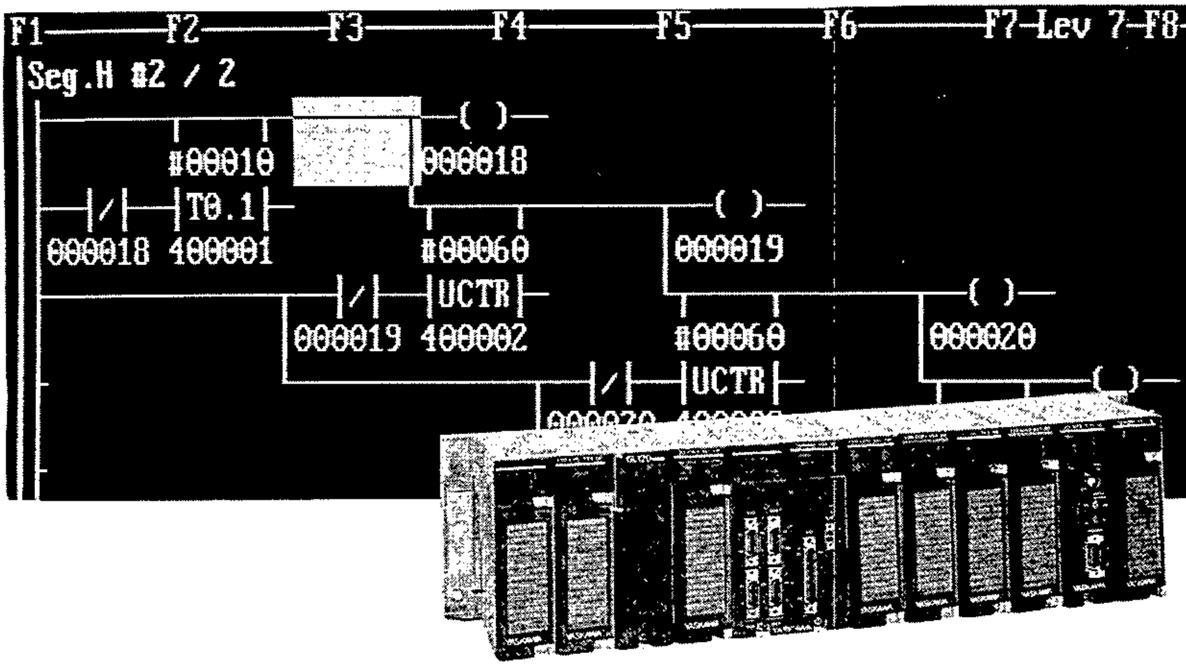
YASKAWA

MEMOCON GL120, GL130

MEMOBUS PLUS SA85 NETWORK ADAPTER

USER'S MANUAL



YASKAWA

MANUAL NO. SIEZ-C825-70.6

Manual Contents

This manual describes the specifications and applications of the SA85 Network Adapter used for the MEMOBUS PLUS industrial communications systems for GL120 and GL130 PLCs.

Please read this manual carefully and be sure you understand the information provided before attempting to use the MEMOBUS PLUS SA85 Network Adapter.

Visual Aids

The following aids are used to indicate certain types of information for easier reference.

☞	Indicates references for additional information.
IMPORTANT	Indicates important information that should be memorized.
◀EXAMPLE▶	Indicates application examples.
INFO	Indicates supplemental information.
♦ SUMMARY	Indicates a summary of the important points of explanations.
Note	Indicates inputs, operations, and other information required for correct operation but that will not cause damage to the device.
TERMS	Indicates definitions of terms used in the manual.

NOTICE

The following conventions are used to indicate precautions in this manual. Failure to heed precautions provided in this manual can result in injury to people or damage to the products.

⚠ **WARNING**	Indicates precautions that, if not heeded, could possibly result in loss of life or serious injury.
⚠ **Caution**	Indicates precautions that, if not heeded, could result in relatively serious or minor injury, damage to the product, or faulty operation.

©Yaskawa, 1996

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form, or by any means, mechanical, electronic, photocopying, recording, or otherwise, without the prior written permission of Yaskawa. No patent liability is assumed with respect to the use of the information contained herein. Moreover, because Yaskawa is constantly striving to improve its high-quality products, the information contained in this manual is subject to change without notice. Every precaution has been taken in the preparation of this manual. Nevertheless, Yaskawa assumes no responsibility for errors or omissions. Neither is any liability assumed for damages resulting from the use of the information contained in this publication.

CONTENTS

CHAPTER 1 Introduction and Precautions 1-1
1.1 Overview of Manual 1-2
1.2 Precautions 1-3
1.2.1 Safety Precautions 1-3
1.2.2 Installation Precautions 1-4
1.2.3 Wiring Precautions 1-4
1.2.4 Application Safety Precautions 1-5
1.2.5 Maintenance 1-6
1.3 Using this Manual 1-7

CHAPTER 2 MEMOBUS PLUS Network Overview 2-1
2.1 MEMOBUS PLUS SA85 Network Adapter 2-2
2.1.1 Software 2-2
2.1.2 README File (README.DOC) 2-2
2.1.3 Sample Programs 2-2
2.2 Overview of MEMOBUS PLUS Network 2-3
2.3 SA85 Network Adapter 2-4
2.4 MEMOBUS PLUS Communications Method 2-5
2.4.1 MEMOBUS PLUS Routing Paths 2-5
2.4.2 Routing Paths for GL120 and GL130 PLCs 2-6
2.4.3 Routing Path for SA85 Network Adapters 2-6
2.4.4 Routing Path for BM85 Bridge/Multiplexer 2-6
2.5 MEMOBUS PLUS Communications Processing 2-7
2.5.1 Path Types 2-7
2.5.2 Number of Paths 2-8
2.5.3 Queuing 2-8
2.6 MEMOBUS Commands 2-9
2.7 Overview of Programs Provided 2-11

CHAPTER 3 SA85 Network Adapter 3-1
3.1 SA85 and Operating Environment 3-2
3.1.1 Hardware Components 3-2
3.1.2 Software Components 3-2
3.1.3 System Design 3-3
3.2 Installation Outline 3-4
3.2.1 Installation Outline 3-4
3.2.2 Adding and Deleting the Node during Operation 3-4
3.3 Switch Layout 3-5
3.4 Setting the MEMOBUS PLUS Address Switch 3-6
3.5 Setting the Memory Address Switch 3-7
3.6 Installing the SA85 Board 3-9
3.7 Indicators 3-10
3.8 MEMOBUS PLUS Port Labels 3-12
3.9 Initializing the SA85 3-13
3.10 SA85 Specifications 3-14
3.10.1 General Specifications 3-14
3.10.2 MEMOBUS PLUS Port Specifications 3-15

CHAPTER 4 Installing Software 4-1
4.1 Installing the Device Driver 4-2

CONTENTS

4.1.1 Copying MBPHOST.SYS 4-2
4.1.2 Changing CONFIG.SYS 4-2
4.1.3 Host Computer Initialization Processing 4-4
4.2 Installing Other Files 4-5
4.2.1 Installing the MBPSTAT and \TESTSRC Programs 4-5
4.2.2 Installing \MBPHINC and \NETLIB Files 4-5

CHAPTER 5 Using the NetBIOS Functions 5-1
5.1 MEMOBUS PLUS NetBIOS Environment 5-2
5.1.1 NetBIOS Environment 5-2
5.1.2 Network Control Block 5-3
5.2 Overview of NetBIOS Commands 5-4
5.2.1 NETLIB Functions 5-4
5.2.2 Path Required 5-4
5.2.3 Message Buffer 5-5
5.2.4 Function Definitions and Prototypes 5-5
5.2.5 NCB Structure Definition and NetBIOS Command Codes 5-5
5.2.6 Function Return Values 5-5
5.3 Return Values 5-6
5.4 NetBIOS Functions 5-8
5.4.1 ncb_cancel () 5-8
5.4.2 ncb_close () 5-9
5.4.3 ncb_open () 5-9
5.4.4 ncb_receive () 5-11
5.4.5 ncb_receive_datagram () 5-12
5.4.6 ncb_receive_wait () 5-13
5.4.7 ncb_reset () 5-14
5.4.8 ncb_sa85off () 5-14
5.4.9 ncb_send () 5-15
5.4.10 ncb_send_datagram () 5-16
5.4.11 ncb_set_slave_login () 5-17
5.4.12 ncb_set_sw_interrupt () 5-18
5.4.13 ncb_status () 5-18

CHAPTER 6 Changing Software Interrupt Number 5C 6-1
6.1 Application Interrupt 6-2
6.2 MBPSTAT Interrupts 6-3
6.3 Sample Program Interrupts 6-3

CHAPTER 7 Include Files 7-1
7.1 NETBIOS.H 7-1
7.2 NETLIB.H 7-5

CHAPTER 8 MBPSTAT Utility 8-1
8.1 MBPSTAT Program 8-2
8.1.1 Starting the MBPSTAT Program 8-2

CONTENTS

8.1.2 Selecting the Network to be Analyzed 8-2
8.1.3 Network Selection Examples 8-3
8.2 MBPSTAT Menu Screen 8-4
8.3 MBPSTAT Menu Items 8-5
8.3.1 Set Routing Parameters 8-5
8.3.2 Monitor Network Activity 8-5
8.3.3 Read Global Data 8-6
8.3.4 Global Data Present Table 8-7
8.3.5 Node Active Station Table 8-7
8.3.6 Node Error Statistics 8-8
8.3.7 Token Station Table 8-8
8.3.8 Token Owner Work Table 8-9
8.3.9 Current Internal Path Transactions 8-9
8.3.10 Show Node Personality 8-10
8.3.11 Quit 8-10

CHAPTER 9 Sample Programs 9-1
9.1 Sample Program Overview 9-2
9.1.1 What is a Sample Program? 9-2
9.1.2 Minimum System Configuration 9-3
9.1.3 Interactive Programs 9-3
9.1.4 Node Address 9-3
9.1.5 5C Interrupt 9-3
9.1.6 Running Sample Programs 9-4
9.2 Sample Program Details 9-5
9.2.1 MBPCHECK 9-5
9.2.2 READNET 9-5
9.2.3 TEST4 9-6
9.2.4 TEST4B 9-6
9.2.5 TEST4C and TESTSLAV 9-7
9.2.6 GLOBTEST 9-8
9.2.7 READNODE 9-8
9.3 Board Reset Programs 9-10
9.3.1 Description 9-10
9.3.2 SA85OFF.EXE 9-10
9.3.3 BDRESET.EXE 9-11

CHAPTER 10 MEMOBUS Commands 10-1
10.1 MEMOBUS Protocol for MEMOBUS PLUS 10-3
10.1.1 Relationship between MEMOBUS Master and Slaves 10-3
10.1.2 Creating MEMOBUS Commands/Response Messages 10-3
10.2 MEMOBUS Communications 10-4
10.3 List of MEMOBUS Commands 10-5
10.4 Specifying Digital and Register References 10-6
10.5 Basic Functions 10-7
10.5.1 Reading Coil Status 10-7
10.5.2 Reading the Status of Specific Coils 10-8

CONTENTS

10.5.3 Modifying Status of a Single Coil 10-9
10.5.4 Modifying Status of Multiple Coils 10-10
10.5.5 Reading Input Relay Status 10-11
10.5.6 Reading Input Register Contents 10-12
10.5.7 Reading Holding Register Contents 10-13
10.5.8 Writing to a Single Holding Register 10-14
10.5.9 Writing to Multiple Holding Registers 10-15
10.5.10 Writing a Mask to Holding Registers 10-17
10.5.11 Reading/Writing to Multiple Holding Registers 10-18
10.6 Expansion Functions 10-20
10.6.1 Reading Link Coil Status 10-20
10.6.2 Modifying Status of a Single Link Coil 10-21
10.6.3 Modifying Status of Multiple Link Coils 10-22
10.6.4 Reading Link Register Contents 10-23
10.6.5 Writing to a Single Link Register 10-25
10.6.6 Writing to Multiple Link Registers 10-26
10.6.7 Reading Constant Register Contents 10-27
10.6.8 Writing to a Single Constant Register 10-28
10.6.9 Writing to Multiple Constant Registers 10-29
10.7 Special Functions 10-31
10.7.1 Loopback Test 10-31
10.7.2 Reading Event Counter 10-32
10.7.3 Reading Communications Status 10-32
10.7.4 Reading Equipment Status 10-34
10.7.5 Reading the FIFO Register Contents 10-37
10.7.6 Reading Special Reference Status 10-38
10.7.7 Modifying Special Reference Status 10-39

Introduction and Precautions

1

This chapter gives precautions and warnings concerning the use of this product and the manual. **You must read this chapter before reading the rest of the manual or using the product.**

1.1 Overview of Manual 1-2

1.2 Precautions 1-3

1.2.1 Safety Precautions 1-3
1.2.2 Installation Precautions 1-4
1.2.3 Wiring Precautions 1-4
1.2.4 Application Safety Precautions 1-5
1.2.5 Maintenance 1-6

1.3 Using this Manual 1-7

1.1 Overview of Manual

- This manual describes how to install and use the SA85 Network Adapter. The SA85 Network Adapter is supplied by Modicon, U.S., through a technical coorperation between Yaskawa and Modicon.

- This manual describes the functions and specifications of the SA85 Network Adapter used for MEMOBUS PLUS communications.

- Read this manual carefully to ensure the proper use of the SA85. Also, keep this manual in a safe place so that it can be used whenever necessary.

- This manual is applicable to the following model of the SA85.
  Model: JAMSC-120NOM21120

- Refer to the following manuals for other information on the MEMOBUS PLUS.

Manual	Manual number	Contents
MEMOCON GL120, GL130 MEMOBUS PLUS Basics User's Manual	• SIEZ-C825-70.5	Describes the functions, specifications, and application procedures for the MEMOBUS PLUS.
MEMOCON GL120, GL130 MEMOBUS PLUS BM85 Bridge/Multiplexer User's Manual	• SIEZ-C825-70.7	Describes the BM85 Bridge/Multiplexer for the MEMOBUS PLUS.

- Thoroughly check the specifications and conditions or restrictions of the product before use.

1.2 Precautions

This section outlines general precautions that apply to using this manual and the product. **You must read this section first before reading the remainder of the manual.**

1.2.1 Safety Precautions 1-3
1.2.2 Installation Precautions 1-4
1.2.3 Wiring Precautions 1-4
1.2.4 Application Safety Precautions 1-5
1.2.5 Maintenance 1-6

1.2.1 Safety Precautions

- MEMOCON was not designed or manufactured for use in devices or systems that concern people's lives. Users who intend to use the product described in this manual for special purposes such as devices or systems relating to transportation, medical, space aviation, atomic power control, or underwater use must contact Yaskawa Electric Corporation beforehand.

- This product has been manufactured under strict quality control guidelines. However, if this product is to be installed in any location in which a failure of MEMOCON involves a life and death situation or in a facility where failure may cause a serious accident, safety devices MUST be installed to minimize the likelihood of any accident.

- The drawings and photographs presented in this manual are typical examples and may not match the product you received.

- This manual is subject to change due to product improvement, specification modification, and manual improvement. When this manual is revised, the manual code is updated and the new manual is published as a next edition. The edition number appears on the front and back covers.

- If the manual must be ordered due to loss or damage, inform your nearest Yaskawa representative or one of the offices listed on the back of this manual.

- If any of the nameplates mounted in the product becomes illegible or damaged, order the nameplates from your nearest Yaskawa representative or one of the offices listed on the back of this manual.

- Yaskawa will not take responsibility for the results of unauthorized modifications of this product. Yaskawa shall not be liable for any damages or troubles resulting from unauthorized modification.

1.2.2 Installation Precautions

Abide by the following precautions when installing MEMOCON systems.

⚠ **Caution** The installation environment must meet the environmental conditions given in the product catalog and manuals. Using the MEMOCON in environments subject to high temperatures, high humidity, excessive dust, corrosive gases, vibration, or shock can lead to electrical shock, fire, or faulty operation. Do not use the MEMOCON in the following locations. Refer to section *3.10 SA85 Specifications.*

- Locations subject to direct sunlight or ambient temperatures not between 0 and 60°C.
- Locations subject to relative humidity in excess of 95%, rapid changes in humidity, or condensation.
- Locations subject to corrosive or flammable gas.
- Locations that would subject the personal computer with SA85 to direct vibration or shock.
- Locations subject to contact with water, oil, chemicals, etc.

⚠ **Caution** Install the MEMOCON as described in this product manual. Improper installation can cause product failure, malfunctions, or Modules or other components to fall off. Refer to section *3.6 Installing the SA85 Board.*

1.2.3 Wiring Precautions

⚠ **Caution** Always connect a power supply that meets the specifications. Refer to section *3.10 SA85 Specifications.* Connecting an inappropriate power supply can cause fires.

⚠ **Caution** Wiring must be performed by qualified personnel. Mistakes in wiring can cause fires, product failure, or malfunctions.

⚠ **WARNING** Always use a protective grounding terminal with a ground resistance of 100 Ω or less. Otherwise, electrical shock or malfunctions may result.

Note Insert the interface cables properly.

Insert the connectors of the various interface cables that are to be connected to the MEMOBUS PLUS communication line and secure them properly. If the connectors are not inserted properly, the GL120 or GL130 may fail to operate properly.

Note Take noise prevention measures for the power supply.

If noise is transmitted from the external power supply, install an isolation transformer or noise filter to prevent noise from being transmitted. If noise prevention measures for the power supply are insufficient, the GL120 or GL130 and the personal computer may fail to operate properly.

Note Select, separate, and lay external cables correctly.

Note the following items when selecting the I/O signal lines (external cables) to connect the GL120 or GL130 to external devices.

- Mechanical strength

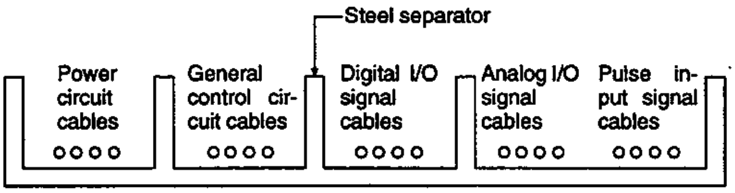
- Noise immunity

- Wiring distance

- Signal voltage, etc.

Always separate the I/O signal lines inside and outside the control panel from the power lines to prevent noise from being transmitted from the power lines. If the I/O signal lines and power lines are not separated properly, malfunctions may result.

Example: Separating I/O Signal Lines from External Cables


Steel separator
Power circuit cables
oooo
General control circuit cables
oooo
Digital I/O signal cables
oooo
Analog I/O signal cables
oooo
Pulse input signal cables
oooo

1.2.4 Application Safety Precautions

External Interlocks for the MEMOCON

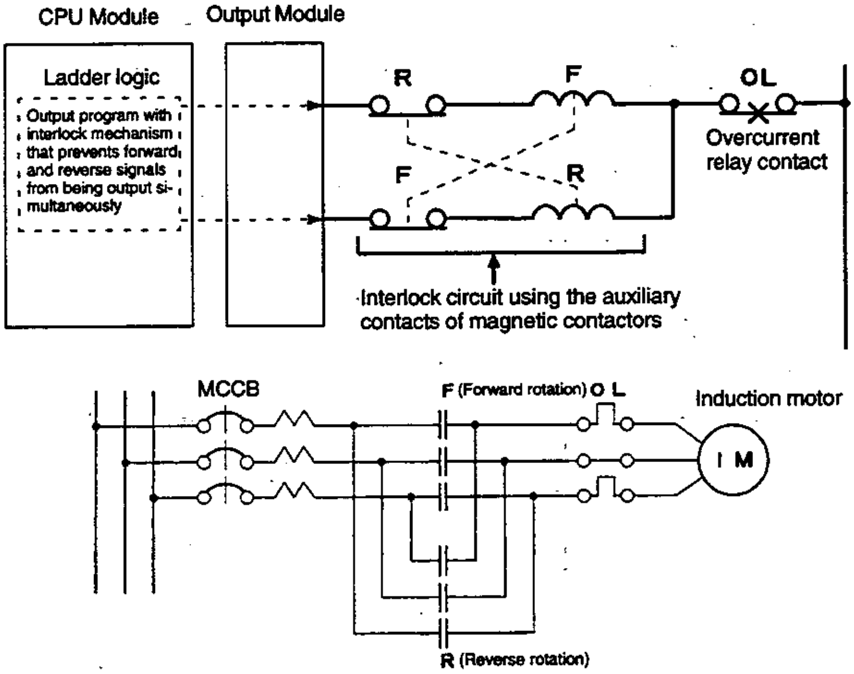
Externally connect an interlock to the MEMOCON if there is any chance that MEMOCON failure could result in bodily harm as well as equipment or accessory damage.

⚠**WARNING** Do not touch any module terminals when the system power is ON. There is a risk of electrical shock.

⚠**WARNING** Externally connect an emergency stop, interlock circuits, and other safety switches to the MEMOCON. Otherwise, MEMOCON failure may damage the equipment or cause an accident.

An example for reciprocal operation for forward and reverse motor operation is shown below.

An interlock circuit is generally written into the MEMOCON ladder logic program to ensure that forward and reverse signals are not simultaneously output. At the same time, an external interlock circuit must be provided using the auxiliary contacts of magnetic contactors for the same purpose.


CPU Module
Output Module
Ladder logic
Output program with interlock mechanism that prevents forward and reverse signals from being output simultaneously
R
F
OL
Overcurrent relay contact
F
R
Interlock circuit using the auxiliary contacts of magnetic contactors
MCCB
F (Forward rotation) O L
Induction motor
I M
R (Reverse rotation)

1.2.5 Maintenance

⚠ **WARNING** Do not attempt to disassemble or modify the MEMOCON in any way. Doing so can cause fires, product failure, or malfunctions.

1.3 Using this Manual

This manual is written for those who already have a basic knowledge of MEMOCON PLCs. We recommend reading the *MEMOCON GL120, GL130 Hardware User's Manual* before attempting to read this manual.

- **Meaning of Basic Terms**

  In this manual, the following terms indicate the meanings as described below, unless otherwise specified.

  - **SA85 = SA85 Network Adapter**
  - **PLC = Programmable (Logic) Controller**
  - **PP = Programming Panel**
  - **GL120, GL130 = MEMOCON GL120, GL130**

- **Description of Technical Terms**

  The bold technical terms in this manual are briefly explained in the **Glossary** provided at the bottom of the page. An example is shown below.

TERMS

Glossary

The following types of terms are described.

- Specific sequence control terms required for explanation of functions.
- Terms that are specific to programmable controllers and electronic devices.

MEMOBUS PLUS Network Overview

2

This chapter outlines the MEMOBUS PLUS network.

2.1 **MEMOBUS PLUS SA85 Network Adapter . . . 2-2**

2.1.1 Software . . . 2-2

2.1.2 README File (README.DOC) . . . 2-2

2.1.3 Sample Programs . . . 2-2

2.2 **Overview of MEMOBUS PLUS Network 2-3**

2.3 **SA85 Network Adapter . . . 2-4**

2.4 **MEMOBUS PLUS Communications Method . 2-5**

2.4.1 MEMOBUS PLUS Routing Paths . . . 2-5

2.4.2 Routing Paths for GL120 and GL130 PLCs . . . 2-6

2.4.3 Routing Path for SA85 Network Adapters . . . 2-6

2.4.4 Routing Path for BM85 Bridge/Multiplexer . . . 2-6

2.5 **MEMOBUS PLUS Communications Processing . . . 2-7**

2.5.1 Path Types . . . 2-7

2.5.2 Number of Paths . . . 2-8

2.5.3 Queuing . . . 2-8

2.6 **MEMOBUS Commands . . . 2-9**

2.7 **Overview of Programs Provided . . . 2-11**

2.1 MEMOBUS PLUS SA85 Network Adapter

This section describes the software contained in the enclosed floppy disks.

2.1.1 Software 2-2

2.1.2 README File (README.DOC) 2-2

2.1.3 Sample Programs 2-2

This section describes the functions and installation method of the MEMOBUS PLUS Network Adapter.

The MEMOBUS PLUS SA85 Network Adapter (Model:JAMSC-120NOM21120) must be installed in an IBM PC/AT or compatible computer (host computer). It functions as a node on the MEMOBUS PLUS network.

The SA85 Network Adapter must be installed in an expansion slot in the IBM PC/AT or compatible computer (host computer). Up to two SA85 Network Adapters can be installed in each host computer.

2.1.1 Software

1) The SA85 Board is provided with 5-inch and 3.5-inch floppy disks.

2) The contents of the floppy disks are as follows:

- A DOS device driver. This driver must be set up according to the host computer's environment.

- A MEMOBUS PLUS network library written in C language.

- Network Diagnostic Utility Software. This software is used to monitor network status, such as active status or error status.

- Sample programs. These programs write and read data to/from each node on the MEMOBUS PLUS network (source programs and executable programs included).

PACKING.LST is a file that contains a list of all programs.

2.1.2 README File (README.DOC)

1) The README file contains the revision history of the Network Device Driver. Read this file before installing the Device Driver.

2) Use the MS-DOS TYPE command to display the file or the PRINT command to print it. A general-purpose text editor can also be used to access the file.

2.1.3 Sample Programs

The enclosed floppy disk contains the library (NETLIB.C) written in C language and the sample programs, including their source codes. Print the sample programs and refer to them when writing application programs.

2.2 Overview of MEMOBUS PLUS Network

▌ This section describes the MEMOBUS PLUS network.

The MEMOBUS PLUS network transfers the right to send (called a "token") to each node in the order of preset node addresses. The SA85 Network Adapter serves as a node on the MEMOBUS PLUS network. The host computer (personal computer) in which the SA85 is installed can send and receive ordinary data, collect network status information, and send and receive global data.

Figure 2.1 shows an example of a MEMOBUS PLUS network configuration.

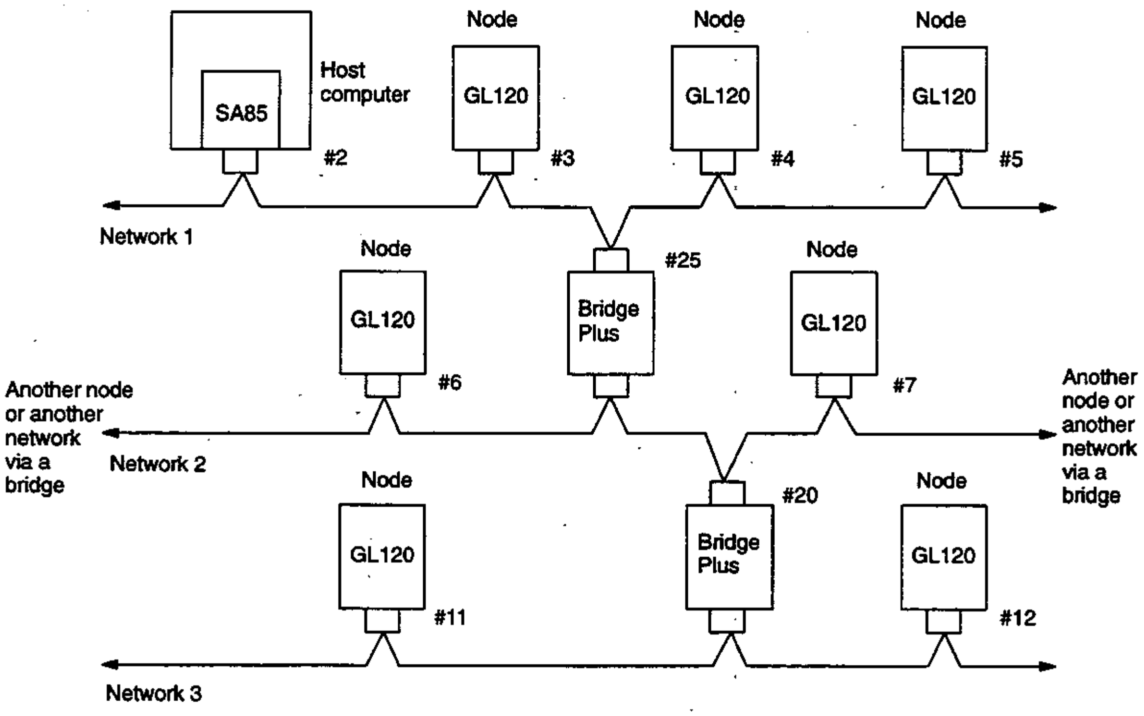


Figure 2.1

2.3 SA85 Network Adapter

This section describes the SA85 Network Adapter.

Network Adapter Application

1) The MEMOBUS PLUS SA85 Network Adapter is installed in an IBM PC/AT or compatible computer. The SA85 Board is a half-size board that can be installed on an eight-bit expansion bus (ISA bus) of an expansion slot in a personal computer. The SA85 is provided with a Modicon-supplied MEMOBUS PLUS communications driver and a network library written in C language.

2) Network adapter applications are briefly explained below.

   a) Data can be read from and written to each node on the network through the user application running on the host computer.

   b) Data can also be read from and written to each node on another network through a bridge.

   c) Network Adapter applications allow the user to program each node, collect status information, and write and read global data.

   d) Typical applications include the following:

      (1) Remote control, monitoring, and checking

      (2) Program loading and saving

      (3) Ladder program testing and debugging

      (4) Network diagnostics

TERMS

Application

A generic name of programs written and run by users

2.4 MEMOBUS PLUS Communications Method

This section describes how to specify MEMOBUS PLUS communications paths.

2.4.1 MEMOBUS PLUS Routing Paths 2-5
2.4.2 Routing Paths for GL120 and GL130 PLCs 2-6
2.4.3 Routing Path for SA85 Network Adapters 2-6
2.4.4 Routing Path for BM85 Bridge/Multiplexer 2-6

2.4.1 MEMOBUS PLUS Routing Paths

The SA85 serves as a master station on the MEMOBUS PLUS network. A unique node address (1 to 64) is assigned to each node. Multiple networks can be connected via bridges.

1) A 5-byte routing path field is used to specify the address of each node.

2) The destination address viewed from the local node is specified in the MEMOBUS PLUS message frame as a routing path. The 5-byte routing path field allows the user to specify the address of a destination node on up to the fourth network viewed from the local node and counting the local network as the first network.

3) The bytes specified in the routing path field depend on the device connected. An example is shown below.

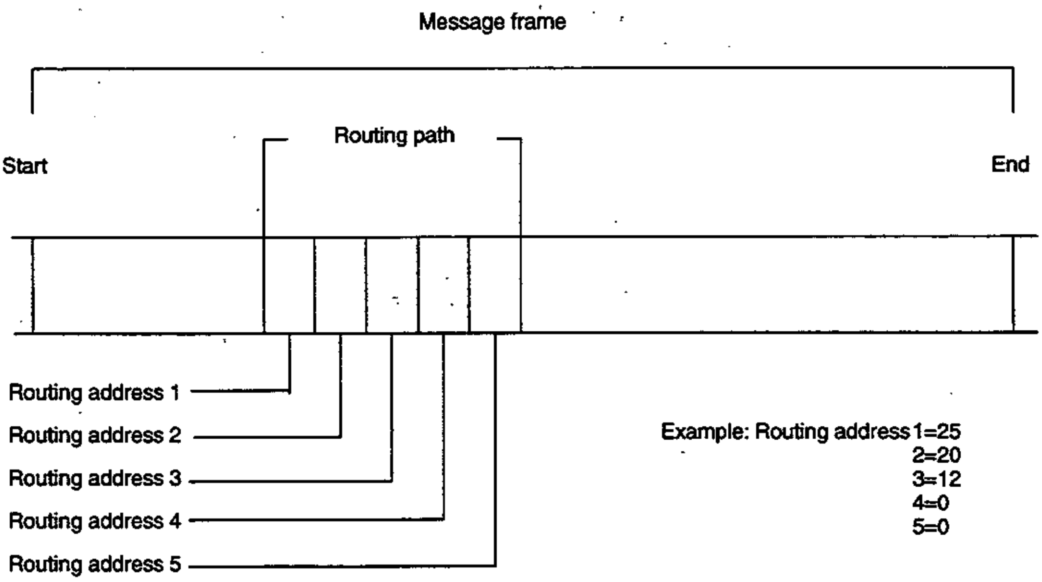


Figure 2.2

Figure 2.2 shows an example of a routing path specified for three networks connected via two bridges (see Figure 2.1). In this path, the message is first transferred to the bridge with node address 25 that is connected to the first network (i.e., the local network). Then, the message is transferred from the above bridge to the bridge with node address 20 that resides on the second network. Finally, the message is transferred from the bridge to the controller with node address 12 that is connected to the third network. The 0's specified in the fourth and fifth bytes indicates the end of the routing path.

4) The bytes specified in the routing path field have different meanings depending on the type of the device specified as the message destination. The routing paths of various device types are outlined in the following manual. Refer to the following manual for details on routing paths:

MEMOCON GL120, GL130 MEMOBUS PLUS Basics User's Manual (SIEZ-C825-70.5)

2.4.2 Routing Paths for GL120 and GL130 PLCs

For the GL120 or GL130, the last nonzero byte in the routing path field is the node address (1 to 64) of the controller. If, for example, the routing path is 5.0.0.0.0, "5" is the node address of the controller on the network.

2.4.3 Routing Path for SA85 Network Adapters

For the SA85 Network Adapters, two bytes are used as the routing path. The second to last nonzero byte is used as the node address (1 to 64) of the Network Adapter. The last nonzero byte is the number (1 to 8) of the internal path or task to which the message is to be assigned.

2.4.4 Routing Path for BM85 Bridge/Multiplexer

1) For the BM85 Bridge/Multiplexer, the bytes specified in the routing path field differ according to the connection mode of slave devices connected to the serial port on the Multiplexer. Either one slave device or a network consisting of multiple slave devices can be connected to the Multiplexer.

2) When only one slave device is connected to the serial port on the Multiplexer, two bytes are used to specify the routing path of the device. The second to last nonzero byte is used as the node address (1 to 64) of the Multiplexer. The last nonzero byte is the port number (1 to 4) of the Multiplexer to which the slave device is connected. If, for example, the node address of the BM85 is 25, 25.1.0.0.0 is the routing path to the slave device connected to port 1 on the Multiplexer.

3) When a network consisting of multiple slave devices is connected to the serial port on the Multiplexer, three bytes are used to specify the routing path of the device. The third to last nonzero byte is used as the node address (1 to 64) of the Multiplexer. The second to last nonzero byte is the port number (1 to 4) of the Multiplexer to which the network is connected. The last nonzero byte is the MEMOBUS slave address (1 to 247). If, for example, the node address of the BM85 is 25, 25.2.200.0.0 is the routing path to the device with slave address 200 on the network connected to port 2 on the Multiplexer.

2.5 MEMOBUS PLUS Communications Processing

▌ This section outlines MEMOBUS PLUS communications processing.

2.5.1 Path Types 2-7
2.5.2 Number of Paths 2-8
2.5.3 Queuing 2-8

MEMOBUS PLUS devices provide several types of internal paths to allow concurrent communications processing. When communications processing starts, one path is used and occupied during the processing. When the communications processing is complete, the path is released and thus becomes available for another communications process.

The source and destination nodes occupy a path for mutual communications processing until processing is complete. If communications processing is performed via a device and a bridge on another network, the bridge occupies the path that extends across the two networks. As mentioned above, a communications path is used between a source node and a destination node and remains occupied until communications processing is complete.

2.5.1 Path Types

The following four path types are provided with MEMOBUS PLUS devices.

1) Data Master (DM) Path

This path is used by the source node to read and write data, and obtain and clear status. It is identified with a path value between 01 and 08 in hexadecimal notation.

2) Data Slave (DS) Path

This path is used by the destination node to read and write data, and obtain and clear status. It is identified with a path value between 41 and 48 in hexadecimal notation.

3) Program Master (PM) Path

This path is specially used for a Programming Panel and used by the source node (mainly MEMOSOFT). It is identified with a path value between 81 and 88 in hexadecimal notation.

4) Program Slave (PS) Path

This path is specially used for a Programming Panel and used by the destination node (mainly controller). It is identified with a path value between C1 and C8 in hexadecimal notation.

2.5.2 Number of Paths

The number of paths differs according to the type of the device connected.

Table 2.1

Path	Device			
	GL120/130	BM85 Bridge/Multiplexer	BP85 Bridge Plus	SA85 Network Adapter
DM	4	4	8	8
DS	4	4	8	8
PM	0	4	8	8
PS	1	4	8	8

2.5.3 Queuing

1) When all the DS paths are occupied, new communications processes are placed in a queue until a DS path is released. When a DS path is released, it is allocated to new communications process. The final data response is not returned to the application in the source device until a complete DS path from the source device to the destination device is available. When the destination device obtains a path, it waits for the right to send (i.e., the token), then requests a command from the source device. The source device resends the command while the destination device is holding the token. Because of this processing, the application need not perform polling between the source device and the destination device.

2) **Queuing at the BP85 Bridge Plus**

When a message passes through multiple bridges, it is queued in the first bridge but need not be queued in the second and subsequent bridges. If an attempt is made to queue the message in the second bridge, an error code will be returned. This error code can be tested with the application in the source node. This mechanism prevents message queuing between networks from delaying communications processing.

2.6 MEMOBUS Commands

▌ This section outlines the MEMOBUS commands.

Communications with PLCs can be performed by pasting MEMOBUS commands into the MEMOBUS PLUS frame. These commands are processed by the PLC. The MEMOBUS commands are used to read and write coil and register data and read status information.

Table 2.2

Function code	Function
01_H	Read coil status
02_H	Read input relay status
03_H	Read holding register contents
04_H	Read input register contents
05_H	Modify status of a single coil
06_H	Write to a single holding register
07_H	Read status of specific coils
08_H	Loopback test
$0B_H$	Read event counter
$0C_H$	Read communications status
$0F_H$	Modify status of multiple coils
10_H	Write to multiple holding registers
11_H	Read equipment status
12_H	Read link coil status
13_H	Read constant register contents
15_H	Read link register contents
16_H	Write a mask to the holding register
17_H	Read and write to multiple holding registers
18_H	Read FIFO register contents
19_H	Modify status of a single link coil
$1A_H$	Write to a single constant register
$1B_H$	Write to a single link register
$1D_H$	Modify status of multiple link coils
$1E_H$	Write to multiple constant registers
$1F_H$	Write to multiple link registers
21_H	Read special reference status
22_H	Modify special reference status

1) Paths Required

The source node must use DM paths to issue all MEMOBUS commands. The paths required for each MEMOBUS PLUS device type are shown in Table 2.1.

2) Sample Programs

The enclosed floppy disk contains sample programs that show how MEMOBUS commands are pasted in the MEMOBUS PLUS frame to communicate with PLCs. The sample programs also show how responses are received from PLCs. The source code and executable file for each sample program are provided.

For example, the sample program READNODE.EXE displays data that is read after the following commands are executed.

a) Read PLC relay and coil status

b) Read PLC register contents

Source file READNODE.C shows the programming method used to achieve these tasks.

2.7 Overview of Programs Provided

▮ This section outlines chapters *5* to *10.*

Chapters *5* to *10* describe the MEMOBUS PLUS application NETBIOS provided by Modicon, U.S.

Chapter *8* describes how to use the MEMOBUS PLUS network utility provided by Modicon, U.S.

Chapter *9* describes the sample programs provided by Modicon, U.S. The floppy disk contains the source code for each sample program.

Chapter *10* describes the MEMOBUS commands.

Refer to the following manual for details on the MEMOBUS PLUS Network:

MEMOCON GL120, GL130 MEMOBUS PLUS Basics User's Manual (SIEZ-C825-70.5)

SA85 Network Adapter

3

This chapter describes how to install and set up the SA85 Network Adapter.

3.1 SA85 and Operating Environment 3-2

3.1.1 Hardware Components 3-2

3.1.2 Software Components 3-2

3.1.3 System Design 3-3

3.2 Installation Outline 3-4

3.2.1 Installation Outline 3-4

3.2.2 Adding and Deleting the Node during Operation 3-4

3.3 Switch Layout 3-5

3.4 Setting the MEMOBUS PLUS Address Switch 3-6

3.5 Setting the Memory Address Switch 3-7

3.6 Installing the SA85 Board 3-9

3.7 Indicators 3-10

3.8 MEMOBUS PLUS Port Labels 3-12

3.9 Initializing the SA85 3-13

3.10 SA85 Specifications 3-14

3.10.1 General Specifications 3-14

3.10.2 MEMOBUS PLUS Port Specifications 3-15

3.1 SA85 and Operating Environment

▌ This section outlines the SA85 hardware and software components.

3.1.1 Hardware Components 3-2
3.1.2 Software Components 3-2
3.1.3 System Design 3-3

3.1.1 Hardware Components

1) Before installing the SA85 in the host computer (personal computer), set the node address and memory address switches.

   a) Assign a node address to the SA85. The node address is a number used to identify the SA85 on the MEMOBUS PLUS network. The token and messages are transferred based on the node address.

   b) Assign a memory address to the SA85. The memory address is used to allocate a 2-kilobyte buffer for communication between the SA85 and the application in the personal computer.

2) After setting the node address and the memory address, install the SA85 into an available expansion slot in the host computer and connect the cables.

3.1.2 Software Components

1) Before starting communication processing with the SA85, always perform the following procedure:

   a) Copy the device driver to the hard disk.

   b) Modify CONFIG.SYS.

2) Device driver setup involves the following tasks:

   a) Setting the adapter number (0 or 1)

   b) Setting the memory address

   c) Setting the software interrupt number

   These parameters must be set according to the application to be used. These parameter settings are particularly important when the personal computer contains multiple optional functions.

3) The source codes, headers, and library contained in the floppy disk provided with the SA85 can be installed. Microsoft C can be used to include these programs in the application program, and compile and link them.

4) The SA85 is provided with the network diagnostic utility and the sample programs which show how to access PLC registers and global data.

5) *Figure 3.1* briefly illustrates how to install the SA85 in the host computer.

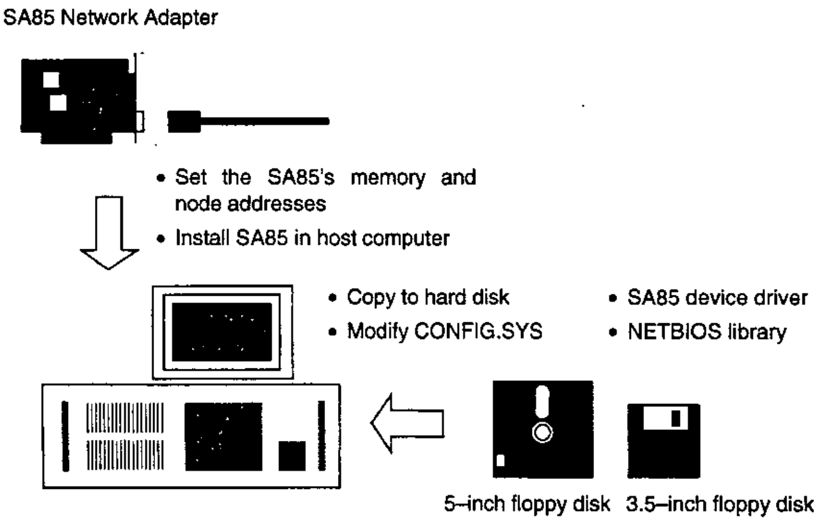


Figure 3.1

3.1.3 System Design

Refer to the following manual for details on how to design the MEMOBUS PLUS Network.

- Manual title: MEMOCON GL120, GL130
  MEMOBUS PLUS User's Manual

- Manual code: SIEZ-C825-70.5

3.2 Installation Outline

▌ This section outlines the SA85 installation procedure.

3.2.1 Installation Outline 3-4
3.2.2 Adding and Deleting the Node during Operation 3-4

3.2.1 Installation Outline

Installing the SA85 Network Adapter involves the following four steps.

1) Set the MEMOBUS PLUS node address switch.

2) Set the memory address switch.

3) Install the Adapter Board and connect it to the MEMOBUS PLUS Network.

4) Install the Device Driver, Software Library, Network Diagnostic Utility Software, and sample programs.

Note Before installing the SA85, the user must understand how to handle Adapter Boards and how to prevent static electricity.

3.2.2 Adding and Deleting the Node during Operation

The SA85 node can be added to and deleted from the MEMOBUS PLUS network during operation without having to turn off other nodes. If the node is deleted, the network protocol will bypass the node. If the node is added, the network protocol will include the node in the network.

Note If the application depends on the Adapter being connected to the network, unpredictable results may arise when the node is added or deleted during operation. Before adding or deleting the node, always decide how to connect the application to the network.

If the SA85 is relocated on the network, be sure the network address switch and the memory address switch are set correctly.

3.3 Switch Layout

This section describes the switch layout.

The following diagram illustrates the MEMOBUS PLUS address switch and memory switch layout.

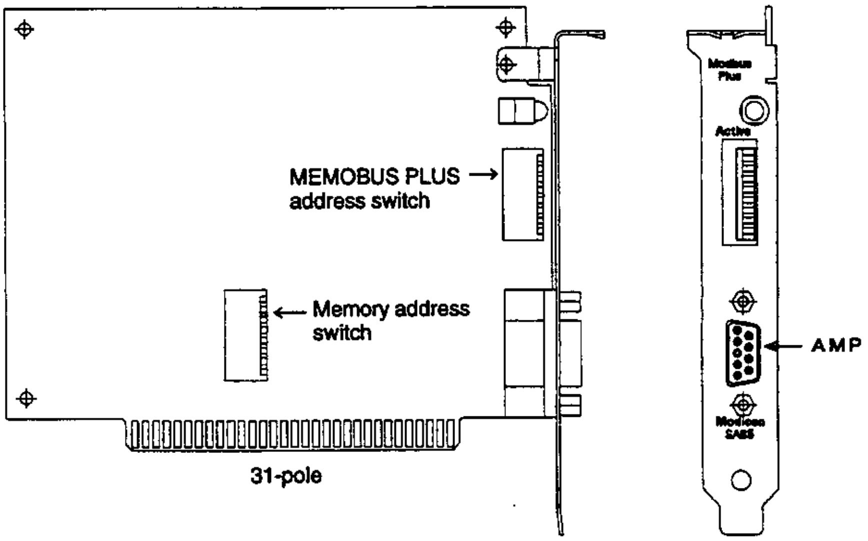
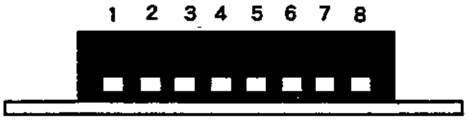


Figure 3.2

3.4 Setting the MEMOBUS PLUS Address Switch

This section describes how to set the MEMOBUS PLUS address switch.

Pins 1to 6 are used to set the node address of MEMOBUS PLUS. Pins 7 and 8 are not used. Each node must be assigned a unique node address. Each node address is a number determined by adding 1 to the binary number set on the switch. We recommend that node address 64 be reserved for use during network maintenance. Also, node address 1 is the initial value of the PLC node address and therefore should not be used to avoid confusion.



All pins in the figure on the left are set to 0.
Pushing the pin up sets it to 1.

Address	Pins						Address	Pins					
	1	2	3	4	5	6		1	2	3	4	5	6
1	0	0	0	0	0	0	33	0	0	0	0	0	1
2	1	0	0	0	0	0	34	1	0	0	0	0	1
3	0	1	0	0	0	0	35	0	1	0	0	0	1
4	1	1	0	0	0	0	36	1	1	0	0	0	1
5	0	0	1	0	0	0	37	0	0	1	0	0	1
6	1	0	1	0	0	0	38	1	0	1	0	0	1
7	0	1	1	0	0	0	39	0	1	1	0	0	1
8	1	1	1	0	0	0	40	1	1	1	0	0	1
9	0	0	0	1	0	0	41	0	0	0	1	0	1
10	1	0	0	1	0	0	42	1	0	0	1	0	1
11	0	1	0	1	0	0	43	0	1	0	1	0	1
12	1	1	0	1	0	0	44	1	1	0	1	0	1
13	0	0	1	1	0	0	45	0	0	1	1	0	1
14	1	0	1	1	0	0	46	1	0	1	1	0	1
15	0	1	1	1	0	0	47	0	1	1	1	0	1
16	1	1	1	1	0	0	48	1	1	1	1	0	1
17	0	0	0	0	1	0	49	0	0	0	0	1	1
18	1	0	0	0	1	0	50	1	0	0	0	1	1
19	0	1	0	0	1	0	51	0	1	0	0	1	1
20	1	1	0	0	1	0	52	1	1	0	0	1	1
21	0	0	1	0	1	0	53	0	0	1	0	1	1
22	1	0	1	0	1	0	54	1	0	1	0	1	1
23	0	1	1	0	1	0	55	0	1	1	0	1	1
24	1	1	1	0	1	0	56	1	1	1	0	1	1
25	0	0	0	1	1	0	57	0	0	0	1	1	1
26	1	0	0	1	1	0	58	1	0	0	1	1	1
27	0	1	0	1	1	0	59	0	1	0	1	1	1
28	1	1	0	1	1	0	60	1	1	0	1	1	1
29	0	0	1	1	1	0	61	0	0	1	1	1	1
30	1	0	1	1	1	0	62	1	0	1	1	1	1
31	0	1	1	1	1	0	63	0	1	1	1	1	1
32	1	1	1	1	1	0	64	1	1	1	1	1	1

Figure 3.3

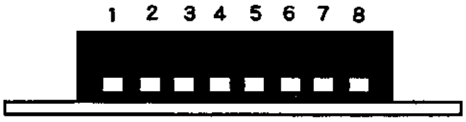
3.5 Setting the Memory Address Switch

▌ This section describes how to set the memory address switch.

1) The SA85 uses the host computer memory as a board status and communications buffer. Specify the buffer address with the memory address switch to prevent conflict between the SA85 Board and other optional boards installed in the computer.

2) The specifiable address range is between C0000 and EF800 in hexadecimal notation. The 2-kilobyte (800 HEX) memory space starting from the specified memory address is allocated as a buffer. Determine available memory space by referring to the user's manual for your computer. Be sure that the specified memory space does not overlap with the area used by other applications. Record the specified memory address because it must be used to modify CONFIG.SYS as described in section *4.1 Installing the Device Driver.*

3) Set the memory address switch by referring to the table below. Pin 8 is not used.



All pins in the figure on the left are set to 0.
Pushing the pin up sets it to 1.

Base Address	Pins							Base Address	Pins						
	1	2	3	4	5	6	7		1	2	3	4	5	6	7
C0000	0	0	0	0	0	0	0	D2800	0	1	0	0	1	0	1
C0800	0	0	0	0	0	0	1	D3000	0	1	0	0	1	1	0
C1000	0	0	0	0	0	1	0	D3800	0	1	0	0	1	1	1
C1800	0	0	0	0	0	1	1	D4000	0	1	0	1	0	0	0
C2000	0	0	0	0	1	0	0	D4800	0	1	0	1	0	0	1
C2800	0	0	0	0	1	0	1	D5000	0	1	0	1	0	1	0
C3000	0	0	0	0	1	1	0	D5800	0	1	0	1	0	1	1
C3800	0	0	0	0	1	1	1	D6000	0	1	0	1	1	0	0
C4000	0	0	0	1	0	0	0	D6800	0	1	0	1	1	0	1
C4800	0	0	0	1	0	0	1	D7000	0	1	0	1	1	1	0
C5000	0	0	0	1	0	1	0	D7800	0	1	0	1	1	1	1
C5800	0	0	0	1	0	1	1	D8000	0	1	1	0	0	0	0
C6000	0	0	0	1	1	0	0	D8800	0	1	1	0	0	0	1
C6800	0	0	0	1	1	0	1	D9000	0	1	1	0	0	1	0
C7000	0	0	0	1	1	1	0	D9800	0	1	1	0	0	1	1
C7800	0	0	0	1	1	1	1	DA000	0	1	1	0	1	0	0
C8000	0	0	1	0	0	0	0	DA800	0	1	1	0	1	0	1
C8800	0	0	1	0	0	0	1	DB000	0	1	1	0	1	1	0
C9000	0	0	1	0	0	1	0	DB800	0	1	1	0	1	1	1
C9800	0	0	1	0	0	1	1	DC000	0	1	1	1	0	0	0
CA000	0	0	1	0	1	0	0	DC800	0	1	1	1	0	0	1
CA800	0	0	1	0	1	0	1	DD000	0	1	1	1	0	1	0
CB000	0	0	1	0	1	1	0	DD800	0	1	1	1	0	1	1
CB800	0	0	1	0	1	1	1	DE000	0	1	1	1	1	0	0
CC000	0	0	1	1	0	0	0	DE800	0	1	1	1	1	0	1
CC800	0	0	1	1	0	0	1	DF000	0	1	1	1	1	1	0
CD000	0	0	1	1	0	1	0	DF800	0	1	1	1	1	1	1
CD800	0	0	1	1	0	1	1	E0000	1	0	0	0	0	0	0
CE000	0	0	1	1	1	0	0	E0800	1	0	0	0	0	0	1
CE800	0	0	1	1	1	0	1	E1000	1	0	0	0	0	1	0
CF000	0	0	1	1	1	1	0	E1800	1	0	0	0	0	1	1
CF800	0	0	1	1	1	1	1	...	.	.	.	.	.	.	.
D0000	0	1	0	0	0	0	0	...	.	.	.	.	.	.	.
D0800	0	1	0	0	0	0	1	EE000	1	0	1	1	1	0	0
D1000	0	1	0	0	0	1	0	EE800	1	0	1	1	1	0	1
D1800	0	1	0	0	0	1	1	EF000	1	0	1	1	1	1	0
D2000	0	1	0	0	1	0	0	EF800	1	0	1	1	1	1	1

Figure 3.4

3.6 Installing the SA85 Board

▌ This section describes how to install the SA85 Board.

Install the SA85 Board and connect the cable as below.

1) Specify the MEMOBUS PLUS network address and memory address. See sections *3.4 Setting the MEMOBUS PLUS Address Switch* and *3.5 Setting the Memory Address Switch.*

2) Refer to the user's manual for your personal computer, turn the power OFF, then unplug the computer.

3) Remove the computer cover. Be careful not to lose the screws and other parts since they must be used to reinstall the computer cover.

4) Locate a vacant expansion slot. Remove the expansion slot cover by removing the screws. Be careful not to lose the screws.

5) Install the SA85 Board properly in the expansion slot.

6) Secure the SA85 Board with the screws that were removed from the expansion slot cover. These screws also serve as the protective grounding terminals for the SA85.

7) Replace the computer cover.

8) Connect the MEMOBUS PLUS network cable connector to the board connector.

9) Plug in the computer and turn ON the power. Make sure that the personal computer properly operates with the SA85 Board installed.

10) Install the Device Driver and other software programs on the hard disk.

3.7 Indicators

▌ This section describes the indicator on the SA85.

The SA85 has a green indicator that indicates the SA85 status in the network.

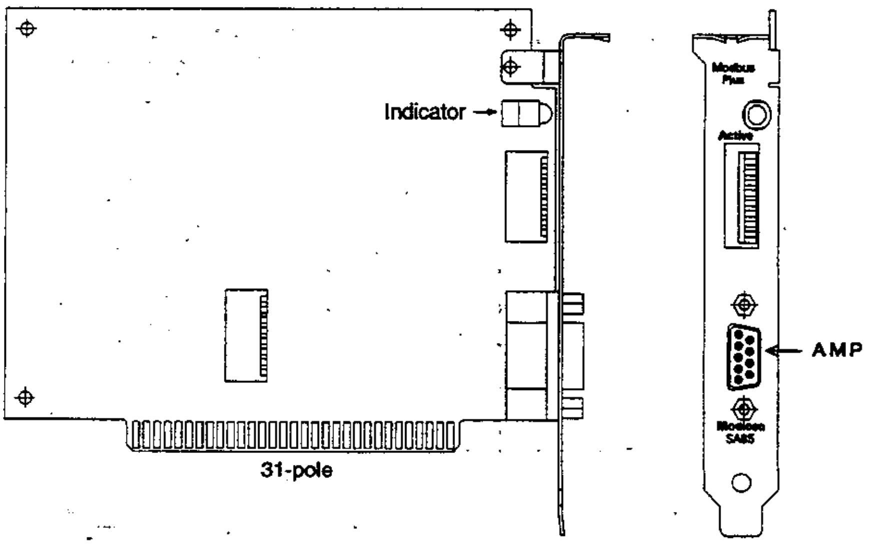


Figure 3.5

MEMOBUS PLUS status is indicated by the indicator flashing patterns.

Table 3.1

Flashing Pattern	Status
6 times per second	Normal operating status: All nodes among which the token is transferred are in normal status.
Once per second	Offline status during power-up or immediately after the status in which the indicator flashes 4 times per second. In this status, the network is being monitored, or a table is being created for active nodes and nodes holding the token. The MEMOBUS PLUS enters a normal status five seconds after this status.
Flashing twice, then OFF for two seconds	The MEMOBUS PLUS can recognize the token at another node but cannot receive the token. Check the network for disconnection or a missing terminator.
Flashing three times, then OFF for 1.7 seconds	The MEMOBUS PLUS cannot recognize the token at another node. It requests the token at regular intervals but cannot recognize the token at another node. Check the network for disconnection or a missing terminator.
Flashing four times, then OFF for 1.4 seconds	The MEMOBUS PLUS has received a message from a node with the same node address. This status lasts until the node with the same node address disappears. If the node with the same node address disappears for five seconds, the MEMOBUS PLUS will enter the status in which the indicator flashes once per second.

MBPSTAT Network Diagnostic Utility

To check whether the MEMOBUS PLUS network is operating normally, it is easier for the user to use the MBPSTAT Network Diagnostic Utility (MBPSTAT.EXE) than to check the network indicator. This utility is contained on the enclosed floppy disk.

Chapter 8 MBPSTAT Utility describes how to use the MBPSTAT utility and select the network diagnostic options. Selecting option number 10 "Show Node Personality" displays the message corresponding to the network indicator flashing pattern. Selecting "Peer Status" displays the status at the "Peer Status" item on the MBPSTAT screen.

The table below shows the correspondence between the MBPSTAT screen messages and the network indicator flashing patterns.

MBPSTAT Message	Indicator Flashing Pattern
Normal Link Operation	6 timers per second
Monitor Link Operation	Once per second
Never Getting Token	Flashing twice, then OFF for two seconds
Sole Station	Flashing three times, then OFF for 1.7 seconds
Duplicate Station	Flashing four times, then OFF for 1.4 seconds

3.8 MEMOBUS PLUS Port Labels

▌ This section describes the MEMOBUS PLUS port labels.

The SA85 is provided with two labels which are used to identify the MEMOBUS PLUS network number and the node address. Attach one label to the unit when MEMOBUS PLUS has been connected to the network. The other label is a spare.

Enter on the label the MEMOBUS PLUS network number and the node address assigned to the SA85, and paste the label in an easy-to-read position on the Unit. *Figure 3.6* shows an example of label entry.

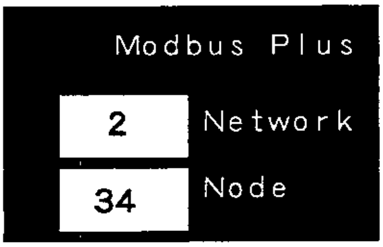


Figure 3.6

3.9 Initializing the SA85

▌ This section describes how to make the SA85 parameter settings valid.

When the SA85 is installed, the SA85 parameters must be set with the switches, including the network address and memory address switches.

To make the SA85 parameter settings valid, the host computer must be turned OFF, then ON. Even if the system is reset by simultaneously pressing the CTRL, ALT, and DEL Keys, the SA85 parameter settings will not be read. The parameter settings can only be made valid by turning the power OFF, then ON.

The MEMOBUS PLUS continues to operate even when the system is reset by simultaneously pressing the CTRL, ALT, and DEL Keys. Token passing and communications processing will continue.

3.10 SA85 Specifications

This section summarizes the specifications of the SA85.

3.10.1 General Specifications 3-14
3.10.2 MEMOBUS PLUS Port Specifications 3-15

3.10.1 General Specifications

The specifications of SA85 Network Adapter are shown below.

Table 3.2 SA85 Specifications

Item		Specifications
Name		MEMOBUS PLUS SA85 Network Adapter
Model		JAMSC-120NOM21120
Dimensions		Standard half-size board: 146 x 107 mm
Mass		0.2 kg
Power Supply		From computer motherboard: 500 mA, 5 VDC max.
Environment Conditions	Ambient Operating Temperature	0 to 60°C
	Ambient Storage Temperature	–40 to 80°C
	Ambient Operating Humidity	0 to 95% (with no condensation)
	EMI Heat Radiation	MIL STD 461B RS02, RS03
	EMI Conductance	MIL STD 461B CS02
Network Connection	Connector	MEMOBUS PLUS Cable
Software	OS	MS-DOS Ver. 3.1 or later (or DOS/V Ver. 5.0 or later)
	C Library	Microsoft C 5. 1 (large model)

3.10.2 MEMOBUS PLUS Port Specifications

1) The specifications of the MEMOBUS PLUS port are shown below.

D-sub Connector (9-pin Female)
Connector type: 747844-4, manufactured by AMP (inch screws are used as set screws)

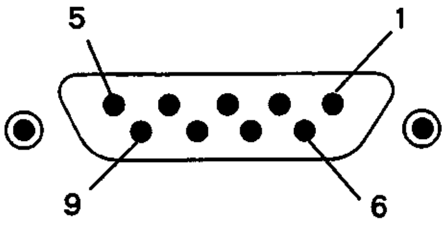


Figure 3.7 RS-232 Connector

2) A D-sub connector (9-pin female) is used as the connector for the MEMOBUS PLUS port. The following table shows the connector pin layout and signal names.

Table 3.3 Connector Pin Layout

Pin Number	Abbreviation	Signal Name
1	SHLD	SHIELD
2	DL	DATA LOW
3	DH	DATA HIGH
4	–	Not used.
5	–	Not used.
6	–	Not used.
7	–	Not used.
8	–	Not used.
9	–	Not used.

Installing Software

4

This chapter describes how to install and set up the Device Driver in the host computer.

4.1 Installing the Device Driver 4-2
4.1.1 Copying MBPHOST.SYS 4-2
4.1.2 Changing CONFIG.SYS 4-2
4.1.3 Host Computer Initialization Processing 4-4

4.2 Installing Other Files 4-5
4.2.1 Installing the MBPSTAT and \TESTSRC Programs ... 4-5
4.2.2 Installing \MBPHINC and \NETLIB Files 4-5

4.1 Installing the Device Driver

This section describes how to install the Device Driver.

4.1.1 Copying MBPHOST.SYS 4-2

4.1.2 Changing CONFIG.SYS 4-2

4.1.3 Host Computer Initialization Processing 4-4

Before installing the Device Driver, always read the README file (README.DOC). This file contains the revision history of the Device Driver. Installing the Device Driver involves the following three steps:

1) Copy the Device Driver to the hard disk in the host computer.

2) Modify the CONFIG.SYS file.

3) Reset the host computer to make the new CONFIG.SYS settings valid.

4.1.1 Copying MBPHOST.SYS

1) MBPHOST.SYS is the Device Driver for the SA85.

2) The Device Driver is contained in the enclosed 5-inch and 3.5-inch floppy disks.

3) Use the MS-DOS COPY command to copy the Device Driver to the hard disk. Copy the Device Driver to the directory specified in CONFIG.SYS.

   The following example shows how to copy the Device Driver to the route directory of drive C.

```
COPY A:\MBPHOST.SYS C:\
```

4.1.2 Changing CONFIG.SYS

1) Use the DEVICE command to include the MEMOBUS PLUS Device Driver into the CONFIG.SYS file for the host computer. Specify in the DEVICE command the directory to which the Device Driver was copied.

2) The following is an example of specifying the device to be added in CONFIG.SYS.

```
DEVICE=A:\MBPHOST.SYS /Mnnnn /Nn /Snn /R2 /B
```

Table 4.1

Switch	Explanation	Example	Initial Value
/Mnnnn	Sets the memory address.	/MD080	D000
/Nn	Sets the adapter number.	/N1	0
/Snn	Sets the software interrupt number.	/S5B	5C
/R2	Sets driver version 2.0.	/R2	None
/B	Resets the board and disables the SA85OFF command.	/B	Enabled

3) The option switches are explained below.

a) /M

The /M switch specifies the same memory address as that set with the DIP switch on the SA85 Board. Specify the first four digits of the five hexadecimal address immediately after /M.

b) /N

The /N switch specifies the device address of the SA85 Board installed in the host computer. Up to two SA85 Boards can be installed in a host computer. Specify 0 or 1. Specify this number in the ncb_open() function of NetBIOS. The default value is 0.

c) /S

The /S switch is used to prevent duplicate software interrupt numbers from being assigned. Specify the software interrupt number as a hexadecimal number between 00 and FF. If two SA85 Boards are installed in the same host computer, the software interrupt number of NetBIOS is always 5C.

INFO Software interrupt number 5C is set in the Network Diagnostic Utility and the sample programs. A software interrupt number other than 5C can also be specified. (See the sample programs.)

d) /R2

The /R2 switch is a standard option switch. Specify this switch when Device Driver Ver. 2.0 is used.

e) /B

The /B switch is used to reset the Board and disable the Board reset function of the SA85OFF command. If this switch is not set, the board is initialized and self-diagnostic processing is executed when the board is reset.

4.1.3 Host Computer Initialization Processing

1) When installing the Device Driver or modifying the MEMOBUS PLUS Device Driver settings in CONFIG.SYS, always reset the host computer system by simultaneously pressing the CTRL, ALT, and DEL Keys. In this way, the modified CONFIG.SYS settings are made valid.

2) If the Device Driver DIP switch settings, software setup, and Device Driver settings are not correct, the host computer may hang up. If this happens, check the settings.

4.2 Installing Other Files

▌ This section describes how to install other files contained in the enclosed floppy disk.

4.2.1 Installing the MBPSTAT and TESTSRC Programs 4-5
4.2.2 Installing \MBPHINC and \NETLIB Files 4-5

4.2.1 Installing the MBPSTAT and \TESTSRC Programs

1) The Network Diagnostic Utility program MBPSTAT.EXE is contained in the enclosed floppy disk. The \TESTSRC directory in this floppy disk contains the sample programs (source codes and executable files) that allow access to controller registers and global data.

2) Copy the utility program and sample programs to the directory in which the MEMOBUS PLUS application is to be executed, tested, or debugged. See chapter *8 MBPSTAT Utility* for an overview of these programs.

4.2.2 Installing \MBPHINC and \NETLIB Files

1) Include files NETLIB.H and NETBIOS.H are contained in the \MBPHINC directory in the enclosed floppy disk. The library file NETLIB.LIB is contained in the \NETLIB directory. The library file MBPLUS.LIB and several XXX.ASM and XXX.MAK files are contained in the \NETLIB directory.

2) These files can be used in an application program by referring to the compiler user's manual. Copy \MBPHINC to the INCLUDE directory on the hard disk. Copy \NETLIB to the LIBRARY directory on the hard disk.

3) NELIB.LIB has been complied using the large model of Borland C Version 3.1. A compatible make file is provided with Microsoft C Version 5.1.

Using the NetBIOS Functions

5

This chapter describes the NetBIOS functions provided by Modicon.

5.1 MEMOBUS PLUS NetBIOS Environment . . . 5-2

5.1.1 NetBIOS Environment 5-2

5.1.2 Network Control Block 5-3

5.2 Overview of NetBIOS Commands 5-4

5.2.1 NETLIB Functions 5-4

5.2.2 Path Required 5-4

5.2.3 Message Buffer 5-5

5.2.4 Function Definitions and Prototypes 5-5

5.2.5 NCB Structure Definition and NetBIOS Command Codes . . 5-5

5.2.6 Function Return Values 5-5

5.3 Return Values 5-6

5.4 NetBIOS Functions 5-8

5.4.1 ncb_cancel () 5-8

5.4.2 ncb_close () 5-9

5.4.3 ncb_open () 5-9

5.4.4 ncb_receive () 5-11

5.4.5 ncb_receive_datagram () 5-12

5.4.6 ncb_receive_wait () 5-13

5.4.7 ncb_reset () 5-14

5.4.8 ncb_sa85off () 5-14

5.4.9 ncb_send () 5-15

5.4.10 ncb_send_datagram () 5-16

5.4.11 ncb_set_slave_login () 5-17

5.4.12 ncb_set_sw_interrupt () 5-18

5.4.13 ncb_status () 5-18

5.1 MEMOBUS PLUS NetBIOS Environment

▌ This section describes the environment required to use the NetBIOS functions.

5.1.1 NetBIOS Environment 5-2

5.1.2 Network Control Block 5-3

5.1.1 NetBIOS Environment

1) The NetBIOS is a PLC network software application that was written in C language on a personal computer. Modicon provides many NetBIOS functions that can be directly and effectively used for the MEMOBUS PLUS network.

2) The main NetBIOS functions include the following:

   a) Initializing the SA85

   b) Opening a path.

   c) Transmitting and receiving data.

   d) Transmitting and receiving global data.

   e) Reading remote node status information.

   f) Reading local node status information.

   The application program can create a MEMOBUS message in the buffer in the host computer and then send it to each node. Received MEMOBUS messages are retained in the buffer in the host computer and used by the application program.

3) For the syntax, arguments, and return values of each function, see the relevant items in this chapter. When using these functions, note the differences between them.

4) Before writing an application program, check the definitions of the NetBIOS functions by referring to the netlib.c source file as required. This file is contained in the enclosed floppy disk. Sample programs are also contained in the floppy disk. These sample programs allow the user to understand how these functions are used in MEMOBUS PLUS applications. If necessary, print these files and use them when writing an application program.

5.1.2 Network Control Block

1) The NetBIOS commands process data in a network control block (NCB). All data is passed to the NCB when the NetBIOS commands are executed.

2) The NCB is 64 bytes long and consists of 14 information storage areas. These areas are shown in the following table.

3) The NCB is defined in header file netbios.h. The source data in the header file is presented in chapter *7 Include Files*.

Table 5.1 Information Storage Areas in the NCB

Offset	Size	Area Name	Contents of Area
0	1	NCB_COMMAND	NetBIOS command code
1	1	NCB_RETCODE	Execution result (0 if normal) See the error code list.
2	1	NCB_LSN	Not used.
3	1	NCB_NUM	Path type used when NetBIOS command is executed This data is set when the NCB is opened. 0x01 to 0x08: DM1 to DM8 0X41 to 0x48: DS1 to DS8 0x81 to 0x88: PM1 to PM8 0xC1 to 0xC8: PS1 to PS8
4	4	NCB_BUFFER	Message buffer address
8	2	NCB_LENGTH	Message buffer size (in bytes)
0A	16	NCB_CALLNAME	Routing path (first seven bytes) used when the NetBIOS command is executed This data is set when the NCB is opened. Bytes 0 and 1: DM, DS, PM, or PS Bytes 2 to 6: Routing address Other bytes are not used.
1A	16	NCB_NAME	User name on the network
2A	1	NCB_RTO	Reception timeout value set in units of 500 ms This area is used for ncb_receive() and ncb_receive_datagram().
3B	1	NCB_STO	Transmission timeout value set in units of 500 ms This area is used for ncb_send() and ncb_send_datagram().
2C	4	NCB_POST_ADDRESS	Not used.
30	1	NCB_LANA_NUM	Adapter number (0 or 1)
31	1	NCB_CMD_CPLT	Not used.
32	14	NCB_RESERVE	Reserved area used by NetBIOS

Offset: Address of information storage area in the NCB (hexadecimal)

Size: Size of information storage area

Area name: Variable name of information storage area

Contents of area: Execution results of each function

5.2 Overview of NetBIOS Commands

This section provides an overview of the NetBIOS commands.

5.2.1 NETLIB Functions 5-4
5.2.2 Path Required 5-4
5.2.3 Message Buffer 5-5
5.2.4 Function Definitions and Prototypes 5-5
5.2.5 NCB Structure Definition and NetBIOS Command Codes 5-5
5.2.6 Function Return Values 5-5

5.2.1 NETLIB Functions

The following table lists the NetBIOS commands of the MEMOBUS PLUS and the NETLIB library functions corresponding to these commands. All functions other than exec_netbios() initialize the NCB structure. The NETLIB functions call exec_netbios() to execute an int86X interrupt.

Table 5.2

Command	NETLIB Function	Explanation
RESET	ncb_reset()	Resets the SA85 and executes self-diagnostic processing.
–	ncb_sa85off	Stops the SA85.
CANCEL	ncb_cancel()	Stops processing.
STATUS	ncb_status()	Reads adapter status.
ADD_NAME	ncb_open()	Opens a path.
DELETE_NAME	ncb_close()	Closes a path.
SEND	ncb_send()	Sends a packet (data).
RECEIVE	ncb_receive()	Receives a packet (data) without wait processing.
RECEIVE_WAIT	ncb_receive_wait()	Receives a packet (data) with reception timeout processing.
SEND_DATAGRAM	ncb_send_datagram()	Sends global data.
RECEIVE_DATAGRAM	ncb_receive_datagram()	Receives global data.
–	ncb_set_slave_login()	Sets slave path login information.
–	ncb_set_sw_interrupt()	Sets interrupt vector.
–	exec_netbios()	Executes a NetBIOS command by using an int86X interrupt. This function cannot be used with applications that send and receive data to/from the Device Driver.

5.2.2 Path Required

Some functions require that the adapter path be specified before the functions are executed. There are four path types: Data Master (DM), Data Slave (DS), Program Master (PM), and Program Slave (PS). The ncb_open() function is used to specify the adapter path and the ncb_close() function is used to release the adapter path. The adapter path need not be specified for other functions; they can be executed regardless of whether the adapter path is specified. See section *5.4 NetBIOS Functions* for details on each function.

5.2.3 Message Buffer

The host application specifies a buffer pointer variable as a pointer to the data buffer used by the function and passes the variable to the ncb_send(), ncb_receive(), and ncb-receive_wait() functions. The buffer is allocated in MEMOBUS message format used for communications with the GL120 or GL130. One pointer is passed on to ncb_send_datagram() or ncb_receive_datagram() to identify the buffer used to transfer global data among the functions.

5.2.4 Function Definitions and Prototypes

The NETLIB functions are defined in the netlib.c file. When writing an application program, print this file as required for reference.

5.2.5 NCB Structure Definition and NetBIOS Command Codes

The NCB structure is defined in the header file netbios.h. This file also contains the NetBIOS command codes.

5.2.6 Function Return Values

The NCB_RETCODE variable is defined in the header file netbios.h. Return values are described in section *5.3 Return Values.*

5.3 Return Values

This section describes the return values of NetBIOS functions.

The execution result of each function is returned as a hexadecimal value to the NCB_RETCODE area. Return codes are defined in the header file netbios.h.

Table 5.3

Code	Variable	Explanation
00	#define ERR_success 0	Normal termination (with no error).
01	#define ERR_bad_buffer_length 1	The transmit command length or status command length is invalid. Transmit command length > 512 bytes Status command length < Minimum value
03	#define ERR_invalid 3	The NetBIOS command is invalid.
05	#define ERR_timeout 5	A timeout occurred before the transmit-receive command or status command terminated.
06	#define ERR_buffer_too_small 6	Reception buffer is too small. The buffer size specified in NCB_LENGTH is insufficient to hold the received data.
08	#define ERR_bad_session_num 8	The local session number specified in NCB_LSN is invalid.
09	#define ERR_no_RAM 9	Resources are insufficient. The adapter does not have enough RAM to execute the command.
0A	#define ERR_session_closed 0xa	The session has been interrupted by the remote node.
0B	#define ERR_cancel 0xb	The command has been canceled. The NCB command has been canceled by the ncb_cancel() command.
0D	#define ERR_dup_local_name 0xd	The local name has been duplicated. The same local name already existed when the ncb_open() command was executed.
0E	#define ERR_name_table_full 0xe	The local name table is full.
0F	#define ERR_active_session 0xf	The ncb_close() command is being executed, but the node with the local name is still active. The local name cannot be deleted until all sessions are closed.
11	#define ERR_sess_table_full 0x11	The local session table is full.
12	#define ERR_no_listen 0x12	A routing error has occurred. No response has been received from the specified node.
13	#define ERR_bad_name_num 0x13	The NCB_NUM variable is illegal.
14	#define ERR_no_answer 0x14	No response to the call has been received or the specified node does not exist.
15	#define ERR_no_local_name 0x15	An attempt was made to write an invalid name to the local name table.

Code	Variable	Explanation
16	#define ERR_duplicate_name 0x16	The same local name already exists in the network.
17	#define ERR_bad_delete 0x17	The local name has been deleted illegally.
18	#define ERR_abnomal_end 0x18	The session has been canceled incorrectly. An attempt was made to connect to an interrupted node.
19	#define ERR_name_error 0x19	Local names conflict. Two nodes are found to have the same local name.
1A	#define ERR_bad_packet 0x1a	An error has occurred in NetBIOS packet on the network.
21	#define ERR_card_busy 0x21	The adapter is busy. The adapter was unable to execute the command.
22	#define ERR_too_many_cmds 0x22	Many commands are in wait state.
23	#define ERR_bad_card_num 0x23	The NCB_LANA_NUM variable is invalid (0 or 1).
24	#define ERR_cancel_done 0x24	The command was completed before ncb_cancel() was executed. The return value of ncb_cancel() is the one that was returned after the previous command terminated normally.
26	#define ERR_no_cancel 0x26	The ncb_cancel() command is invalid. The desired NCB was not found.
40 to FE	–	A hardware error has occurred.
FF	#define ERR_busy 0xff	The command is being executed.

5.4 NetBIOS Functions

This section describes the NetBIOS functions.

5.4.1 ncb_cancel () 5-8
5.4.2 ncb_close () 5-9
5.4.3 ncb_open () 5-9
5.4.4 ncb_receive () 5-11
5.4.5 ncb_receive_datagram () 5-12
5.4.6 ncb_receive_wait () 5-13
5.4.7 ncb_reset () 5-14
5.4.8 ncb_sa85off () 5-14
5.4.9 ncb_send () 5-15
5.4.10 ncb_send_datagram () 5-16
5.4.11 ncb_set_slave_login () 5-17
5.4.12 ncb_set_sw_interrupt () 5-18
5.4.13 ncb_status () 5-18

5.4.1 ncb_cancel ()

1) Corresponding NetBIOS Command

CANCEL

2) Explanation

The ncb_cancel() function cancels the current processing before it terminates. This function does not clear the existing path. To clear the path, issue ncb_close() after ncb_cancel() is completed.

3) Path Required

DM, DS, PM, or PS path is required.

4) Syntax

```
int ncb_cancel (*ncbp)
NCB *ncbp;
```

a) Arguments

ncbp	NCB pointer required to cancel processing

b) Return Values

`ERR_success`	Always this value

5.4.2 ncb_close ()

1) Corresponding NetBIOS Command

DELETE_NAME

2) Explanation

The ncb_close() function closes the path opened by ncb_open(). Any queued processing on the path will be canceled.

3) Path Required

DM, DS, PM, or PS path is required.

4) Syntax

```
int ncb_close (*ncbp)
NCB *ncbp;
```

a) Arguments

`ncbp`	Pointer to the NCB allocated by ncb_open()

b) Return Values

`ERR_success`	Normal termination
`ERR_bad_name_num`	Invalid path

5.4.3 ncb_open ()

1) Corresponding NetBIOS Command

ADD_NAME

2) Explanation

The ncb_open() function opens the path for a node to communicate with the SA85. This function must be called before a node begins communicating with another node. A character string representing a path type and routing information is passed to the argument of this function.

a) One of the following can be specified as the path type:

DM: Data master path
DS: Data slave path
PM: Program master path
PS: Program slave path

b) Specify a character string in the following format:

"PP. R1. R2. R3. R4. R5"

PP:	SA85 path type (DM, DS, PM, or PS)
R1 to R5:	MEMOBUS PLUS routing to the destination node

A period is used to separate each portion of a character string. As shown in the following examples, unused portions must be padded with 0.

3) Examples

The MEMOBUS PLUS routing path is described in chapter *NO TAG MEMOBUS PLUS Network Overview.* The following are two examples of network routing to the SA85 and the GL120.

Routing to SA85:	DM.9.2.0.0.0 is a data master path to task 2 of the SA85 with node address 9.
Routing to GL120:	DM.8.3.0.0.0 is a data master path to the GL120 with node address 3 via the bridge with node address 8.

4) Example of Call

```
ncb_open ("DM.8.3.0.0.0") ;
```

"DM.8.3.0.0.0" is a character string representing a path type and routing information. 0 is the adapter number of the SA85.

5) NCB Pointer

This function returns a NCB pointer, or a NULL pointer if an error has occurred. The pointer is passed on to the following commands:

```
ncb_send()       ncb_receive_wait()      ncb_close()
ncb_receive()    ncb_set_slave_login()   ncb_cancel()
```

6) Path Required

The new path to be opened (DM, DS, PM, or PS) must be specified.

7) Syntax

```
NCB *ncb_open (*name. lan)
char *name;
int lan;
```

a) Arguments

`name`	Character-string pointer representing the path type and routing information
`lan`	Adapter number of SA85 (0 or 1)

b) Return Values

`NCB pointer`	See *NCB Pointer* in *5)* above.
`NULL`	When error occurs

5.4.4 ncb_receive ()

1) Corresponding NetBIOS Command

RECEIVE

2) Explanation

The ncb_receive() function receives a message from the specified path. If there is response data, the function will return the response data. If there is no response data, the function will immediately return ERR_timeout. Under a multitasking environment, this function provides better throughput than ncb_receive_wait().

3) Path Required

DM, DS, PM, or PS path is required.

4) Syntax

```
int ncb_receive (*ncbp, *buffer)
NCB *ncbp;
char *buffer;
```

a) Arguments

`ncbp`	Pointer to the NCB allocated by ncb_open()
`buffer`	Pointer to the received message buffer

b) Return Values

`ERR_success`	Normal termination
`ERR_bad_session_num`	NCB_NUM is 0
`ERR_bad_name_num`	Invalid path
`ERR_timeout`	No response received
`ERR_invalid`	No response in the queue
`ERR_busy`	Invalid path (NCB_NUM)
`ERR_invalid`	Returned data length is zero (NCB_LENGTH)
`ERR_no_listen`	Specified node does not exist

5.4.5 ncb_receive_datagram ()

1) Corresponding NetBIOS Command

RECEIVE_DATAGRAM

2) Explanation

The ncb_receive_datagram() function stores a specified amount of received global data in the buffer. For this function, allocate a NCB different from the NCB allocated by ncb_open(). NCB_LENGTH returns the number of received global data items.

3) Path Required

A path need not be specified for this function.

4) Syntax

```
int ncb_receive_datagram (*ncbp, node, *buffer, timeout,
adaptno)
NCB *ncbp;
int node;
char *buffer;
unsigned char timeout;
int adaptno;
```

a) Arguments

`ncbp`	Pointer to NCB
`node`	Address of the node to receive global data (1 to 64)
`buffer`	Pointer to the buffer to store received data
`timeout`	Timeout value in units of 0.5 second
`adaptno`	Adapter number of SA85 (1 or 0)

b) **Return Values**

ERR_success	Normal termination
ERR_card_busy	Specified node does not exist
ERR_no_answer	No global data

5.4.6 ncb_receive_wait ()

1) Corresponding NetBIOS Command

RECEIVE_WAIT

2) Explanation

This function receives a message from the specified path. Unlike ncb_receive(), this function waits until a message is received or a timeout error occurs.

3) Path Required

DM, DS, PM, or PS path is required.

4) Syntax

```
int ncb_receive_wait (*ncbp, *buffer, timeout)
NCB *ncbp;
char *buffer;
unsigned char timeout;
```

a) **Arguments**

ncbp	Pointer to the NCB allocated by ncb_open()
buffer	Pointer to the received message buffer
timeout	Timeout value set in units of 0.5 second

b) **Return Values**

ERR_success	Normal termination
ERR_bad_session_num	NCB_NUM is 0
ERR_bad_name_num	Invalid path
ERR_timeout	No response received
ERR_invalid	No response in the queue
ERR_busy	Invalid path (NCB_NUM)
ERR_invalid	Returned data length is zero (NCB_LENGTH)
ERR_no_listen	Specified node does not exist

5.4.7 ncb_reset ()

1) Corresponding NetBIOS Command

RESET

2) Explanation

The ncb_reset() function resets the SA85, executes self-diagnostic processing, and returns the SA85 to its initial state. It takes at least 100 ms to execute the function. This function can be enabled or disabled by setting the switch in CONFIG.SYS in the Device Driver setup.

3) Path Required

A path need not be specified for this function.

4) Syntax

```
ini ncb_reset (adaptno)
int adaptno;
```

a) Arguments

`adaptno`	Adapter number of the SA85 to be reset (0 or 1)

b) Return Values

`ERR_success`	Normal termination
`ERR_invalid`	Invalid adapter number
`0x80`	Self-diagnostic processing error

5.4.8 ncb_sa85off ()

1) Corresponding NetBIOS Command

None

2) Explanation

The ncb_sa85off() function stops the SA85.

Use this function when the application program is not to be used for a short while or when a program using another terminal is to be used. The Device Driver is used to stop SA85 polling. Use the ncb_reset() function to reset and restart the SA85.

3) Path Required

A path need not be specified for this function.

4) Syntax

```
int ncb_sa85off (adaptno)
int adaptno;
```

a) Arguments

adaptno	Adapter number of the SA85 to be stopped (0 or 1)

b) Return Values

ERR_success	Normal termination
ERR_bad_card_num	Invalid adapter number

5.4.9 ncb_send ()

1) Corresponding NetBIOS Command

SEND

2) Explanation

The ncb_send() function sends a message to the specified path.

For a data master path, an error will be returned if ncb_send() is executed before ncb_receive(). For a data slave path, an error will be returned if ncb_receive() is executed before ncb_send().

3) Path Required

DM, DS, PM, or PS path is required.

4) Syntax

```
int ncb_send (*ncbp, length, *buffer, timeout)
NCB *ncbp;
int length:
char *buffer;
unsigned char timeout;
```

a) Arguments

ncbp	Pointer to the NCB allocated by ncb_open()
length	Message buffer length (in bytes)
buffer	Pointer to the sent message buffer
timeout	Timeout value set in units of 0.5 second

b) Return Values

ERR_success	Normal termination
ERR_bad_session_num	NCB_NUM is 0
ERR_invalid	Reception processing by the data master path is in pending state or there is no previous reception processing by the data salve path

5.4.10 ncb_send_datagram ()

1) Corresponding NetBIOS Command

SEND_DATAGRAM

2) Explanation

The ncb_send_datagram() function sends a specified amount of global data. For this function, allocate a NCB different from the NCB allocated by ncb_open(). NCB_LENGTH returns the number of received global data items.

3) Path Required

A path need not be specified for this function.

4) Syntax

```
int  ncb_send_datagram  (*ncbp,  length,  *buffer,  timeout,
adaptno)
NCB *ncbp;
int length:
char *buffer;
unsigned char timeout;
int adaptno;
```

a) Arguments

ncbp	Pointer to NCB (empty NCB)
length	Message buffer length (in bytes)
buffer	Pointer to the sent message buffer
timeout	Timeout value set in units of 0.5 second
adaptno	Adapter number of SA85 (0 or 1)

b) Return Values

`ERR_success`	Normal termination
`ERR_card_busy`	Specified adapter was not found
`ERR_bad_card_num`	Invalid adapter number
`ERR_bad_buffer_length`	Buffer length > 64 bytes

5.4.11 ncb_set_slave_login ()

1) Corresponding NetBIOS Command

None

2) Explanation

The ncb_set_slave_login() function sets or clears the login status of a program slave path for the SA85. Once a path is established, the login command and the next programming command can be used.

3) Path Required

A PS path is required.

4) Syntax

```
int ncb_set_slave_login (*ncbp, login_status)
NCB *ncbp;
unsigned char login_status;
```

a) Arguments

`ncbp`	Pointer to the NCB allocated by ncb_open()
`login_status`	Desired login information (1: Sets login, 0: Clears login)

b) Return Values

`ERR_success`	Normal termination
`ERR_busy`	Error or adapter busy
`ERR_bad_name_num`	NCB_NUM data error
`ERR_no_listen`	Path not specified

5.4.12 ncb_set_sw_interrupt ()

1) Corresponding NetBIOS Command

None

2) Explanation

The ncb_set_sw_interrupt() function sets the software interrupt number to be used for interaction with the Device Driver.

3) Path Required

None

4) Syntax

```
int ncb_set_sw_interrupt (swInterrupt)
int swInterrupt
```

a) Arguments

`swInterrupt` Desired software interrupt number
Any number between 3 and 7 or between 10 and FF
The specified value must match the value specified in the device setting in CONFIG.SYS.

b) Return Values

`ERR_success`	Normal
`ERR_invalid`	Previous setting cannot be changed

5.4.13 ncb_status ()

1) Corresponding NetBIOS Command

STATUS

2) Explanation

The ncb_status() function returns the SA85 configuration information. For this function, allocate a NCB different from the NCB allocated by ncb_open(). NCB_LENGTH returns the number of received global data items.

This function receives 110 bytes of status information. The message buffer pointer in NCB_BUFFER must point to a buffer of at least 110 bytes.

The first and last bytes of the data stored in the reception buffer are the data patterns for self-diagnostic processing. OxAA, Ox55 (binary digits 0 and 1 are set alternately) This check sets and resets each bit.

Status information is set between bytes 2 and 109.

3) Path Required

A path need not be specified for this function.

4) Syntax

```
int ncb_status (*ncbp, adaptno)
NCB *ncbp;
int adaptno
```

a) Arguments

ncbp	Pointer to the set NCB
adaptno	Adapter number of SA85 (0 or 1)

b) Return Values

ERR_success	Normal
ERR_invalid	Self-diagnostic processing error

Changing Software Interrupt Number 5C

6

This chapter describes how to change the software interrupt number when using an application program or sample program.

6.1 Application Interrupt 6-2

6.2 MBPSTAT Interrupts 6-3

6.3 Sample Program Interrupts 6-3

6.1 Application Interrupt

This section describes how to change the software interrupt number when using an application program.

The software interrupt number used for interaction with the device driver is included in sw_interrupt, which is declared in the NETLIB.C file.

1) To change the software interrupt number for the application program, install the software as described in this manual and change the interrupt parameter specified for DEVICE in CONFIG.SYS. Then, change the software interrupt number in sw_interrupt to a new number.

2) The following list shows the process for modifying the software interrupt number.

Example:

```
/* main.c - arg1 is the software interrupt number
usage: main_/s5c */


extern int sw_interrupt;


int main ( int argc, char *argv [] )
{
/* read command line parameter - if not assigned,
assign default */


if (sscanf( argv[1], "/S%x", &sw_interrupt) !=1 )
     sw_interrupt = 0x5c ;
}
```

3) The ncb_set_sw_interrupt() function can also be used to change the software interrupt number. Sample program BDRESET.C shows an example of using this function.

6.2 MBPSTAT Interrupts

This section describes how to change the software interrupt number when using a Network Diagnostic Utility program.

The /S switch can be used with the Network Diagnostic Utility program MBPSTAT to change the software interrupt number. When using MBPSTAT, specify in the /S switch the software interrupt number specified in DEVICE in CONFIG.SYS.

6.3 Sample Program Interrupts

This section describes how to change the software interrupt number when using a sample program.

1) The /S switch can be used with SA85 reset programs SA85OFF and BDRESET to change the software interrupt number.

2) Since other sample programs are provided as references for users to write programs, the software interrupt number is set at 5C.

   a) Before executing sample programs, refer to the notes on setting the software interrupt number and node address in chapter *9 Sample Programs.*

   b) To execute a sample program using a different software interrupt number, add the sample program shown in section *6.1 Application Interrupt* to the source code.

Include Files

7

This chapter presents the include files for the NetBIOS functions provided by Modicon.

7.1 NETBIOS.H 7-1

7.2 NETLIB.H 7-5

7.1 NETBIOS.H

This section presents NETBIOS.H.

```
/=========================================================================
 *     MODICON, Inc., 1 High Street, North Andover, MA 01845, USA
 *     Copyright (c) $Date: 1994/07/15 13:16:45 $ - All rights reserved.
 *
 *     No part of this document may be reproduced in any form
 *     without the express written consent of MODICON, Inc.
 *
 *=fdoc====================================================================
 *
 *  File:     netbios. h
 *
 *
 *  Notes:    Include file for NetBIOS calls and definitions
 *
 *  Author:
 *
 *  RCSID:
 *
 *  RCS:
 *  $Revision: 3.34 $
 *  $State: R3_41 $
 *  $Locker: $
 *
 *  Revision History Log:
 *
 *  Revision 3.32   03/11/94
 *  Baseline
 *
 *=fdoc==================================================================*/

#ifndef _NETBIOS_H
#define _NETBIOS_H

#ifdef __cplusplus
    extern "C" {
#endif



#if defined (__OS2__) && defined (__32BIT__)
    #pragma seg16 ( NCB )
    #define FARPTR * _Seg16
    #define LINKAGE _Optlink
    #define OS216ENTRY _Far16 _Cdecl
#else
    #define FARPTR far *
    #define LINKAGE
    #define OS216ENTRY
```

```
#endif


/* Structure of Network Control Block (NCB) */
typedef struct
    {
    unsigned char NCB_COMMAND;       /* Command */
    unsigned char NCB_RETCODE;       /* Function return code */
    unsigned char NCB_LSN;           /* Local session number */
    unsigned char NCB_NUM;           /* Number of network name */
    char FARPTR NCB_BUFFER;          /* Far pointer to message buffer */
    unsigned short NCB_LENGTH;       /* Length of message buffer */
    unsigned char NCB_CALLNAME[16];  /* Name of session user is talking to */
    unsigned char NCB_NAME[16];      /* User's network name */
    unsigned char NCB_RTO;           /* Receive timeout in 500 ms increments. */
    unsigned char NCB_STO;           /* Send timeout - 500 ms increments */
    char FARPTR NCB_POST_ADDRESS;    /* Offset of "no-wait" interrupt call */
    unsigned char NCB_LANA_NUM;      /* Adapter number (must be 0 or 1) */
    unsigned char NCB_CMD_CPLT;      /* Command completion status */
    unsigned char NCB_RESERVE[14];   /* Reserved area for token ring */
    } NCB;


/* NetBIOS error return codes - returned in NCB_RETCODE */
#define ERR_success 0              /* NetBIOS command completed normally */
#define ERR_bad_buffer_length 1    /* Bad send or status buffer size */
#define ERR_invalid 3              /* Invalid NetBIOS command */
#define ERR_timeout 5              /* Command time-out has expired */
#define ERR_buffer_too_small 6     /* Receive buffer not big enough */
#define ERR_bad_session_num 8      /* Bad value in NCB_LSN */
#define ERR_no_RAM 9               /* LAN card doesn't have enough memory */
#define ERR_session_closed 0xa     /* This session is closed */
#define ERR_cancel 0xb             /* Command has been closed */
#define ERR_dup_local_name 0xd     /* Name already exists for this PC */
#define ERR_name_table_full 0xe    /* Local name table is full */
#define ERR_active_session 0xf     /* Can't delete name - used in session */
#define ERR_sess_table_full 0x11   /* Local session table is full */
#define ERR_no_listen 0x12         /* Remote PC not listening for call */
#define ERR_bad_name_num 0x13      /* Bad value in NCB_NUM field */
#define ERR_no_answer 0x14         /* No answer to CALL or no such remote */
#define ERR_no_local_name 0x15     /* No such name in local name table */
#define ERR_duplicate_name 0x16    /* Name is in use elsewhere on net */
#define ERR_bad_delete 0x17        /* Name incorrectly deleted */
#define ERR_abnormal_end 0x18      /* Session aborted abnormally */
#define ERR_name_error 0x19        /* Two or more identical names in use! */
#define ERR_bad_packet 0x1a        /* Bad NetBIOS packet on network */
#define ERR_card_busy 0x21         /* Network card is busy */
#define ERR_too_many_cmds 0x22     /* Too many NetBIOS commands queued */
#define ERR_bad_card_num 0x23      /* Bad NCB_LANA_NUM - must be 0 or 1 */
#define ERR_cancel_done 0x24       /* Command finished while cancelling */
#define ERR_no_cancel 0x26         /* Command can't be cancelled */
#define ERR_busy 0xff              /* Still processing command */

#ifndef NONETBIOS
```

```
/*  NetBIOS functions list - "WAIT" calls wait until command completes */
/*  While the others jump to the routine in NCB_POST when the NetBIOS */
/*  command completes and does an interrupt. */
#define RESET 0x32         /*  Reset adapter card and tables */
#define CANCEL 0x35        /*  Cancel command. NCB_BUFFER = cmd. */

#define STATUS 0xb3        /*  Status information for adapter */
#define STATUS_WAIT 0x33

#define TRACE 0xf9         /*  Token-Ring protocol trace */
#define TRACE_WAIT 0x79

#define UNLINK 0x70        /*  Unlink from IBM Remote Program */

#define ADD_NAME 0xb0             /*  Add name to name table */
#define ADD_NAME_WAIT 0x30

#define ADD_GROUP_NAME 0xb6       /*  Add group name to name table */
#define ADD_GROUP_NAME_WAIT 0x36

#define DELETE_NAME 0xb1          /*  Delete name from name table */
#define DELETE_NAME_WAIT 0x31

#define CALL 0x90          /*  Start session with NCB_NAME name */
#define CALL_WAIT 0x10

#define LISTEN 0x91        /*  Listen for call */
#define LISTEN_WAIT 0x11

#define HANG_UP 0x92       /*  End session with NCB_NAME name */
#define HANG_UP_WAIT 0x12

#define SEND 0x94          /*  Send data via NCB_LSN */
#define SEND_WAIT 0x14

#define SEND_NO_ACK 0xf1          /*  Send data without waiting for ACK */
#define SEND_NO_ACK_WAIT 0x71

#define CHAIN_SEND 0x97           /*  Send multiple data buffers */
#define CHAIN_SEND_WAIT 0x17

#define CHAIN_SEND_NO_ACK 0xf2    /*  Send multiple buffers without ACK */
#define CHAIN_SEND_NO_ACK_WAIT 0x72

#define RECEIVE 0x95              /*  Receive data from a session */
#define RECEIVE_WAIT 0x15

#define RECEIVE_ANY 0x96          /*  Receive data from any session */
#define RECEIVE_ANY_WAIT 0x16

#define SESSION_STATUS 0xb4       /*  Status of all sessions for name */
#define SESSION_STATUS_WAIT 0x34

#define SEND_DATAGRAM 0xa0        /*  Send un-ACKed message */
#define SEND_DATAGRAM_WAIT 0x20
```

```
#define SEND_BCST_DATAGRAM 0xa2   /*  Send broadcast message */
#define SEND_BCST_DATAGRAM_WAIT 0x22

#define RECEIVE_DATAGRAM 0xa1     /*  Receive un-ACKed message */
#define RECEIVE_DATAGRAM_WAIT 0x21

#define RECEIVE_BCST_DATAGRAM 0xa3 /*  Receive broadcast message */
#define RECEIVE_BCST_DATAGRAM_WAIT 0x23

#define SA85_OFF 0x37 /* Turn SA85 driver off (RESET to turn on) */
#define SA85_MASK 0x38 /* Return status of mask bit */
#define SA85_ION 0x39 /* Enable sa85 interrupts */
#define SA85_IOFF 0x3a /* Disable sa85 interrupts */

#define SET_SLAVE_LOGIN 0x3b /* Set/clear slave login status */

#endif

#ifdef __cplusplus
    }
#endif

#endif                     /* _NETBIOS_H */
```

7.2 NETLIB.H

This section presents NETLIB.H.

```
/*=========================================================================
 *     MODICON, Inc., 1 High Street, North Andover, MA 01845, USA
 *     Copyright (c) $Date: 1994/07/15 13:16:45 $ - All rights reserved.
 *
 *     No part of this document may be reproduced in any form
 *     without the express written consent of MODICON, Inc.
 *
 *=fdoc=====================================================================
 *
 *  File:     This header file contains the NETBIOS Modbus PLUS
 *                                   library prototypes.
 *
 *  Notes:
 *
 *  Author:
 *
 *  RCS:
 *  $Revision: 1.2 $
 *  $State: R3_41 $
 *  $Locker: $
 *
 *  Revision History Log:
 *
 *=fdoc===================================================================*/

#ifndef _NETLIB_H
#define _NETLIB_H

#ifdef __cplusplus
    extern "C" {
#endif

int far ncb_reset (int adaptno);
int far ncb_sa85off (int adaptno);
int far ncb_status (NCB far *ncbp, int adaptno);
int far ncb_send (NCB far *ncbp, int length, char far *buffer, unsigned char timeout);
int far ncb_receive_wait (NCB far *ncbp, char far *buffer, unsigned char timeout);
NCB far *far ncb_open (char far *name, int lan);
int far ncb_receive (NCB far *ncbp, char far *buffer);
int far ncb_close (NCB far *ncbp);
int far ncb_send_datagram (NCB far *ncbp, int length, char far *buffer, unsigned char time-
out, int adaptno);
int far ncb_receive_datagram (NCB far *ncbp, int node, char far *buffer, unsigned char time-
out, int adaptno);
int far ncb_cancel (NCB far *ncbp);
int far ncb_set_slave_login (NCB far *ncbp, unsigned char login_status);
int far ncb_set_sw_interrupt (int swInterrupt);
void exec_netbios (NCB far *ncbstruct);
```

```
#ifndef SW_INTERRUPT
#define SW_INTERRUPT         0x5c /*  Actual SW interrupt used by NETBIOS */
#endif

#ifdef __cplusplus
    {
#endif

#endif                       /* _NETLIB_H */
```

MBPSTAT Utility

8

This chapter describes how to use the MBPSTAT utility provided by Modicon.

8.1 MBPSTAT Program 8-2

8.1.1 Starting the MBPSTAT Program 8-2

8.1.2 Selecting the Network to be Analyzed 8-2

8.1.3 Network Selection Examples 8-3

8.2 MBPSTAT Menu Screen 8-4

8.3 MBPSTAT Menu Items 8-5

8.3.1 Set Routing Parameters 8-5

8.3.2 Monitor Network Activity 8-5

8.3.3 Read Global Data 8-6

8.3.4 Global Data Present Table 8-7

8.3.5 Node Active Station Table 8-7

8.3.6 Node Error Statistics 8-8

8.3.7 Token Station Table 8-8

8.3.8 Token Owner Work Table 8-9

8.3.9 Current Internal Path Transactions 8-9

8.3.10 Show Node Personality 8-10

8.3.11 Quit 8-10

8.1 MBPSTAT Program

MBPSTAT is the MEMOBUS PLUS Network Diagnostic Utility. It generates a list of active nodes, monitors the activity status of the entire network, and displays node error statistics.

8.1.1 Starting the MBPSTAT Program 8-2
8.1.2 Selecting the Network to be Analyzed 8-2
8.1.3 Network Selection Examples 8-3

8.1.1 Starting the MBPSTAT Program

1) The MBPSTAT program is started with software interrupt number 5C, which is the initial value of NetBIOS. The software interrupt number can be changed through the command line using the /S switch. Always specify in the /S switch the software interrupt number specified in DEVICE in CONFIG.SYS. Switch to the directory in which MBPSTAT.EXE resides. To execute the program with interrupt number 5C (initial value), type MBPSTAT. To execute the program with an interrupt number other than 5C, type in the interrupt number with the /S switch. For example, type MBPSTAT /S5B if the interrupt number is 5B.

INFO

If a DOS/V machine is used, always switch to an English environment before starting MBPSTAT. To switch to an English environment, type CHEV US on the command line and press the Enter key. To switch from the English environment to a Japanese environment, type CHEV JP on the command line and press the Enter key.

2) Board Number

After executing the program, the user is prompted to specify a board number to select the SA85 network to be analyzed. Enter the SA85 board number (0 or 1) specified in CONFIG.SYS.

8.1.2 Selecting the Network to be Analyzed

1) The MBSTAT program prompts the user to specify the routing path to the MEMOBUS PLUS network to be analyzed. For example, the user can specify the local network or another network connected via several bridges. The user can select only one network. The last entered network address indicates the network to be analyzed.

   a) Up to four bytes can be used to specify routing through up to four networks.

   b) The last nonzero number indicates the target network.

   c) The first zero is the end of the routing path.

2) Network routing examples are presented in the following section.

8.1.3 Network Selection Examples

1) If all zeros are specified as a routing path, routing to other networks will be prohibited.

2) Analysis information for the local MEMOBUS PLUS network (to which the SA85 is connected) can be displayed.

3) If 22 00 00 00 is specified as the routing path, analysis information for the adjacent network connected via bridge 22 will be displayed.

4) If 22 24 00 00 is specified as the routing path, analysis information for the third network connected via bridges 22 and 24 Will be displayed.

5) While the program is being executed, the current routing path can be switched to another routing path by specifying the path in the program menu.

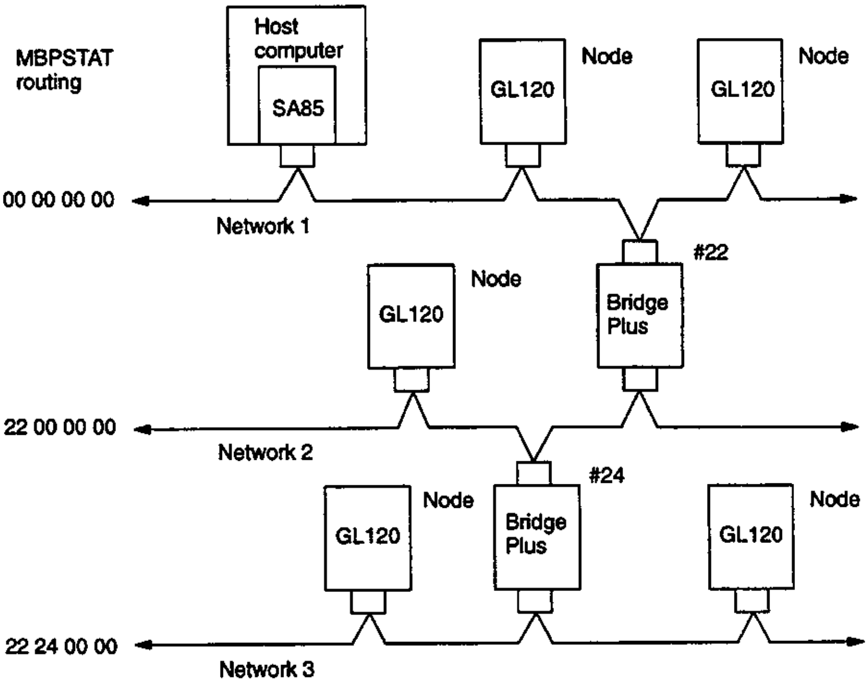
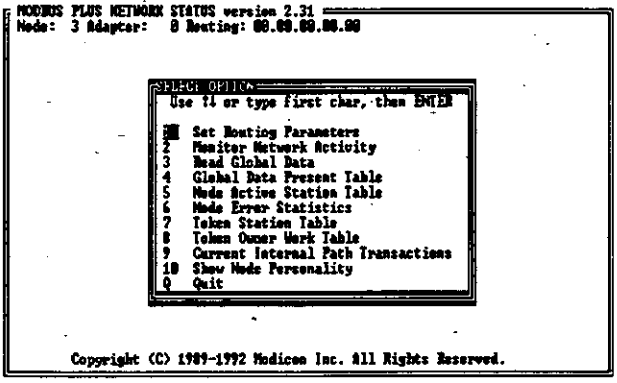


Figure 8.1

8.2 MBPSTAT Menu Screen

This section describes the MBPSTAT menu screen.

1) The MBPSTAT menu screen is displayed by specifying a board number and a network routing path. It displays the SA85 node address, the board number, and the routing path of the network being analyzed.



```
MODBUS PLUS NETWORK STATUS version 2.31
Node:  3 Adapter:   0 Routing: 00.00.00.00.00

            SELECT OPTION
             Use ↑↓ or type first char, then ENTER

            1   Set Routing Parameters
            2   Monitor Network Activity
            3   Read Global Data
            4   Global Data Present Table
            5   Node Active Station Table
            6   Node Error Statistics
            7   Token Station Table
            8   Token Owner Work Table
            9   Current Internal Path Transactions
            10  Show Node Personality
            Q   Quit

       Copyright (C) 1989-1992 Modicon Inc. All Rights Reserved.
```

Figure 8.2 MBPSTAT Menu Screen

To select an item, move the cursor to the desired item and press the Enter Key. Alternatively, enter the corresponding number (1 to 10) or Q, then press the Enter Key. To return from the item to the menu screen, press the Esc Key.

2) Entering a Node Address

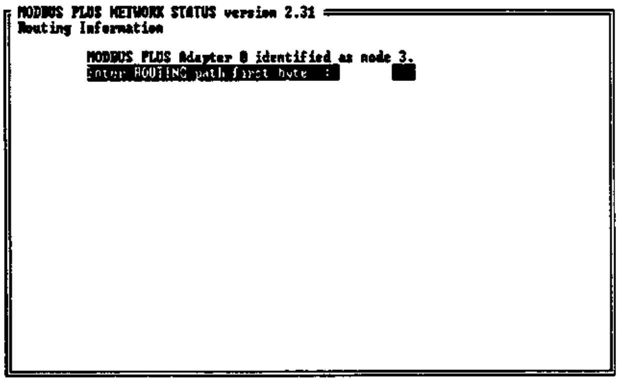
For some items, the entire network is analyzed. Therefore, a node address need not be specified. For other nodes, the state of one node is analyzed. The user is prompted to specify the address of the node to be analyzed. The node to be analyzed must always reside on the network with the specified network routing path.

8.3 MBPSTAT Menu Items

This section explains the MBPSTAT menu items.

8.3.1 Set Routing Parameters 8-5
8.3.2 Monitor Network Activity 8-5
8.3.3 Read Global Data 8-6
8.3.4 Global Data Present Table 8-7
8.3.5 Node Active Station Table 8-7
8.3.6 Node Error Statistics 8-8
8.3.7 Token Station Table 8-8
8.3.8 Token Owner Work Table 8-9
8.3.9 Current Internal Path Transactions 8-9
8.3.10 Show Node Personality 8-10
8.3.11 Quit 8-10

8.3.1 Set Routing Parameters

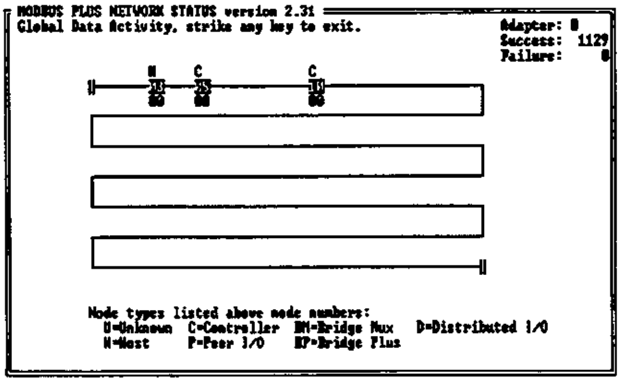


```
MODBUS PLUS NETWORK STATUS version 2.31
Routing Information

        MODBUS PLUS Adapter 0 identified as node 3.
        Enter ROUTING path first byte  :
```

Figure 8.3

This item allows the user to specify a new routing path without having to stop and restart MBPSTAT.

8.3.2 Monitor Network Activity



```
MODBUS PLUS NETWORK STATUS version 2.31
Global Data Activity. strike any key to exit.                    Adapter: 0
                                                                 Success:  1129
                                                                 Failure:     0

Node types listed above node numbers:
  U=Unknown  C=Controller  BM=Bridge Mux   D=Distributed I/O
  H=Host     P=Peer I/O    BP=Bridge Plus
```

Figure 8.4

1) This item displays the nodes that reside on the network with the specified routing path. The MBPSTAT program attempts to communicate with all nodes that have node addresses 1 to 64 and checks whether response can be received from each node.

2) This program also attempts to read global data from each node and displays the global data length.

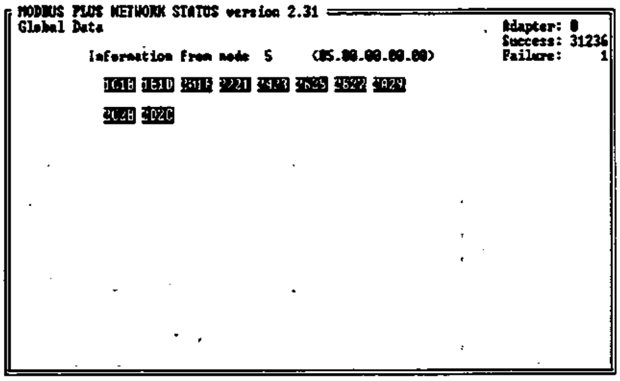
   a) The global data length is displayed below the node address.

   b) If no global data exists, "00" is displayed.

   c) When the status of another network connected via bridges is displayed, the program indicates that global data for that network cannot be accessed.

3) This item can be quit only by pressing the Esc Key.

8.3.3 Read Global Data



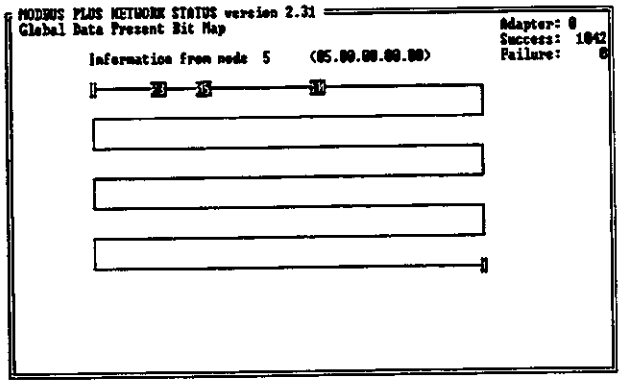
```
MODBUS PLUS NETWORK STATUS version 2.31
Global Data                                                    Adapter: 0
                                                               Success: 31236
          Information from node  5     (05.00.00.00.00)        Failure:     1

            1C1B 1E1D 201F 2221 2423 2625 2827 2A29

            2C2B 2D2C
```

Figure 8.5

1) This item reads global data from the specified node address and displays the data. If the global data length is 0 but global data exists, NULL data will be displayed.

2) This item can be quit only by pressing the Esc Key.

8.3.4 Global Data Present Table



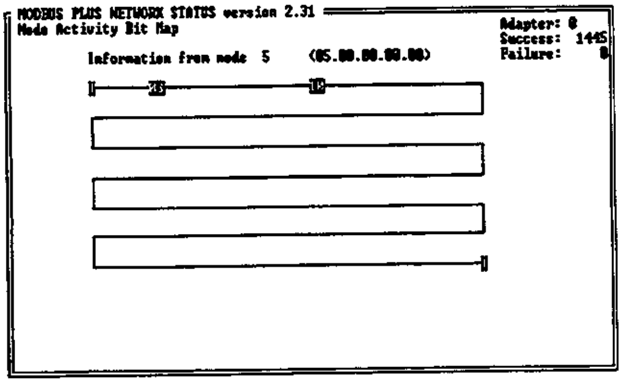
```
MODBUS PLUS NETWORK STATUS version 2.31                          Adapter: 0
Global Data Present Bit Map                                       Success:  1042
              Information from node  5     (05.00.00.00.00)       Failure:     0
```

Figure 8.6

1) This item displays a table map for the nodes that can be referred to by the specified node and from which global data has been written to the specified node. This table displays the addresses of the nodes from which the specified node has received global data since the diagnostic test was started. This table does not necessarily display only one node on the network.

2) This item can be quit only by pressing the Esc Key.

8.3.5 Node Active Station Table



```
MODBUS PLUS NETWORK STATUS version 2.31                          Adapter: 0
Node Activity Bit Map                                             Success:  1445
              Information from node  5     (05.00.00.00.00)       Failure:     0
```

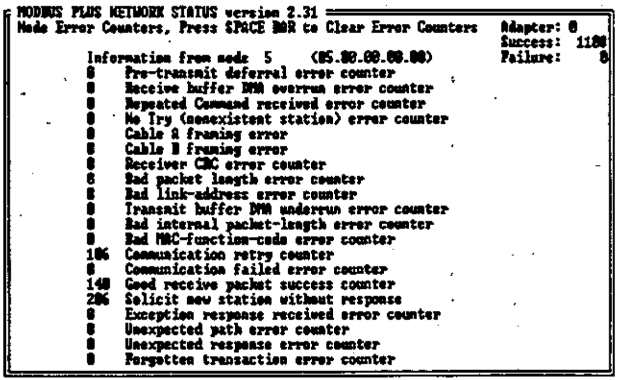
Figure 8.7

1) This item displays other active nodes that can be referenced by the specified node.

2) The nodes normally displayed on the screen were active during the test. The nodes highlighted on the screen were active at least once and then became inactive during the test.

3) This item can be quit only by pressing the Esc Key.

8.3.6 Node Error Statistics



```
MODBUS PLUS NETWORK STATUS version 2.31
Node Error Counters, Press SPACE BAR to Clear Error Counters    Adapter: 0
                                                                Success:  1100
        Information from node  5     (05.00.00.00.00)           Failure:     0
        0    Pre-transmit deferral error counter
        0    Receive buffer DMA overrun error counter
        0    Repeated Command received error counter
        0    No Try (nonexistent station) error counter
        0    Cable A framing error
        0    Cable B framing error
        0    Receiver CRC error counter
        0    Bad packet length error counter
        0    Bad link-address error counter
        0    Transmit buffer DMA underrun error counter
        0    Bad internal packet-length error counter
        0    Bad MAC-function-code error counter
        106  Communication retry counter
        0    Communication failed error counter
        140  Good receive packet success counter
        206  Solicit new station without response
        0    Exception response received error counter
        0    Unexpected path error counter
        0    Unexpected response error counter
        0    Forgotten transaction error counter
```

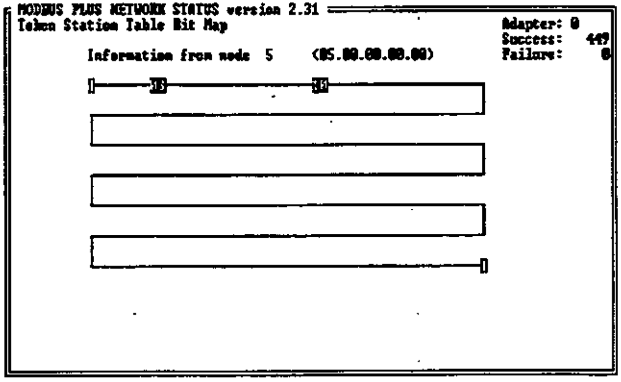
Figure 8.8

1) This item displays error statistics for the specified node. Pressing the space bar during operation clears the displayed data to zeros.

2) If the MBPSTAT utility is used in a single-cable network, framing errors that occur in one channel are also added up for the other channel. If, for example, a single-cable network is constructed using a cable A connector, cable B framing errors will also be added up for cable A. In this case, ignore the error count.

3) This item can be quit only by pressing the Esc Key.

8.3.7 Token Station Table



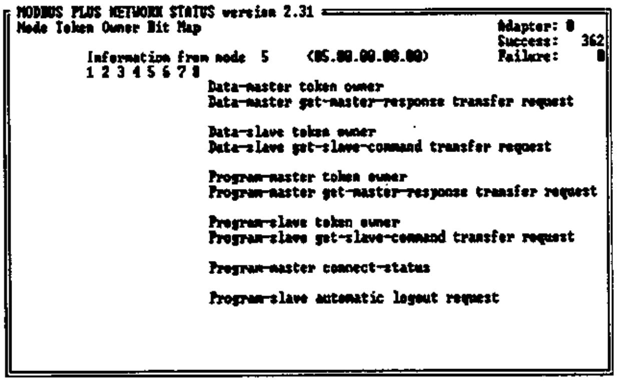
```
MODBUS PLUS NETWORK STATUS version 2.31
Token Station Table Bit Map                                     Adapter: 0
                                                                Success:   449
        Information from node  5     (05.00.00.00.00)           Failure:     0
```

Figure 8.9

1) This item displays the nodes from/to which the specified node receives and passes the token.

2) This item can be quit only by pressing the Esc Key.

8.3.8 Token Owner Work Table



```
MODBUS PLUS NETWORK STATUS version 2.31
Node Token Owner Bit Map                                          Adapter: 0
                                                                  Success:   362
        Information from node  5    (05.00.00.00.00)              Failure:     0
        1 2 3 4 5 6 7 8
                        Data-master token owner
                        Data-master get-master-response transfer request

                        Data-slave token owner
                        Data-slave get-slave-command transfer request

                        Program-master token owner
                        Program-master get-master-response transfer request

                        Program-slave token owner
                        Program-slave get-slave-command transfer request

                        Program-master connect-status

                        Program-slave automatic logout request
```

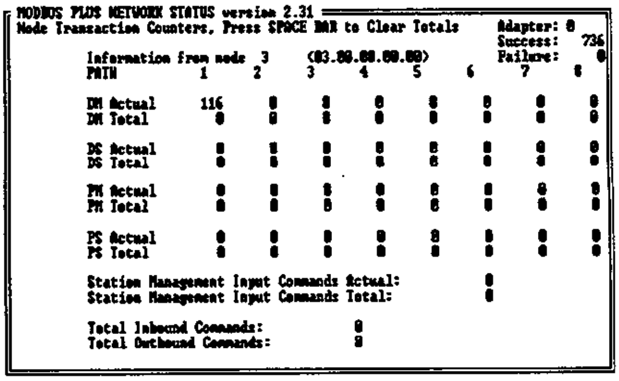
Figure 8.10

1) This item requests that the specified node provide a list of active paths. Each path on the list is a path between data inside the node and the program.

2) When a path becomes active, its number is displayed on the left of the row. The paths normally displayed on the screen were active during the test. The paths highlighted on the screen were active at least once and then became inactive during the test.

3) This item can be quit only by pressing the Esc Key.

8.3.9 Current Internal Path Transactions



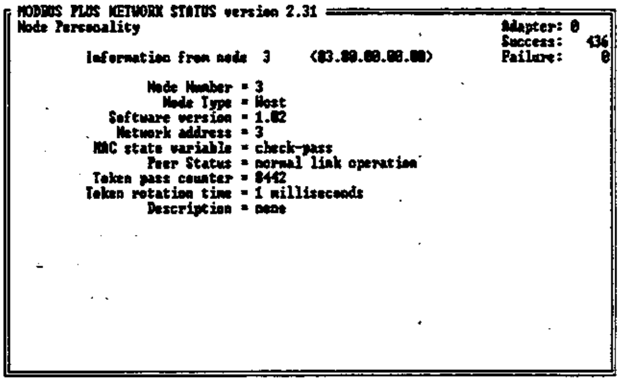
```
MODBUS PLUS NETWORK STATUS version 2.31
Node Transaction Counters, Press SPACE BAR to Clear Totals        Adapter: 0
                                                                  Success:   736
        Information from node  3    (03.00.00.00.00)              Failure:     0
        PATH          1     2     3     4     5     6     7     8

        DM Actual   116     0     0     0     0     0     0     0
        DM Total      0     0     0     0     0     0     0     0

        DS Actual     0     0     0     0     0     0     0     0
        DS Total      0     0     0     0     0     0     0     0

        PM Actual     0     0     0     0     0     0     0     0
        PM Total      0     0     0     0     0     0     0     0

        PS Actual     0     0     0     0     0     0     0     0
        PS Total      0     0     0     0     0     0     0     0

        Station Management Input Commands Actual:       0
        Station Management Input Commands Total:        0

        Total Inbound Commands:           0
        Total Outbound Commands:          0
```

Figure 8.11

1) This item displays the number of times each internal path for the specified node was used. Pressing the space bar during operation clears the displayed data to zeros.

2) This item can be quit only by pressing the Esc Key.

8.3.10 Show Node Personality



```
MODBUS PLUS NETWORK STATUS version 2.31
Node Personality                                              Adapter: 0
                                                              Success:   436
        Information from node  3    (03.00.00.00.00)          Failure:     0

                Node Number = 3
                  Node Type = Host
           Software version = 1.02
            Network address = 3
         MAC state variable = check-pass
                Peer Status = normal link operation
         Token pass counter = 8442
        Token rotation time = 1 milliseconds
                Description = none
```

Figure 8.12

1) This item displays the node type, software version, node operation status on the network, and other information for the specified node.

2) This item can be quit only by pressing the Esc Key.

8.3.11 Quit

This item quits MBPSTAT and returns to MS-DOS.

Sample Programs

9

This chapter describes how to use the sample programs provided by Modicon.

9.1 Sample Program Overview 9-2
9.1.1 What is a Sample Program? 9-2
9.1.2 Minimum System Configuration 9-3
9.1.3 Interactive Programs 9-3
9.1.4 Node Address 9-3
9.1.5 5C Interrupt 9-3
9.1.6 Running Sample Programs 9-4

9.2 Sample Program Details 9-5
9.2.1 MBPCHECK 9-5
9.2.2 READNET 9-5
9.2.3 TEST4 9-6
9.2.4 TEST4B 9-6
9.2.5 TEST4C and TESTSLAV 9-7
9.2.6 GLOBTEST 9-8
9.2.7 READNODE 9-8

9.3 Board Reset Programs 9-10
9.3.1 Description 9-10
9.3.2 SA85OFF.EXE 9-10
9.3.3 BDRESET.EXE 9-11

9.1 Sample Program Overview

This section describes the sample programs.

9.1.1 What is a Sample Program? 9-2
9.1.2 Minimum System Configuration 9-3
9.1.3 Interactive Programs 9-3
9.1.4 Node Address 9-3
9.1.5 5C Interrupt 9-3
9.1.6 Running Sample Programs 9-4

9.1.1 What is a Sample Program?

1) Sample programs are contained in the enclosed floppy disk. Use these sample programs as references when writing an application program for data communications between MEMOBUS PLUS network nodes.

2) The sample programs are shown below. Each sample program is provided as an executable file and a source code file written in C language. When writing an application program, use these sample programs as references to ensure correct operation.

a) TEST4.EXE

b) TEST4B.EXE

c) TEST4C.EXE

d) TESTSLAV.EXE

e) GLOBTESTS.EXE

f) READNODE.EXE

g) MBPCHECK.EXE

h) READNET.EXE

Note The sample programs are provided as coding examples. The sample programs cannot be run on the network unless modified. Before using a sample program, always check the source codes in the program, change the software interrupt number in the program and the node address based on the network configuration, then recompile the program. If the sample program is not modified properly based on the network configuration, unexpected operation will occur.

9.1.2 Minimum System Configuration

1) MBPCHECK and READNET

To run these programs, one or two SA85 Boards must be installed in the host computer and the network must be enabled.

2) GLOBTEST, TESTSLAV and TEST4C

To run these programs, one SA85 Board must be installed in each of two host computers and the network must be enabled.

3) TEST4 and READNODE

To run these programs, one SA85 Board must be installed in the host computer, the network must be enabled, and at least two PLCs must reside on the network.

4) TEST4B

To run this program, one SA85 Board must be installed in each of two host computers, the network must be enabled, and at least two PLCs must reside on the network.

9.1.3 Interactive Programs

1) GLOBTEST

GLOBTEST generates global data for the local host node. This program is used to write global data and test it in the node in which the Network Diagnostic Utility is running.

2) TEST4C and TESTSLAV

TEST4C and TESTSLAV are simultaneously run on two host nodes. These programs operate by transferring data between two nodes.

9.1.4 Node Address

1) For all programs other than GLOBTEST, specify a node address as an argument after the command. GLOBTEST does not require a node address.

2) These programs process nodes on the network by referring to their node addresses during operation.

9.1.5 5C Interrupt

1) The sample programs using NetBIOS use 5C interrupt. If another interrupt is to be used, modify and recompile the sample program by using the program shown in section *6.1 Application Interrupt*.

2) If another interrupt is specified using SA85OFF and BDRESET in the SA85 reset program, the sample program can be executed without modifications.

9.1.6 Running Sample Programs

1) To run a sample program, switch to the directory in which the executable program is stored. Then, specify the command name, option switches, and parameters.

2) See section *9.2 Sample Program Details* for details on option switches and parameters. If the sample program is executed with incorrect option switches and parameters specified, it will display the correct option switches and parameter setting methods.

INFO

If a DOS/V machine is used, always switch to an English environment before starting MBPSTAT. To switch to an English environment, type CHEV US on the command line and press the Enter key. To switch from the English environment to a Japanese environment, type CHEV JP on the command line and press the Enter key.

TERMS

Option switch

An option switch is an optional argument that is specified to run a command.

Parameter

A parameter is a mandatory argument that is specified to run a command.

9.2 Sample Program Details

This section details each sample program.

9.2.1 MBPCHECK 9-5
9.2.2 READNET 9-5
9.2.3 TEST4 9-6
9.2.4 TEST4B 9-6
9.2.5 TEST4C and TESTSLAV 9-7
9.2.6 GLOBTEST 9-8
9.2.7 READNODE 9-8

9.2.1 MBPCHECK

1) Start Command

MBPCHECK

2) Function

a) This program checks whether one or two SA85 Boards exist in a host computer. It checks the adapter number (0 or 1) and node address (1 to 64).

b) Message Example

```
MODBUS PLUS Adapter 0 identified as node 5.
```

9.2.2 READNET

1) Start Command

READNET
READNET routing-path

2) Function

a) This program checks all active nodes on the network. It checks the node address and the version of the software residing at the node.

b) The routing path of the network to be tested must be specified in the routing-path parameter. Specify 5-byte data as a network routing path. Use a period to separate each byte of data.

c) Enter "READNET ?" to display how to specify data correctly.

d) If the parameter is not specified, the program will check the local network. If the parameter is specified, the program will check the specified network.

e) Message Examples

```
Station 5 is a Host, Software version 2.01.
Station 9 is a Controller, Software version 2.02.
```

9.2.3 TEST4

1) Start Command

TEST4 slave_node

2) Function

a) This program reads the contents of the 125-word holding register from the specified slave node (PLC). It reads the contents of the holding registers with addresses 400001 to 400125.

b) This program continues to run until the user presses the CTRL and C keys or the CTRL and Break keys simultaneously.

c) If 125-word information is read normally, the data will be displayed.

9.2.4 TEST4B

1) Start Command

TEST4B 1st slave_node 2nd slave_node [/d]

2) Function

a) If the /d switch (enable flag for two SA85 Boards) is not specified, one SA85 will communicate with two PLCs. The program reads the contents of the 125-word holding register from each slave node specified in slave_node.

b) If the /d switch (enable flag for two SA85 Boards) is specified, two SA85 Boards (0 and 1) will be used. The program reads the contents of the 125-word holding register from each of the nodes specified in the first and second slave_node parameters for board 0.

c) The program reads holding register information (addresses 400001 to 400125) from the two PLCs.

d) This program continues to run until the user presses the CTRL and C keys or the CTRL and Break keys simultaneously.

e) If the request to read data from the first slave node is normal, "+" is displayed. If 125-word data is correctly read from the node of board 0, "–" is displayed. If the request to read data from the second slave node is normal, "<" is displayed. If 125-word data is correctly read from the node of board 1, ">" is displayed. Therefore, if processing continues normally, the following will be displayed: +<–>+<–>+<–>

9.2.5 TEST4C and TESTSLAV

1) Start Command

```
TEST4C slave_address
TESTSLAV
```

2) Function

a) These programs simultaneously run on two host devices. TEST4C reads the contents of the 125-word holding register from the SA85 in which TESTSLAV operates. The host device in which TESTSLAV operates is a MEMOBUS slave which is waiting for a data read request. This program continues to run until the user presses the CTRL and C keys or the CTRL and Break keys simultaneously.

b) If the request to read the contents of the 125-word register is normal, TEST4C displays "+". If the contents of the 125-word register are read normally, TEST4C displays "–". Therefore, if processing continues normally, the following will be displayed: +–+–+–+–

c) When processing is normal, TESTSLAV displays "+". If processing continues normally, the following will be displayed: +++++++

d) When these programs are not running simultaneously, the active program displays an error message until the other program is executed.

e) When TEST4C accesses a node in which TESTSLAV is not operating, "receive error = 18" is displayed as a routine error message.

f) When TEST4C is not running (or TESTSLAV does not receive a request in 15 seconds), TESTSLAV displays "Status = 0x204".

9.2.6 GLOBTEST

1) Start Command

```
GLOBTEST
```

2) Function

This program continues to generate global data from SA85 board 0. The main purpose of this program is to generate global data that can be read and displayed using the “Read Global Data” option on the MBPSTAT menu screen (for Network Diagnostic Utility MBPSTAT). This program continues to run until the user presses the CTRL and C keys or the CTRL and Break keys simultaneously.

9.2.7 READNODE

1) Start Command

```
READNODE ruthing_path reference length count
```

2) Function

a) This program reads input relay, coil, input register, and holding register information from the specified PLC and displays the information. This program continues to run until the value specified in the count parameter is reached or until the user presses the CTRL and C keys or the CTRL and Break keys simultaneously.

b) Parameter

(1) ruthing_path

This parameter specifies the network routing path of the node from which data is to be read. Specify the routing path of the destination node with one to five bytes of data. Use a period to separate each byte of data.

For example, if 10 is specified in the parameter, it means that node 10 on the local area network is selected as the target node. If 24.10 is specified in the parameter, it means that node 10 on the adjacent network connected via bridge 24 is selected as the target node. If 24.22.10 is specified in the parameter, it means that node 10 on the third network connected via bridges 24 and 22 is selected as the target node.

(2) reference

This parameter specifies a reference leading number (0x, 1x, 3x, 4x).

(3) length

This parameter specifies the data size from the reference leading number.

(4) count

This parameter specifies the number of times information is to be read from the node. The program terminates when the count specified in the parameter is reached. If 0 is specified in the parameter, the program will continue to run until the user presses the CTRL and C keys or the CTRL and Break keys simultaneously.

3) Entry Example

```
READNODE 24.5 400101 10 1
```

Data between reference numbers 400101 and 400110 is read only once from the node with node address 5 on the network connected via bridge 24.

9.3 Board Reset Programs

This section describes the board reset programs.

9.3.1 Description 9-10
9.3.2 SA85OFF.EXE 9-10
9.3.3 BDRESET.EXE 9-11

9.3.1 Description

1) When the SA85 is ready to operate, it remains in wait state until it actually operates. Therefore, if the SA85 is not to be used for a while, it can be stopped to prevent it from interfering with other new programs.

2) The following two programs are provided to stop and restart the SA85. These programs can be directly run from the MS-DOS command line. Alternatively, they can also be called from an application program.

3) The board reset function can be disabled using the /B switch of MBPHOST.SYS in CONFIG.SYS.

9.3.2 SA85OFF.EXE

1) This program disables the SA85 Board. Normally, use this program when the SA85 is not to be used for a while or when the host computer is to be used for another purpose. The SA85 can be restarted by executing BDRESET.EXE.

2) The main purpose of this program is to prevent the SA85 from interfering with the terminal emulation program executed in the host computer. If the SA85 is enabled, the serial port will not be handled properly.

3) The following two option switches are available.

a) /S: Specifies an interrupt number.

b) /N: Specifies a SA85 Board number (0 or 1).

Example:

```
SA85OFF /S5B /N1
```

This command string specifies interrupt number 5B and disables SA85 Board 1. The default values for the /S and /N option switches are interrupt number 5C and board 0, respectively.

9.3.3 BDRESET.EXE

1) This program initializes the SA85 Board, performs self-diagnostic processing, and sets the SA85 to its initial state. If the SA85 Board is initialized normally, it will be involved in network operations. This program performs the same processing as that performed by turning ON the host computer.

2) The following two option switches are available.

a) /S: Specifies an interrupt number.

b) /N: Specifies a SA85 Board number (0 or 1).

Example:

```
BDRESET /S5B /N1
```

This command string specifies interrupt number 5B and disables SA85 Board 1. The default values for the /S and /N option switches are interrupt number 5C and board 0, respectively.

MEMOBUS Commands

10

This chapter describes the MEMOBUS commands used for MEMOBUS PLUS.

10.1 MEMOBUS Protocol for MEMOBUS PLUS 10-3

10.1.1 Relationship between MEMOBUS Master and Slaves . 10-3

10.1.2 Creating MEMOBUS Commands/Response Messages . 10-3

10.2 MEMOBUS Communications 10-4

10.3 List of MEMOBUS Commands 10-5

10.4 Specifying Digital and Register References .. 10-6

10.5 Basic Functions 10-7

10.5.1 Reading Coil Status 10-7

10.5.2 Reading the Status of Specific Coils 10-8

10.5.3 Modifying Status of a Single Coil 10-9

10.5.4 Modifying Status of Multiple Coils 10-10

10.5.5 Reading Input Relay Status 10-11

10.5.6 Reading Input Register Contents 10-12

10.5.7 Reading Holding Register Contents 10-13

10.5.8 Writing to a Single Holding Register 10-14

10.5.9 Writing to Multiple Holding Registers 10-15

10.5.10 Writing a Mask to Holding Registers 10-17

10.5.11 Reading/Writing to Multiple Holding Registers 10-18

10.6 Expansion Functions 10-20

10.6.1 Reading Link Coil Status 10-20
10.6.2 Modifying Status of a Single Link Coil 10-21
10.6.3 Modifying Status of Multiple Link Coils 10-22
10.6.4 Reading Link Register Contents 10-23
10.6.5 Writing to a Single Link Register 10-25
10.6.6 Writing to Multiple Link Registers 10-26
10.6.7 Reading Constant Register Contents 10-27
10.6.8 Writing to a Single Constant Register 10-28
10.6.9 Writing to Multiple Constant Registers 10-29

10.7 Special Functions 10-31

10.7.1 Loopback Test 10-31
10.7.2 Reading Event Counter 10-32
10.7.3 Reading Communications Status 10-32
10.7.4 Reading Equipment Status 10-34
10.7.5 Reading the FIFO Register Contents 10-37
10.7.6 Reading Special Reference Status 10-38
10.7.7 Modifying Special Reference Status 10-39

10.1 MEMOBUS Protocol for MEMOBUS PLUS

This section describes the MEMOBUS protocol used for MEMOBUS PLUS.

10.1.1 Relationship between MEMOBUS Master and Slaves 10-3
10.1.2 Creating MEMOBUS Commands/Response Messages 10-3

This section describes the MEMOBUS commands used for MEMOBUS PLUS. It is intended for programmers who write application programs that enable communications between the SA85 and Yaskawa PLCs. Refer to the following manual for details on the MEMOBUS commands:

MEMOCON GL120, GL130 MEMOBUS User's Manual (SIEZ-C825-70.13)

The GL120 and GL130 use MEMOBUS protocol to transfer data and read status information. MEMOBUS PLUS uses MEMOBUS protocol to transfer data between the SA85 and the GL120/GL130 over a network.

10.1.1 Relationship between MEMOBUS Master and Slaves

1) The MEMOBUS commands allow only one master device for multiple slave devices on the network. The MEMOBUS PLUS 1:1 communications protocol allows multiple master devices for multiple slave devices. Simultaneous communications are processed by using several different types of paths within the devices.

2) The MEMOBUS protocol allows commands and responses to be transferred between the master and slave devices. One communication consists of a command sent from the MEMOBUS master device to a slave device and a response returned from the slave device to the master device.

10.1.2 Creating MEMOBUS Commands/Response Messages

The application in the host computer creates a MEMOBUS query message in the data buffer. The message contains a MEMOBUS function code and each element data described in this chapter. When the message is sent, the buffer is used by the NetBIOS function. Likewise, the slave application in the host computer uses the NetBIOS function to receive the query message in the buffer. The application can also create a response message in the buffer and send it from the buffer. The NetBIOS functions and network hardware perform the process for obtaining the network path of the MEMOBUS PLUS destination node and the process for pasting the MEMOBUS message in the MEMOBUS PLUS message packet.

10.2 MEMOBUS Communications

This section describes the MEMOBUS communications method.

1) Each MEMOBUS query message and response message consists of 8-bit data that can be a maximum of 252 bytes in length.

   a) Query Message

   The function code in a query message specifies the processing to be performed by the specified slave device. Other data in the query message contains the information required for the slave device to perform the specified processing. If, for example, function code 03 is specified to request the slave device to read holding register data, the leading address of the holding register and the number of read times must be specified in the query message.

   b) Response Message

   When the slave device returns a normal response, the function code in the response message is identical with the function code in the query message. The response message contains register data, status information, and other data collected by the slave device. When an error occurs, the function code is changed to indicate that the response message is faulty. The response message also contains an error code.

2) Apart from the following two items, MEMOBUS PLUS messages are almost the same as MEMOBUS messages.

   a) A MEMOBUS message contains a slave address. Before the MEMOBUS message is converted into a MEMOBUS PLUS message, the slave address is removed from the MEMOBUS message and replaced with a destination field for the MEMOBUS PLUS MAC field.

   b) The CRC/LRC error check area is removed from the MEMOBUS message. An error check is performed in the HDLC level CRC-16 field in the MEMOBUS PLUS message.

10.3 List of MEMOBUS Commands

- This section presents a list of MEMOBUS commands.

Table 10.1

Function code	Function
01_H	Read coil status
02_H	Read input relay status
03_H	Read holding register contents
04_H	Read input register contents
05_H	Modify status of a single coil
06_H	Write to a single holding register
07_H	Read status of specific coils
08_H	Loopback test
$0B_H$	Read event counter
$0C_H$	Read communications status
$0F_H$	Modify status of multiple coils
10_H	Write to multiple holding registers
11_H	Read equipment status
12_H	Read link coil status
13_H	Read constant register contents
15_H	Read link register contents
16_H	Write a mask to the holding register
17_H	Read and write to multiple holding registers
18_H	Read FIFO register contents
19_H	Modify status of a single link coil
$1A_H$	Write to a single constant register
$1B_H$	Write to a single link register
$1D_H$	Modify status of multiple link coils
$1E_H$	Write to multiple constant registers
$1F_H$	Write to multiple link registers
21_H	Read special reference status
22_H	Modify special reference status

10.4 Specifying Digital and Register References

▌ This section describes how to specify digital and register references.

In a MEMOBUS message, a function code is used to specify the reference type (such as internal coils, input relays, and holding registers). In a MEMOBUS message, all digital and register addresses are represented in hexadecimal notation and specified as addresses relative to the leading address (0). Each address is specified as an offset address (subtract the leading address of the specified reference).

For example, the leading address of the holding register for the GL120 is 400001. For the "read holding register contents" function, this address must be specified as 0000H. Likewise, the leading address of the coil is 000001. For the "read coil status" function, this address must be specified as 0000H. For example, the address of coil 000127 must be specified as 007E (hexadecimal) because the offset address is 000126.

Note In MEMOBUS, all numbers must be specified in hexadecimal notation and in higher-place and lower-place bytes. For coil 000127 (offset address 007E in hexadecimal notation) in the preceding example, specify 00 in the higher-place bytes and 7E in the lower-place bytes.

10.5 Basic Functions

This section describes the basic functions of the MEMOBUS system.

10.5.1 Reading Coil Status 10-7
10.5.2 Reading the Status of Specific Coils 10-8
10.5.3 Modifying Status of a Single Coil 10-9
10.5.4 Modifying Status of Multiple Coils 10-10
10.5.5 Reading Input Relay Status 10-11
10.5.6 Reading Input Register Contents 10-12
10.5.7 Reading Holding Register Contents 10-13
10.5.8 Writing to a Single Holding Register 10-14
10.5.9 Writing to Multiple Holding Registers 10-15
10.5.10 Writing a Mask to Holding Registers 10-17
10.5.11 Reading/Writing to Multiple Holding Registers 10-18

10.5.1 Reading Coil Status

1) Function

The status of a series of coils is read starting from a specified coil number up to the number of coils specified. The slave organizes data for the status of groups of 8 coils in numerical order, and returns a response message. If the number of coils is a multiple of 8, then the MSB (most significant bit) of the last data represents the status of the last coil. If the number of coils is not a multiple of 8, then the unused bits in the last data on the MSB side all become zeroes.

- Function code: 01_H

- No. of coils handled by a single message: 2,000

2) Example Message

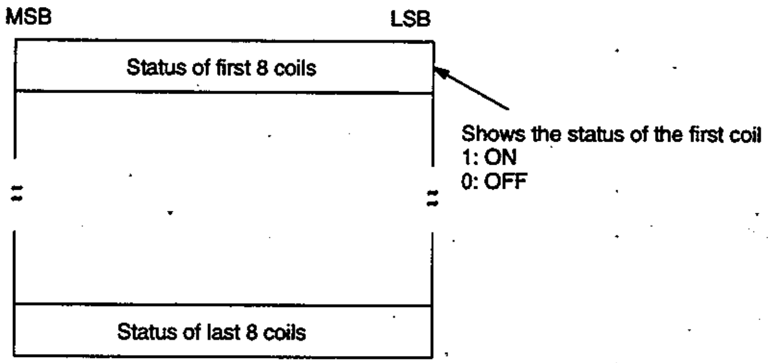
In this example, the status is read for a total of 37 coils numbered 000020 to 000056 from slave 2 ($20 = 14_H$, $37 = 25_H$).

◀EXAMPLE▶ **Query Message**

Function code		01_H	
Starting No.	High byte	00_H	
	Low byte	13_H	
No. of coils	High byte	00_H	1 to 2,000
	Low byte	25_H	

Note Starting No.
First coil number for GL120 or GL130: 000001

Data Read Order and Meaning



Response Message (Normal Operation)

Function code	01_H	
No. of data	05_H	} (N+7)/8 (see note)
Status of first 8 coils	CD_H	
Status of next 8 coils	$6B_H$	
Status of next 8 coils	$B2_H$	
Status of next 8 coils	$0E_H$	
Status of next 8 coils	$1B_H$	

Note N is the number of coils, and (N + 7)/8 is the maximum integer value.

10.5.2 Reading the Status of Specific Coils

1) **Function**

The status of a specified group of 8 coils (000001 to 000008) is read. The slave organizes the coil status data, and returns a response message.

- Function code: 07_H

- Maximum No. of coils handled by a single message: 8

2) **Example Message**

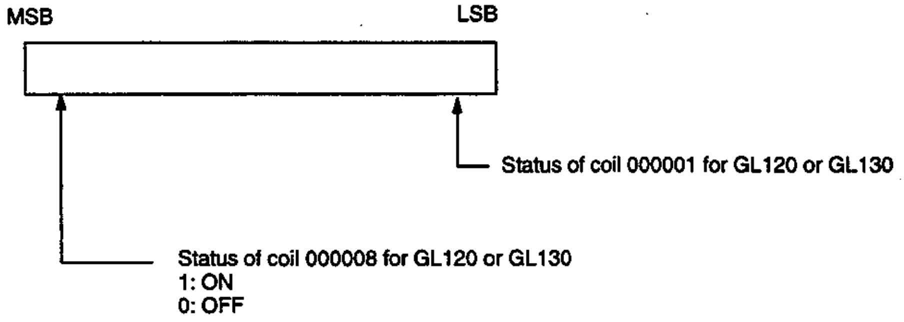
In this example, the status is read for specified coils numbered 000001 to 000008 from slave 1.

◀EXAMPLE▶

Query Message

Function code	07_H

Data Read Order and Meaning


MSB
LSB
Status of coil 000001 for GL120 or GL130
Status of coil 000008 for GL120 or GL130
1: ON
0: OFF

Response Message (Normal Operation)

Function code	07_H
Status of 8 coils	03_H

10.5.3 Modifying Status of a Single Coil

1) **Function**

A specified coil status is changed to a specified state (ON or OFF). The master specifies ON or OFF, and sends a query message.

- Function code: 05_H

- Maximum No. of coils handled by a single message: 1

2) **Example Message**

In this example, the status is modified for slave coil 000173 (173 = AD_H).

◀EXAMPLE▶ **Query Message**

Function code		05_H
Specified no.	High byte	00_H
	Low byte	AC_H
Specified status	High byte	FF_H
	Low byte	00_H

$FF00_H$: ON
0000_H: OFF

Note Specified No.
Coil no.: 000001

Response Message (Normal Operation)

Function code		05_H
Specified no.	High byte	00_H
	Low byte	AC_H
Specified status	High byte	FF_H
	Low byte	00_H

Note The coil that you wish to modify should always be set in the Disable mode. The instruction can be executed in the Enable mode, but be very careful because the coil status may be overwritten despite the internal logic decoding results when the coil resides in the ladder.

10.5.4 Modifying Status of Multiple Coils

1) **Function**

The status of a series of coils starting from a specified coil number up to the number of coils specified is modified to a specified state (ON or OFF). The master organizes data specifying ON or OFF for groups of 8 coils in numerical order, and sends a query message. If the number of coils is a multiple of 8, then the MSB of the last data specifies the status of the last coil, but if the number of coils is not a multiple of 8, then unused bits are generated for the last data on the MSB side. The unused bits, however are ignored on the slave side.

- Function code: $0F_H$

- Maximum No. of coils handled by a single message: 800

2) **Example Message**

In this example, the status is modified for a total of 10 coils numbered 000020 to 000029 in slave 1 (20 = 14_H, 10 = $0A_H$).

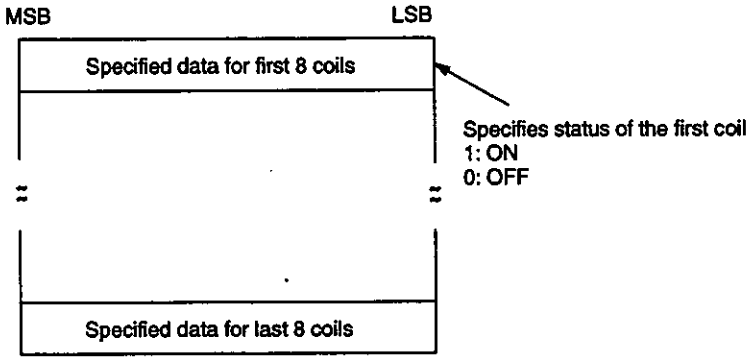
◀EXAMPLE▶ **Query Message**

Function code		$0F_H$	
Starting No.	High byte	00_H	
	Low byte	13_H	
No. of coils	High byte	00_H	1 to 800
	Low byte	$0A_H$	
No. of data		02_H	(N+7)/8 (see note a)
Specified data for first 8 coils		00_H	
Specified data for next 8 coils		00_H	

Note (a) N is the number of coils, and (N + 7)/8 is the maximum integer value.

(b) **Starting No.**
First coil number: 000001

Specified Data Order and Meaning


MSB
LSB
Specified data for first 8 coils
Specifies status of the first coil
1: ON
0: OFF
Specified data for last 8 coils

Response Message (Normal Operation)

Function code		$0F_H$
Starting No.	High byte	00_H
	Low byte	13_H
No. of coils	High byte	00_H
	Low byte	$0A_H$

Note The coil that you wish to modify should always be set in the Disable mode. The instruction can be executed in the Enable mode, but be very careful because the coil status may be overwritten despite the internal logic decoding results.

10.5.5 Reading Input Relay Status

1) Function

The status of a series of input relays is read starting from a specified input relay number up to the number of relays specified. The slave organizes data for the status in groups of 8 relays in numerical order, and returns a response message. If the number of relays is a multiple of 8, then the MSB of the last data represents the status of the last relay, but if the number of relays is not a multiple of 8, then the unused bits in the last data on the MSB side all become zeroes.

- Function code: 02_H

- Maximum No. of relays handled by a single message: 2,000

2) Example Message

In this example, the status is read for a total of 22 input relays numbered 100197 to 100218 from slave (197 = $C5_H$, 22 = 16_H).

◀EXAMPLE▶

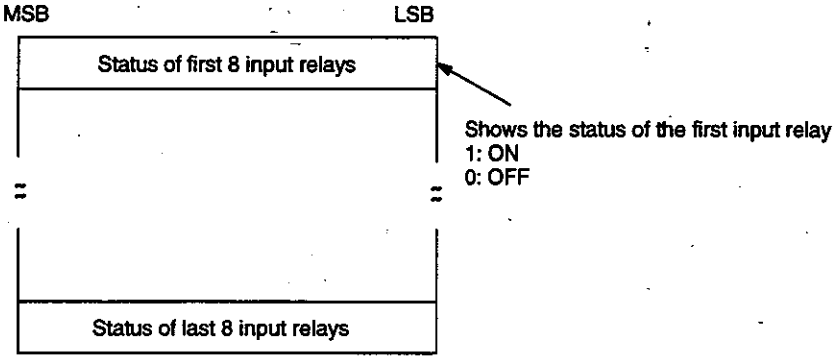
Query Message

Function code		02_H
Starting No.	High byte	00_H
	Low byte	$C4_H$
No. of relays	High byte	00_H
	Low byte	16_H

} 1 to 2,000

Note Starting No.
First input relay number:100001

Data Read Order and Meaning


MSB
LSB
Status of first 8 input relays
Shows the status of the first input relay
1: ON
0: OFF
Status of last 8 input relays

Response Message (Normal Operation)

Function code	02_H	
No. of data	03_H	(N+7)/8 (see note)
Status of first 8 input relays	AC_H	
Status of next 8 input relays	DB_H	
Status of next 8 input relays	35_H	

Note N is the number of coils, and (N + 7)/8 is the maximum integer value.

10.5.6 Reading Input Register Contents

1) Function

The contents of a series of input registers is read starting from a specified input register number up to the number of registers specified. The slave splits the contents of the input registers into the most significant 8 bits and the least significant 8 bits, and then organizes the input register data in numerical order before returning a response message.

- Function code: 04_H

- Maximum No. of registers handled by a single message: 125

2) Example Message

In this example, the contents are read for slave input register 300009.

◀EXAMPLE▶ **Query Message**

Function code		04_H	
Starting No.	High byte	00_H	
	Low byte	08_H	
No. of registers	High byte	00_H	1 to 125
	Low byte	01_H	

Note Starting No.

First input register number: 300001

Data Read Order and Meaning

MSB LSB

Most significant 8-bit data for the first input register
Least significant 8-bit data for the first input register
Most significant 8-bit data for the last input register
Least significant 8-bit data for the last input register

Response Message (Normal Operation)

Function code		04_H
No. of data		02_H
Input register contents	High byte	05_H
	Low byte	39_H

No. of input registers × 2

10.5.7 Reading Holding Register Contents

1) **Function**

The contents of a series of holding registers is read starting from a specified holding register number up to the number of registers specified. The slave splits the contents of the holding register into the most significant 8 bits and the least significant 8 bits, and then organizes the holding register data in numerical order before returning a response message.

- Function code: 03_H

- Maximum No. of registers handled by a single message: 125

2) **Example Message**

In this example, the contents are read for a total of three holding registers numbered 400108 to 400110 from slave (108 = $6C_H$).

◀EXAMPLE▶ **Query Message**

Function code		03_H	
Starting No.	High byte	00_H	
	Low byte	$6B_H$	
No. of registers	High byte	00_H	1 to 125
	Low byte	03_H	

Note Starting No.
First holding register number: 400001

Data Read Order and Meaning

MSB LSB

Most significant 8-bit data for the first holding register
Least significant 8-bit data for the first holding register
Most significant 8-bit data for the last holding register
Least significant 8-bit data for the last holding register

Response Message (Normal Operation)

Function code		03_H	
No. of data		06_H	No. of input registers × 2
First holding register contents	High byte	02_H	
	Low byte	$2B_H$	
Next holding register contents	High byte	00_H	
	Low byte	00_H	
Next holding register contents	High byte	00_H	
	Low byte	63_H	

10.5.8 Writing to a Single Holding Register

1) Function

Specified data is written to a specified holding register. The master splits and organizes the write data into the most significant 8 bits and the least significant 8 bits, and then sends a query message.

- Function code: 06_H

- Maximum No. of registers handled by a single message: 1

2) **Example Message**

In this example, the master writes to slave holding register 400136 (136 = 88_H)

◀EXAMPLE▶ **Query Message**

Function code		06_H
Specified no.	High byte	00_H
	Low byte	87_H
Write data	High byte	03_H
	Low byte	$9E_H$

} Write data

Note Specified No.
Holding register no.: 400001

Data Write Order and Meaning

MSB LSB

Most significant 8-bit write data
Least significant 8-bit write data

Response Message (Normal Operation)

Function code		06_H
Specified no.	High byte	00_H
	Low byte	87_H
Write data	High byte	03_H
	Low byte	$9E_H$

10.5.9 Writing to Multiple Holding Registers

1) **Function**

Specified data is written to a series of holding registers starting from a specified holding register number up to the number of registers specified. The master splits the write data into the most significant 8 bits and the least significant 8 bits, and then organizes the write data in numerical order before sending a query message.

- Function code: 10_H

- Maximum No. of registers handled by a single message: 100

2) Example Message

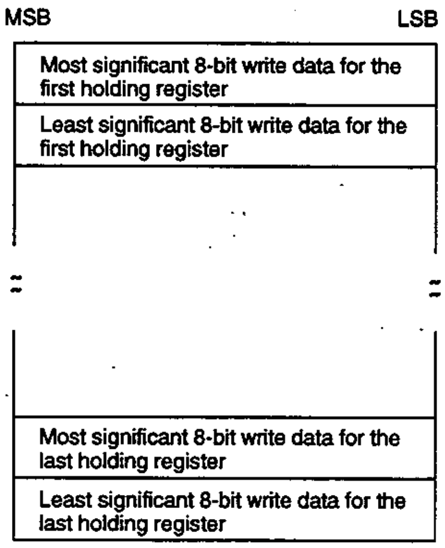
In this example, data is written to holding registers 400136 and 400137 at slave (136 = 88_H).

◀EXAMPLE▶ **Query Message**

Function code		10_H	
Starting No.	High byte	00_H	
	Low byte	87_H	
No. of registers	High byte	00_H	1 to 100
	Low byte	02_H	
No. of data		04_H	No. of holding registers × 2
Data written to the first holding register	High byte	00_H	
	Low byte	$0A_H$	
Data written to the next holding register	High byte	01_H	
	Low byte	02_H	

Note Starting No.
First holding register number: 400001

Data Write Order and Meaning


MSB LSB

Most significant 8-bit write data for the first holding register

Least significant 8-bit write data for the first holding register

Most significant 8-bit write data for the last holding register

Least significant 8-bit write data for the last holding register

Response Message (Normal Operation)

Function code		10_H
Starting No.	High byte	00_H
	Low byte	87_H
No. of registers	High byte	00_H
	Low byte	02_H

10.5.10 Writing a Mask to Holding Registers

1) Function

A specified bit pattern is written to the contents of a specified holding register. With an AND mask, each bit in the holding register is ANDed with the corresponding bit in the AND mask, and with an OR mask, each bit in the holding register is ORed with the corresponding bit in the OR mask. The holding register is modified accordingly.

Note An OR mask is valid for only the bits that have been set to 0 by an AND mask.

- Function code: 16_H

- Maximum No. of registers handled by a single message: 1

2) Example Message

In this example, an AND mask is written to slave holding register 400136 (136 = 88_H).

◀EXAMPLE▶ **Query Message**

Function code		16_H
Specified no.	High byte	00_H
	Low byte	87_H
AND mask data	High byte	05_H
	Low byte	05_H
OR mask data	High byte	00_H
	Low byte	00_H

Note Specified No.
Holding register no.: 400001 for GL120 or GL130

Mask Data Order and Meaning

MSB LSB

Most significant 8-bit mask data
Least significant 8-bit mask data

Response Message (Normal Operation)

Function code		16_H
Specified no.	High byte	00_H
	Low byte	87_H
AND mask data	High byte	05_H
	Low byte	05_H
OR mask data	High byte	00_H
	Low byte	00_H

10.5.11 Reading/Writing to Multiple Holding Registers

1) Function

The contents of a series of holding registers is read starting from a specified holding register number up to the number of registers specified. At the same time, the function writes the respective data to a series of holding registers starting from a specified holding register number up to the number of registers specified. The master splits the write data into the most significant 8 bits and the least significant 8 bits, and then organizes the holding register data in numerical order before sending a query message. On the other side, the slave splits the contents of the holding register into the most significant 8 bits and the least significant 8 bits, and then organizes the holding register data in numerical order before returning a response message.

- Function code: 17_{H}

- Maximum No. of registers handled by a single message: 125 read and 100 write

2) Example Message

In this example, the contents are read for a total of three holding registers numbered 400108 to 400110 from slave, and data is written to holding registers 400136 and 400137 ($108 = 6C_{H}$, $136 = 88_{H}$).

◀EXAMPLE▶ **Query Message**

Function code		17_{H}	
Read start no.	High byte	00_{H}	
	Low byte	$6B_{H}$	
No. of read units	High byte	00_{H}	1 to 125
	Low byte	03_{H}	
Write start no.	High byte	00_{H}	
	Low byte	87_{H}	
No. of write units	High byte	00_{H}	1 to 100
	Low byte	02_{H}	
No. of data		04_{H}	No. of write holding registers × 2
Write data to the first holding register	High byte	00_{H}	
	Low byte	$0A_{H}$	
Write data to the next holding register	High byte	01_{H}	
	Low byte	02_{H}	

Note (a) **Read Starting No.**

First holding register number: 400001

(b) **Write Starting No.**

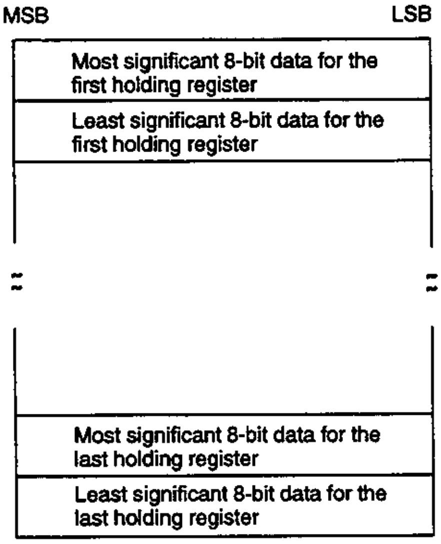
First holding register number: 400001

Response Message (Normal Operation)

Function code		17_H
No. of data		06_H
First holding register contents	High byte	02_H
	Low byte	$2B_H$
Next holding register contents	High byte	00_H
	Low byte	00_H
Next holding register contents	High byte	00_H
	Low byte	63_H

No. of read holding registers × 2

Data Read and Write, Order and Meaning


MSB
LSB
Most significant 8-bit data for the first holding register
Least significant 8-bit data for the first holding register
Most significant 8-bit data for the last holding register
Least significant 8-bit data for the last holding register

10.6 Expansion Functions

This section describes link coils, constant registers and other expansion functions and presents their message examples.

10.6.1 Reading Link Coil Status 10-20
10.6.2 Modifying Status of a Single Link Coil 10-21
10.6.3 Modifying Status of Multiple Link Coils 10-22
10.6.4 Reading Link Register Contents 10-23
10.6.5 Writing to a Single Link Register 10-25
10.6.6 Writing to Multiple Link Registers 10-26
10.6.7 Reading Constant Register Contents 10-27
10.6.8 Writing to a Single Constant Register 10-28
10.6.9 Writing to Multiple Constant Registers 10-29

10.6.1 Reading Link Coil Status

1) Function

The status of a series of link coils is read starting from a specified link coil number up to the number of coils specified. The slave organizes data for the status of groups of 8 link coils in numerical order, and returns a response message. If the number of link coils is a multiple of 8, then the MSB (most significant bit) of the last data represents the status of the last link coil, but if the number of link coils is not a multiple of 8, then the unused bits in the last data on the MSB side all become zeroes.

- Function code: 12_H

- No. of coils handled by a single message: 2,000

2) Example Message

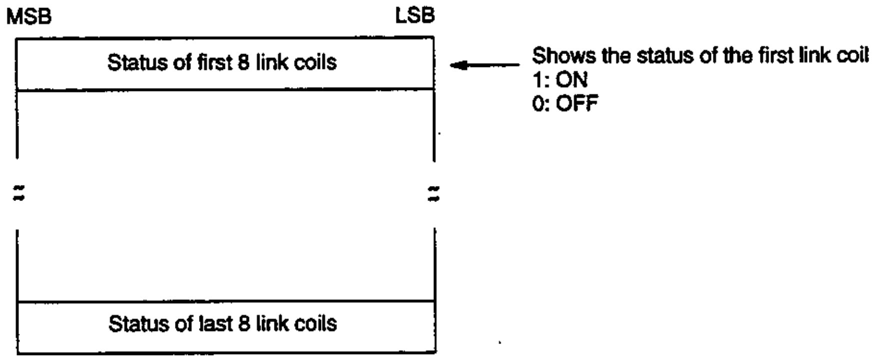
In this example, the status is read for a total of 37 link coils numbered D10020 to D10056 from slave (GL120 or GL130) (10020 = 2724_H, 37 = 25_H).

◀EXAMPLE▶ **Query Message**

Function code		12_H	
Starting No.	High byte	27_H	
	Low byte	23_H	
No. of coils	High byte	00_H	1 to 2,000
	Low byte	25_H	

Note Starting No.
First link coil number: D00001

Data Read Order and Meaning


MSB
LSB
Status of first 8 link coils
Shows the status of the first link coil
1: ON
0: OFF
Status of last 8 link coils

Response Message (Normal Operation)

Function code	12_H	
No. of data	05_H	} (N+7)/8 (see note)
Status of first 8 link coils	CD_H	
Status of next 8 link coils	$6B_H$	
Status of next 8 link coils	$B2_H$	
Status of next 8 link coils	DE_H	
Status of next 8 link coils	$1B_H$	

Note N is the number of coils, and (N + 7)/8 is the maximum integer value.

10.6.2 Modifying Status of a Single Link Coil

1) **Function**

A specified link coil status is changed to a specified status (ON or OFF). The master specifies ON or OFF status, and sends a query message.

- Function code: 19_H

- Maximum No. of coils handled by a single message: 1

2) **Example Message**

In this example, the status is modified for slave (GL120 or GL130) link coil D10173 (10173 = $27BD_H$).

◀EXAMPLE▶ **Query Message**

Function code		19_H	
Specified no.	High byte	27_H	
	Low byte	BC_H	
Specified status	High byte	FF_H	} ON with FF00 OFF with 0000
	Low byte	00_H	

Note Specified No.
Link coil no.: D00001

Response Message (Normal Operation)

Function code		19_H
Specified no.	High byte	27_H
	Low byte	BC_H
Specified status	High byte	FF_H
	Low byte	00_H

Note The link coil that you wish to modify should always be set in the Disable mode. The instruction can be executed in the Enable mode, but be very careful because the link coil status may be overwritten despite the internal logic decoding results.

10.6.3 Modifying Status of Multiple Link Coils

1) Function

The status of a series of link coils starting from a specified link coil number up to the number of coils specified is modified to a specified status (ON or OFF). The master organizes data specifying ON or OFF status for groups of 8 link coils in numerical order, and sends a query message. If the number of link coils is a multiple of 8, then the MSB of the last data specifies the status of the last link coil, but if the number of link coils is not a multiple of 8, then unused bits are generated in the last data on the MSB side. The unused bits, however, are ignored on the slave side.

- Function code: $1D_H$

- Maximum No. of coils handled by a single message: 800

2) Example Message

In this example, the status is modified for a total of 10 link coils numbered D10020 to D10029 at slave (GL120 or GL130) (10020 = 2724_H, 10 = $0A_H$).

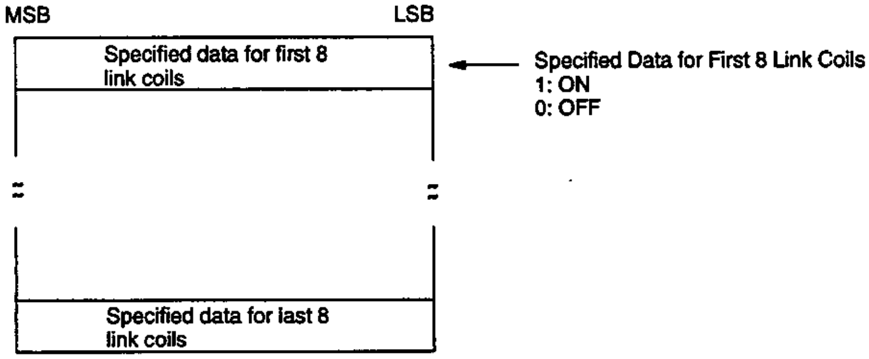
◀EXAMPLE▶ **Query Message**

Function code		$1D_H$	
Starting No.	High byte	27_H	
	Low byte	23_H	
No. of coils	High byte	00_H	1 to 800
	Low byte	$0A_H$	
No. of data		02_H	(N+7)/8 (see note b)
Specified data for first 8 link coils		00_H	
Specified data for next 8 link coils		00_H	

Note (a) **Starting No.**
First link coil number: D00001

(b) N is the number of link coils, and (N + 7)/8 is the maximum integer value.

Data Read Order and Meaning


MSB
LSB
Specified data for first 8 link coils
Specified Data for First 8 Link Coils
1: ON
0: OFF
Specified data for last 8 link coils

Response Message (Normal Operation)

Function code		$1D_H$
Starting No.	High byte	27_H
	Low byte	23_H
No. of coils	High byte	00_H
	Low byte	$0A_H$

Note The link coil that you wish to modify should always be set in the Disable mode. The instruction can be executed in the Enable mode, but be very careful because the link coil status may be overwritten despite the internal logic decoding results.

10.6.4 Reading Link Register Contents

1) Function

The contents of a series of link registers is read starting from a specified link register number up to the number of registers specified. The slave splits the contents of the link register into the most significant 8 bits and the least significant 8 bits, and then organizes the link register data in numerical order before returning a response message.

- Function code: 15_H

- Maximum No. of registers handled by a single message: 125

2) Example Message

In this example, the contents are read for a total of three link registers numbered R10001 to R10003 from slave (GL120 or GL130) ($10001 = 2711_H$).

◀EXAMPLE▶

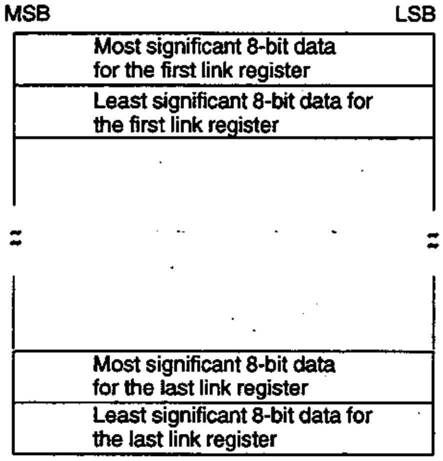
Query Message

Function code		15_H	
Starting No.	High byte	27_H	
	Low byte	10_H	
No. of registers	High byte	00_H	} 1 to 125
	Low byte	03_H	

Note Starting No.

First link register number: R00001

Data Read Order and Meaning



MSB LSB

Most significant 8-bit data for the first link register

Least significant 8-bit data for the first link register

Most significant 8-bit data for the last link register

Least significant 8-bit data for the last link register

Response Message (Normal Operation)

Function code		15_H	
No. of data		06_H	} No. of link registers × 2
First link register contents	High byte	12_H	
	Low byte	34_H	
Next link register contents	High byte	56_H	
	Low byte	78_H	
Next link register contents	High byte	$9A_H$	
	Low byte	BC_H	

10.6.5 Writing to a Single Link Register

1) Function

Specified data is written to a specified link register. The master splits and organizes the write data into the most significant 8 bits and the least significant 8 bits, and then sends a query message.

- Function code: $1B_H$

- Maximum No. of registers handled by a single message: 1

2) Example Message

In this example, the master writes to slave (GL120 or GL130) link register R10136 (10136 = 2798_H).

◀EXAMPLE▶

Query Message

Function code		$1B_H$	
Specified no.	High byte	27_H	
	Low byte	97_H	
Write data	High byte	03_H	Write data
	Low byte	$9E_H$	

Note Specified No.
Link register no.: R00001

Data Write Order and Meaning

MSB	LSB
Most significant 8-bit write data	
Least significant 8-bit write data	

Response Message (Normal Operation)

Function code		$1B_H$
Specified no.	High byte	27_H
	Low byte	97_H
Write data	High byte	03_H
	Low byte	$9E_H$

10.6.6 Writing to Multiple Link Registers

1) Function

Specified data is written to a series of link registers starting from a specified link register number up to the number of registers specified. The master splits the write data into the most significant 8 bits and the least significant 8 bits, and then organizes the write data in numerical order before sending a query message.

- Function code: $1F_H$

- Maximum No. of registers handled by a single message: 100

2) Example Message

In this example, data is written to link registers R10136 and R10137 at slave (GL120 or GL130) ($10136 = 2798_H$).

◀EXAMPLE▶ **Query Message**

Function code		$1F_H$	
Starting No.	High byte	27_H	
	Low byte	97_H	
No. of registers	High byte	00_H	1 to 100
	Low byte	02_H	
No. of data		04_H	No. of link registers × 2
Data written to the first link register	High byte	00_H	
	Low byte	$0A_H$	
Data written to the next link register	High byte	01_H	
	Low byte	02_H	

Note Starting No.
First link register number: R00001

Data Write Order and Meaning

MSB LSB

Most significant 8-bit write data for the first link register
Least significant 8-bit write data for the first link register
Most significant 8-bit write data for the last link register
Least significant 8-bit write data for the last link register

Response Message (Normal Operation)

Function code		$1F_H$
Starting No.	High byte	27_H
	Low byte	97_H
No. of registers	High byte	00_H
	Low byte	02_H

10.6.7 Reading Constant Register Contents

1) **Function**

The contents of a series of constant registers is read starting from a specified constant register number up to the number of registers specified. The slave splits the contents of the constant register into the most significant 8 bits and the least significant 8 bits, and then organizes the constant register data in numerical order before returning a response message.

- Function code: 13_H

- Maximum No. of registers handled by a single message: 125

2) **Example Message**

In this example, the contents are read for a total of three constant registers numbered 700108 to 700110 from slave (GL120 or GL130) (108 = $6C_H$).

◀EXAMPLE▶

Query Message

Function code		13_H	
Starting No.	High byte	00_H	
	Low byte	$6B_H$	
No. of registers	High byte	00_H	1 to 125
	Low byte	03_H	

Note Starting No.
First constant register number: 700001

Data Read Order and Meaning

MSB	LSB
Most significant 8-bit data for the first constant register	
Least significant 8-bit data for the first constant register	
⋮	⋮
Most significant 8-bit data for the last constant register	
Least significant 8-bit data for the last constant register	

Response Message (Normal Operation)

Function code		13_H	
No. of data		06_H	} No. of constant registers × 2
First constant register contents	High byte	02_H	
	Low byte	$2B_H$	
Next constant register contents	High byte	00_H	
	Low byte	00_H	
Next constant register contents	High byte	00_H	
	Low byte	63_H	

10.6.8 Writing to a Single Constant Register

1) Function

Specified data is written to a specified constant register. The master splits and organizes the write data into the most significant 8 bits and the least significant 8 bits, and then sends a query message.

- Function code:$1A_H$

- Maximum No. of registers handled by a single message: 1

2) Example Message

In this example, the master writes to slave (GL120 or GL130) constant register 700136 (136 = 88_H).

◀EXAMPLE▶ **Query Message**

Function code		$1A_H$	
Specified no.	High byte	00_H	
	Low byte	87_H	
Write data	High byte	03_H	} Write data
	Low byte	$9E_H$	

Note Specified No.
Constant register no.: 700001

Data Write Order and Meaning

MSB LSB

Most significant 8-bit write data
Least significant 8-bit write data

Response Message (Normal Operation)

Function code		1A$_H$
Specified no.	High byte	00$_H$
	Low byte	87$_H$
Write data	High byte	03$_H$
	Low byte	9E$_H$

10.6.9 Writing to Multiple Constant Registers

1) **Function**

Specified data is written to a series of constant registers starting from a specified constant register number up to the number of registers specified. The master splits the write data into the most significant 8 bits and the least significant 8 bits, and then organizes the write data in numerical order before sending a query message.

- Function code: 1E$_H$

- Maximum No. of registers handled by a single message: 100

2) **Example Message**

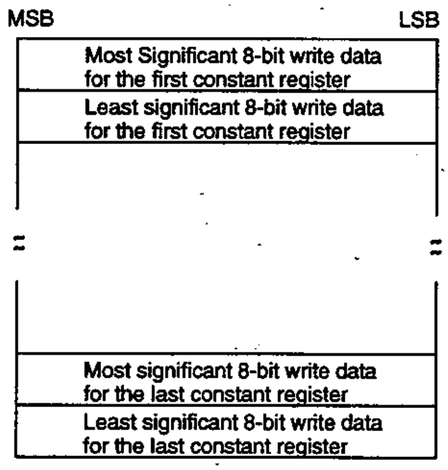
In this example, data is written to constant registers 700136 and 700137 at slave (GL120 or GL130) (136 = 88$_H$).

◀EXAMPLE▶ **Query Message**

Function code		1E$_H$	
Starting No.	High byte	00$_H$	
	Low byte	87$_H$	
No. of registers	High byte	00$_H$	1 to 100
	Low byte	02$_H$	
No. of data		04$_H$	No. of constant registers × 2
Data written to the first constant register	High byte	00$_H$	
	Low byte	0A$_H$	
Data written to the next constant register	High byte	01$_H$	
	Low byte	02$_H$	

Note Starting No.
First constant register number: 700001

Data Write Order and Meaning



Most Significant 8-bit write data for the first constant register

Least significant 8-bit write data for the first constant register

Most significant 8-bit write data for the last constant register

Least significant 8-bit write data for the last constant register

Response Message (Normal Operation)

Function code		$1E_H$
Starting No.	High byte	00_H
	Low byte	87_H
No. of registers	High byte	00_H
	Low byte	02_H

10.7 Special Functions

This section describes the loopback test, status, reference and other special functions.

10.7.1 Loopback Test 10-31
10.7.2 Reading Event Counter 10-32
10.7.3 Reading Communications Status 10-32
10.7.4 Reading Equipment Status 10-34
10.7.5 Reading the FIFO Register Contents 10-37
10.7.6 Reading Special Reference Status 10-38
10.7.7 Modifying Special Reference Status 10-39

10.7.1 Loopback Test

1) Function

The slave returns the query message unaltered as a response message. The test is used for a signal transmission check between the master and slaves.

- Function code: 08_H

2) Example Message

In this example, a loopback test is conducted with slave.

◀EXAMPLE▶ **Query Message**

Function code		08_H
Test code	High byte	00_H
	Low byte	00_H
Data	High byte	$A5_H$
	Low byte	37_H

} Any data

Note Test Code
The test code must be set to 00_H.

Response Message

Function code		08_H
Test code	High byte	00_H
	Low byte	00_H
Data	High byte	$A5_H$
	Low byte	37_H

10.7.2 Reading Event Counter

1) Function

The event counter of the slave is read. The event counter is a counter that adds +1 each time a query message is successfully executed. A +1 is added with all function codes except this one ($0B_H$). The slave splits the contents of the event counter into the most significant 8 bits and the least significant 8 bits, and then organizes the contents in numerical order before sending a response message.

- Function code: $0B_H$

2) Example Message

In this example, the event counter is read from slave.

◀EXAMPLE▶ **Query Message**

Function code	$0B_H$

Response Message

Function code		$0B_H$
Status	High byte	00_H
	Low byte	00_H
Event counter contents	High byte	00_H
	Low byte	03_H

Note Status
$FFFF_H$: Event executing
0000_H: Event completed

10.7.3 Reading Communications Status

1) Function

The event counter, message counter and event log contents of the slave are read.

- Function code: $0C_H$

The query message and response message configurations are given below:

Query Message

Function code	} $0C_H$

Response Message

Function code		$0C_H$
No. of data		46_H
Status	High byte	
	Low byte	
Event counter contents	High byte	
	Low byte	
Message counter contents	High byte	
	Low byte	
Event log (64th byte)		
Event log (63rd byte)		
• • •		
Event log (1st byte)		

Note Status
$FFFF_H$: Event executing
0000_H: Event completed

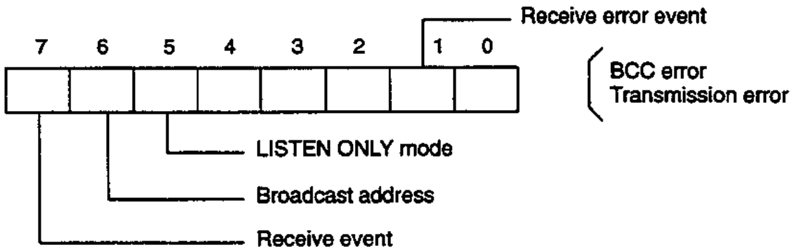
2) The event counter is a counter that adds +1 each time a query message is successfully executed.

3) The message counter is a counter that adds +1 when a query message is properly received. However, a +1 is also added when a BCC or a transmission error is generated in a message received at the CPU Module RS-232C port or the MEMOBUS PLUS port.

4) An event log is a table that uses a byte structure (8 bits) to store the status of past communications.

Details of the Contents

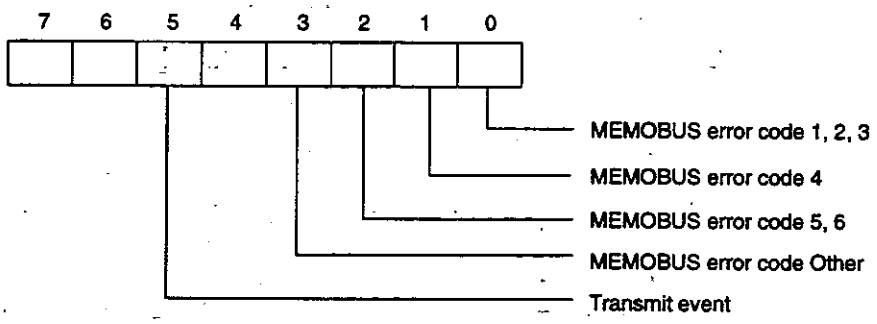
a) **Receive Events**



With normal reception, we get 80_H.
With error reception, we get 82_H.

Bits 5 and 6 are set depending on the mode.

b) **Transmit Events**



Bits 0 to 3 are set for a MEMOBUS error response.

5) The slave splits the contents into the most significant 8 bits and the least significant 8 bits, and then organizes the data in numerical order before returning a response message.

10.7.4 Reading Equipment Status

1) **Function**

The ID, LED status, memory capacity, number of segments, equipment status, and stop status are read.

- Function code: 11_H

The query message and response message configurations are shown below:

Query Message

Function code	11_H

Response Message

Function code		11_H
No. of data		09_H
ID contents		10_H (GL120,GL130)
LED status		FFH: ON (CPU RUN) 00H: OFF (CPU STOP)
Memory capacity 1		
Memory capacity 2		
No. of segments		
Equipment status contents	High byte	
	Low byte	
Stop status contents	High byte	
	Low byte	

2) The individual responses during normal operation are given below:

- ID is the type of slave.
- LED status is the status of the RUN indicator of the slave.
- Memory capacity 1 is the total capacity of user and system memory.
- Memory capacity 2 is the status memory capacity.
- Number of segments is the number of segments allotted to the ladder program.
- Machine and equipment status are various types of MEMOCON status.

3) The slave splits the contents into the most significant 8 bits and the least significant 8 bits, and then organizes the data in numerical order before returning a response message.

4) Equipment Status

	MSB															LSB
Word n	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

Equipment status data contains the following information about the CPU and Power Supply Modules.

Table 10.2 Equipment Status Contents

Bit No.	Name	Meaning	
		When Bit is Set to 1	**When Bit is Set to 0**
1	High-speed scan single sweep mode	High-speed scan single sweep mode is set.	High-speed scan single sweep mode is not set.
6	Constant sweep mode	Constant sweep mode is set.	Constant sweep mode is not set.
7	Ordinary scan single sweep mode	Ordinary scan single sweep mode is set.	Ordinary scan single sweep mode is not set.
9	Power Supply Module OK	The Power Supply Module is operating normally.	The Power Supply Module is not operating normally.
10	RUN indicator	The RUN indicator is extinguished (CPU Module is stopped).	The RUN indicator is lit (CPU Module is operating).
11	Memory protect mode	Memory protect mode is turned OFF.	Memory protect mode is turned ON.
12	Abnormal battery voltage	Battery voltage is outside the normal range (battery needs to be replaced).	Battery voltage is within the normal range.

5) Stop Status

	MSB															LSB
Word n	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

Stop status data contains the cause of CPU Module shutdown.

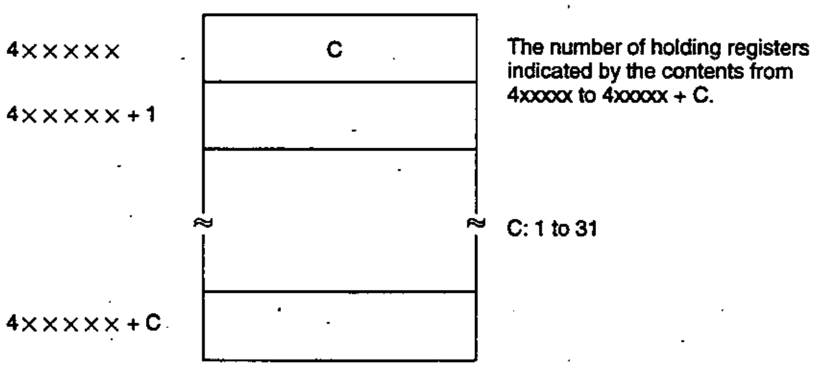
Table 10.3 Stop Status Contents

Bit No.	Name	When Bit is Set to 1
1	Controller stop	The CPU Module has been stopped by the Programming Panel.
2	Allocated memory error	Allocated memory contains abnormal data.
3	Load request	Loading of the system configuration table is requested.
4	Segment scheduler error	The segment scheduler table contains an error.
7	Power-down checksum error	• The contents of the system configuration memory at power-down do not match the contents of the system configuration memory at power-up. • The calendar circuit is faulty. • Hardware errors have been detected.
8	EOL error	Data in the last address in the ladder program memory contains an error.
9	Watchdog timer error	Ordinary scan time has exceeded 240 ms.
12	Remote I/O driver error	The remote I/O driver module is faulty.
13	Node type error	The ladder program memory contains abnormal data.
14	Ladder program checksum error	Ladder program memory data has been modified by means other than write processing from the Programming Panel.
16	System configuration error	The system configuration memory contains abnormal data.

10.7.5 Reading the FIFO Register Contents

1) Function

The contents of holding registers used as an FIFO register is read. The status of a series of FIFO registers is read starting from a specified holding register number up to the number stored in the counter for the holding register.


4×××××
4××××× + 1
4××××× + C
C
The number of holding registers indicated by the contents from 4xxxxx to 4xxxxx + C.
C: 1 to 31

The slave splits the contents of the FIFO register into the most significant 8 bits and the least significant 8 bits, and then organizes the data in numerical order before returning a response message.

- Function code: 18_H

- Maximum No. of registers handled by a single message: 31

2) Example Message

In this example, the contents are read for a total of three holding registers used as a FIFO. The holding registers are 400108 to 400110 at slave. Two FIFO registers are stored in holding register 400108 (108 = $6C_H$).

◀EXAMPLE▶

Query Message

Function code		18_H
Starting No.	High byte	00_H
	Low byte	$6B_H$

Note Starting No.
First holding register number: 400001

Data Read Order and Meaning

MSB LSB

Most significant 8-bit data for the first holding register

Least significant 8-bit data for the first holding register

≈ ≈

Most significant 8-bit data for the last holding register

Least significant 8-bit data for the last holding register

Response Message

Function code		18_H	
No. of data		06_H	} No. of holding registers × 2
First holding register contents	High byte	00_H	
	Low byte	02_H	
Next holding register contents	High byte	00_H	
	Low byte	00_H	
Next holding register contents	High byte	00_H	
	Low byte	63_H	

10.7.6 Reading Special Reference Status

1) Function

The status of motion control-related relays and coils of slaves are read.

- Function code: 21_H.
- No. of coils/relays handled by a single message: digital 256

The query message and response message configurations are shown below:

Query Message

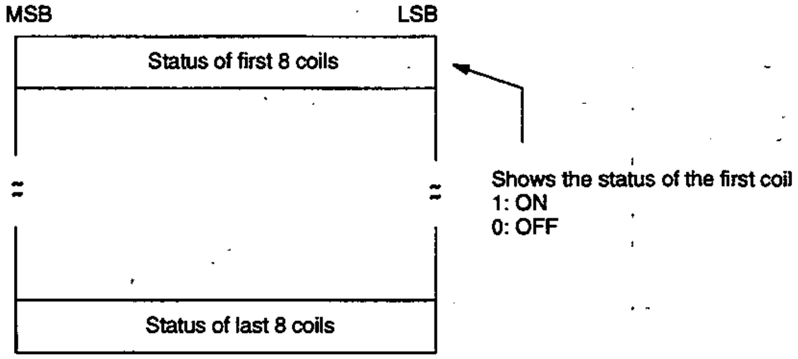
Function code		21_H
Sub function code		
Starting No.	High byte	
	Low byte	
No. of coils/relays	High byte	1 to 1,984
	Low byte	

Note (a) **Sub Function Code**
01_H: MC coil
02_H: MC control coil
03_H: MC relay
04_H: MC control relay
05_H: MC code relay

(b) **Starting No.**

First MC coil number:	Y00001
First MC control coil number:	Q00001
First MC relay number:	X00001
First MC control relay number:	P00001
First MC code relay number:	M00001

Data Read Order and Meaning



Response Message

Function code	} 21_H
Sub Function code	
No. of data	} (N+7)/8 (see note)
Status of first 8 coils	
Status of next 8 coils	
• • •	
Status of last 8 coils	

Note N is the number of coils, and (N + 7)/8 is the maximum integer value.

10.7.7 Modifying Special Reference Status

1) Function

The status of the motion control-related coils of slaves are changed.

- Function code: 22_H
- No. of coils handled by a single message: digital 256

The query message and response message configurations are shown below:

Query Message

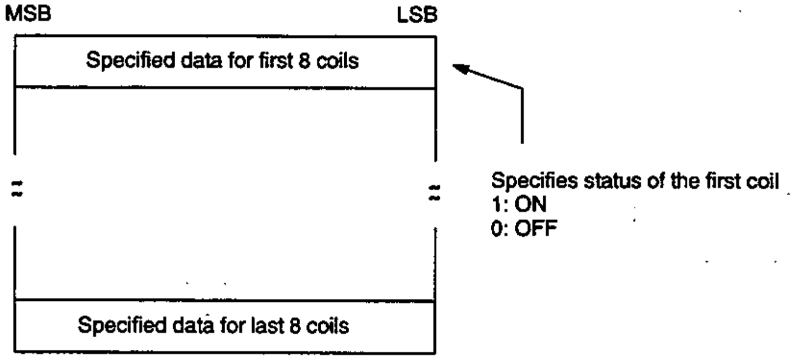
Function code		22_H
Sub Function code		
Starting No.	High byte	
	Low byte	
No. of coils	High byte	1 to 800
	Low byte	
No. of data		(N+7)/8 (see note a)
Specified data for first 8 coils		
Specified data for next 8 coils		
● ● ●		
Specified data for last 8 coils		

Note (a) N is the number of coils, and (N+7)/8 is the maximum integer value.

(b) **Sub Function Code**
01_H: MC coil
02_H: MC control coil

(c) **Starting No.**
First MC coil number: Y00001
First MC control coil number: Q00001

Specified Data Order and Meaning


MSB
LSB
Specified data for first 8 coils
Specified data for last 8 coils

Specifies status of the first coil
1: ON
0: OFF

Response Message

Function code		} 22_H
Sub function code		
Starting No.	High byte	
	Low byte	
No. of coils	High byte	
	Low byte	

Note The link coil that you wish to modify should always be set in the Disable mode. The instruction can be executed in the Enable mode, but be very careful because the relay (coil) status may be overwritten despite the internal logic decoding results.

MEMOBUS PLUS SA85 NETWORK ADAPTER
USER'S MANUAL

TOKYO OFFICE
New Pier Takeshiba South Tower, 1-16-1, Kaigan, Minatoku, Tokyo 105-6891 Japan
Phone 81-3-5402-4511 Fax 81-3-5402-4580

YASKAWA ELECTRIC AMERICA, INC.
2121 Norman Drive South, Waukegan, IL 60085, U.S.A.
Phone 1-847-887-7000 Fax 1-847-887-7370

MOTOMAN INC. HEADQUARTERS
805 Liberty Lane West Carrollton, OH 45449, U.S.A.
Phone 1-937-847-6200 Fax 1-937-847-6277

YASKAWA ELÉTRICO DO BRASIL COMÉRCIO LTDA.
Avenida Fagundes Filho, 620 Bairro Saude-Sao Pãulo-SP, Brazil CEP: 04304-000
Phone 55-11-5071-2552 Fax 55-11-5581-8795

YASKAWA ELECTRIC EUROPE GmbH
Am Kronberger Hang 2, 65824 Schwalbach, Germany
Phone 49-6196-569-300 Fax 49-6196-888-301

Motoman Robotics Europe AB
Box 504 S38525 Torsås, Sweden
Phone 46-486-48800 Fax 46-486-41410

Motoman Robotec GmbH
Kammerfeldstraβe1, 85391 Allershausen, Germany
Phone 49-8166-900 Fax 49-8166-9039

YASKAWA ELECTRIC UK LTD.
1 Hunt Hill Orchardton Woods Cumbernauld, G68 9LF, United Kingdom
Phone 44-1236-735000 Fax 44-1236-458182

YASKAWA ELECTRIC KOREA CORPORATION
Kfpa Bldg #1201, 35-4 Youido-dong, Yeongdungpo-Ku, Seoul 150-010, Korea
Phone 82-2-784-7844 Fax 82-2-784-8495

YASKAWA ELECTRIC (SINGAPORE) PTE. LTD.
151 Lorong Chuan, #04-01, New Tech Park Singapore 556741, Singapore
Phone 65-282-3003 Fax 65-289-3003

YASKAWA ELECTRIC (SHANGHAI) CO., LTD.
4F No.18 Aona Road, Waigaoqiao Free Trade Zone, Pudong New Area, Shanghai 200131, China
Phone 86-21-5866-3470 Fax 86-21-5866-3869

YATEC ENGINEERING CORPORATION
Shen Hsiang Tang Sung Chiang Building 10F 146 Sung Chiang Road, Taipei, Taiwan
Phone 886-2-2563-0010 Fax 886-2-2567-4677

YASKAWA ELECTRIC (HK) COMPANY LIMITED
Rm. 2909-10, Hong Kong Plaza, 186-191 Connaught Road West, Hong Kong
Phone 852-2803-2385 Fax 852-2547-5773

BEIJING OFFICE
Room No. 301 Office Building of Beijing International Club, 21
Jianguomenwai Avenue, Beijing 100020, China
Phone 86-10-6532-1850 Fax 86-10-6532-1851

TAIPEI OFFICE
Shen Hsiang Tang Sung Chiang Building 10F 146 Sung Chiang Road, Taipei, Taiwan
Phone 886-2-2563-0010 Fax 886-2-2567-4677

SHANGHAI YASKAWA-TONGJI M & E CO., LTD.
27 Hui He Road Shanghai China 200437
Phone 86-21-6531-4242 Fax 86-21-6553-6060

BEIJING YASKAWA BEIKE AUTOMATION ENGINEERING CO., LTD.
30 Xue Yuan Road, Haidian, Beijing P.R. China Post Code: 100083
Phone 86-10-6233-2782 Fax 86-10-6232-1536

SHOUGANG MOTOMAN ROBOT CO., LTD.
7, Yongchang-North Street, Beijing Economic Technological Investment & Development Area,
Beijing 100076, P.R. China
Phone 86-10-6788-0551 Fax 86-10-6788-2878

YASKAWA ELECTRIC CORPORATION

YASKAWA

MANUAL NO. SIEZ-C825-70.6

Specifications are subject to change without notice
for ongoing product modifications and improvements.

© Printed in Japan November 2000 97-3
99-7③
95-81192, 95-C82-031