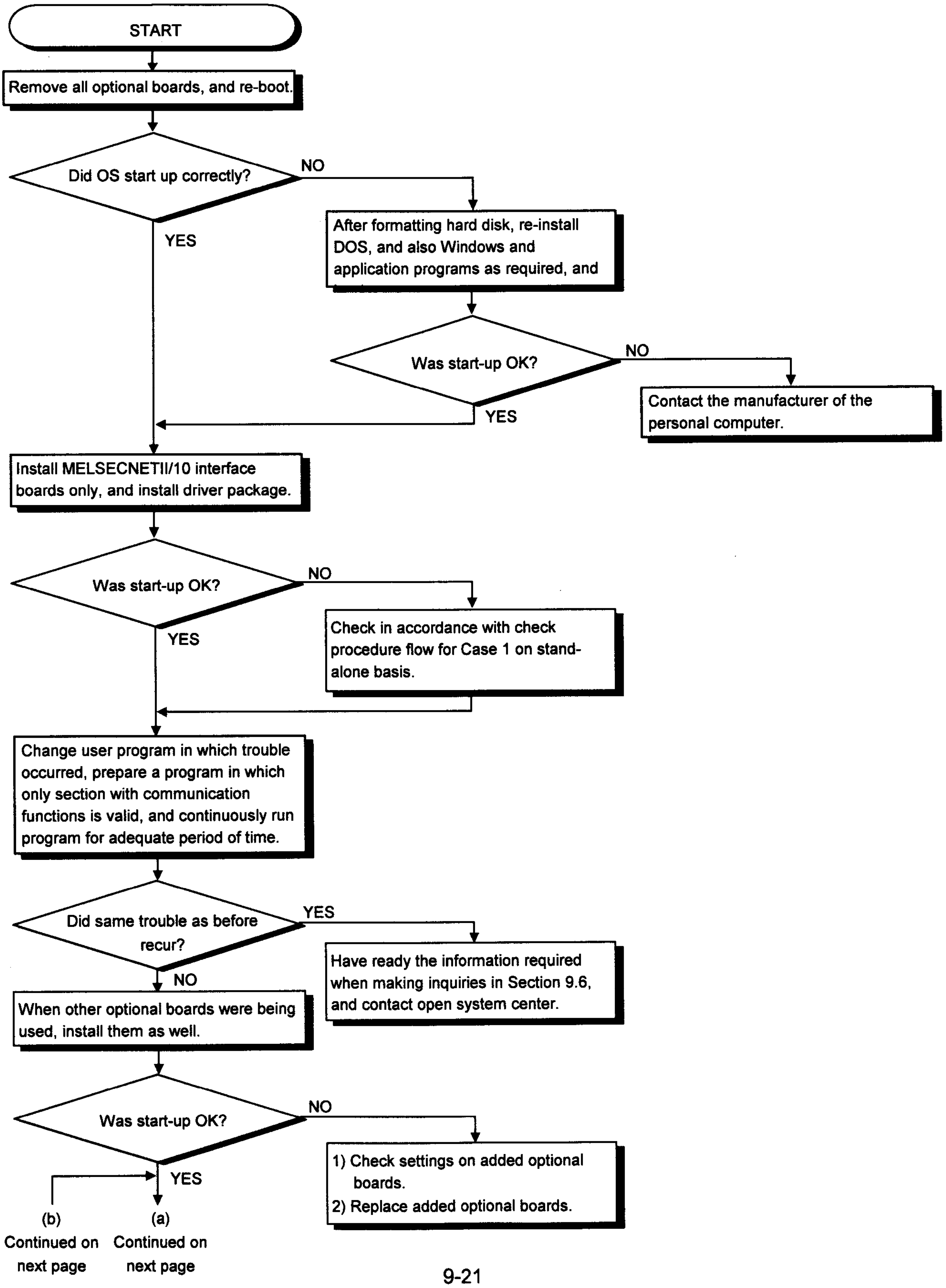
Q3: Why is communication disabled from time to time while the user program is running?

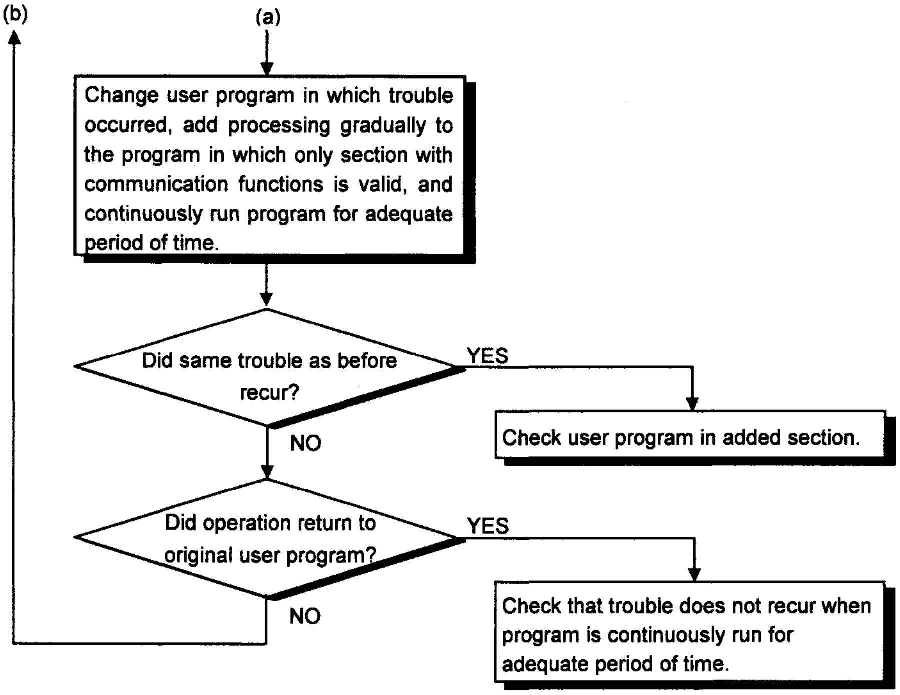
A3: Conduct checks in accordance with the check procedure flow given below.


START
Has error code been returned by user program?
YES
Check error details in error code table in Chapter 8, and check user program and access stations.
Start network monitor of monitor function.
Any trouble stations?
YES
Replace link cable or trouble station.
NO
Configure system with control station and own station only.
Link OK?
YES
Replace link cable or stations additionally connected.
NO
Add neighboring stations in sequence and monitor link over adequate period of time.

Q4: Why did system resetting or system hang-up occur while the user program was running in DOS/Windows 3.1?

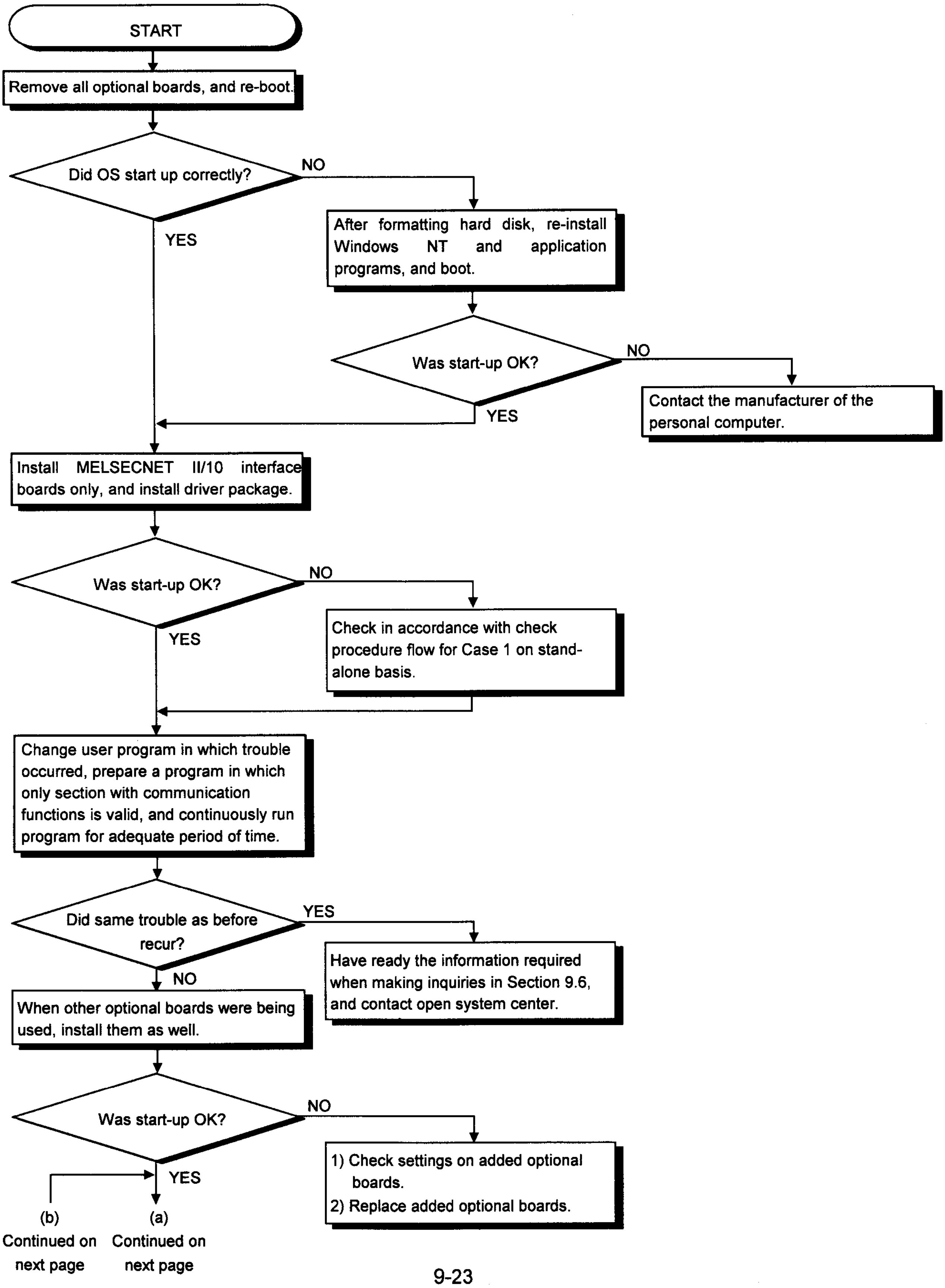
A4: Conduct checks in accordance with the flow below.

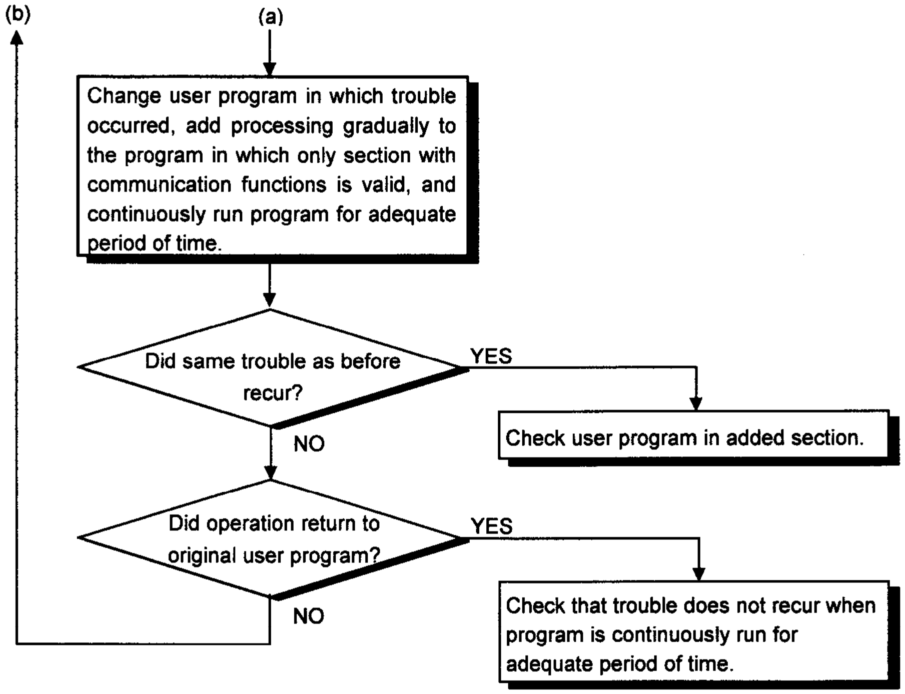

START
Remove all optional boards, and re-boot.
Did OS start up correctly?
NO
After formatting hard disk, re-install DOS, and also Windows and application programs as required, and
Was start-up OK?
NO
Contact the manufacturer of the personal computer.
YES
YES
Install MELSECNETII/10 interface boards only, and install driver package.
Was start-up OK?
NO
Check in accordance with check procedure flow for Case 1 on stand-alone basis.
YES
Change user program in which trouble occurred, prepare a program in which only section with communication functions is valid, and continuously run program for adequate period of time.
Did same trouble as before recur?
YES
Have ready the information required when making inquiries in Section 9.6, and contact open system center.
NO
When other optional boards were being used, install them as well.
Was start-up OK?
NO
1) Check settings on added optional boards.
2) Replace added optional boards.
YES
(b)
Continued on next page
(a)
Continued on next page


(b)
(a)
Change user program in which trouble occurred, add processing gradually to the program in which only section with communication functions is valid, and continuously run program for adequate period of time.
Did same trouble as before recur?
YES
Check user program in added section.
NO
Did operation return to original user program?
YES
Check that trouble does not recur when program is continuously run for adequate period of time.
NO

Q5: Why did system down (blue screen) or system resetting occur while the user program was running in Windows NT?

A5: Conduct checks in accordance with the check procedure flow given below.


START
Remove all optional boards, and re-boot.
Did OS start up correctly?
NO
YES
After formatting hard disk, re-install Windows NT and application programs, and boot.
Was start-up OK?
NO
YES
Contact the manufacturer of the personal computer.
Install MELSECNET II/10 interface boards only, and install driver package.
Was start-up OK?
NO
YES
Check in accordance with check procedure flow for Case 1 on stand-alone basis.
Change user program in which trouble occurred, prepare a program in which only section with communication functions is valid, and continuously run program for adequate period of time.
Did same trouble as before recur?
YES
NO
Have ready the information required when making inquiries in Section 9.6, and contact open system center.
When other optional boards were being used, install them as well.
Was start-up OK?
NO
YES
1) Check settings on added optional boards.
2) Replace added optional boards.
(b)
Continued on next page
(a)
Continued on next page


(b)
(a)
Change user program in which trouble occurred, add processing gradually to the program in which only section with communication functions is valid, and continuously run program for adequate period of time.
Did same trouble as before recur?
YES
NO
Check user program in added section.
Did operation return to original user program?
YES
NO
Check that trouble does not recur when program is continuously run for adequate period of time.

9.6 Information Required When Making Inquiries

When making inquiries with Mitsubishi Electric to obtain information on what to do when the board appears to have failed, please be ready to supply details on the conditions and statuses listed below.

(1) Trouble description (provide specific details)
Example: When starting up the board after the power was turned on, the "No Response From Board" message was displayed, and the board failed to start up.

(2) Manufacturer of personal computer, name and model name of personal computer

(3) Name of operating system such as Windows NT3.51 or Windows 3.1

(4) Position of slot in which board is installed

(5) Whether optional boards of other makes are present

(6) If optional boards of other makes are present, provide the following details for each board concerned.

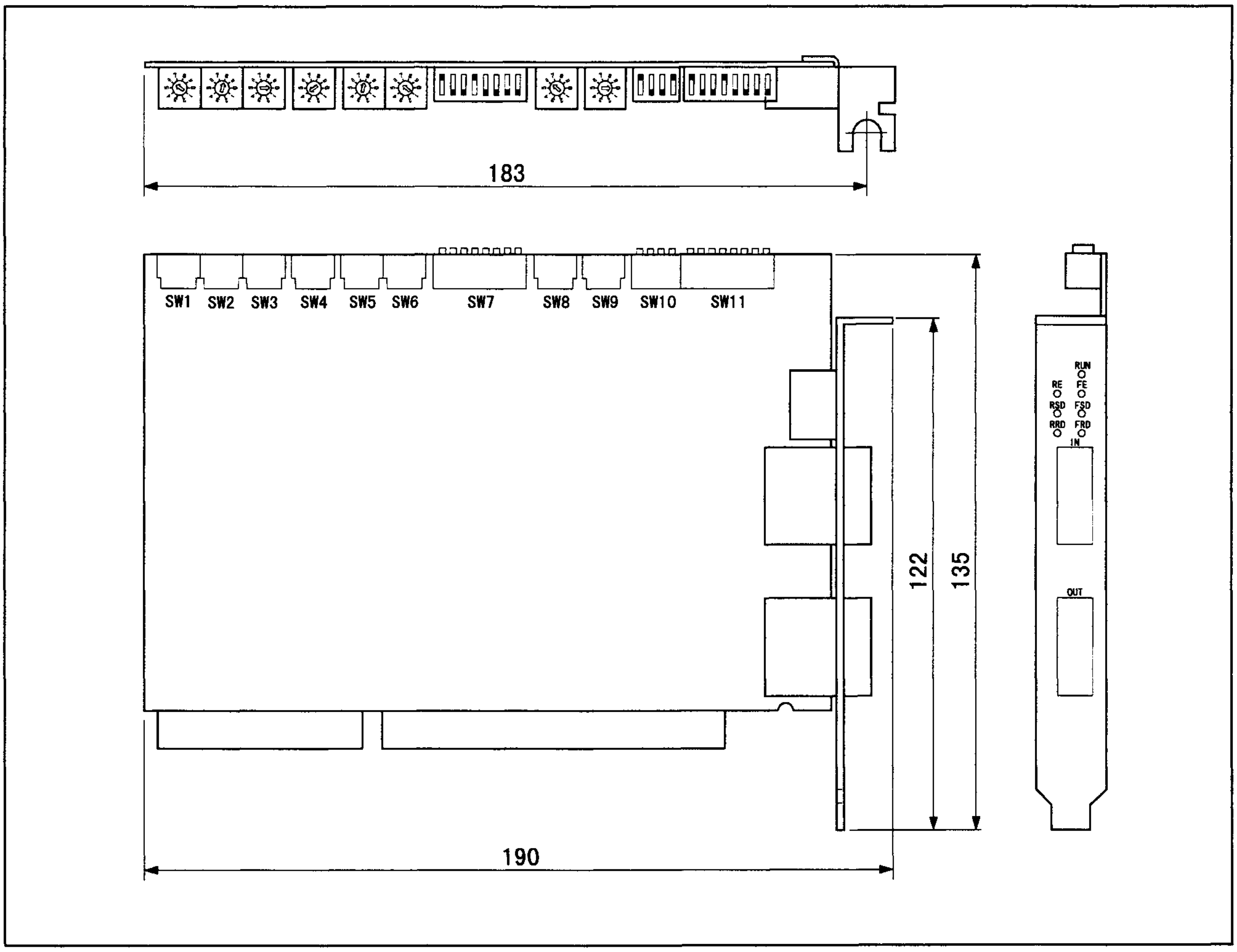
- Model name of board, manufacturer of board
- Memory addresses (head address and size occupied), I/O addresses (head address and size occupied), IRQ numbers, DMA numbers

(7) Whether checks have been conducted using other personal computers

APPENDIX

Appendix 1 Outline Drawings

Appendix 1.1 A70BDE-J71LP23GE



Trademarks

Mitsubishi Electric offers no guarantees are made for any software products sold on the market which are compatible with DOS or Windows NT and which are introduced by Mitsubishi Electric.

WARRANTY

Please confirm the following product warranty details before using this product.

1. Gratis Warranty Term and Gratis Warranty Range

If any faults or defects (hereinafter "Failure") found to be the responsibility of Mitsubishi occurs during use of the product within the gratis warranty term, the product shall be repaired at no cost via the sales representative or Mitsubishi Service Company.
However, if repairs are required onsite at domestic or overseas location, expenses to send an engineer will be solely at the customer's discretion. Mitsubishi shall not be held responsible for any re-commissioning, maintenance, or testing on-site that involves replacement of the failed module.

[Gratis Warranty Term]

The gratis warranty term of the product shall be for one year after the date of purchase or delivery to a designated place.
Note that after manufacture and shipment from Mitsubishi, the maximum distribution period shall be six (6) months, and the longest gratis warranty term after manufacturing shall be eighteen (18) months. The gratis warranty term of repair parts shall not exceed the gratis warranty term before repairs.

[Gratis Warranty Range]

(1) The range shall be limited to normal use within the usage state, usage methods and usage environment, etc., which follow the conditions and precautions, etc., given in the instruction manual, user's manual and caution labels on the product.

(2) Even within the gratis warranty term, repairs shall be charged for in the following cases.

1. Failure occurring from inappropriate storage or handling, carelessness or negligence by the user. Failure caused by the user's hardware or software design.
2. Failure caused by unapproved modifications, etc., to the product by the user.
3. When the Mitsubishi product is assembled into a user's device, Failure that could have been avoided if functions or structures, judged as necessary in the legal safety measures the user's device is subject to or as necessary by industry standards, had been provided.
4. Failure that could have been avoided if consumable parts (battery, backlight, fuse, etc.) designated in the instruction manual had been correctly serviced or replaced.
5. Failure caused by external irresistible forces such as fires or abnormal voltages, and Failure caused by force majeure such as earthquakes, lightning, wind and water damage.
6. Failure caused by reasons unpredictable by scientific technology standards at time of shipment from Mitsubishi.
7. Any other failure found not to be the responsibility of Mitsubishi or that admitted not to be so by the user.

2. Onerous repair term after discontinuation of production

(1) Mitsubishi shall accept onerous product repairs for seven (7) years after production of the product is discontinued. Discontinuation of production shall be notified with Mitsubishi Technical Bulletins, etc.

(2) Product supply (including repair parts) is not available after production is discontinued.

3. Overseas service

Overseas, repairs shall be accepted by Mitsubishi's local overseas FA Center. Note that the repair conditions at each FA Center may differ.

4. Exclusion of loss in opportunity and secondary loss from warranty liability

Regardless of the gratis warranty term, Mitsubishi shall not be liable for compensation of damages caused by any cause found not to be the responsibility of Mitsubishi, loss in opportunity, lost profits incurred to the user by Failures of Mitsubishi products, special damages and secondary damages whether foreseeable or not , compensation for accidents, and compensation for damages to products other than Mitsubishi products, replacement by the user, maintenance of on-site equipment, start-up test run and other tasks.

5. Changes in product specifications

The specifications given in the catalogs, manuals or technical documents are subject to change without prior notice.

6. Product application

(1) In using the Mitsubishi MELSEC programmable controller, the usage conditions shall be that the application will not lead to a major accident even if any problem or fault should occur in the programmable controller device, and that backup and fail-safe functions are systematically provided outside of the device for any problem or fault.

(2) The Mitsubishi programmable controller has been designed and manufactured for applications in general industries, etc. Thus, applications in which the public could be affected such as in nuclear power plants and other power plants operated by respective power companies, and applications in which a special quality assurance system is required, such as for Railway companies or Public service purposes shall be excluded from the programmable controller applications.
In addition, applications in which human life or property that could be greatly affected, such as in aircraft, medical applications, incineration and fuel devices, manned transportation, equipment for recreation and amusement, and safety devices, shall also be excluded from the programmable controller range of applications.
However, in certain cases, some applications may be possible, providing the user consults their local Mitsubishi representative outlining the special requirements of the project, and providing that all parties concerned agree to the special circumstances, solely at the users discretion.

For IBM-PC/AT Computers Model A70BDE-J71LP23GE
MELSECNET/10 Interface Board

User’s Manual

MODEL	A70BDE-J71LP23GEU
MODEL CODE	13J898
IB(NA)-66749-D(0706)MEE	

MITSUBISHI ELECTRIC CORPORATION

HEAD OFFICE : TOKYO BUILDING, 2-7-3 MARUNOUCHI, CHIYODA-KU, TOKYO 100-8310, JAPAN
NAGOYA WORKS : 1-14 , YADA-MINAMI 5-CHOME , HIGASHI-KU, NAGOYA , JAPAN

When exported from Japan, this manual does not require application to the Ministry of Economy, Trade and Industry for service transaction permission.

Specifications subject to change without notice.