

C200PC-ISA01-E
C200PC-ISA_2-DRM-E
C200PC-ISA_2-SRM-E
C200PC-EXP01

SYSMAC Board

OPERATION MANUAL

OMRON

C200PC-ISA01-E

C200PC-ISA□2-DRM-E

C200PC-ISA□2-SRM-E

C200PC-EXP01

SYSMAC Board




Operation Manual

Produced December 1998

Notice:

OMRON products are manufactured for use according to proper procedures by a qualified operator and only for the purposes described in this manual.

The following conventions are used to indicate and classify precautions in this manual. Always heed the information provided with them. Failure to heed precautions can result in injury to people or damage to property.

-  **DANGER** Indicates an imminently hazardous situation which, if not avoided, will result in death or serious injury.
-  **WARNING** Indicates a potentially hazardous situation which, if not avoided, could result in death or serious injury.
-  **Caution** Indicates a potentially hazardous situation which, if not avoided, may result in minor or moderate injury, or property damage.

OMRON Product References

All OMRON products are capitalized in this manual. The word “Unit” is also capitalized when it refers to an OMRON product, regardless of whether or not it appears in the proper name of the product.

The abbreviation “Ch,” which appears in some displays and on some OMRON products, often means “word” and is abbreviated “Wd” in documentation in this sense.

In this manual “PLC” is used as the abbreviation for Programmable Controller.

Visual Aids

The following headings appear in the left column of the manual to help you locate different types of information.

Note Indicates information of particular interest for efficient and convenient operation of the product.

1, 2, 3... 1. Indicates lists of one sort or another, such as procedures, checklists, etc.

Trademarks and Copyrights

Microsoft, MS, MS-DOS, Windows, Windows 95, and Microsoft C/C++ are registered trademarks of the Microsoft Corporation.

IBM is a registered trademark of the IBM Company.

DeviceNet is a registered trademark of the Open DeviceNet Vendor Association, Inc.

The copyright to software for the CompoBus/D Master functionality provided on the SYSMAC Board belongs to S–S Technologies Inc.

© OMRON, 1998

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 OMRON.

No patent liability is assumed with respect to the use of the information contained herein. Moreover, because OMRON 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, OMRON 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.

TABLE OF CONTENTS

PRECAUTIONS	xi
1 Intended Audience	xii
2 General Precautions	xii
3 Safety Precautions	xii
4 Operating Environment Precautions	xii
5 Application Precautions	xiii
SECTION 1	
Introduction	1
1-1 Features	2
1-2 List of Available Models	4
1-3 System Configuration	5
1-4 Package Contents	10
1-5 Programmable Controller Functions	11
1-6 CompoBus/D Master Function	15
1-7 CompoBus/S Master Function	19
1-8 Computer Board Functions	22
1-9 Differences from Standard PLCs	24
1-10 Specifications	25
1-11 Setup Procedures	30
SECTION 2	
Hardware Setup and Installation	33
2-1 Setup Confirmations	34
2-2 Components	35
2-3 Board Switch Settings	44
2-4 Support Board Installation	49
2-5 External Connections	52
2-6 Required Files	59
SECTION 3	
SYSMAC Board Communications Library	61
3-1 Using the PLC Communications Library	62
3-2 Library Functions	64
pcinit() CHANGE SETTINGS	64
pcopen() PLC OPEN	65
pcclose() PLC CLOSE	66
pcmode() PLC MODE	66
pcstat() PLC STATUS	68
pcread() PLC READ	76
pcwrite() PLC WRITE	82
SECTION 4	
Read/Write Performance	89
4-1 Measurement Program	90
4-2 Measurement Results	92
SECTION 5	
Troubleshooting	101
Appendices	
A Replacing the Battery	103
B Handling Power Interruptions	105
Index	107
Revision History	109

About this Manual:

This manual describes the installation and operation of SYSMAC Board PLCs (Programmable Controllers). The SYSMAC Boards are ISA bus-compatible boards that provide functionality equivalent to a SYSMAC C200HX/HG/HE PLC.

The following manuals may also be required to install and run a PLC System based on the SYSMAC Board.

Manual	Contents	Cat. No. (suffixes omitted)
SYSMAC Board Operation Manual	Mainly describes information required to connect a SYSMAC Board to an ISA bus. Refer to the <i>C200HX/HG/HE PLC Installation Guide</i> and <i>Operation Manual</i> for descriptions of PLC functionality and to the <i>CompoBus/D (DeviceNet) Operation Manual</i> for descriptions of the CompoBus/D Network functionality.	V201 (this manual)
C200HX/HG/HE (-E/-ZE) PLC Installation Guide	Describes handling procedures for C200HX/HG/HE hardware, including installation and wiring. This manual is required for the SYSMAC Board when connecting an Expansion I/O Rack.	W302
C200HX/HG/HE-ZE PLC Operation Manual	Describes programming for C200HX/HG/HE PLCs. Use this manual as reference with programming the SYSMAC Boards.	W322
CompoBus/D (DeviceNet) Operation Manual	Describes CompoBus/D specifications and functions, including Slaves and overall Network operation. This manual is required to use the CompoBus/D functionality of the SYSMAC Boards.	W267
CompoBus/D Configurator Operation Manual	Required to make special settings for a CompoBus/D Network, such as user-set allocations for remote I/O.	W328
CompoBus/S Operation Manual	Describes CompoBus/S specifications and functions, including Slaves and overall Network operation. This manual is required to use the CompoBus/S functionality of the SYSMAC Boards.	W266

Please read this manual and all other related manuals carefully and be sure you understand the information provided before attempting to install and operate a SYSMAC Board. You will also need the manuals for your personal computer.

This manual includes the following sections.

Section 1 describes the functions and specifications of the SYSMAC Boards. Read this section before reading the rest of the manual.


Section 2 describes the components and functions of the SYSMAC Boards. It also describes hardware installation and settings, cable connections, etc.

Section 3 describes the PLC communications library used to control the SYSMAC Boards from the personal computer. Read this section to write applications for the SYSMAC Boards.

Section 4 provides measurement programs and actual measurement results for determining the processing time required to read and write SYSMAC Board data from the personal computer. Use this section for reference when precise timing is required.

Section 5 describes troubleshooting errors

The **Appendices** describe how to replace the battery built into the SYSMAC Boards and how to handle power interruptions.

 **WARNING** Failure to read and understand the information provided in this manual may result in personal injury or death, damage to the product, or product failure. Please read each section in its entirety and be sure you understand the information provided in the section and related sections before attempting any of the procedures or operations given.

PRECAUTIONS

This section provides general precautions for using the SYSMAC Board Programmable Controller (PLC) and related devices. **The information contained in this section is important for the safe and reliable application of the PLC. You must read this section and understand the information contained before attempting to set up or operate a PLC System.**

1 Intended Audience	xii
2 General Precautions	xii
3 Safety Precautions	xii
4 Operating Environment Precautions	xii
5 Application Precautions	xiii

1 Intended Audience

This manual is intended for the following personnel, who must also have knowledge of electrical systems (an electrical engineer or the equivalent).

- Personnel in charge of installing FA systems.
- Personnel in charge of designing FA systems.
- Personnel in charge of managing FA systems and facilities.


2 General Precautions

The user must operate the product according to the performance specifications described in the operation manuals.


Before using the product under conditions which are not described in the manual or applying the product to nuclear control systems, railroad systems, aviation systems, vehicles, combustion systems, medical equipment, amusement machines, safety equipment, and other systems, machines, and equipment that may have a serious influence on lives and property if used improperly, consult your OMRON representative.


Make sure that the ratings and performance characteristics of the product are sufficient for the systems, machines, and equipment, and be sure to provide the systems, machines, and equipment with double safety mechanisms.


This manual provides information for programming and operating OMRON PLCs. Be sure to read this manual before attempting to use the software and keep this manual close at hand for reference during operation.

 **WARNING** It is extremely important that a PLC and all PLC Units be used for the specified purpose and under the specified conditions, especially in applications that can directly or indirectly affect human life. You must consult with your OMRON representative before applying a PLC System to the abovementioned applications.

3 Safety Precautions

 **WARNING** Do not attempt to disassemble or touch the inside of the SYSMAC Board while the power is being supplied. Doing so may result in electric shock.

 **Caution** Execute online edit only after confirming that no adverse effects will be caused by extending the cycle time. Otherwise, the input signals may not be readable.


 **Caution** Confirm safety at the destination node before transferring a program to another node or changing the I/O memory area. Doing either of these without confirming safety may result in injury.

4 Operating Environment Precautions

Do not operate the system in the following places.


- Locations subject to direct sunlight.
- Locations subject to temperatures or humidity outside the range specified in the specifications.
- Locations subject to condensation as the result of severe changes in temperature.

- Locations subject to corrosive or flammable gases.
- Locations subject to dust (especially iron dust) or salts.
- Locations subject to shock or vibration.
- Locations subject to exposure to water, oil, or chemicals.
- Take appropriate and sufficient countermeasures when installing systems in the following locations.
 - Locations subject to static electricity or other forms of noise.
 - Locations subject to strong electromagnetic fields.
 - Locations subject to possible exposure to radioactivity.
 - Locations close to power supplies.


 **Caution** The operating environment of the PLC System can have a large effect on the longevity and reliability of the system. Improper operating environments can lead to malfunction, failure, and other unforeseeable problems with the PLC System. Be sure that the operating environment is within the specified conditions at installation and remains within the specified conditions during the life of the system.

5 Application Precautions

Observe the following precautions when using the SYSMAC Board.

 **WARNING** Failure to abide by the following precaution could lead to serious or possibly fatal injury. Always heed this precaution.

- Always ground the system to 100 Ω or less when installing the system to protect against electrical shock.

 **Caution** Failure to abide by the following precautions could lead to faulty operation of the SYSMAC Board or the system. Always heed these precautions.

- Fail-safe measures must be taken by the customer to ensure safety in the event of incorrect, missing, or abnormal signals caused by broken signal lines, momentary power interruptions, or other causes.
- Always use the power supply voltage specified in the operation manuals. An incorrect voltage may result in malfunction or burning.
- Install external breakers and take other safety measures against short-circuiting in external wiring. Insufficient safety measures against short-circuiting may result in burning.
- Always turn off the power supply to the SYSMAC Board before attempting any of the following. Performing any of the following with the power supply turned ON may lead to electrical shock.
 - Assembling the system.
 - Setting DIP switches or rotary switches.
 - Connecting or wiring cables.
 - Connecting or disconnecting any connectors.
- Do not attempt to disassemble, repair, or modify the SYSMAC Board. Any attempt to do so may result in malfunction, fire, or electric shock.
- Be sure that all the mounting screws, terminal screws, and cable connector screws are tightened to the torque specified in the relevant manuals. Incorrect tightening torque may result in malfunction.
- Double-check all the wiring before turning ON the power supply. Incorrect wiring may result in burning.

- Be sure that the connectors, expansion cables, and other items with locking devices are properly locked into place. Improper locking may result in malfunction.
- Check the user program for proper execution before actually running it on the system. Not checking the program may result in an unexpected operation.
- Confirm that no adverse effect will occur in the system before changing the operating mode of the SYSMAC Board. Not doing so may result in an unexpected operation.
- Confirm that no adverse effect will occur in the system before force-setting/force-resetting any bit in memory. Not doing so may result in an unexpected operation.
- Confirm that no adverse effect will occur in the system before changing the present value. Not doing so may result in an unexpected operation.
- Confirm that no adverse effect will occur in the system before changing the set value. Not doing so may result in an unexpected operation.
- Whenever a SYSMAC Board is replaced, be sure to transfer all required programs and data to the new SYSMAC Board before starting operation.
- Do not pull on or bend the communications cables beyond their natural limit. Doing either of these may break the cables.
- Do not place objects on top of the communications cables. Doing so may break the cables.
- The SYSMAC Board retains HR, AR, CNT, DM, EM, and clock data even when the power is OFF. Data may not be properly retained unless the power is turned OFF after switching to PROGRAM mode. Therefore, be sure to change the SYSMAC Board mode to PROGRAM mode using the “pcmode” (operating mode change) function.
- Take appropriate measures to ensure that the specified power with the rated voltage and frequency is supplied. Be particularly careful in places where the power supply is unstable. An incorrect power supply may result in malfunction.

SECTION 1

Introduction

This section describes the features, specifications, and configuration of the SYSMAC Board.

1-1	Features	2
1-2	List of Available Models	4
1-2-1	SYSMAC Boards	4
1-2-2	Expansion Option Board	4
1-3	System Configuration	5
1-4	Package Contents	10
1-5	Programmable Controller Functions	11
1-6	CompoBus/D Master Function	15
1-7	CompoBus/S Master Function	19
1-8	Computer Board Functions	22
1-9	Differences from Standard PLCs	24
1-10	Specifications	25
1-10-1	General Specifications	25
1-10-2	SYSMAC Board CPU Specifications	25
1-10-3	CompoBus/D Communications Specifications	27
1-10-4	CompoBus/S Communications Specifications	28
1-10-5	Board Specifications	28
1-10-6	RS-232C Port Cable Specifications	28
1-10-7	Expansion Option Board	29
1-11	Setup Procedures	30

1-1 Features

A Programmable Controller within a Personal Computer

A SYSMAC Board is an ISA bus-compatible board that provides functionality equivalent to a SYSMAC C200HG or C200HX PLC. It is installed as an expansion board in an IBM PC/AT or compatible computer. The features of the SYSMAC Board are described next.

Programmed Like a C200HX/HG PLC

The SYSMAC Board is a fully functional PLC that operates within a personal computer. It is functionally equivalent to the SYSMAC C200HG-CPU43-E or C200HX-CPU64-E PLC. The equivalent PLC depends on the type of SYSMAC Board used.

Control with C Language Library

A Programming Device can be connected to the RS-232C port on the Board to program it the same as a C200HX or C200HG PLC. Programming Devices can be connected to either the RS-232C port or the peripheral port when an Expansion Option Board is used.

A C language library for the SYSMAC Board can be used in user applications running on the computer to read/write Board memory, monitor Board status, and control Board operation. In this sense, the computer can serve as a host computer for the SYSMAC Board.

The Microsoft C/C++ Ver. 7.0 C library is used, and small, medium, large, huge, and compact memory models are supported. Windows Drivers, called the FinsGateway for Windows 95 and FinsGateway for Windows NT, are sold separately.

Connect Up To Three Expansion I/O Racks

There are no I/O points provided on the Board itself, but it can be connected to up to three C200HX or C200HG Expansion I/O Racks to enable I/O through mounted I/O Units.

Standard 16- or 32-K Word EEPROM

The Board is provided with 16K or 32K words of EEPROM built into it. This EEPROM can be used exactly like the C200HW-ME16K and C200HW-ME32K Memory Cassettes mounted to C200HX or C200HG PLCs. (The EEPROM cannot be removed from the Board.)

Serial Communications with Expansion Option Boards

If a C200PC-EXP01 Expansion Option Board is used, the same RS-232C connector and peripheral connector as the C200HX or C200HG PLCs can be used. The use of an Expansion Option Board enables the SYSMAC Board to be connected to other PLCs and Programming Console, and allows a rich variety of communications, such as Host Link, RS-232C no-protocol, 1:1 PLC Link, 1:1 NT Link, and 1:n NT Link communications, just as for C200HX or C200HG PLCs. When an Expansion Option Board is used, however, the RS-232C port on the SYSMAC Board cannot be used.

Use C200HX/HG/HE Special I/O Units

The following Special I/O Units for use with C200HX/HG/HE PLCs can be mounted to the Expansion I/O Racks connected to the Board.

Unit	SYSMAC Board	
	C200PC-ISA01-E C200PC-ISA02-DRM-E C200PC-ISA02-SRM-E	C200PC-ISA12-DRM-E C200PC-ISA12-SRM-E
Host Link Units	2 Units max.	
PLC Link Units	2 Units max.	
SYSMAC BUS Remote I/O Master Units (Note 1)	2 Units max.	
Group-2 B7A Interface Units	10 Units max. (see note 2)	
Group-2 High-density I/O Units	10 Units max. (see note 2)	16 Units max. (see note 2)
Other Special I/O Units CompoBus/D Master Units (Note 3) CompoBus/S Master Units (Note 4) High-speed Counter Units Position Control Units ASCII Units Analog I/O Units ID Sensor Units Fuzzy Logic Units Temperature Control Units Cam Positioner Units Temperature Sensor Units Voice Units	10 Units max.	16 Units max.

- Note**
1. If the SYSMAC Board uses fixed allocations for CompoBus/D Master function or the CompoBus/D Master Unit, the words allocated in memory will overlap with the area used for remote I/O, so the SYSMAC BUS Remote I/O Master Unit cannot be used at the same time.
 2. The Group-2 B7A Interface Units and Group-2 High-density I/O Units are available in 32-point or 64-point models. Since 64-point models are allocated the words for two 32-point models, the maximum number of 64-point models that can be mounted is half the number of 32-point models.
 3. The maximum number of CompoBus/D Master Units that can be mounted includes the SYSMAC Board's CompoBus/D Master function.
If the SYSMAC Board uses fixed allocations for the CompoBus/D Master function and CompoBus/D Master Unit, the words allocated in memory will overlap with the area used for remote I/O. Therefore, only one CompoBus/D Master Unit can be mounted (refer to *1-6 CompoBus/D Master Unit Function*), and SYSMAC BUS Remote I/O Master Units cannot be used at the same time. The CompoBus/D Configurator must be used to allocated other words if the fixed allocations are not used.
 4. The maximum number of CompoBus/S Master Units that can be mounted includes the SYSMAC Board's CompoBus/S Master function.
If the SYSMAC Board's CompoBus/S Master function and CompoBus/S Master Unit are used when the number of Units that can be connected is set to IN 0 to 15/OUT 0 to 15, twice the number of words will be used in the Special I/O Unit Area, so the maximum number of CompoBus/S Masters that can be mounted will be halved. (Refer to *1-7 CompoBus/S Master Unit Function*.)

Note The following Units cannot be used with a SYSMAC Board.

- SYSMAC NET Link Units
- SYSMAC LINK Units
- PC Card Units
- Controller Link Units
- Any other Units that must be mounted to a CPU Backplane

Automatic Data Exchange with CompoBus/D and CompoBus/S Slaves

The SYSMAC Board is available in models that are equipped with the same functionality as the C200HW-DRM21-EV1 CompoBus/D Master Unit and the C200HW-SRM21 CompoBus/S Master Unit. The Board can, therefore, be connected to CompoBus/D or CompoBus/S Slaves to control distributed remote I/O.

The status of I/O to and from CompoBus/D or CompoBus/S Slaves is continuously updated in the memory of the SYSMAC Board. Slave I/O can, therefore, be read or written merely by reading/writing data from/to the Board's memory.

Note CompoBus/D is OMRON's implementation of the international DeviceNet specification. The DeviceNet is a floor-level factory automation network. A single cable can be used to connect remote Slaves to automatically exchange I/O data or messages. Refer to the *CompoBus/D (DeviceNet) Operation Manual (W267)* for details. Many manufacturers market DeviceNet masters and slaves.

CompoBus/S is OMRON's communications system for controlling I/O by connecting remote slave devices, such as I/O devices, in various locations in the system to the Master using a single cable. Refer to the *CompoBus/S (DeviceNet) Operation Manual (W266)* for details.

1-2 List of Available Models

The following table provides a list of Boards that are currently available.

1-2-1 SYSMAC Boards

SYSMAC Board	Equivalent CPU Unit	Max. No. of Expansion Racks	Built-in EEPROM	CompoBus/D communications	CompoBus/S communications	Expansion Option Board
C200PC-ISA01-E	C200HG-CPU43-E	2	16 K words	No	No	No
C200PC-ISA02-DRM-E				Yes	No	Yes
C200PC-ISA02-SRM-E				No	Yes	Yes
C200PC-ISA12-DRM-E	C200HX-CPU64-E	3	32 K words	Yes	No	Yes
C200PC-ISA12-SRM-E				No	Yes	Yes

The CompoBus/D communications function is equivalent to that provided by a C200HW-DRM21-EV1 CompoBus/D Master Unit.

The CompoBus/S communications function is equivalent to that provided by a C200HW-SRM21 CompoBus/S Master Unit.

The other similarities and differences between Units not provided here are the same as those for C200HX-CPU64-E and C200HG-CPU43-E CPU Units. Refer to the *C200HX/HG/HE-ZE Programmable Controllers Operation Manual (W322)*.

1-2-2 Expansion Option Board

Model	Peripheral port	RS-232C port
C200PC-EXP01	Equivalent to C200HX and C200HG PLCs	Equivalent to C200HX and C200HG PLCs (female D-sub 9-pin)

The SYSMAC Board is connected to the Expansion Option Board using the Flat Cable that is provided with the Expansion Option Board.

The SYSMAC Board has a built-in RS-232C port. If an Expansion Option Board is not used, the RS-232C port can be used to connect external devices or computers running Support Software.

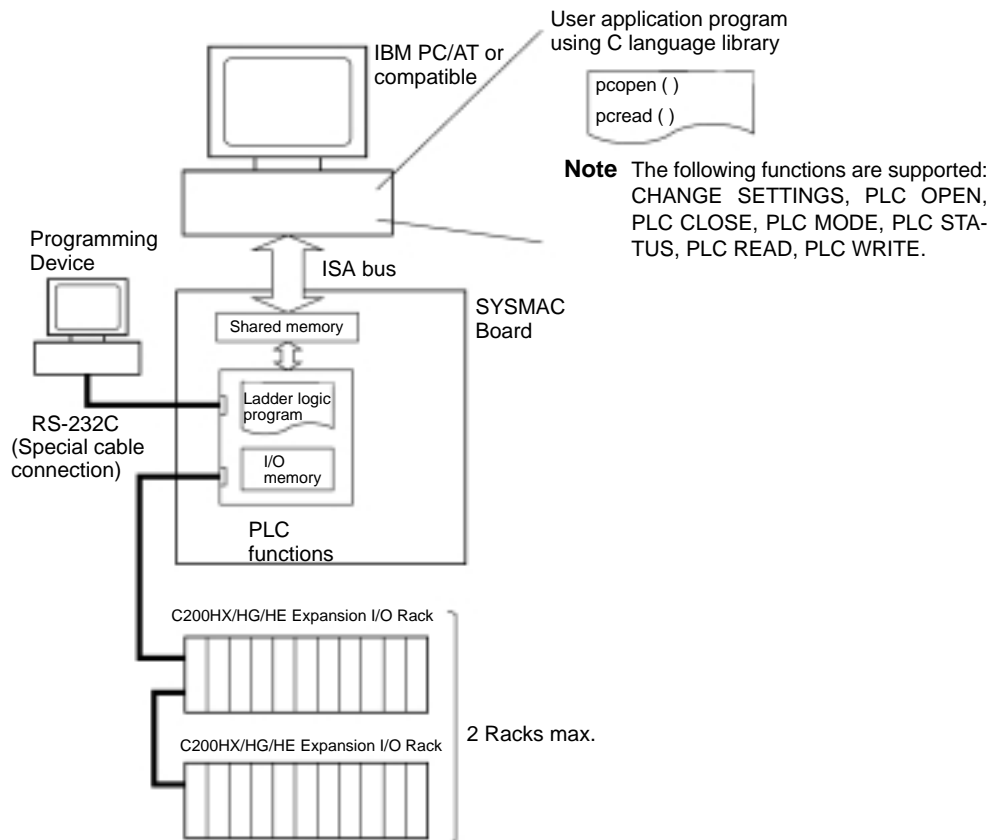
The Expansion Option Board is mounted in one slot on the ISA (XT) bus, but it does not use any resources (I/O ports, memory, or IRQ).

Note When an Expansion Option Board is connected, the SYSMAC Board's RS-232C port cannot be used.

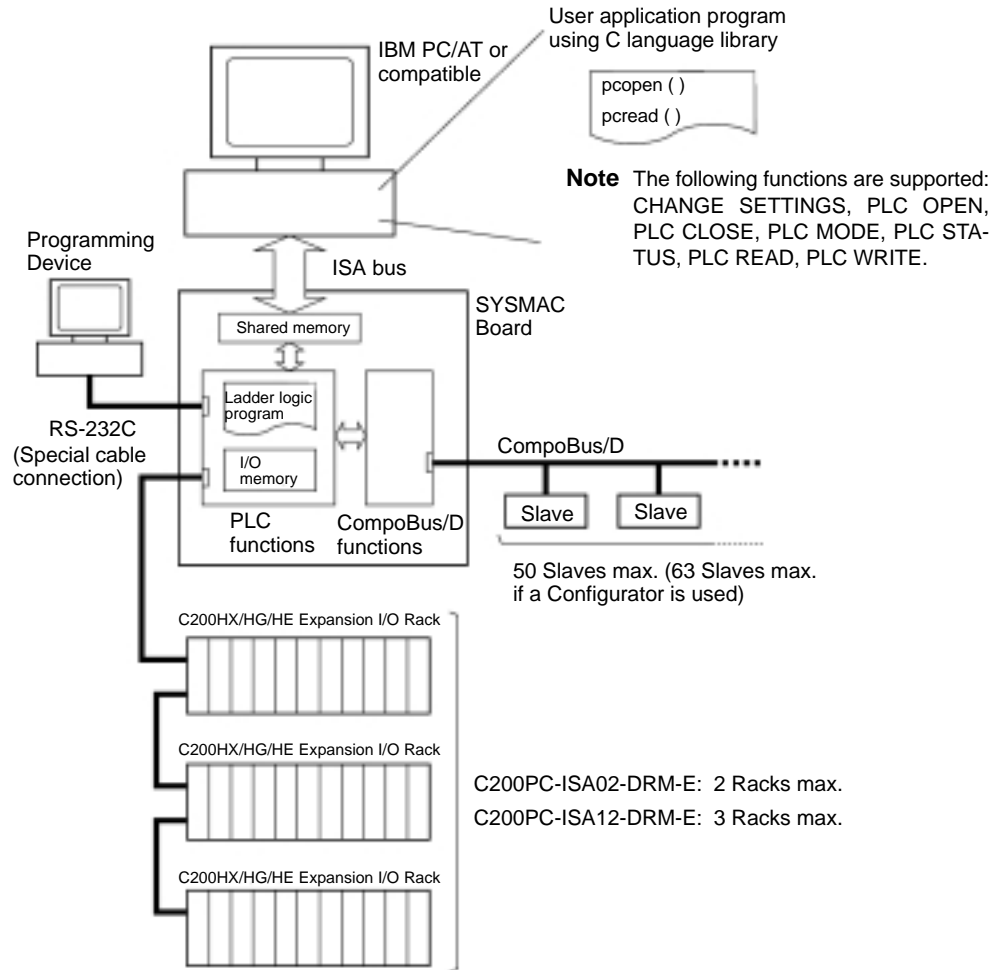
1-3 System Configuration

A system configuration for the SYSMAC Board is shown in the following illustration.

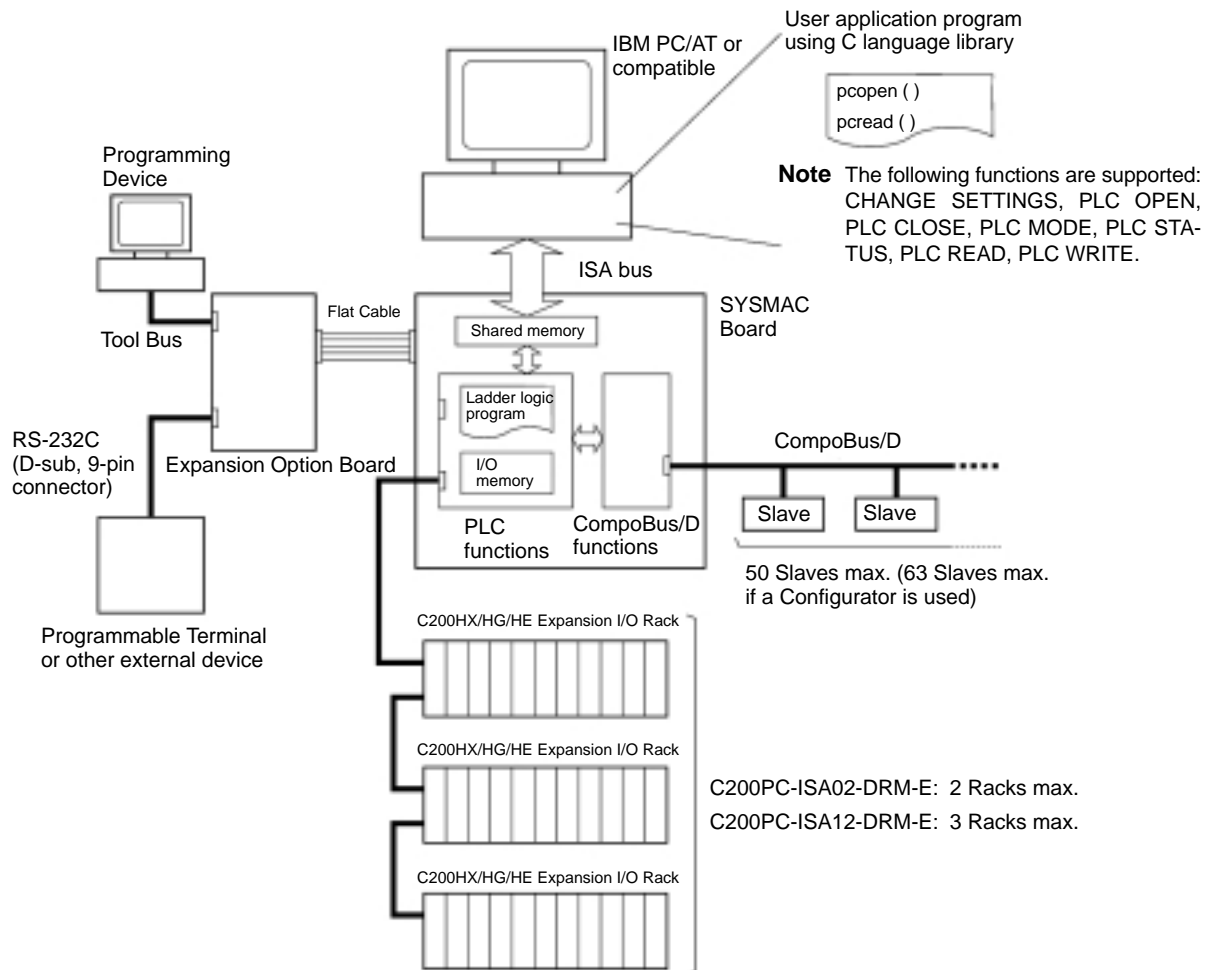
C200PC-ISA-01-E SYSMAC Board without CompoBus



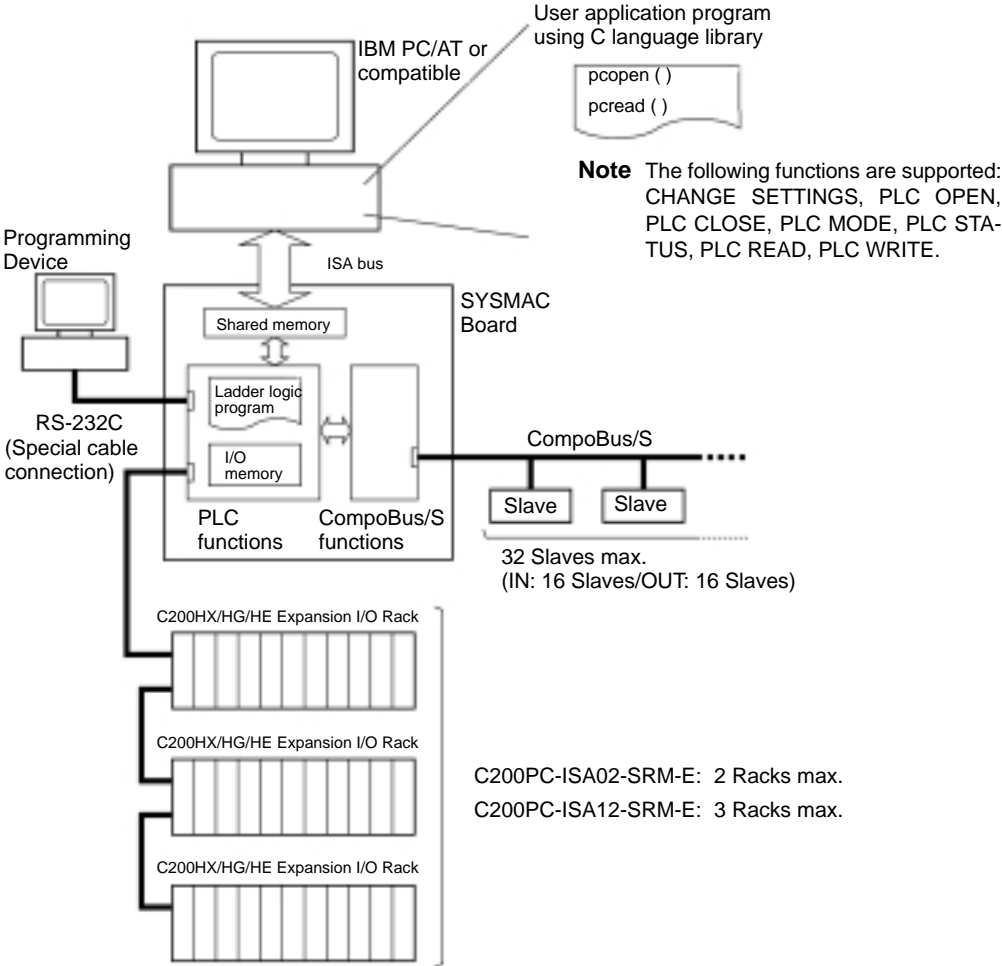
C200PC-ISA□2-DRM-E SYSMAC Board with CompoBus/D



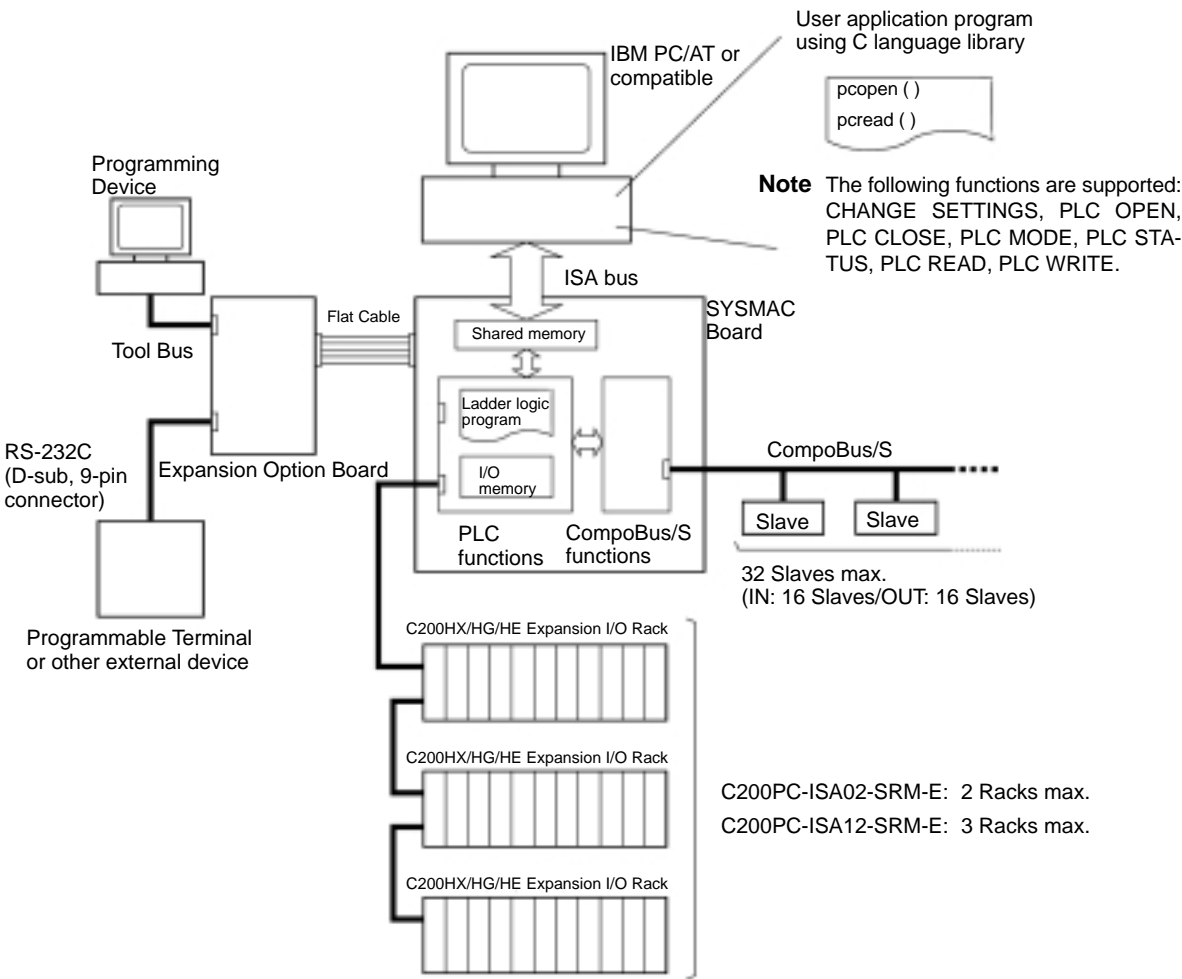
C200PC-ISA□2-DRM-E SYSMAC Board with CompoBus/D Plus C200PC-EXP01 Expansion Option Board



C200PC-ISA□2-SRM-E SYSMAC Board with CompoBus/S



C200PC-ISA□2-SRM-E SYSMAC Board with CompoBus/S Plus C200PC-EXP01 Expansion Option Board



Personal Computer
System Requirements





The following environment is required for the SYSMAC Board.

Item	Specification
Personal computer	IBM PC/AT or compatible computer
OS	Microsoft C/C++ Ver. 7.0 must be running.
Hard disk drive	2 Mbytes of space must be available.
Floppy disk drive	Must be able to read 1.44-Mbyte 3.5" floppy disk.

1-4 Package Contents



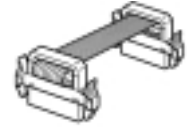

Confirm the contents of the package as soon as you open it to be sure everything is included and in proper condition.

SYSMAC Boards

	SYSMAC Board	1
	C language library on 3.5" floppy disk	1
	SYSMAC Board Operation Manual (this manual)	1
	Connector (attached to the SYSMAC Board) C200PC-ISA□2-DRM-E for CompoBus/D C200PC-ISA□2-SRM-E for CompoBus/S (Not attached to the C200PC-ISA01-E.)	1

Note The operating system for the personal computer is not included with the SYSMAC Board and must be provided by the user.

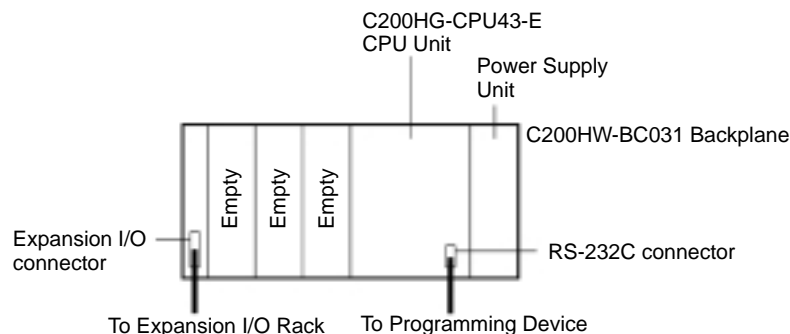
Expansion Option Boards

	Expansion Option Board	1
	D-SUB 9-pin connector (plug section)	1
	D-SUB 9-pin connector cover (cover section)	1
	Flat cable	1
	Safety Precautions	1

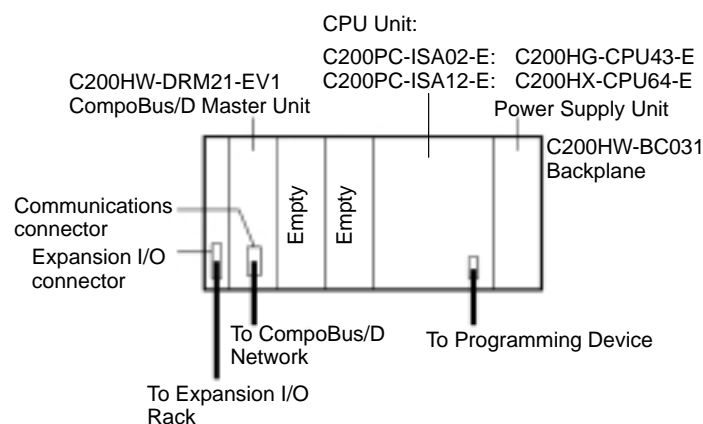
1-5 Programmable Controller Functions

The SYSMAC Boards contain the same functions as the PLC illustrated in the following diagrams. The specific functions vary with the model.

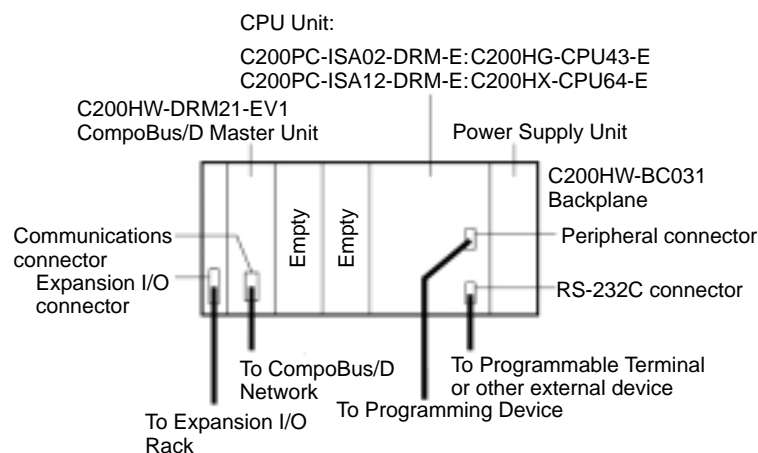
C200PC-ISA01-E SYSMAC Board without CompoBus

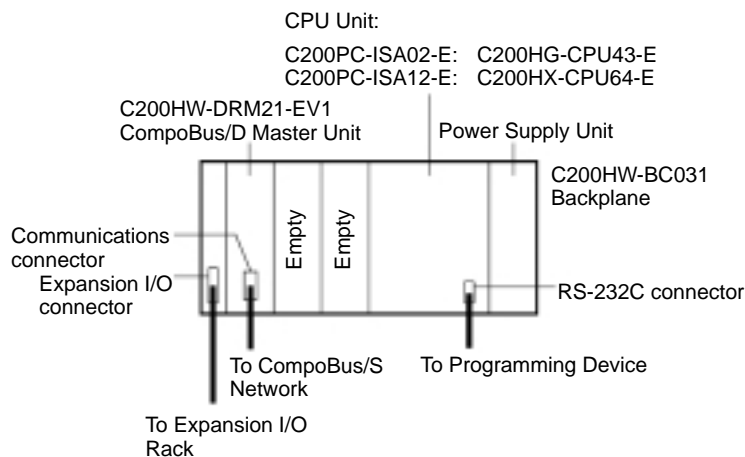
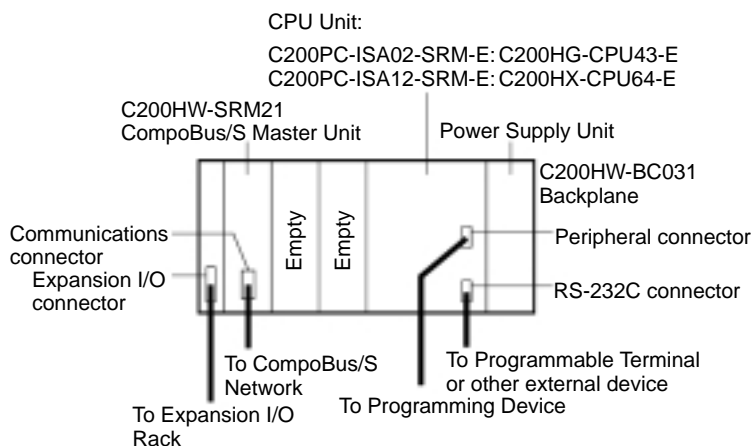


C200PC-ISA□2-DRM-E SYSMAC Board with CompoBus/D



C200PC-ISA□2-DRM-E SYSMAC Board with CompoBus/D Plus C200PC-EXP01 Expansion Option Board



C200PC-ISA□2-SRM-E SYSMAC Board with CompoBus/S

C200PC-ISA□2-SRM-E SYSMAC Board with CompoBus/S Plus C200PC-EXP01 Expansion Option Board


Note The SYSMAC Board is essentially the same as either a C200HG-CPU43-E or C200HX-CPU64-E CPU Unit, depending to the model used. This manual does not contain descriptions of the memory areas, programming instructions, Host Link commands, or other functionality covered in the manuals for C200HX/HG/HE PLCs. Refer to the manuals for C200HX/HG/HE PLCs for information not provided in this manual. Any differences between the SYSMAC Board and the CPU Unit functionality are outlined in *1-9 Differences from a C200HG or C200HX PLC*.

Expansion I/O Racks

The maximum number of C200HX/HG/HE Expansion I/O Racks that can be connected depends on the SYSMAC Board used, as shown in the following table. Refer to the *C200HX/HG/HE (-E/-ZE) Programmable Controllers Installation Guide (W302)* for details on Expansion I/O Racks.

SYSMAC Board	Maximum No. of Expansion Racks
C200PC-ISA01-E	2
C200PC-ISA02-DRM-E	
C200PC-ISA02-SRM-E	
C200PC-ISA12-DRM-E	3
C200PC-ISA12-SRM-E	

Expansion I/O Cables

Use the C200PC-CN221 (2 m) Expansion I/O Cable (sold separately) between the SYSMAC Board and the C200HX/HG/HE Expansion I/O Rack.

Mountable Units

Use the same C200H-CN□□□ Expansion I/O Cable between Expansion I/O Racks as used for the C200HX/HG/HE PLCs. Refer to the *C200HX/HG/HE (-E/-ZE) Programmable Controllers Installation Guide (W302)* for details.

All C200HX/HG/HE Power Supply, I/O, Special I/O, and Communications Units can be mounted to the C200HX/HG/HE Expansion I/O Racks connected to the SYSMAC Board **except** for those listed in the following table. Refer to the *C200HX/HG/HE (-E/-ZE) Programmable Controllers Installation Guide (W302)* for details on the Units to be mounted.

Unit	Reason
SYSMAC NET Link Unit	The SYSMAC Board corresponds to the CPU Rack, so Communications Boards cannot be used. Therefore, the Units listed here cannot be used with a SYSMAC Board.
SYSMAC LINK Unit	
PC Card Unit	
Controller Link Unit	
Other Units that must be mounted to the CPU Rack	

Note When fixed allocations are used to allocate I/O words for the CompoBus/D Master function, the words that are allocated will overlap if more than one CompoBus/D Master Unit (including the CompoBus/D Master function of the C200PC-ISA□2-DRM-E SYSMAC Board) cannot be used at the same time. SYSMAC BUS Remote I/O Master Units are also allocated words in the same area, and they also cannot be used if fixed I/O word allocations are used for the CompoBus/D Master function. When a CompoBus/D Configurator is used, I/O word allocations can be set by the users to prevent the same word from being allocated more than one, enabling the use of multiple CompoBus/D Master Units, and the use of both the SYSMAC BUS Remote I/O function and CompoBus/D Master function at the same time.

Programming Devices

Connecting to SYSMAC Board's RS-232C Port

Any C200HX/HG/HE Programming Device/Software that can be connected to the RS-232C port can be used for the SYSMAC Board. The special Connecting Cables for the SYSMAC Board must be used to connect to the SYSMAC Board. Refer to the *C200HX/HG/HE (-E/-ZE) Programmable Controllers Installation Guide (W302)*, *C200HX/HG/HE-ZE Programmable Controllers Operation Manual (W322)*, the *SYSMAC Support Software Operation Manuals (W247, W248, and W249)*, and the *SYSMAC-CPT Support Tool Quick Start Guide and Users Manuals (W332 and W333)* for details on Programming Devices.

Software	Model number	Specifications	Connecting Cables
SYSMAC Support Software	C500-ZL3AT1-E	3.5" 2HD floppies IBM PC/AT or compatible	ES100-CT23-202 (9-pin type at Programming Device end)
SYSMAC-CPT Support Tool	WS01-CPB1-E	3.5" floppies/ CD-ROM Windows 3.1/95	

Using an Expansion Option Board

Connect the same Programming Devices and Connecting Cables that are used for the C200HX/HG/HE PLCs to the peripheral port and RS-232C port of the Expansion Option Board. Refer to the *C200HX/HG/HE (-E/-ZE) Programmable Controllers Installation Guide (W302)* and the applicable operation manual for the Programming Device to be used.

Note The SYSMAC Board's RS-232C port cannot be used when the Expansion Option Board is connected.

Serial Communications

The following serial communications functions for C200HX and C200HG PLCs can be used when a C200PC-EXP01 Expansion Option Board is connected. Refer to the *C200HX/HG/HE-ZE Programmable Controllers Operation Manual* (W322) for details.

Protocol	Applicable Devices	Connection ratio	Overview	Applicable communications instructions
Host Link	PLC to personal computer PLC to Programmable Terminal	1:1 1:N	Host Link communications are used between host computers and PLCs. The PLC's operating status and the operation of each bit can be monitored from the computer. The data stored in the IOM area of the PLC can be transferred to the host computer using the TXD(236) instruction.	Host Link commands TXD(236) for PLC-initiated communications
No protocol (custom)	PLC to personal computer PLC to components	1:1 1:1	Uses communications port I/O instructions TXD(236) and RXD(235) for simple transmission sequences, such as inputting data from Barcode Readers and outputting data to printers. Uses instructions to control RS, CS, ER, and DR signals.	TXD(236) and RXD(235)
1:1 Link	PLC to PLC	1:1	Allows one PLC to be connected to another PLC. Allows link bits to be shared.	---
NT Link	PLC to Programmable Terminal	1:1 1:N	Allows the PLC to be connected to one or more Programmable Terminals to send and receive data.	---

CompoBus/D Slaves

The number of CompoBus/D Slaves that can be connected to a C200HW-DRM21-EV1 CompoBus/D Master Unit used with a C200PC-ISA□2-DRM-E SYSMAC Board depends on whether or not a Configurator is used, as shown in the following table. Refer to the *CompoBus/D (DeviceNet) Operation Manual* (W267) for details.

Condition	Maximum No. of Slaves connected to SYSMAC Board
CompoBus/D Configurator not used.	50 Slave nodes
CompoBus/D Configurator used.	63 Slave nodes

CompoBus/S Slaves

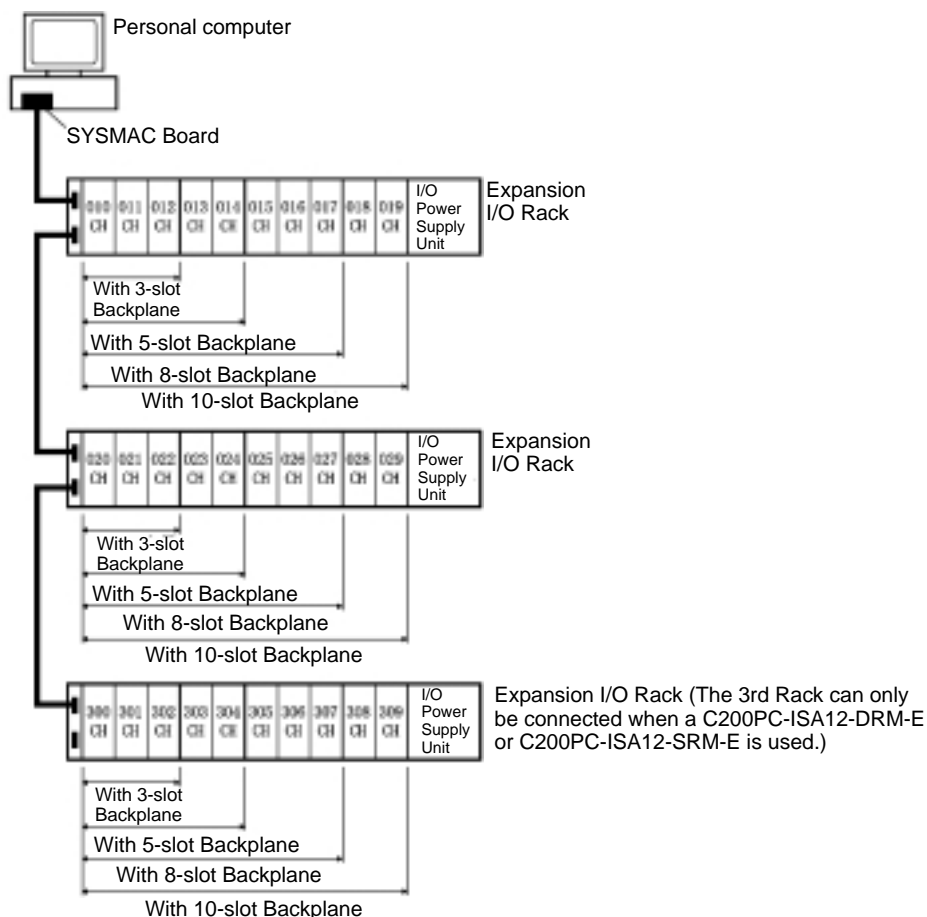
The node addresses and number of CompoBus/S Slaves that can be connected to a C200HW-SRM21 CompoBus/S Master Unit used with a C200PC-ISA□2-SRM SYSMAC Board depends on the setting of the communications setting switch is shown in the following table. Refer to the *CompoBus/S (DeviceNet) Operation Manual* (W266) for details.

CompoBus/S communications parameter DIP switch	Connectable node address	Max. No. of Slaves connected to SYSMAC Board
Pin 1: OFF	IN 0 to 7/OUT 0 to 7	16 Slave nodes
Pin 1: ON	IN 0 to 15/OUT 0 to 15	32 Slave nodes

I/O Allocations

Allocation of I/O words for the SYSMAC Board is the same as for the C200HX/C200HG PLCs, i.e., Units can be mounted in any slot, but the word allocated to

each slot is fixed. The word allocations to slots on the Expansion I/O Racks are shown in the following diagram.



Word allocations begin from the leftmost slot on the first Expansion I/O Rack beginning with IR 010 and continuing one word per slot through the end of the Rack. Words are allocated on the second and third Expansion I/O Rack in the same way, but they begin with IR 020 for the second Rack and IR 300 (IR Area 2) for the third Rack.

I/O Units that required less than 16 I/O bits are allocated only the number of bits they require beginning from bit 00. Any bits not allocated to real I/O can be used as work bits in programming.

The SYSMAC Board corresponds to the CPU Rack, but because it provides no I/O connections, the words that would normally be allocated to it, i.e., IR 000 to IR 009, can be used as work bits.

Word allocations to Special I/O Units are the same as for the C200HX-CPU64-E and C200HG-CPU43-E.

Refer to the *C200HX/HG/HE-ZE Programmable Controllers Operation Manual* (W322) for more details on memory allocation.

1-6 CompoBus/D Master Function

The C200PC-ISA□2-DRM-E SYSMAC Board with CompoBus/D is equipped with essentially the same remote I/O functionality as a C200HW-DRM21-EV1 CompoBus/D Master Unit. I/O can, therefore, be conducted with Slaves without special programming in the SYSMAC Board.

Note This manual does not provide detailed information on CompoBus/D functions. Refer to the *CompoBus/D (DeviceNet) Operation Manual* (W267) for details.

Maximum I/O Capacity and Maximum Number of Slaves

The following table shows the maximum number of I/O points and Slaves that can be connected to the SYSMAC Board.

Item	Without a CompoBus/D Configurator	With a CompoBus/D Configurator
Max. No. of I/O points	1,600 points (50 input/50 output words)	Without message communications: 4,800 points With message communications: 1,600 points
Max. No. of Slaves	50 Slaves (node addresses 0 to 49)	63 Slaves (node addresses 0 to 62)

Baud Rate and Communications Distance

The communications distance in a CompoBus/D Network is restricted by the baud rate, as shown in the following table.

Baud rate	Maximum network length		Drop line length	Total drop line length
	Thick Cable	Thin Cable		
500 kbps	100 m max.	100 m max.	6 m max.	39 m max.
250 kbps	250 m max.		6 m max.	78 m max.
125 kbps	500 m max.		6 m max.	156 m max.

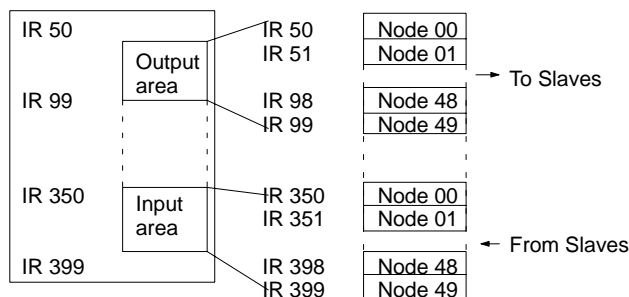
Remote I/O Communications

Remote I/O communications enable automatic transfer of I/O data between Slaves and the SYSMAC Board without any special programming. To achieve this, each Slave is allocated words in the I/O memory of the CPU Unit. Words are allocated as described next.

Fixed Allocations

Unless a Configurator is used to change the I/O allocations, words in the CPU Unit are allocated in order of CompoBus/D node addresses starting from node 00. The words are divided into an output area and an input area and allocated as shown in the following diagram.

Each node address is allocated one input and one output word. If a Slave requires more than one input or one output word, then it is assigned more than one node address. If a Slave requires less than one word, it simply uses the rightmost bits in the word allocated to it.

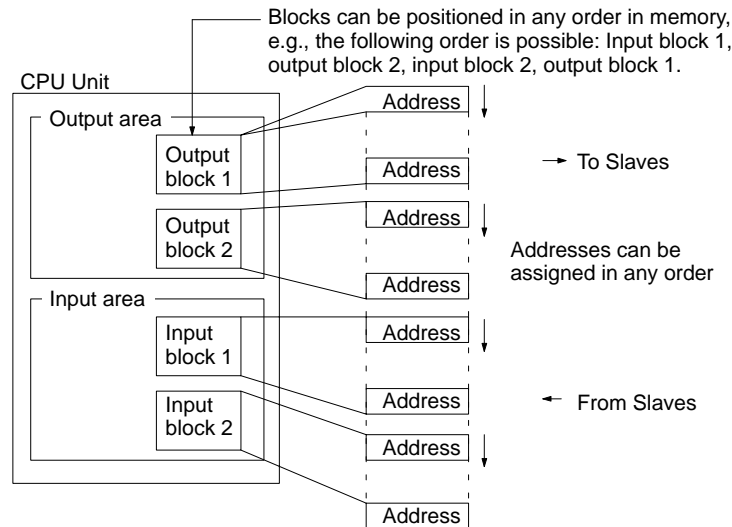


Note If a CompoBus/D Configurator is not used, I/O can be allocated using fixed allocations only. The SYSMAC Board does not provide CompoBus/D Configurator ISA Board functions. To use a CompoBus/D Configurator, you must either use a special ISA Board or use another personal computer to run the Configurator.

User-set Allocations

A Configurator can be used to allocate blocks 1 and 2 in the output area and input blocks 1 and 2 in the input area in any order to the node addresses in any order.

Each node is allocated at least one byte (leftmost or rightmost). If a Slave requires more than one input or one output word, then can be allocated more than one input or output word. If a Slave requires less than one word, it is allocated either the right or left byte in a word.



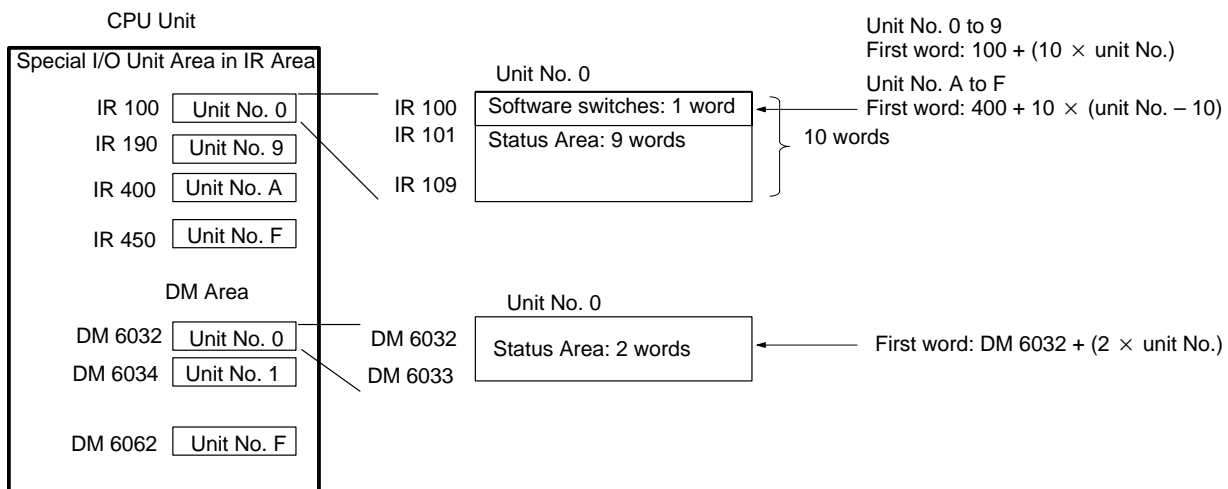
Message Communications

The C200PC-ISA□2-DRM-E SYSMAC Board also supports message communications. Message communications can be used to send messages between CompoBus/D masters and slaves (either those manufactured by OMRON or by other manufacturers). Messages can be sent based on execution conditions to achieve the following: Data transfers, reading/writing clocks or error logs, force-setting/resetting bits, etc.

Messages are implemented for the SYSMAC Board by sending FINS commands or explicit DeviceNet messages using ladder diagram instructions (IOWR(223)).

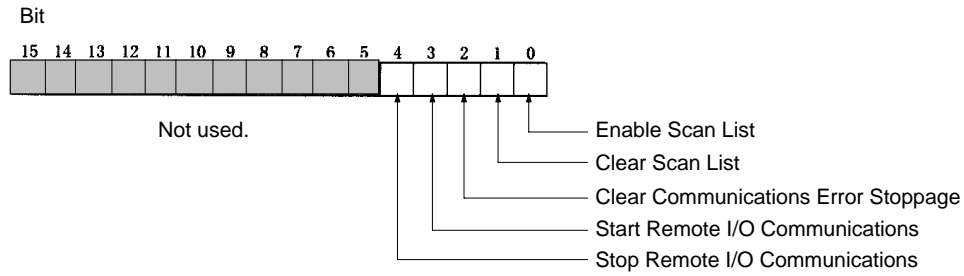
Software Switches and Status Area

The software switches and Status Area are allocated in IR words according to the Master Unit's unit number as shown in the following diagram.

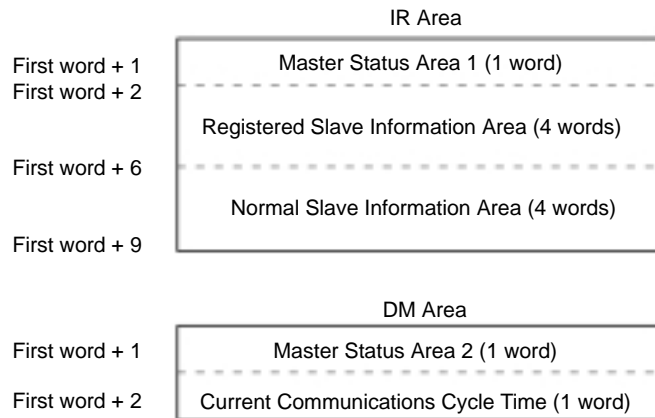


A brief explanation of the function of each area is provided here. For details, refer to *CompoBus/D (DeviceNet) Operation Manual (W267)*.

Software Switches

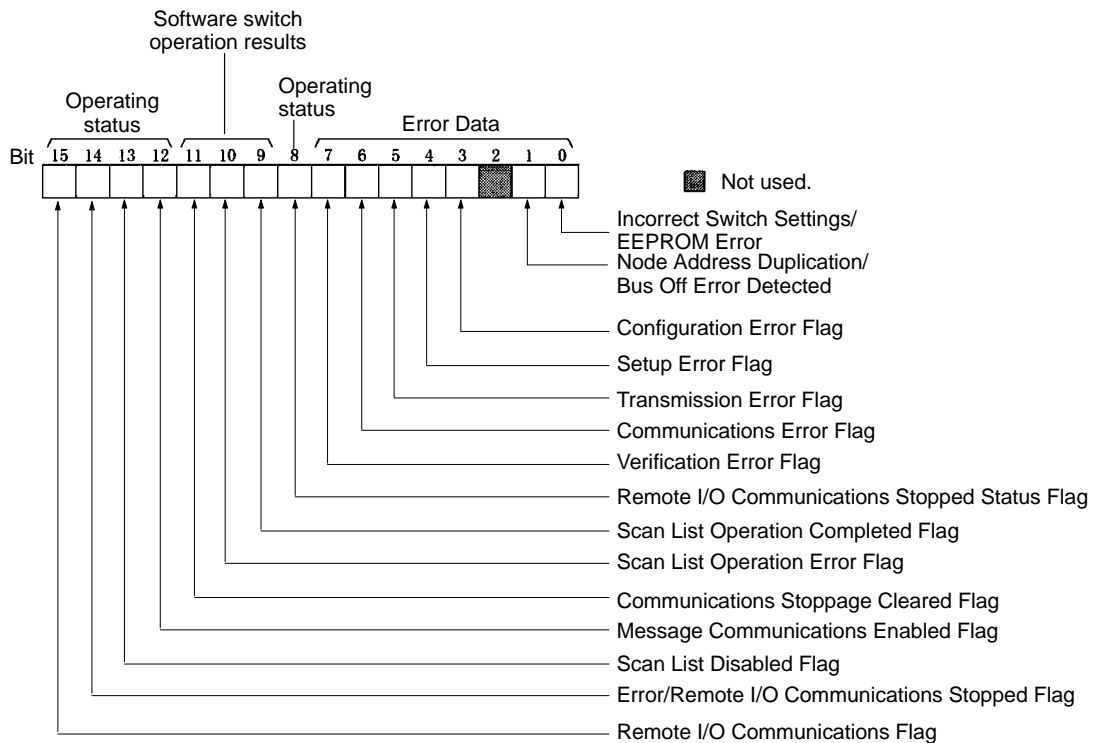


Status Area



Master Status Area 1

This word contains flags that show the operating status of the network, the results of software switch operations, and current error data.



Registered Slave Data

When the Master is operating with the scan list enabled, these flags indicate the Slaves that are registered in the scan list. When the Master is operating with the scan list disabled, these flags indicate the Slaves that have participated in communications even one time. Each bit is allocated to one Slave. When the Enable Scan List software switch is turned ON to create the scan list, the Slave for which bits are ON in the Registered Slave Data will be registered in the scan list.

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

Normal Slave Data

These flags indicate which Slaves are communicating normally. Each bit is allocated to one Slave.

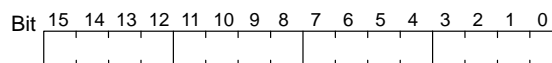
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
First word + 6	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
First word + 7	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
First word + 8	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
First word + 9	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48

Master Status Area 2

This word contains status data, such as whether there is an error history.

**Current Communications Cycle Time**

This word contains the the most recent communications cycle time.



1-7 CompoBus/S Master Function

The C200PC-ISA□2-SRM-E SYSMAC Board contains essentially the same remote I/O functionality as a C200HW-SRM21 CompoBus/S Master Unit. Therefore, I/O can be conducted with Slaves without special programming in the SYSMAC Board.

Note This manual does not provide detailed information on the CompoBus/S functions. Refer to the *CompoBus/S (DeviceNet) Operation Manual (W266)* for details.

Maximum I/O Capacity and Maximum Number of Slaves

The following table shows the maximum number of I/O points and Slaves that can be connected to the C200PC-ISA□2-SRM-E SYSMAC Board.

Item	Details
Max. No. of I/O points	256 points (128 inputs/ 128 outputs) or 128 points (64 inputs/64 outputs) (The maximum number of I/O points depends on setting for the number of nodes connectable.)
No. of I/O points per node	8 points
No. of nodes connectable	IN 0 to 7/OUT 0 to 7 or IN 0 to 15/OUT 0 to 15 (Set using the DIP switch.)

Note Slaves with 16 points use 2 node addresses. Slaves with 4 points use 8 bits starting from the rightmost bit.

Communications Distance

Using VCTF Cable

Main line	Drop line length	Total drop line length
100 m max.	3 m max.	50 m max.

Using Flat Cable

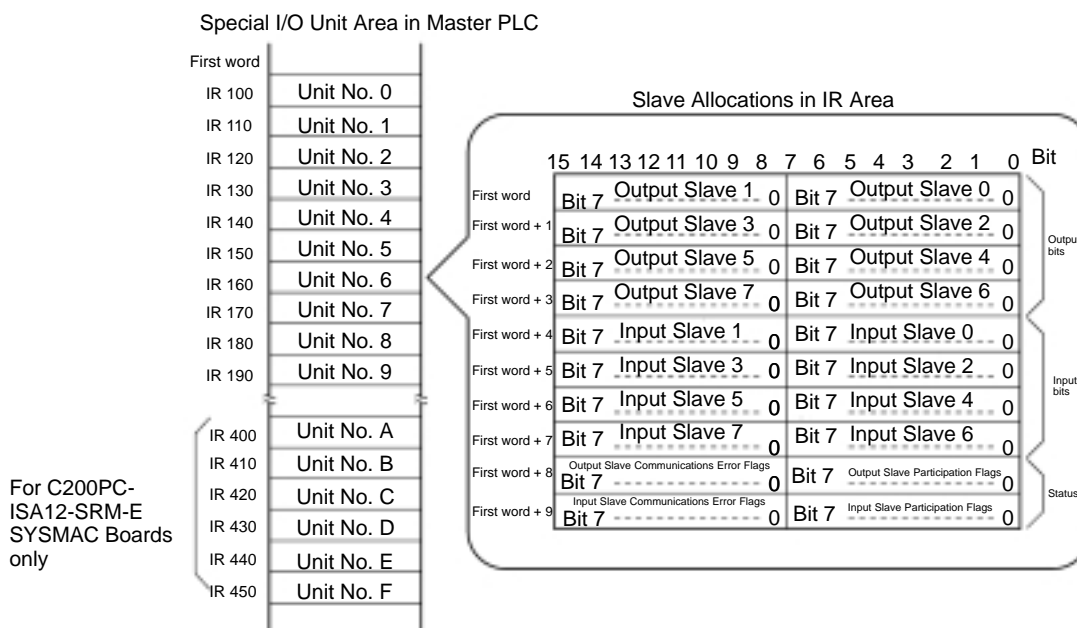
Main line	Drop line length	Total drop line length
30 m max.	3 m max.	30 m max.

If 16 Slaves or less are connected, the main line length is 100 m max, and the total drop line length is 50 m max.

Remote I/O Communications

I/O Allocations for IN 0 to 7/OUT 0 to 7

Slaves are allocated I/O using remote I/O allocations, as shown in the following diagram.



Slaves with 16 outputs or 16 inputs use 2 node addresses, and I/O is allocated in the same words, as follows:

If the Slave is set to a node address with an even number, the previous node address will also be used.

If the Slave is set to a node address with an odd number, the next node address will also be used.

Example

If a Slave with 16 output points is set to node address 3, it will use the words shown above for Output Slave 2 and 3. If the Slave is set to node address 6, it will use the words shown above for Output Slave 6 and 7.

If the Slave required both input and output allocations, it needs to only have the node address set; inputs and outputs for the same node address will be used.

Example

If a Slave with 8 points (4 inputs and 4 outputs) is connected to node address 2, it will use both input node address 2 and output node address 2.

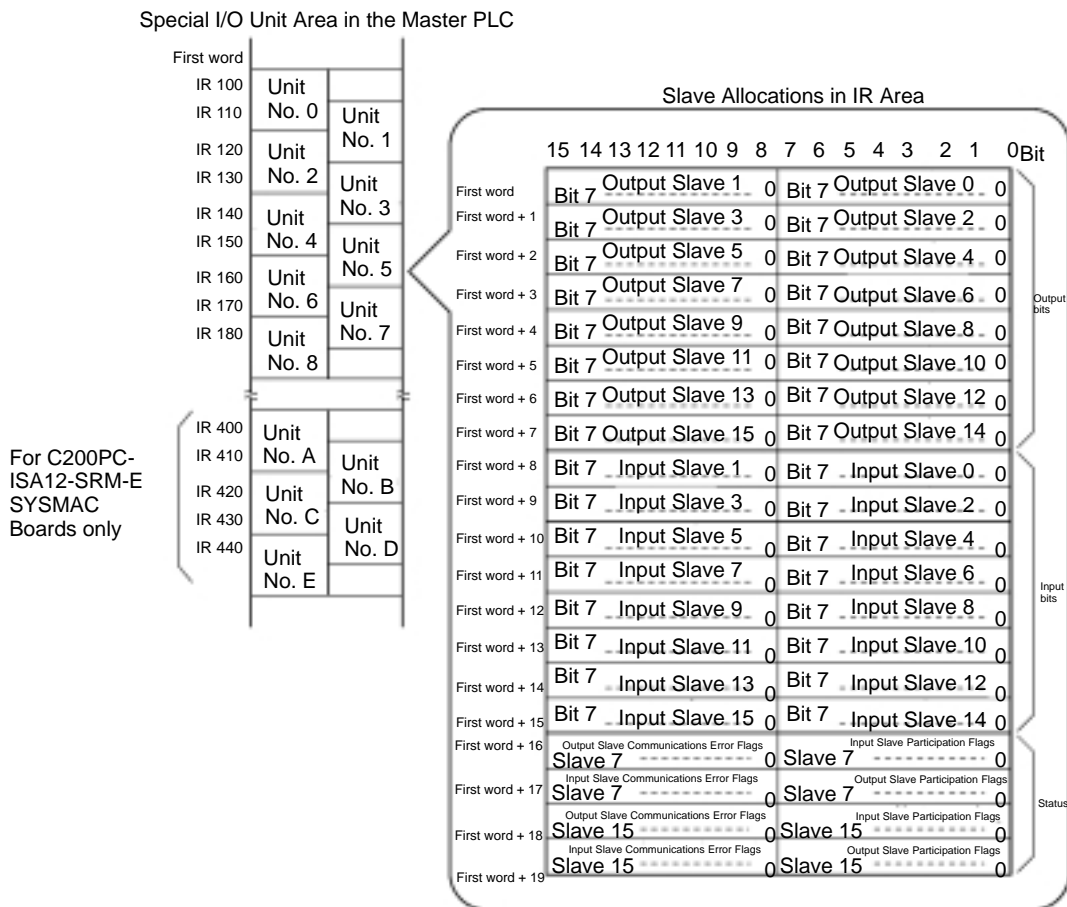
Slaves with 4 outputs or 4 inputs will occupy one node address, as follows:

If the Slave is set to a node address with an even number, it will use bits 08 to 11. Bits 12 to 15 will not be used.

If the Slave is set to a node address with an odd number, it will use bits 00 to 03. Bits 04 to 07 will not be used.

I/O Allocations for IN 0 to 15/OUT 0 to 15

When the connectable node addresses is set to IN 0 to 15/OUT 0 to 15, 20 words are used, so twice the amount of Units use the IR area.



Slaves with 16 outputs or 16 inputs used 2 node addresses, and I/O is allocated in the same words, as follows:

If the Slave is set to a node address with an even number, the previous node address will also be used.

If the Slave is set to a node address with an odd number, the next node address will also be used.

Example

If a Slave with 16 output points is set to node address 3, it will use the words shown above for Output Slave 2 and 3. If the Slave is set to node address 6, it will use the words shown above for Output Slave 6 and 7.

Slaves with 4 outputs or 4 inputs will use one node address, as follows:

If the Slave is set to a node address with an even number, it will use bits 08 to 11. Bits 12 to 15 will not be used.

If the Slave is set to a node address with an odd number, it will use bits 00 to 03. Bits 04 to 07 will not be used.

Note When the connectable node addresses are set to IN 0 to 15/OUT 0 to 15, unit numbers 9 to F cannot be set.

Status Area

The flags used in the Status Area are as follows:

Output Slave Participation Flags

These flags show whether the communications have been established for each Output Slave (Output Slaves 0 to 15) in one bit for each Slave.

0 (OFF): Communications not established. (No communications yet.)

1 (ON): Participated in communications. (Will not change once turned ON.)

All bits are automatically cleared when the power is turned ON again or the Slave is restarted.

Input Slave Participation Flags

These flags show whether the communications have been established for each Input Slave (Input Slaves 0 to 15) in one bit for each Slave.

0 (OFF): Communications not established. (No communications yet.)

1 (ON): Participated in communications. (Will not change once turned ON.)

All bits are automatically cleared when the power is turned ON again or the Slave is restarted.

Output Slave Communications Error Flags

These flags show the communications status of each Output Slave (Output Slaves 0 to 15) using one bit for each Slave.

0 (OFF): Communications are normal or have not been established.

1 (ON): Communications have been released (communications error).

All bits are automatically cleared when the power is turned ON again or the Slave is restarted.

Input Slave Communications Error Flags

These flags show the communications status of each Input Slave (Input Slaves 0 to 15) in one bit for each Slave.

0 (OFF): Communications are normal or have not been established.

1 (ON): Communications have been released (communications error).

All bits are automatically cleared when the power is turned ON again or the Slave is restarted.

Note At CompoBus/S startup, inputs from Input Slaves and outputs to Output Slaves will be enabled as soon as the corresponding Participation Flag turns ON. When the Slave is going to be used, check to be sure that the Communications Error Flag is OFF after the Participation Flag turns ON in ladder logic used to process I/O for the Slave.

1-8 Computer Board Functions

The SYSMAC Board normally functions independently from the personal computer it is installed in based on a user program downloaded from a Programming

PLC Communications Library Functions

Device. The personal computer, however, can function as a host computer for the SYSMAC Board to read/write memory areas on the SYSMAC Board, to read SYSMAC Board status, or to control SYSMAC Board operation.

Communications between the personal computer and the SYSMAC Board are implemented using C language library functions written into a user application running on the personal computer. The PLC communications library includes the following functions.

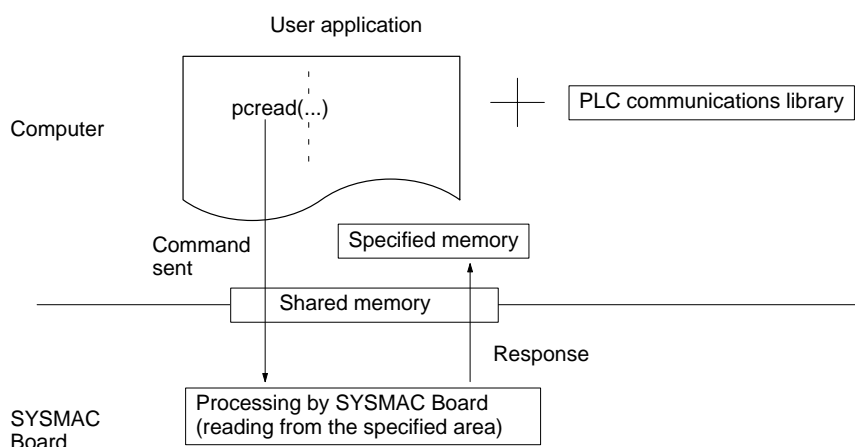
Name	Operation	Reference
pcinit()	Sets the I/O port address and base address in shared memory to use for communications between the SYSMAC Board and computer.	p. 64
pcopen()	Initializes communications for the SYSMAC Board and starts communications between the SYSMAC Board and the computer.	p. 65
pcclose()	Ends communications between the SYSMAC Board and the computer.	p. 66
pcmode()	Changes the SYSMAC Board operating mode (PROGRAM, MONITOR, RUN).	p. 66
pcstat()	Accesses and controls SYSMAC Board status, including the following: Reading status Clearing errors Reading the error history Clearing the error history	p. 68
pcread()	Reads data from a SYSMAC Board memory area.	p. 76
pcwrite()	Writes data into a SYSMAC Board memory area.	p. 82

Note A special driver is not required to use the SYSMAC Board.

Use either FinsGateway for Windows 95 or FinsGateway for Windows NT (sold separately) when using Windows 95 or Windows NT.

Communications

The following diagram illustrates communications between the SYSMAC Board and the computer using `pcread` as an example.



- 1, 2, 3... 1. The `pcread` function from the PLC communications library is executed in the user application on the computer to send a command to the SYSMAC Board via shared memory.
2. The SYSMAC Board processes the command and returns a response to the computer via shared memory to store the results of the command in the memory location specified in the `pcread` function.

Note 1. The PLC communications library is created using Microsoft C/C++ Ver. 7. Refer to *Section 3 PLC Communications Library* for details.

2. The SYSMAC Board does not use hardware interrupts.

1-9 Differences from Standard PLCs

Although functionally a SYSMAC Board is essentially the same as either the C200HX-CPU64-E or C200HG-CPU43-E, depending on the model, there are a few differences resulting from differences in hardware. These differences are described next. These differences must be allowed for when programming or operating the SYSMAC Board.

CPU Rack

The SYSMAC Board corresponds to the CPU Rack, making it impossible to add Units and other hardware, such as Communications Boards, to the CPU Rack. This also limits the maximum I/O capacity to the I/O provided on the Expansion I/O Racks, which is either 320 points maximum for 2 Racks or 480 points maximum for 3 Racks.

Communications Boards

Communications Boards cannot be mounted. This means that protocol macros and expansions to other CPU Bus Units is not possible.

Peripheral Port

There is no peripheral port on the SYSMAC Board. This means that Programming Consoles and other peripheral devices that must be connected to a peripheral port cannot be used.

If the C200PC-EXP01 Expansion Option Board is connected, its peripheral port, which is equivalent to the peripheral port of a C200HX or C200HG PLC, can be used.

Programming Devices Connected to RS-232C Port

The RS-232C port uses different connector, so special Connecting Cables must be used to connect Programming Devices. Only Programming Devices/Software can be connected to the RS-232C port.

If, however, a C200PC-EXP01 Expansion Option Board is connected, its RS-232C port, which is equivalent to the RS-232C port of a C200HX or C200HG PLC, can be used. When the Expansion Option Board's RS-232C port is used, the SYSMAC Board's RS-232C port cannot be used.

PLC Operating Mode

The SYSMAC Board will always start up in PROGRAM mode. To execute the user program, you must switch to RUN mode either from a Programming Device or by executing `pcmode` from a user application on the computer.

Holding Data Area Contents

The SYSMAC Board will maintain HR area, AR area, CNT area, DM area, EM area, and time data when the power supply is turned off. The operating mode, however, must be changed to PROGRAM mode before turning off the power to ensure that data area contents are saved correctly. Before turning off the power supply, you must switch to PROGRAM mode either from a Programming Device or by executing `pcmode` from a user application on the computer.

1-10 Specifications

1-10-1 General Specifications

Item	Specifications
Power supply voltage	4.875 to 5.25 VDC
Power consumption	0.5 A max. 0.8 A max. if the Programming Console is connected to the C200PC-EXP01 Expansion Option Board.
Vibration resistance	JIS C0911 conforming, 10 to 57 Hz; 0.075 mm amplitude, 57 to 150 Hz (see note 2); acceleration: 1 G, in X, Y, and Z directions, for 80 minutes each (sweep time 8 min x 10 sweeps = 80 min); (When mounted on DIN track, 2 to 55 Hz, 0.3 G, in X, Y, and Z directions, for 20 minutes each)
Shock resistance	JIS C0912 conforming, 15 G (147 m/S ²) in X, Y, and Z directions, 3 times each
Ambient temperature	Operating: 0° to 55°C; storage: -20° to 75°C (without battery)
Humidity	10% to 90% (without condensation)
Atmosphere	Must be free of corrosive gases.
Weight	SYSMAC Board: 350 g max. Expansion Option Board: 250 g max.
Dimensions	SYSMAC Board: 106.7 x 163.0 x 16.3 mm (W x H x D) (The depth measurement is the height at the battery.) Expansion Option Board: 106.7 x 128.7 x 13.5 mm (W x H x D) (The depth measurement is the height of the connector guide.)

1-10-2 SYSMAC Board CPU Specifications

Item	Specifications
Control method	Stored program
I/O control method	Cyclic scan and immediate processing are both possible.
Programming method	Ladder diagram
Instruction length	1 address/instruction, 1 to 4 words/instruction
Number of instructions	14 basic instructions + 231 special instructions
Execution time	C200PC-ISA01-E, C200PC-ISA02-DRM-E, C200PC-ISA02-SRM-E: Basic instructions: 0.15 μs min. Special instructions: 0.6 μs min. (for MOV(021)) C200PC-ISA12-DRM-E, C200PC-ISA12-SRM-E: Basic instructions: 0.10 μs min. Special instructions: 0.4 μs min. (for MOV(021))
Program capacity	C200PC-ISA01-E, C200PC-ISA02-DRM-E, C200PC-ISA02-SRM-E: 15.2K words max. C200PC-ISA12-DRM-E, C200PC-ISA12-SRM-E: 31.2K words max.
User program/I/O data storage area	C200PC-ISA01-E, C200PC-ISA02-DRM-E, C200PC-ISA02-SRM-E: 16 K words of built-in EEPROM (equivalent to C200HW-ME16K Memory Cassette). C200PC-ISA12-DRM-E, C200PC-ISA12-SRM-E: 32 K words of built-in EEPROM (equivalent to C200HW-ME32K Memory Cassette). Max. No. of writes: 50,000 times (Cannot be removed.)
IR Area: I/O bits	C200PC-ISA01-E, C200PC-ISA02-DRM-E, C200PC-ISA02-SRM-E: 320 bits (IR 01000 through IR 02915) C200PC-ISA12-DRM-E, C200PC-ISA12-SRM-E: 480 bits (IR 01000 through IR 02915, IR 03000 through IR 30915)

Item	Specifications
IR Area: Work bits	C200PC-ISA01-E, C200PC-ISA02-DRM-E, C200PC-ISA02-SRM-E: 6,784 bits (IR 00000 to IR 00915, IR 03000 through IR 23115, IR 30000 through IR 51115) C200PC-ISA12-DRM-E, C200PC-ISA12-SRM-E: 6,624 bits (IR 00000 through IR 00915, IR 03000 through IR 23115, IR 31000 through IR 51115)
SR bits	1,080 (IR 23600 through IR 25507 and IR 25600 through IR 29915)
TR bits	8 (TR 0 through TR 7)
HR bits	1,600 (HR 0000 through HR 9915)
AR bits	448 (AR 0000 through AR 2715)
LR bits	1,024 (LR 0000 through LR 6315)
Timers/Counters	512 (TIM/CNT 000 through TIM/CNT 511)
DM words	Read/write: 6,144 (DM 0000 through DM 6143) Read only: 512 (DM 6144 through DM 6655) Expansion DM: 3,000 words max. (DM 7000 through DM 9999) (The Expansion DM Area must be set from a Programming Device.)
EM words	C200PC-ISA01-E, C200PC-ISA02-DRM-E, C200PC-ISA02-SRM-E: Read/write: 6,144 words (EM 0000 to EM 6143) C200PC-ISA12-DRM-E, C200PC-ISA12-SRM-E: Read/write: 6,144 words (EM 0000 to EM 6143) x 3 banks
Power failure backup function	Holds HR, AR, CNT, and DM, EM and clock (RTC) contents. The above data may not be saved correctly if power is turned off when the SYSMAC Board is not in PROGRAM mode.
Memory backup time	The battery service life is five years at 25°C. The service life will be shortened if the battery is used at higher temperatures. Replace the battery within one week after the battery alarm indicator starts blinking. When replacing the battery, install the new battery within 5 minutes after removing the old one. (Refer to page 103.)
Self-diagnostic function	CPU errors (watchdog timer), I/O verification errors, host link errors, memory errors, battery errors, I/O bus errors, etc.
Program check function	Carries out program checks at the beginning of operation for items such as no END(01) instruction, instruction errors, and so on. Program can also be checked from Programming Device; three checking levels are available.

1-10-3 CompoBus/D Communications Specifications

CompoBus/D communications specifications conform to the DeviceNet specification.

Item		Specification
Baud rate		500 kbps, 250 kbps, or 125 kbps (switchable)
Communications distances	500 kbps	Network length: 100 m max. Drop line length: 6 m max. Total drop line length: 39 m max.
	250 kbps	Network length: 250 m max. (see note 1) Drop line length: 6 m max. Total drop line length: 78 m max.
	125 kbps	Network length: 500 m max. (see note 1) Drop line length: 6 m max. Total drop line length: 156 m max.
Max. number of Slaves		Without Configurator: 50 Slaves With Configurator: 63 Slaves
Max. I/O capacity		Without Configurator: 1,600 pts (50 input/50 output words) With Configurator: Without messages: 4,800 pts With messages: 1,600 pts
Error control checks		CRC error check, node address duplications, scan list verification
Cables		5-wire cable: 2 signal lines, 2 power supply lines, 1 shield line

- Note**
1. Indicates the length when Thick Cables are used. Reduce the network length to 100 m max. when using Thin Cables. When using both Thick and Thin Cables together, refer to the *CompoBus/D (DeviceNet) Operation Manual* (W267) for details.
 2. The CompoBus/D connector on the C200PC-ISA□2-DRM-E SYSMAC Board is different from that for the C200HW-DRM21-EV1 CompoBus/D Master Unit. Use the cable provided with the SYSMAC Board.

1-10-4 CompoBus/S Communications Specifications

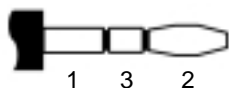
Item	Specifications			
Communications	CompoBus/S protocol			
Baud rate	750 kbps			
Modulation	Base band			
Code	Manchester code			
Error control checks	Manchester code check, frame length check, parity check			
Cables	VCTF Cable: VCTF JIS C 3306 twisted-pair, nominal cross section: 0.75 mm ² (2 signal wires) Flat Cable: 0.75 mm ² x 4 (2 signal wires, 2 power lines)			
Communications distance	VCTF Cable	Main line	Drop lines	Total drop line
		100 m max.	3 m max. each	50 m max.
	Flat Cable	Main line	Drop lines	Total drop line
		30 m max.	3 m max. each	30 m max.
Maximum I/O points	64 inputs/64 outputs		128 inputs/128 outputs	
No. of connectable node addresses	IN 0 to 7/OUT 0 to 7		IN 0 to 15/OUT 0 to 15	
Communications cycle	0.5 ms		0.8 ms	

Note If Flat Cables are used when a maximum of 16 Slaves are connected, the maximum main line length will be 100 m, and a maximum total drop line length will be 50 m.

1-10-5 Board Specifications

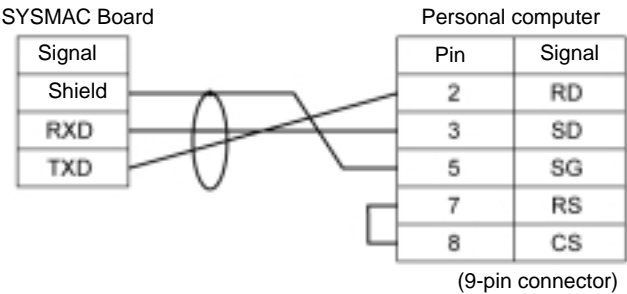
Item	Specification
I/O port addresses	3X0H to 3X4H The base address is set to one of the following hexadecimal values via the SYSMAC Board DIP switch: 0370, 0380, 0390, or 03A0 (default: 03A0).
Memory addresses	8 Kbytes required. The base address is set to between 0C8000 and 0E5FFF hexadecimal using an I/O port register (default: 0C8000 hexadecimal).

1-10-6 RS-232C Port Cable Specifications



Pin	Signal
1	0 V
2	RXD
3	TXD

The connections for the RS-232C port are shown below.

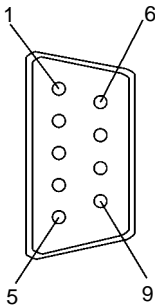


1-10-7 Expansion Option Board

Communications Specifications

Item	Specifications
Communications	Half duplex
Synchronization	Start-stop
Baud rate	1,200, 2,400, 4,800, 9,600, or 19,200 bps
Transmission method	Point to point
Transmission distance	15 m max.
Interface	Conforms to EIA RS-232C

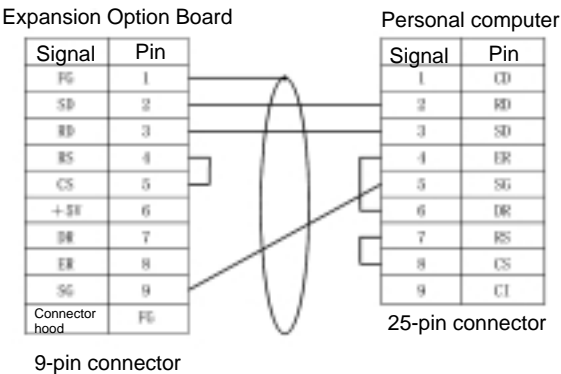
Pin Assignments



Pin	Abbreviation	Name	Direction
1	FG	Field ground	---
2	SD (TXD)	Send data	Output
3	RD (RXD)	Receive data	Input
4	RS (RTS)	Request to send	Output
5	CS (CTS)	Clear to send	Input
6	+5V	Power supply	---
7	DR (DSR)	Data set ready	Input
8	ER (DTR)	Data terminal ready	Output
9	SG	Signal ground	---
Connector hood	FG	Field ground	---

Connection Method

The RS-232C port on the Expansion Option Board can be connected to a personal computer as shown in the following diagram.



Connectors and Cables

Use the following connectors and cables or equivalent.

Name	Model	Manufacturer
Plug	SM2A-0901	OMRON
Hood	XM2S-0911	OMRON
Cable	AWG28 × 5P IFVV-SB	Fujikura Ltd.
	CO-MA-VV-SB 5P × AWG28	Hitachi Cable, Ltd.

The maximum cable length is 15 m.

1-11 Setup Procedures

Hardware Setup

Refer to *Section 2 Hardware Setup and Installation* for details. Make the following settings before mounting the SYSMAC Board into the computer.

Item	Setting	Description
User memory protection	PLC DIP switch, pin 1	Enables or disables writing to the user memory area.
EEPROM transfer at startup	PLC DIP switch, pin 2	Specifies whether or not to automatically load the contents of EEPROM when the SYSMAC Board is started.
Programming Console message display language	PLC DIP switch, pin 3	Sets the Programming Console message display to Japanese or English.
Expansion instructions setting	PLC DIP switch, pin 4	Specifies whether or not expansion instructions are to be set.
RS-232C port parameters	PLC DIP switch, pin 5	Specifies whether or not default communications parameters are to be used.
Programming Console mode/Expansion Terminal mode	PLC DIP switch, pin 6	Sets the Programming Console mode to the console mode or expansion terminal mode.
I/O port address	PLC DIP switch, pins 7, 8	Sets the first address of the I/O port to be used by SYSMAC Board registers.
EEPROM protection	EEPROM write protect switch	Enables or disables writing to EEPROM.
CompoBus/D node address (C200PC-ISA□2-DRM only)	CompoBus/D node address DIP switch	Sets the node address of the SYSMAC Board's CompoBus/D Master Unit.
CompoBus/D unit number (C200PC-ISA□2-DRM only)	CompoBus/D unit number switch	Sets the unit number of the SYSMAC Board's CompoBus/D Master Unit.
CompoBus/D communications parameters (C200PC-ISA□2-DRM only)	CompoBus/D communications parameter DIP switch	Sets the baud rate and the handling of communications error for CompoBus/D communications.
CompoBus/S unit number (C200PC-ISA□2-SRM only)	CompoBus/S unit number switch	Sets the unit number of the SYSMAC Board's CompoBus/S Master Unit.
CompoBus/S communications parameters (C200PC-ISA□2-SRM only)	CompoBus/S communications parameter DIP switch	Sets the node addresses that can be connected for CompoBus/S communications.

Installation

Refer to *Section 2 Hardware Setup and Installation* for details.

The SYSMAC Board must be mounted in the computer. Any Expansion I/O Racks, the CompoBus/D Network, the CompoBus/S Network, or other external devices must then be wired.

This manual covers only the specific functions of the SYSMAC Board. Refer to the *C200HX/HG/HE (-E/-ZE) Programmable Controllers Installation Guide* (W302), the *CompoBus/D (DeviceNet) Operation Manual* (W267), and the *CompoBus/S (DeviceNet) Operation Manual* (W266) for details on system installation.

Programming

Refer to *Section 3 PLC Communications Library* for details.

The programs to control the SYSMAC Board must be written.

This manual describes using the PLC communications library to control the SYSMAC Board from user applications running on the computer. Refer to the following manuals for further details.

- The *C200HX/HG/HE-ZE Programmable Controllers Operation Manual* (W322) for information on writing the ladder diagram program to control the SYSMAC Board
- The *CompoBus/D (DeviceNet) Operation Manual* (W267) for details on remote I/O and message communications in a CompoBus/D Network.
- The *CompoBus/S (DeviceNet) Operation Manual* (W266) for details on remote I/O in a CompoBus/S Network.

Note Thoroughly check the user program after writing it before using it to control device or system.

SECTION 2

Hardware Setup and Installation

This section describes the hardware settings on the SYSMAC Board, SYSMAC Board installation methods, CompoBus/D wiring methods, and CompoBus/S wiring methods. Refer to the *CompoBus/D Operation Manual* (W267) and the *C200HX/HG/HE (-E/-ZE) Installation Guide* (W302) for details on CompoBus/D and Expansion I/O Rack installation.

2-1	Setup Confirmations	34
2-1-1	Confirming Available Resources	34
2-1-2	Reserving Resources	34
2-2	Components	35
2-3	Board Switch Settings	44
2-3-1	PLC DIP Switch (Same for All Boards)	44
2-3-2	EEPROM Write Protect Switch (Save for All Boards)	46
2-3-3	CompoBus/D Node Address DIP Switch (C200PC-ISA□2-DRM-E)	46
2-3-4	CompoBus/D Unit Number Switch (C200PC-ISA□2-DRM-E)	47
2-3-5	CompoBus/D Communications DIP Switch (C200PC-ISA□2-DRM-E)	47
2-3-6	CompoBus/S Unit Number Switch (C200PC-ISA□2-SRM-E)	48
2-3-7	CompoBus/S Communications DIP Switch (C200PC-ISA□2-DRM-E)	49
2-4	Support Board Installation	49
2-5	External Connections	52
2-5-1	C200HX/HG/HE Expansion I/O Racks	53
2-5-2	Connecting CompoBus/D Communications Cables (C200PC-ISA□2-DRM-E) .	53
2-5-3	Connecting CompoBus/S Communications Cables (C200PC-ISA□2-SRM-E) .	56
2-5-4	Connecting the SYSMAC Board's RS-232C Port (With No Expansion Option Board)	58
2-5-5	Connecting the Expansion Option Board's RS-232C Port and Peripheral Port ..	59
2-6	Required Files	59

2-1 Setup Confirmations

Confirm the following items before starting to set up or install the SYSMAC Board.

2-1-1 Confirming Available Resources

I/O Ports

One of the following ranges of I/O ports is used on the computer by the SYSMAC Board. Confirm that one of these is available for use by the SYSMAC Board. If none of these I/O ports are available, reset another resource so that one of the ranges can be used by the SYSMAC Board.

0370 to 0374

0380 to 0384

0390 to 0394

03A0 to 03A4

Memory Addresses

One of the ranges of memory addresses is used on the computer by the SYSMAC Board. Confirm that one of these is available for use by the SYSMAC Board. If none of these ranges is available, reset another resource so that one of them can be used by the SYSMAC Board.

C8000 to C9FFF

CA000 to CBFFF

CC000 to CDFFF

CE000 to CFFFF

D0000 to D1FFF

D2000 to D3FFF

D4000 to D5FFF

D6000 to D7FFF

D8000 to D9FFF

DA000 to DBFFF

DC000 to DDFFF

DE000 to DFFFF

E0000 to E1FFF

E2000 to E3FFF

E4000 to E5FFF

Note Do not set the memory area used by the SYSMAC Board to Shadow RAM. When using Windows 95, confirm the available resources by opening the System Icon from the Control Panel and then display the Computer Properties using the Device Manager.

2-1-2 Reserving Resources

Use the following procedure to reserve the resources (e.g., I/O ports and memories) for the SYSMAC Board so that they will not be used for other applications.

- 1, 2, 3... 1. Click the **Start** button in Windows 95 and select **Settings** and then **Control Panel**.
2. Double-click the **System** icon in the Control Panel Window. The System Properties Window will be displayed.
3. Display the **Device Manager** tab and then select **Computer** and click **Properties**. The Computer Properties Window will be displayed.
4. Display the **Resource Reservations** and then select **I/O Port Address** or **Memory** and click the **Add** button.
5. Enter the range to be used by the SYSMAC Board and then click the **OK** button. Be sure to enter the values within the ranges that have been confirmed to be available.

Example: Set the following when I/O ports 0390 to 0394H and the memory addresses D0000 to D1FFFFH are to be used.

I/O Ports

Beginning value: 0390

Ending value: 0394

Memory Addresses

Beginning value: 000D0000

Ending value: 000D1FFF

6. Press the **OK** button in the Computer Properties Window and in the System Properties Window to reserve and confirm the set I/O and memory addresses.

Note When using EMM386.EXE, modify the contents of the CONFIG.SYS file to reserve memory. Using the EMM386.EXE parameter, set the memory area to be used by the SYSMAC Board as prohibited addresses.

Parameter Setting: X=mmmm–nnnn

mmmm: Beginning address (leftmost 4 digits)

nnnn: Ending address (leftmost 4 digits)

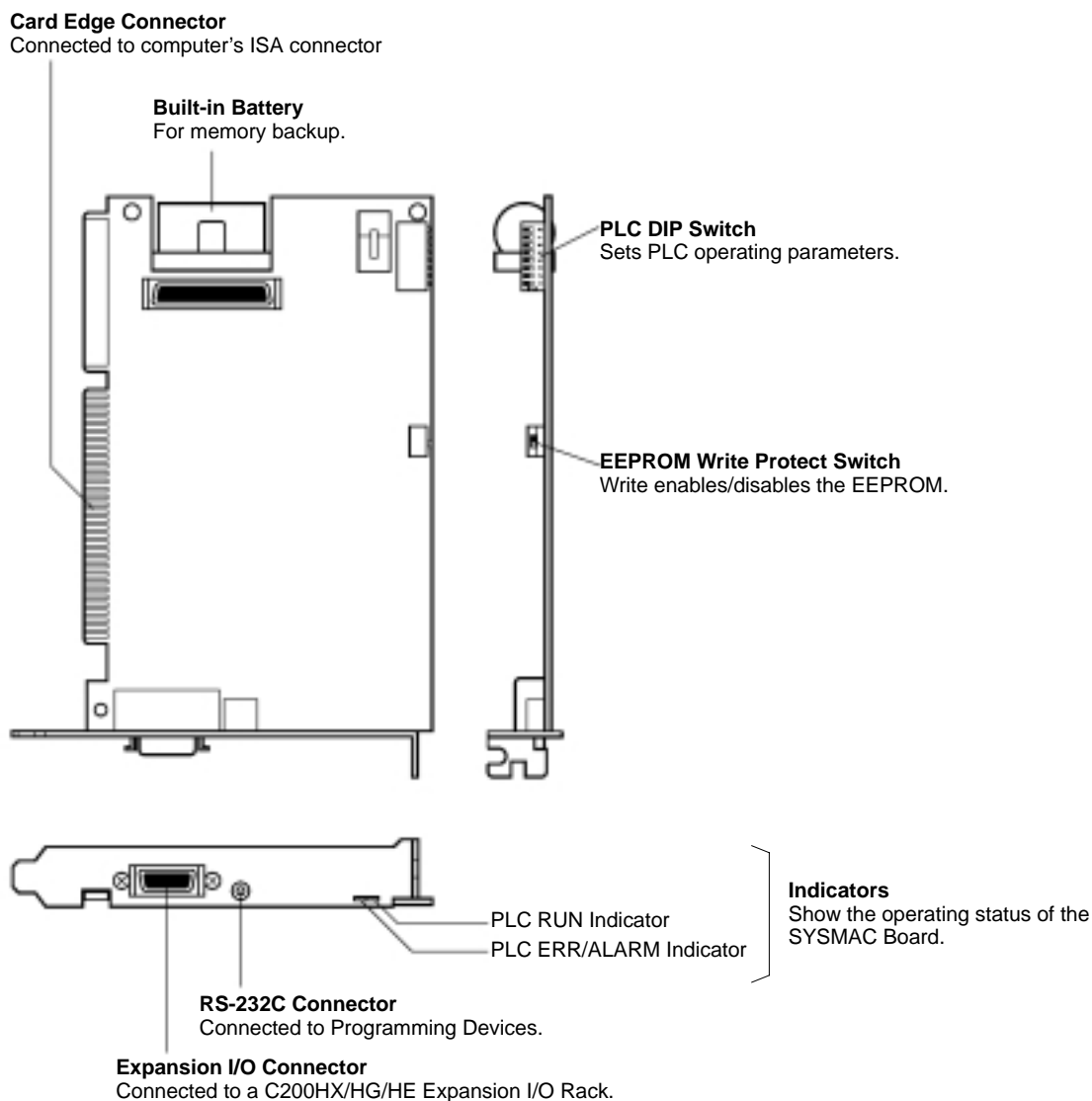
Example: Set the following when memory addresses D0000 to D1FFFH are to be used.

X=D000–D1FF

2-2 Components

The names and functions of the SYSMAC Board components are given in this section.

C200PC-ISA01-E SYSMAC Board



C200PC-ISA□2-DRM-E**Card Edge Connector**

Connected to computer's ISA connector

Built-in Battery

For memory backup.

PLC DIP Switch

Sets PLC operating parameters.

Expansion Option Board Connector

Used to connect a flat cable to the Expansion Option Board.

EEPROM Write Protect Switch

Write enables/disables the EEPROM.

Expansion Function Connector

Not used. Do not connect anything.

CompoBus/D DIP Switch

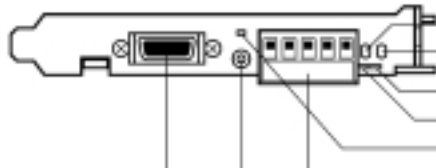
Sets CompoBus/D communications parameters.

CompoBus/D Node Address DIP Switch

Sets the node address of the CompoBus/D Master.

CompoBus/D Unit Number Switch

Sets the unit number of the CompoBus/D Master.



CompoBus/D NS Indicator
CompoBus/D MS Indicator
PLC RUN Indicator
PLC ERR/ALARM Indicator
PLC COM Indicator

Indicators

Show the operating status of the SYS-MAC Board, the CompoBus/D Master, and the CompoBus/D Network.

CompoBus/D Connector

Connected to the CompoBus/D Network. The cable connector is mounted in the connector at the factory.

RS-232C Connector

Connected to Programming Devices.

Cannot be used when the Expansion Option Board is connected.

Expansion I/O Connector

Connected to the C200HX/HG/HE Expansion I/O Rack.

C200PC-ISA□2-SRM-E**Card Edge Connector**

Connected to computer's ISA connector

Built-in Battery

For memory backup.

PLC DIP Switch

Sets PLC operating parameters.

Expansion Option Board Connector

Used to connect a flat cable to the Expansion Option Board.

EEPROM Write Protect Switch

Write enables/disables the EEPROM.

Expansion Function Connector

Not used. Do not connect anything.

CompoBus/D DIP Switch

Sets CompoBus/D communications parameters.

CompoBus/D Unit Number Switch

Sets the unit number of the CompoBus/D Master.

CompoBus/S Status Indicators
(RUN, ERC, SD, RD)

CompoBus/S Error Slave Indicators
(IN/OUT, 8, 4, 2, 1)

PLC RUN Indicator

PLC ERR/ALARM Indicator

PLC COM Indicator

Indicators

Show the operating status of the SYSMAC Board, the CompoBus/D Master, and the CompoBus/D Network.

CompoBus/D Connector

Connected to the CompoBus/D Network. The cable connector is mounted in the connector at the factory.

RS-232C Connector

Connected to Programming Devices.

Cannot be used when the Expansion Option Board is connected.

Expansion I/O Connector

Connected to the C200HS/HG/HE Expansion I/O Rack.

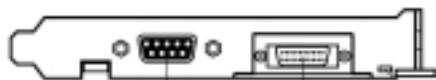
C200PC-EXP01 Expansion Option Board**Card Edge Connector**

Connected to computer's ISA connector.

With the Expansion Option Board, data is not directly transmitted with the computer and the Connector is used only to secure the Board to the computer.

**SYSMAC Board Connector**

Used to connect a flat cable to the SYSMAC Board.



PLC COMM Indicator

Peripheral Connector

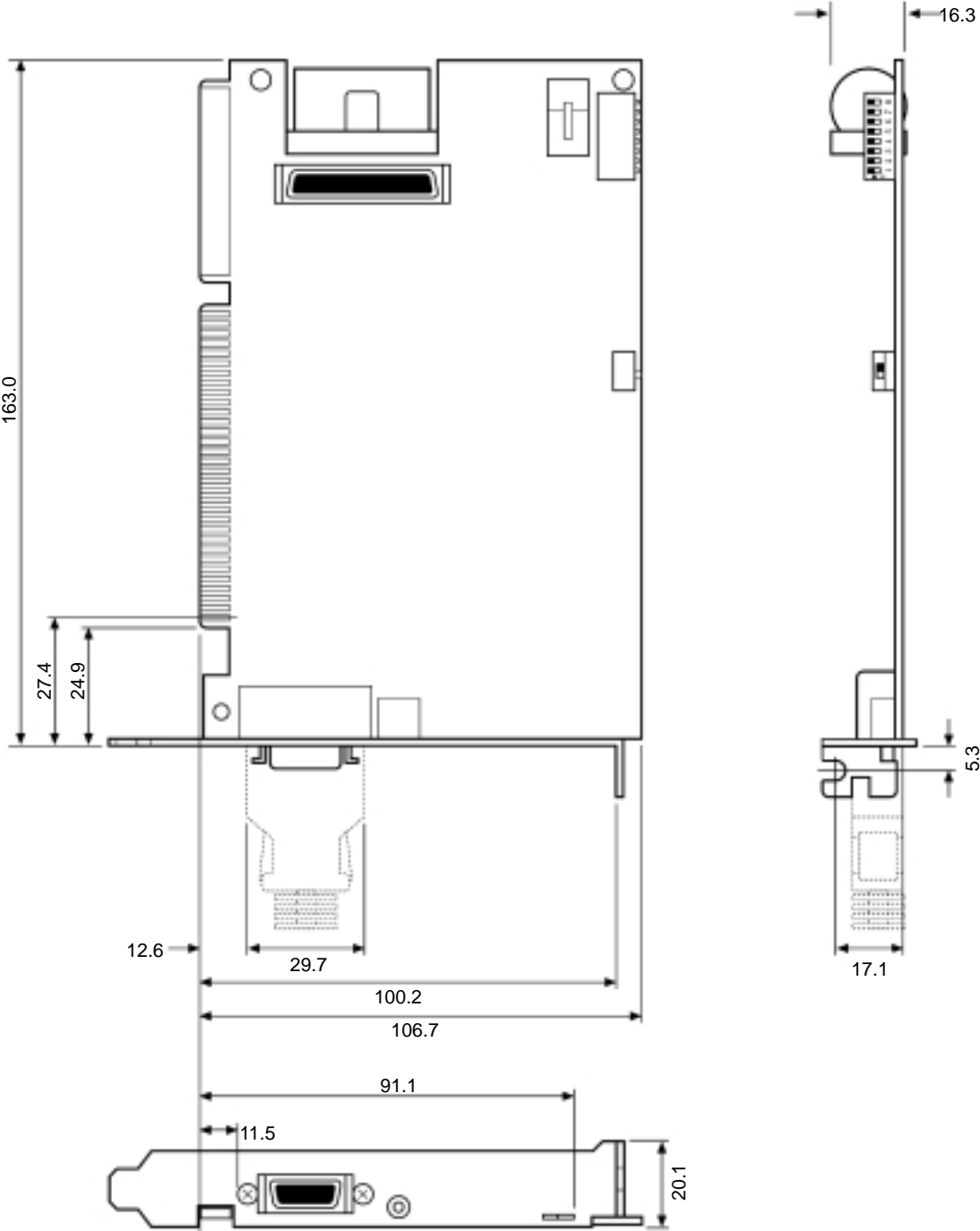
Connected to a Programming Device.

RS-232C Connector

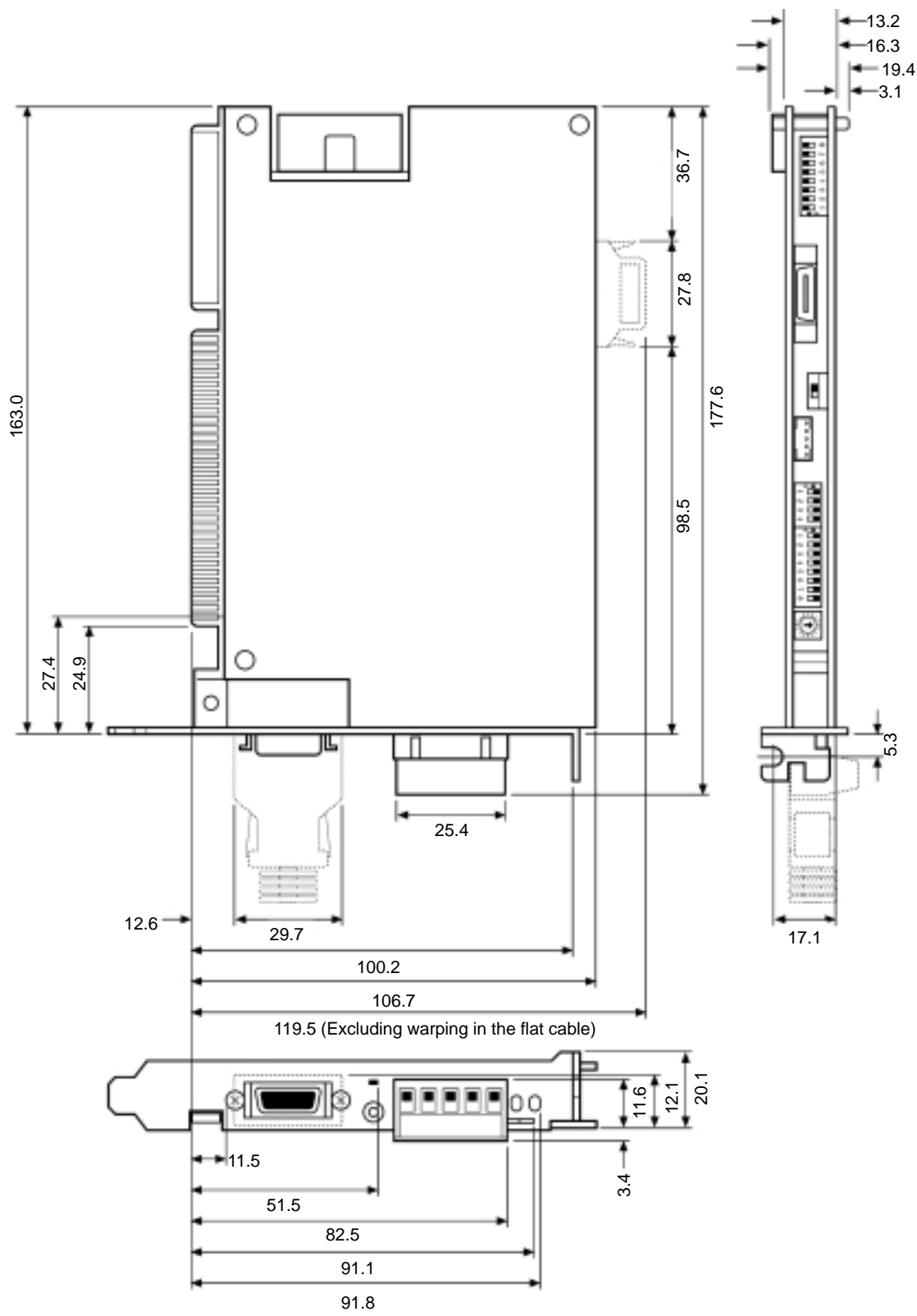
Connected to a Programmable Terminal or other external device with a RS-232C port.

Dimensions (Unit: mm)

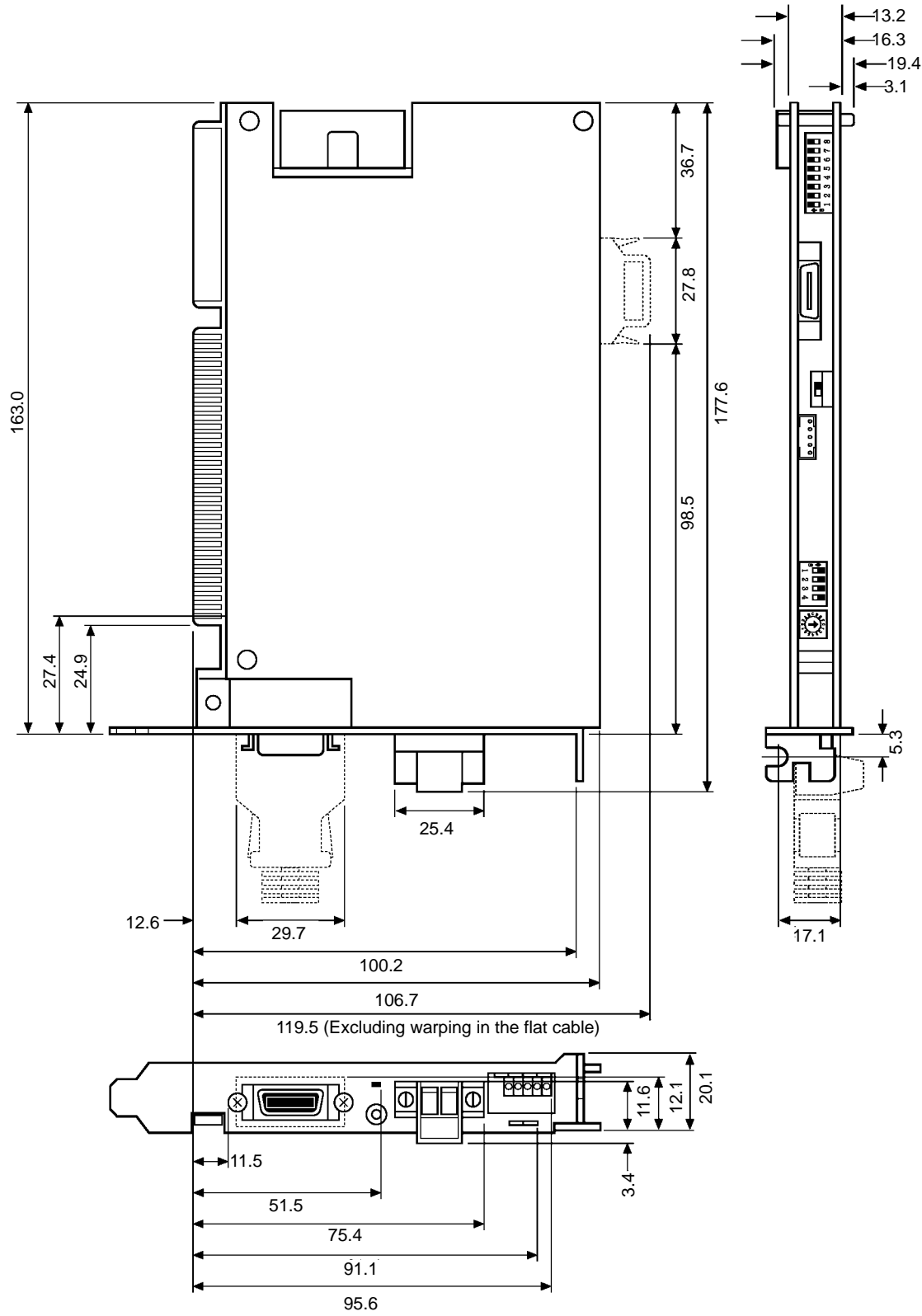
C200PC-ISA01-E SYSMAC Board



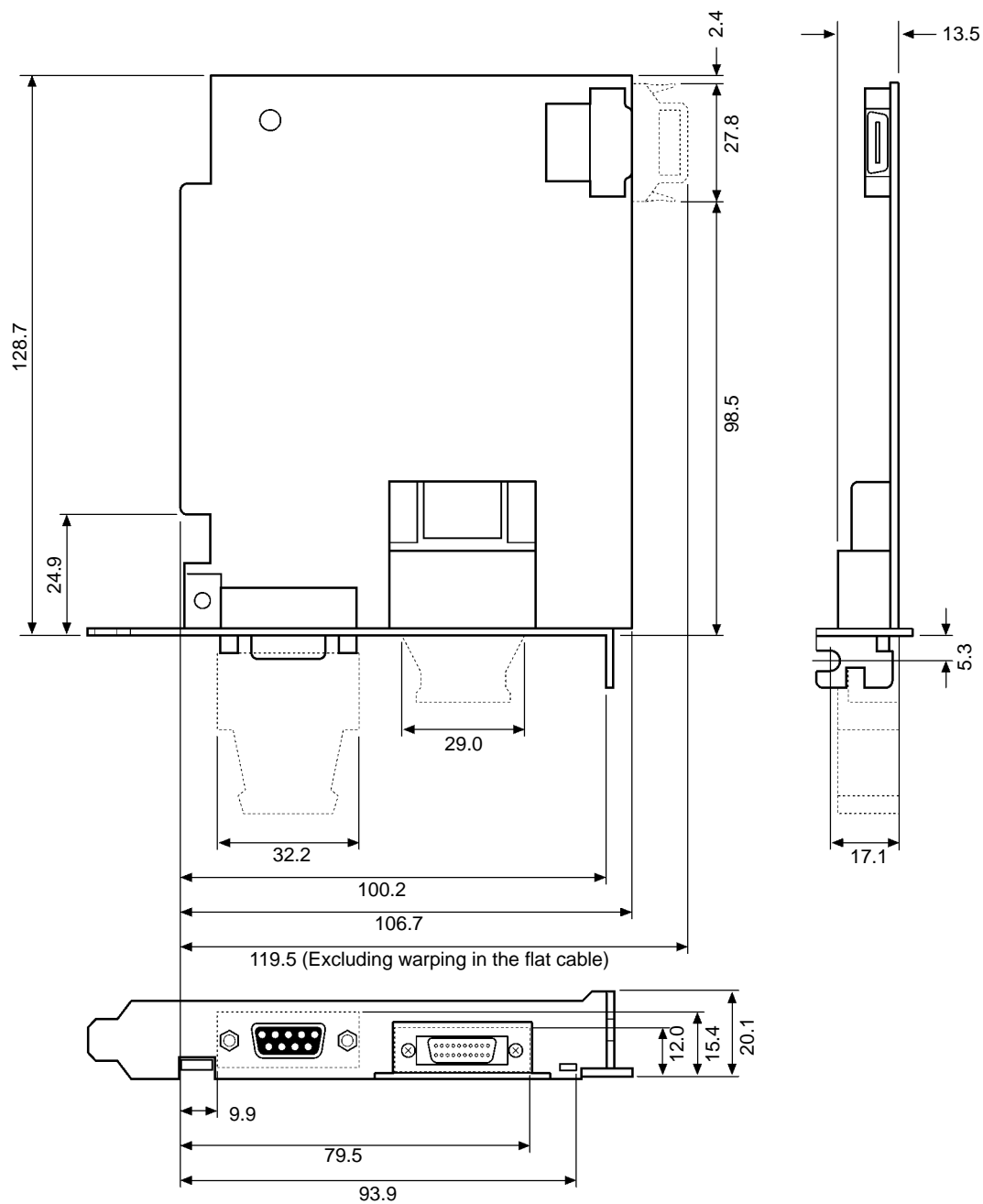
C200PC-ISA□2-DRM-E SYSMAC Board



C200PC-ISA□2-SRM-E



C200PC-EXP01 Expansion Option Board



Indicators

Refer to the *CompoBus/D Operation Manual (W267)* for further details on the NS and MS indicators and refer to the *CompoBus/S Operation Manual (W266)* for further details on the RUN, SD, RD, ERC, IN/OUT, 8, 4, 2, and 1 indicators.

Indicator	Color	Status	Meaning
PLC RUN	Green	ON	The program is being executed normally in either MONITOR or RUN mode.
	---	OFF	One of the following: The SYSMAC Board is in PROGRAM mode. A fatal error has occurred and program execution has stopped. Power is not being supplied to the SYSMAC Board.
PLC ERR/ALARM	Red	ON	A fatal error has occurred, program execution has stopped, and all outputs from Output Units have been turned OFF.
		Flashing	A non-fatal error has occurred (program is still being executed).
	---	OFF	Either the operation is normal or power is not being supplied to the SYSMAC Board.
PLC COMM (*1)	Orange	ON	Communications are being executed through the RS-232C port/peripheral port.
		OFF	Communications through the RS-232C port/peripheral port are stopped.
	Red	ON	A fatal communications error has occurred. A node address duplication or Bus Off error was detected. (These errors make communications impossible.)
CompoBus/D NS (Node status)	Green	ON	Normal Network online status (Communications connection established.)
		Flashing	The Network is normal and online, but a communications connection isn't established.
	Red	ON	A fatal communications error has occurred. A node address duplication or Bus Off error was detected. (These errors make communications impossible.)
		Flashing	A non-fatal communications error (a communications error in a Slave) has occurred.
	---	OFF	A Network error has occurred and is offline or power is not turned on. For example, the Master is the only node in the Network.
CompoBus/D MS (Master status)	Green	ON	The Unit is operating normally.
		Flashing	Switch settings are being read.
	Red	ON	A fatal error (Unit hardware error) has occurred.
		Flashing	A non-fatal error, such as a switch setting error, has occurred.
	---	OFF	The Master Unit's power is OFF, it is resetting, or it is waiting to start initialization.
CompoBus/S RUN	Green	ON	The CompoBus/S is operating normally.
		OFF	Indicates one of the following for the CompoBus/S: Power not turned ON, I/O setting error, CPU standby, or unit number setting error.
CompoBus/S SD	Yellow	ON	Data being transferred.
		OFF	Data transfer stopped.
CompoBus/S RD	Yellow	ON	Data being received.
		OFF	Data reception stopped.
CompoBus/S ERC	Red	ON	One or more Slaves not communicating (communications error).
		OFF	Slaves communicating normally.
CompoBus/S IN/OUT	Red	ON	Error in an Output Slave.
		OFF	Error occurred in the input slave or all the slaves under normal conditions.
CompoBus/S 8421	Red	ON	The number of the Slave with an error is indicated as a binary number. Read lit indicates as "1" and not lit indicators as "0"

Note A PLC COMM indicator is provided both on the SYSMAC Board and on the Expansion Option Board.

2-3 Board Switch Settings

The following settings must be made on the SYSMAC Board. These settings should be made before the SYSMAC Board is installed inside the computer. CompoBus/D settings are only for the C200PC-ISA□2-DRM-E and CompoBus/S settings are only for the C200PC-ISA□2-SRM-E.

Item	Setting	Description	Page
User memory protection	PLC DIP switch, pin 1	Enables or disables writing to the user memory area.	p. 45
EEPROM reading at startup	PLC DIP switch, pin 2	Specifies whether or not to automatically load the contents of EEPROM when the SYSMAC Board is started.	p. 45
Programming Console message display language	PLC DIP switch, pin 3	Keep set to ON (for English).	p. 45
Expansion instructions setting	PLC DIP switch, pin 4	Specifies whether or not expansion instructions are set.	p. 45
RS-232C port parameters	PLC DIP switch, pin 5	Specifies whether or not default communications parameters are to be used.	p. 45
Programming Console mode/Expansion terminal mode selection	PLC DIP switch, pin 6	Keep set to OFF.	p. 45
I/O port address	PLC DIP switch, pin 7, 8	Sets the base address of the I/O port to be used by SYSMAC Board registers.	p. 46
EEPROM protection	EEPROM write protect switch	Enables or disables writing to EEPROM.	p. 46
CompoBus/D node address	CompoBus/D node address DIP switch	Set the node address of the SYSMAC Board's CompoBus/D Master Unit.	p. 46
CompoBus/D unit number	CompoBus/D node number switch	Set the unit number of the SYSMAC Board's CompoBus/D Master Unit.	p. 47
CompoBus/D communications parameters	CompoBus/D communications parameter DIP switch	Sets the baud rate and the handling of communications error for CompoBus/D communications.	p. 47
CompoBus/S unit number	CompoBus/D node number switch	Set the unit number of the SYSMAC Board's CompoBus/S Master Unit.	p. 48
CompoBus/S communications parameters	CompoBus/D communications parameter DIP switch	Sets the baud rate and the handling of communications error for CompoBus/S communications.	p. 49

- Note**
1. The above settings should be made before the SYSMAC Board is installed into the computer. Although you may be able to change the settings after installation, doing so will require that you open up the computer case to access the SYSMAC Board.
 2. Always turn off the power supply to the computer and all peripheral devices connected to the computer before opening the case to change switch settings. Also, turn off the power supply to all CompoBus/D Slaves and the CompoBus/D Network's communications power supply.

2-3-1 PLC DIP Switch (Same for All Boards)

The PLC DIP switch is used to set operating parameters for the SYSMAC Board as a PLC. The I/O port addresses used for communications between the SYSMAC Board and the computer are also set.



(Factory defaults shown in diagram.)

Pin 1: User Memory Write Protect

Pin 1 can be set to either enable or disable writing to the user memory area.

Pin	Setting	Function
1	ON	The UM area cannot be overwritten from a Peripheral Device.
	OFF	The UM area can be overwritten from a Peripheral Device.

Note Factory default: OFF (user memory not write protected).

Pin 2: EEPROM Memory Transfer at Startup

Pin 2 can be turned ON to automatically transfer the contents of the EEPROM to SYSMAC Board memory at startup.

Pin	Setting	Function
2	ON	The contents of the EEPROM will be automatically transferred to the internal RAM at start-up.
	OFF	The contents will not be automatically transferred.

Note Factory default: OFF (EEPROM contents not transferred).

Pin 3: Programming Console Message Display Language

Used to select the language for the display on the Programming Console.

Pin	Setting	Function
3	ON	English display
	OFF	Japanese display

Note Factory default: OFF (EEPROM contents not transferred).

Pin 4: Expansion Instruction Setting

Pin 4 can be turned ON to enable setting function codes for expansion instructions from a Programming Device. If pin 4 is OFF, you will not be able to set the function codes and all expansion instructions will be set to their default function codes.

Pin	Setting	Function
4	ON	Expansion instructions can be set by user.
	OFF	Expansion instructions set to default function codes.

Note

1. Factory default: OFF (default function codes).
2. Pin 4 must be turned OFF and pin 1 must be turned OFF (to enable writing to the user memory) to use function codes allocated to expansion instructions from a Peripheral Device. If the settings of pins 1 and 4 are incorrect, unexpected operation could result from program execution. Check these settings carefully.

Pin 5: RS-232C Communications Parameters

The setting of pin 5 determines if default communications parameters are used or if user-set parameters are used. The communications parameters can be set from a Programming Device.

Pin	Setting	Function
5	ON	Default communications parameters for RS-232C port: Start bits: 1; data length: 7 bits; parity: even; stop bits: 2; baud rate: 9,600 bps
	OFF	User-set communications parameters used.

Note

1. Factory default: OFF (user-set parameters).
2. The following RS-232C port will be used depending on whether the Expansion Option Board is connected or not:
Not Connected: RS-232C port on the SYSMAC Board
Connected: RS-232C port on the Expansion Board (In this case, the RS-232C port on the SYSMAC Board cannot be used.)

Pin 6: Programming Console Mode/Expansion Terminal Mode Selection

The Programming Console supports the following modes: Console Mode for normal setting, monitoring, or programming and Expansion Terminal Mode for

message display and keyboard mapping.

When set to the message display, MSG instruction or LMSG instruction messages are displayed. When set to the keyboard mapping function, pressing a key on the Programming Console will turn ON the corresponding auxiliary memory bit. For further details, refer to the *C200HX/HG/HE-ZE Operation Manual (W322)*.

Pin	Setting	Function
6	ON	The Programming Console set to expansion terminal mode.
	OFF	The Programming Console set to console mode.

Note Factory default: OFF (user-set parameters).

Pins 7, 8: I/O Port Base Address

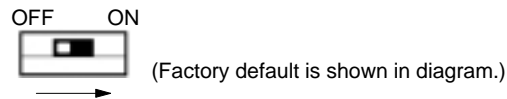
Pins 7 and 8 are used to set the base I/O port address used for SYSMAC Board registers. A total of 5 addresses are used being with the one set on these pins.

Pin 8	Pin 7	First I/O port address
ON	ON	0370
ON	OFF	0380
OFF	ON	0390
OFF	OFF	03A0

Note Factory default: OFF, OFF (03A0).

2-3-2 EEPROM Write Protect Switch (Save for All Boards)

The EEPROM write protect switch is used to enable or disable writing to the EEPROM (equivalent to C200HW-ME16K or C200HW-ME32K Memory Cassette).



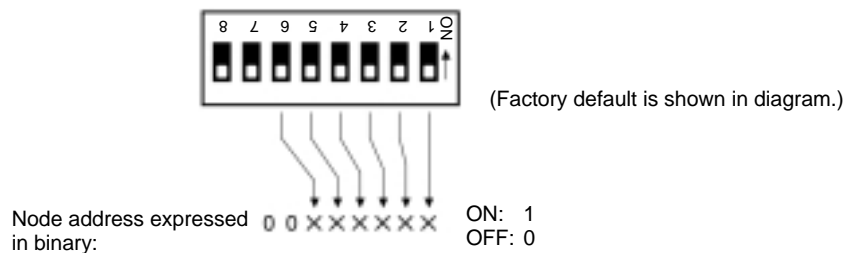
Setting	Function
ON	EEPROM write protected
OFF	EEPROM write enabled

Note Factory default: OFF (EEPROM write enabled).

2-3-3 CompoBus/D Node Address DIP Switch (C200PC-ISA□2-DRM-E)

The node address of the CompoBus/D Master is set using this DIP switch. Node addresses are used to differentiate nodes on the CompoBus/D Network.

Pins 1 through 6 represent binary digits which are 1 when the pin is ON, 0 when the pin is OFF.



The following ranges can be used for the node addresses for Slaves and Masters on the CompoBus/D Network. The same node address cannot be set for more than one node on the same Network. It won't be possible to start up the CompoBus/D Network if the same node address is used for more than one node (node address duplication error).

Condition	Node address range
Configurator not used.	0 to 49
Configurator used.	0 to 63

The Master Unit is shipped with the node address set to 0.

Pins 7 and 8 are reserved for system use. Leave these pins set to OFF (0).

2-3-4 CompoBus/D Unit Number Switch (C200PC-ISA□2-DRM-E)

The unit number of the CompoBus/D Master is set using this rotary switch. Unit numbers are used to differentiate Special I/O Units mounted to the same PLC (including those mounted to the Expansion I/O Racks).



Any unit number shown below can be set as long as it hasn't been set on another Special I/O Unit within the system of the SYSMAC Board (including Expansion I/O Racks).

Model	Unit number range
C200PC-ISA02-DRM-E	0 to 9 (decimal: 0 to 9)
C200PC-ISA12-DRM-E	0 to F (decimal: 0 to 15)

- Note**
1. Factory default: 0
 2. The unit number determines which words in the PLC are allocated for the software switches and status area for the CompoBus/D Master. (Refer to *CompoBus/D Operation Manual (W267)*.)
 3. If the same unit number is used for the more than one Unit, a duplicate number error will occur in the PLC and the SYSMAC Board will not start properly.

2-3-5 CompoBus/D Communications DIP Switch (C200PC-ISA□2-DRM-E)

The CompoBus/D communications DIP switch is used to set the communications baud rate and whether communications will be continued or stopped when a communications error occurs.



(Factory setting shown in diagram.)

Baud rate
Continue/Stop communications for communications error
Reserved (Always OFF.)

- Note** Pin 4 is not used and must always be set to OFF.

Pins 1, 2: Baud Rate

Pins 1 and 2 are used to set the baud rate as shown in the following table.

Pin 1	Pin 2	Baud rate
OFF	OFF	125 kbps
ON	OFF	250 kbps
OFF	ON	500 kbps
ON	ON	Not allowed. (This setting causes an "incorrect switch setting" error.)

- Note**
1. Factory default: OFF, OFF (125 kbps).
 2. Set the same baud rate on all of the nodes (Master and Slaves) in the Network. Any Slaves with baud rates different from the Master's rate won't be able to participate in communications.

Pin 3: Communications Error Handling

Pin 3 is used to set the whether or not communications will stop after a communications error.

Pin 3	Function
OFF	Continue communications.
ON	Stop communications.

If pin 3 is ON, remote I/O communications will be stopped if a transfer error, transmission timeout, or Network power supply error occurs. Remote I/O communications will remain stopped even if the error is cleared until the Clear Communications Error Stoppage Bit is turned ON. (Message communications will continue.) Refer to the *CompoBus/D Operation Manual (W267)* for further details.

If pin 3 is OFF, remote I/O communications will stop if a transmission timeout or Network power supply error occurs, but will restart automatically when the cause of the error is cleared.

2-3-6 CompoBus/S Unit Number Switch (C200PC-ISA□2-SRM-E)

The unit number of the CompoBus/S Master is set using this rotary switch. Unit numbers are used to differentiate Special I/O Units mounted to the same PLC (including those mounted to the Expansion I/O Racks).



The setting range for the unit number is as shown below depending on SYSMAC Board model and the connectable node address described in the next section. Any unit number shown in the table can be set as long as it hasn't been set on another Special I/O Unit within the system of the SYSMAC Board (including Expansion I/O Racks).

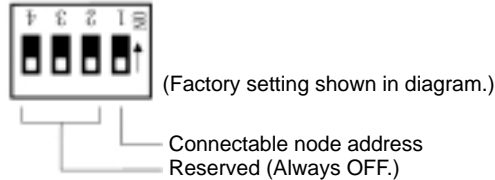
Model	Connectable node address	Unit number setting range
C200PC-ISA02-SRM	IN 0 to 7/OUT 0 to 7	0 to 9 (decimal: 0 to 9)
	IN 0 to 15/OUT 0 to 15	0 to 8 (decimal: 0 to 8)
C200PC-ISA12-SRM	IN 0 to 7/OUT 0 to 7	0 to 9 (decimal: 0 to 9) A to F (decimal: 0 to 15)
	IN 0 to 15/OUT 0 to 15	0 to 8 (decimal: 0 to 8) A to E (decimal: 10 to 14)

When setting the number, carefully use a small Philips screwdriver and do not damage the rotary switches.

- Note**
1. Factory default: 0
 2. The unit number determines which words in the Special I/O Unit Area are allocated to the CompoBus/S Master. (Refer to *CompoBus/S Operation Manual (W266)*).
 3. If the connectable node address is set to IN0 to 15/OUT0 to 15, the Special I/O Unit Area words for two Units will be allocated and the the number next to the set unit number will also be used by the CompoBus/S Master. Be careful not to use the second unit number for another Special I/O Unit.
 4. If the same unit number is used for the more than one Unit, a duplicate number error will occur in the PLC and the SYSMAC Board will not start properly.

2-3-7 CompoBus/S Communications DIP Switch (C200PC-ISA□2-DRM-E)

The CompoBus/S communications DIP switch is used to set the communications baud rate and whether communications will be continued or stopped when a communications error occurs.



Set pin 1 on the DIP switch as shown below.

SW1	Connectable node address setting
ON	IN0 to 15/OUT0 to 15 (Total of inputs and outputs, 32 Slave Units max.)
OFF	IN0 to 7/OUT0 to 7 (Total of inputs and outputs, 16 Slave Units max.)

Note Factory default: OFF

- Note**
1. Always leave pins 2 to 4 set to OFF.
 2. When setting the switch with the SYSMAC Board connected to a personal computer, be sure to turn OFF the power supply to the personal computer or peripheral devices. If Slave Units are connected, be sure to turn OFF the power supply to the Slave Units and the communications power supply.
 3. Pin 1 is factory set to OFF (IN0 to 7/OUT0 to 7).
 4. The connectable node address setting determines a total number of words allocated to the CompoBus/S Master in the Special I/O Unit Area and also the communications cycle with the Slave Units. (Refer to *CompoBus/S Operation Manual (W266)*.)
 5. If the connectable node address is set to IN0 to 15/OUT0 to 15, the Special I/O Unit Area words for two Units will be allocated and the the number next to the set unit number will also be used by the CompoBus/S Master. Be careful not to use next unit number for another Special I/O Unit.

2-4 Support Board Installation

Once the SYSMAC Board's switches have been set, it can be installed in an expansion slot of the computer. The following procedure is a generalize procedure to install the SYSMAC Board in an IBM PC/AT or compatible computer. Refer to the user's manual for your computer and install the SYSMAC Board correctly.

⚠ WARNING Never touch the Board or the interior of any other Unit or attempt to disassemble a Unit while power is being supplied. Doing so may result in serious electrical shock or electrocution.

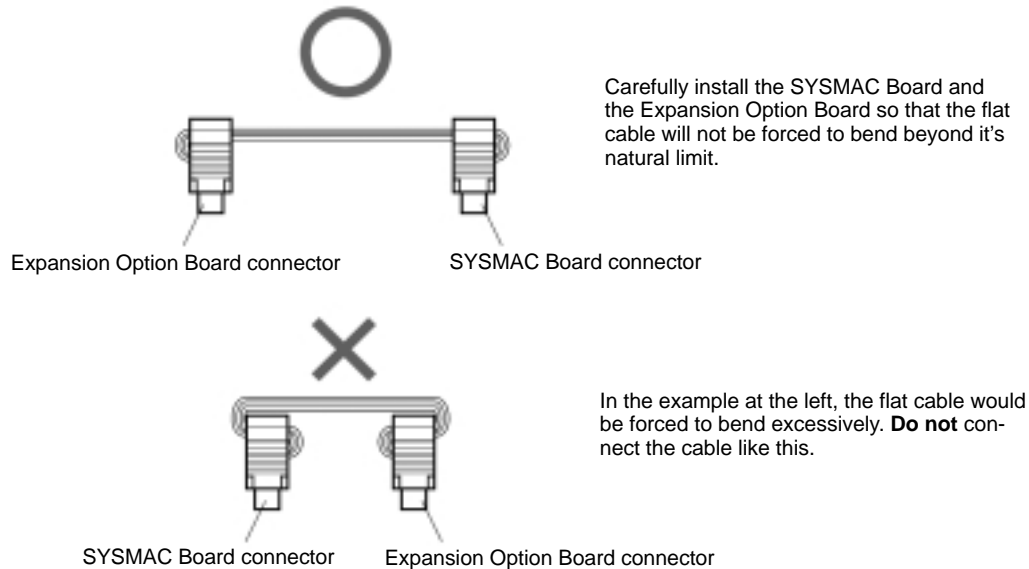
Mounting Restrictions

Only one SYSMAC Board can be installed in each computer.

The SYSMAC Board must be installed in an ISA bus connector. The ISA bus connector should match the connector on the SYSMAC Board.

When using the Expansion Option Board, carefully position the Units so that no excessive force is applied to the special flat cable connected to the SYSMAC Board.

Cable Viewed from the Side



When using the Expansion Option Board, carefully position the Units so that the flat cable can be easily connected.

Mounting Procedure

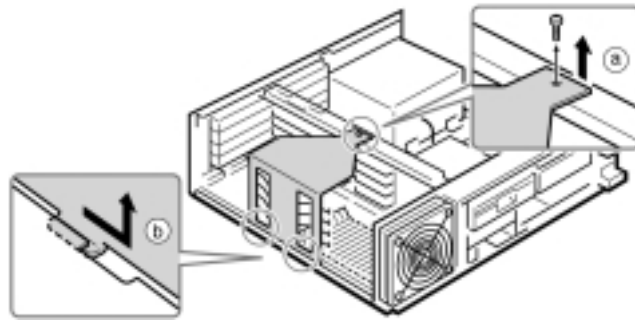
The following is a generalize mounting procedure. Refer to your computer's user's manual for details and differences.

- Note**
1. Turn the power off to the computer and all peripheral devices before starting to install the SYSMAC Board.
 2. Be extremely careful of static electricity when installing the SYSMAC Board. Static electricity can destroy the SYSMAC Board or your computer.
 3. Be extremely careful not to damage any other components inside your computer.

- 1, 2, 3...**
1. Remove the computer's cover by unscrewing the mounting screws and lifting it off.

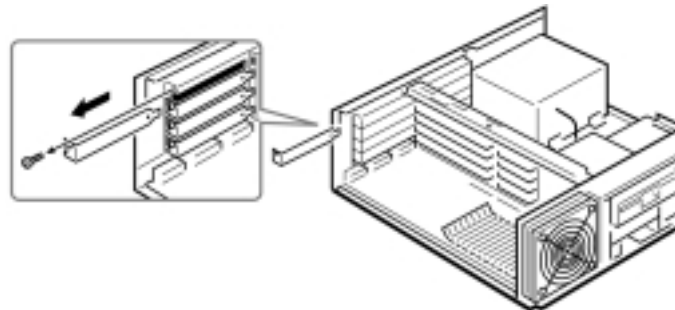


2. Remove the slot cover. Set aside the screw(s) that held the slot cover; it will be used to hold the SYSMAC Board. Store the slot cover; it may be needed sometime in the future.

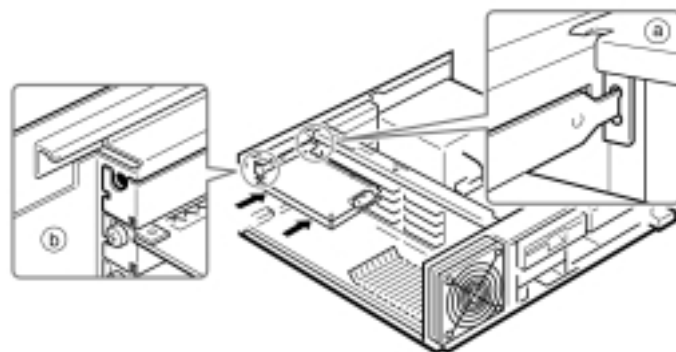


3. Install the SYSMAC Board in the computer. When installing the Board, be sure to insert the connector into the slot in the proper direction.

Insert the Board until the metal mounting bracket fits into the computer, and is flush with the edge of the computer.

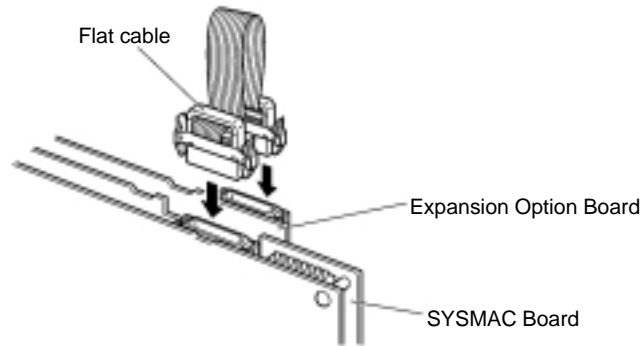


4. Replace the screw(s) that was set aside in step 2 to secure the Board to the computer.



5. Replace the computer's cover and screw in the mounting screws.
6. When using the Expansion Option Board, mount the Expansion Option Board before mounting the SYSMAC Board. Connect the Expansion Option Board to the SYSMAC Board using the flat cable attached to the Expansion

Option Board. Push the flat cable connector into the Board connector until it clicks. (The levers on the connector sides will open.)



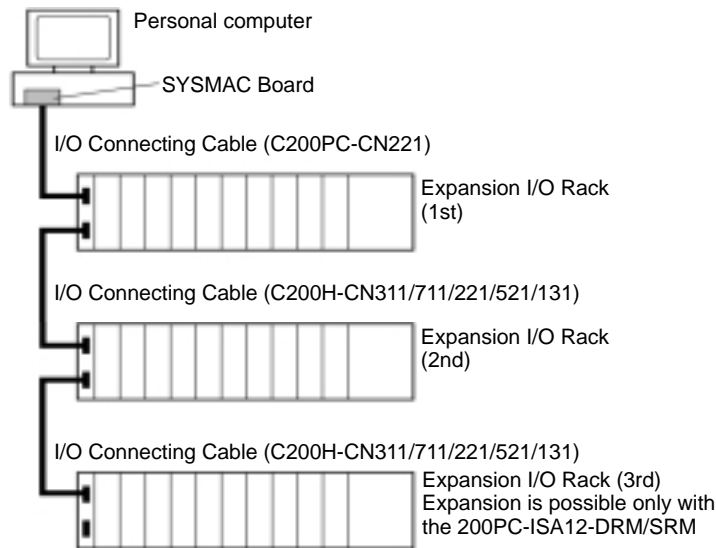
2-5 External Connections

This section describes how to connect Expansion I/O Racks, CompoBus/D Slaves, CompoBus/S Slaves, and Programming Devices to the SYSMAC Board.

- Note**
1. Turn the power supply OFF to the PCs and all Slaves, and turn OFF the communications power supply before connecting communications cables.
 2. Tighten the screws holding the cables to the CompoBus/D and CompoBus/S connectors to a torque of 0.5 N•m and the screws holding the CompoBus/S connectors to a torque of 0.2 N•m.
 3. The tensile strength of the CompoBus/S connector is 50 N. Do not pull the connector with a force exceeding 50 N.
 4. Be sure to confirm that the terminal blocks, expansion cables, and other items with locking devices are properly locked into place.
 5. Do not pull on the communications cables.
 6. Do not forcibly bend the communications cables.
 7. Do not place heavy objects on the communications cables.
 8. Always use crimp terminals when connecting communications cables. Connecting bare twisted wires can result in disconnection, broken wires, faulty operation, or damage to the product.
 9. Use an appropriate crimp tool and appropriate methods to attach the crimp terminals. Consult the manufacturer of the crimp tool for specific instructions. Inappropriate crimp tools or crimping methods can result in broken wires.
 10. Be sure to connect the signal lines, power lines, and shield lines correctly and do not get them mixed up.
 11. Be sure that the signal lines, power lines, and shield line do not become disconnected during communications.
 12. Double-check all wiring before turning the power supplies back ON.

2-5-1 C200HX/HG/HE Expansion I/O Racks

C200HX/HG/HE Expansion I/O Racks are connected as shown in the following diagram.



Up to three Expansion I/O Racks can be connected. If only one or two Expansion I/O Racks are connected, the lower left connector on the last Expansion I/O Rack must be covered with a protective cap.

- Note**
1. Always turn off the power supply to the computer, all peripheral devices, and the Expansion I/O Racks when connecting or disconnecting the I/O Connecting Cable.
 2. The total length of I/O Connecting Cable must be 12 m max.
 3. Be careful not to connect the wrong connectors.
 4. Always secure the I/O Connecting Cable connectors with the lock screws.

Precautions

The following precautions must be followed when connecting Expansion I/O Racks.

- Do not place the I/O Connecting Cables in the same ducts as I/O line and power supply lines.
- Connect the I/O Connecting Cables properly into the connectors of the Expansion I/O Rack or SYSMAC Board and secure them with the lock screw. If an I/O Connecting Cable disconnects, an I/O bus error will occur and operation will stop.
- A hole diameter of 53 mm is normally required to pass the I/O Connecting Cable through. This can be reduced to 33 mm by removing the connector cover, but be sure to replace the cover properly and tighten the cover screws.
- The tensile strength of the cable is 50 N. Do not pull or allow the cables to be pulled at a force greater than 50 N.

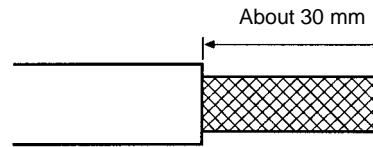
2-5-2 Connecting CompoBus/D Communications Cables (C200PC-ISA□2-DRM-E)

The CompoBus/D communications cables can be installed as soon as the SYSMAC Board is installed in the computer. This manual provides information only on wiring the communications cables to the SYSMAC Board. Refer to the *CompoBus/D Operation Manual (W267)* for details on wiring communications cables and Slaves.

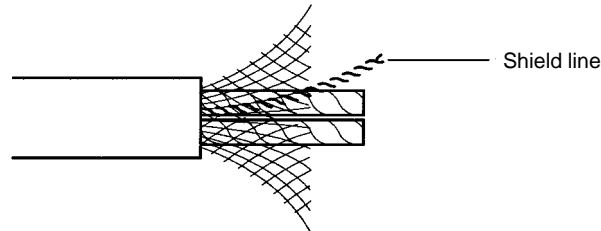
Preparing Cables

This section explains how to prepare and connect the communications cables to connectors for the CompoBus/D Network. Use the following procedure to prepare and connect the communications cables to the connectors.

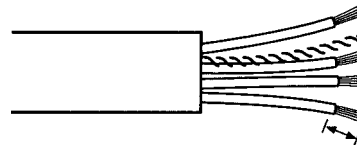
- 1, 2, 3...** 1. Remove about 30 mm of the cable covering being careful not to damage the shield weaving underneath. Do not remove more than about 30 mm; removing too much of the covering can result in short-circuits.



2. Carefully peel back the weaving. You'll find the signal lines, the power lines, and the shield line. The shield line will be loose on the outside of the other lines, but it is harder than the weaving and should be easily identified.



3. Remove the exposed weaving, remove the aluminum tape from the signal lines and the power lines, and strip the covering from the signal lines and power lines to the proper length for the crimp terminal connectors. Twist together the wires of each of the signal lines and each of the power lines.



Strip to match the crimp terminals

4. Connect the crimp terminal to the lines and then cover any exposed areas of the cable and lines with electricians tape or heat-shrinking tubes.

We recommend the following crimp terminals.

Phoenix Contact K.K., AI-series Crimp Terminals



Insert the line into the terminal and then crimp.

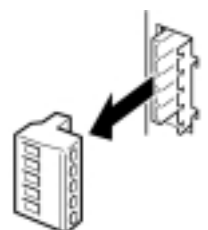
The following crimp tool is also available.

Phoenix Contact K.K., ZA3 Crimp Tool

Connecting Communications Cables

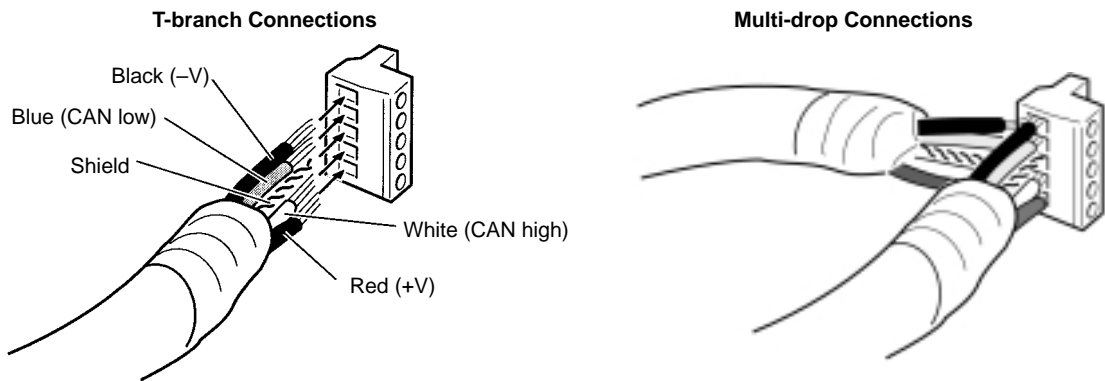
Turn off the power supply to the computer and Slaves, and turn off the communications power supply before connecting CompoBus/D Communications Cables.

- 1, 2, 3...** 1. Remove the connector from the CompoBus/D communications connector on the SYSMAC Board.



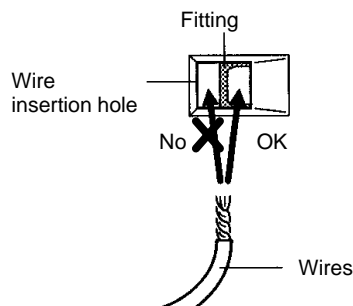
Note The connector can be wired even when connected to the SYSMAC Board as long as it is accessible.

2. Orient the connector properly, loosen the line lock screws, and then insert the lines into the holes in order: Black, blue, shield, white, and then red.



For multi-drop connections with Thin Cables, two lines are inserted into each hole, as shown in the diagram on the right above. Be sure to place crimp terminals on each of the lines. (Do not use a multi-drop connection with Thick Cables; use a T-branch Connector.)

Note Be sure the line lock screws are sufficiently loosened before attempting to insert the lines. If these screws are not loose, the lines will enter the gaps in the back of the connector and will not lock properly.

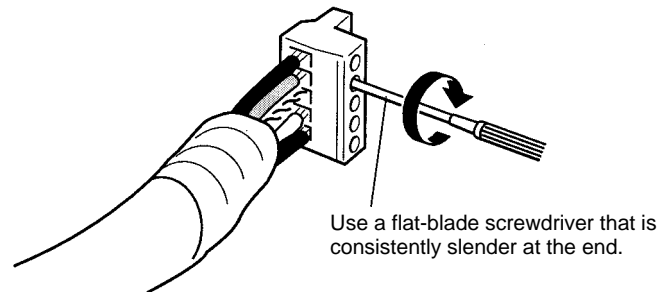


The cable colors are as follows:

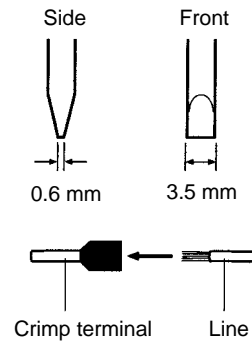
Color	Signal
Black	Power line, negative voltage (-V)
Blue	Communications line, low (CAN low)
---	Shield
White	Communications line, high (CAN high)
Red	Power line, positive voltage (+V)

3. Tighten the line lock screws for each line in the connector.
- You will not be able to tighten these screws with a normal screwdriver, which narrows to a point at the end. You will need a screwdriver that is consistently

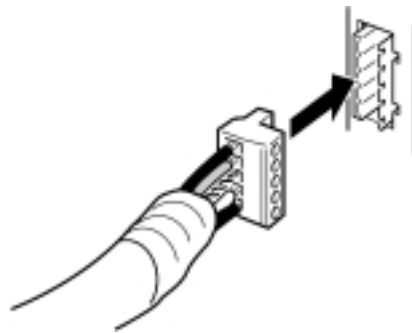
thin for the entire length. The line lock screws must be tightened to a torque of 0.5 N·m.



The following OMRON XW4Z-00C Screwdriver is available to tighten the line lock screws. The shape of this Screwdriver is shown in the following illustration.



4. Connect the connector back to the SYSMAC Board, being sure to orient the connector in the correct direction. Press the connector all the way to the back.



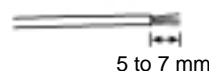
2-5-3 Connecting CompoBus/S Communications Cables (C200PC-ISA□2-SRM-E)

The CompoBus/S communications cables can be installed as soon as the SYSMAC Board is installed in the computer. This manual provides information only on wiring the communications cables to the SYSMAC Board. Refer to the *CompoBus/S Operation Manual* (W266) for details on wiring communications cables and Slaves.

Preparing Cables

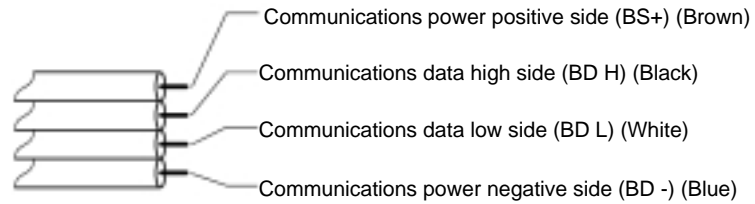
Use the following procedure to prepare and connect the communications cables to the connectors.

- 1, 2, 3... 1. Peel off the sheath of the two signal lines by 5 to 7 mm. Twist the exposed signal lines.

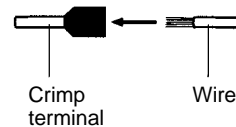


When using commercially available VCTF cables, determine which signal lines to use for the BD H (high side) and BD L (low side).

Special flat cable consists of the following signal lines.



We recommend the following crimp terminals.
Phoenix Contact K.K., AI-series Crimp Terminals



Insert the line into the terminal and then crimp.

The following crimp tool is also available.

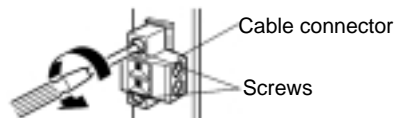
Phoenix Contact K.K., ZA3 Crimp Tool

2. When using the special flat cable, insulate the two signal lines for the communications power supply.

Connecting Communications Cables

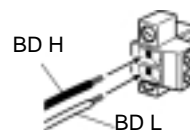
Turn off the power supply to the computer and Slaves, and turn off the communications power supply before connecting CompoBus/S Communications Cables. Do not touch the communications connectors while the power is being supplied in order to prevent any malfunction or breakdown due to static electricity.

- 1, 2, 3... 1. Loosen the two screws fixing the CompoBus/S communications connectors to the SYSMAC Board and then remove the cable connector.

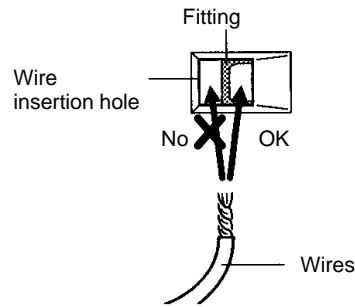


Note If the Communications Cable can be connected with the screws on, there is no need to remove the screws.

2. Loosen the signal line lock screws on the cable side connector and then insert the two signal lines into each hole while paying attention to the orientation of the terminal block.



Note Be sure the line lock screws are sufficiently loosened before attempting to insert the lines. If these screws are not loose, the lines will enter the gaps in the back of the connector and will not lock properly.

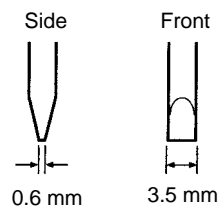


3. Tighten the line lock screws for each line in the connector.

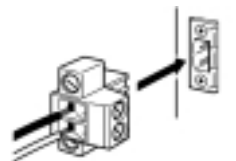
You will not be able to tighten these screws with a normal screwdriver, which narrows to a point at the end. You will need a screwdriver that is consistently thin for the entire length. The line lock screws must be tightened to a torque of 0.5 N·m.



The following OMRON XW4Z-00C Screwdriver is available to tighten the line lock screws. The shape of this Screwdriver is shown in the following illustration.



4. Connect the connector back to the SYSMAC Board, being sure to orient the connector in the correct direction. Press the connector all the way to the back.



5. Tighten the two screws that were loosened in step 1, above, using a flat-blade screwdriver to a torque of 0.2 N·m.

Note The tensile strength of the CompoBus/S is 50 N. Do not pull the connector with a force exceeding 50 N.

2-5-4 Connecting the SYSMAC Board's RS-232C Port (With No Expansion Option Board)

A Programming Device is connected to the RS-232C connector on the SYSMAC Board to adjust PLC settings and program/monitor the PLC.

Note When the Expansion Option Board is connected, the RS-232C port on the SYSMAC Board cannot be used.

The following Connecting Cable must be used to connect the Programming Device to the SYSMAC Board.

Model	Specification
ES100-CT023-202	9-pin connector for IBM PC/AT or compatible

For pin arrangements of the above cable, refer to *1-10-6 RS-232C Port Cable Specifications*. Refer to the *SYSMAC-CPT Controller Programming Tool User Manual* (W333) or the *SYSMAC Support Software Operation Manual: Basics* (W247) for connection methods.

Note Be sure to double-check the wiring before using the connection cables.

2-5-5 Connecting the Expansion Option Board's RS-232C Port and Peripheral Port

By connecting the C200PC-EXP01 Expansion Option Board to the SYSMAC Board, the same connectors and cables can be used for the RS-232C port or peripheral port as on the C200HX/HG/HE.

For further details, refer to the *C200HX/HG/HE (-E/-ZE) Installation Guide* (W302) or *C200HX/HG/HE-ZE Operation Manual* (W322).

- Note**
1. Be sure to double-check the wiring before using the connection cables.
 2. When connecting cables to the peripheral port, be careful not to break the connector pins.

2-6 Required Files

Once the hardware setup has been completed, the PLC communications library files must be copied to the hard disk. The following files are provided on the floppy disk. It is necessary to copy only the files required for the user applications.

- Compact models: CLIBC.LIB
- Small models: CLIBS.LIB
- Medium models: CLIBL.LIB
- Large models: CLIBM.LIB
- Huge models: CLIBH.LIB

The files can be copied to any directory, but it is best to copy them to a directory that is easily linked with the user applications.

- Note**
1. The PLC communications library uses Microsoft C/C++ Ver. 7.
 2. A software driver is not required for the SYSMAC Board. It can be used by writing a user application with the PLC communications library functions and then linking the application to the PLC communications library files.
 3. For applications on Windows 95 or Windows NT, purchase the FinsGateway for Windows 95 or the FinsGateway for Windows NT separately.

SECTION 3

SYSMAC Board Communications Library

This section describes the PLC communications library functions used to write user applications on the computer to communicate with the SYSMAC Board.

3-1	Using the PLC Communications Library	62
3-2	Library Functions	64
	pcinit() CHANGE SETTINGS	64
	pcopen() PLC OPEN	65
	pcclose() PLC CLOSE	66
	pcmode() PLC MODE	66
	pcstat() PLC STATUS	68
	pcread() PLC READ	76
	pcwrite() PLC WRITE	82

3-1 Using the PLC Communications Library

This section provides an overview of the PLC communications library and how to use it.

The PLC communications library contains C functions used to control the SYSMAC Board from user applications running on the computer. The library file depends on the program model. Be sure the proper file is available to the user application. (It is necessary to copy only the files required for the user applications.)

- Compact models: CLIBC.LIB
- Small models: CLIBS.LIB
- Medium models: CLIBL.LIB
- Large models: CLIBM.LIB
- Huge models: CLIBH.LIB

- Note**
1. The PLC communications library uses Microsoft C/C++ Ver. 7.
 2. When using the PLC communications library attached to the SYSMAC Board, execute the application at the MS-DOS prompt or in the DOS mode.
 3. The PLC communications library attached to the SYSMAC Board cannot be used on Windows NT.
 4. Driver libraries for the Windows 95 or Windows NT (FinsGateway for Windows 95 and FinsGateway for Windows NT) are separately sold.

PLC Communications Library

The following functions are available in the PLC communications library.

Name	Operation	Page
pcinit()	Sets the I/O port address and base address in common memory to use for communications between the SYSMAC Board and computer.	p. 64
pcopen()	Initializes communications for the SYSMAC Board and starts communications between the SYSMAC Board and the computer.	p. 65
pcclose()	Ends communications between the SYSMAC Board and the computer.	p. 66
pcmode()	Changes the SYSMAC Board operating mode (PROGRAM, MONITOR, RUN).	p. 66
pcstat()	Accesses and controls SYSMAC Board status, including the following: Reading status Clearing errors Reading the error history Clearing the error history	p. 68
pcread()	Reads data from a SYSMAC Board memory area.	p. 76
pcwrite()	Writes data into a SYSMAC Board memory area.	p. 82

- Note** The source files for the sample programs used in this sections are provided in the SAMPLE directory on the floppy disk as shown below.

Source Files

PCMODE.C
PCOPEN.C
PCREAD.C
PCSTAT_1.C
PCSTAT_2.C
PCSTAT_3.C
PCSTAT_4.C
PCWRITE.C

Microsoft C/C++ 7.0 Make Files

PCMODE.MAK
PCOPEN.MAK
PCREAD.MAK
PCSTAT_1.MAK
PCSTAT_2.MAK
PCSTAT_3.MAK
PCSTAT_4.MAK
PCWRITE.MAK

Using the Library

The following procedure is required to use the PLC communications library.

- 1, 2, 3...**
1. Use the functions in the PLC communications library to write a user application to control the SYSMAC Board.

2. Link the proper PLC communication library to the user program and then compile the program to create the executable file. The following program lines show an example where the user program source file SAMPLE.C is used to create the executable file SAMPLE.EXE (small model).

```
CL/C /AS SAMPLE.C
```

Creates the object.

```
LINK SAMPLE.OBJ, SAMPLE.EXE, NUL, CSLIBS.LIB
```

Links the library.

3-2 Library Functions

This section provides details on the PLC communications library functions.

Notation

The following headings are used to provide information on the PLC communications library functions.

Purpose	Gives an outline of the purpose of the function.
Format	Shows how the function is formatted in a program.
Parameters	Describes the parameters used in the function
Description	Provides details on the operation of the function.
Return Values	Describes the values returned by the function.
Example	Provides an example of programming with the function.

pcinit()

CHANGE SETTINGS

Purpose	Sets the base addresses of the I/O port and common memory to use for communications with the SYSMAC Board.		
Format	<pre>unsigned int pcinit(port_addr,dpram_addr); unsigned char port_addr; unsigned char dpram_addr;</pre>		
Parameters	port_addr	I/O port base address	
	dpram_addr	Common memory base address	

Parameter	Type	IN/OUT	Setting
port_adr	Integer	IN	One of the following. 1: 0370 _H 2: 0380 _H 3: 0390 _H 4: 03A0 _H
dpram_addr	Integer	IN	One of the following. 1: C8000 _H 2: CA000 _H 3: CC000 _H 4: CE000 _H 5: D0000 _H 6: D2000 _H 7: D4000 _H 8: D6000 _H 9: D8000 _H 10: DA000 _H 11: DC000 _H 12: DE000 _H 13: E0000 _H 14: E2000 _H 15: E4000 _H

Description `pcinit` is used to set the base addresses for the common memory and I/O port used for communications with the SYSMAC Board using PLC communications library functions.

The following defaults will be used when `pcinit` is not used.

Common memory base address: C8000_H

I/O port base address: 03A0_H

The same I/O port base address must be used as the one set on the SYSMAC Board's DIP switch. If the default DIP switch setting is not used, `pcinit` must be executed to set the same address as the one set on the DIP switch.

`pcinit` cannot be executed while the PLC communications library is open. The base addresses that are set will remain valid until reset using `pcinit` or until power to the SYSMAC Board is turned off or the SYSMAC Board is reset.

The base addresses are set in the SYSMAC Board when `pcopen` is executed.

Return Values

Value	Meaning
0	Ended normally.
5	Illegal parameter.
9	Communications already open; cannot be executed.

Example

Refer to the example for `pcopen`.

pcopen ()**PLC OPEN****Purpose**

Opens communications with the SYSMAC Board.

Format

```
unsigned int pcopen();
```

Parameters

None.

Description

`pcopen` initializes communications with the SYSMAC Board. `pcopen` must be executed before any other PLC communications library function can be used, with the exception of `pcinit` and `pcopen` itself.

`pcclose` is used to end communications with the SYSMAC Board. Once communications have been closed with `pcclose`, they can then be reopened with `pcopen`.

The following I/O port and common memory base addresses will be used if `pcinit` has not been executed to change them.

Common memory base address: C8000_H

I/O port base address: 03A0_H

Return Values

Value	Meaning
0	Ended normally.
2	Communications already open; cannot be executed.
10	Communications timeout. A communications timeout will occur when SYSMAC Board initialization has not been completed within 10 s. Check the settings made with <code>pcinit</code> .

Example

```

/*****
/* Opening Communications */
/*****
extern unsigned int pcinit();
extern unsigned int pcopen();
extern unsigned int pcclose();

void main(void)
{
    int ret ;
    unsigned char port_addr=1 ; /* I/O port base address = 0370 */
    unsigned char dpram_addr=7 ; /* DPRAM base address = D4000 */

    ret = pcinit ( port_addr, dpram_addr ) ; /* SYSMAC Board addresses set */
    if ( ret != 0 ) {
        printf ("PC initialization error: %d\n", ret ) ;
        exit () ;
    }

    printf ("Opening communications\n");

    switch( pcopen() ) {
        /* Communications opened */

```



```

        case 0:
            printf("Opened communications\n");
            break;
        case 2:
            printf("Already open\n");
            break;
        case 10:
            printf("PLC communications timed out\n");
            break;
    }

    pcclose(); /* Communications closed */
}

```

pcclose()**PLC CLOSE**

Purpose Ends communications with the SYSMAC Board.

Format unsigned int pcclose();

Parameters None

Description pcclose is used to end communications with the SYSMAC Board. Communications will not be possible after pcclose is executed. Always execute pcclose before turning off or resetting the computer. Communications are opened with pccopen. Once communications have been closed with pcclose, they can then be reopened with pccopen.

Return Values

Value	Meaning
0	Ended normally.
1	Communications have not been opened; execution not possible.

Example Refer to the example for pccopen.

pcmode()**PLC MODE**

Purpose Changes the operating mode of the SYSMAC Board.

Note Change the operating mode of the SYSMAC Board only after confirming that no adverse effects will result.

Format unsigned int pcmode(ne, no, mode);
 unsigned char ne;
 unsigned char no;
 unsigned char mode;

Parameters ne Remote network address. This parameter is not used; set it to 0.
 no Remote node address. This parameter is not used; set it to 0.
 mode SYSMAC Board operating mode.

Parameter	Type	IN/OUT	Setting
ne	Integer	IN	Not used; set to 0.
no	Integer	IN	Not used; set to 0.
mode	Integer	IN	One of the following. 0: PROGRAM mode 2: MONITOR mode 3: RUN mode

Description pcmode is used to change the operating mode of the SYSMAC Board. Always use pcmode before turning off or resetting the computer to set the SYSMAC Board to PROGRAM mode. If the SYSMAC Board is not in PROGRAM mode when the computer is turned off or reset, the data in SYSMAC Board memory

that should be maintained while power is off (HR, AR, Counter, DM, and EM Areas) may not be saved correctly.

Communications must be opened with the SYSMAC Board before `pcmode` can be executed.

Return Values

Value	Meaning
0	Ended normally.
1	Communications have not been opened; execution not possible
5	Illegal parameter.
7	Ended in an error (response error).
10	Communications timeout (no response within 10 s).

Example

```

/*****
/* Changing Operating Mode */
*****/
extern unsigned int  pcinit() ;
extern unsigned int  pccopen() ;
extern unsigned int  pcclose() ;
extern unsigned int  pcmode() ;

void main(void)
{
    int  ret ;
    unsigned char  ne=0 ;
    unsigned char  no=0 ;
    unsigned int  mode=0 ;
    unsigned char  port_addr=1 ;
    unsigned char  dpram_addr=7 ;

    /* Return value */
    /* Network address */
    /* Node address */
    /* Mode */
    /* I/O port base address = 0370 */
    /* DPRAM base address = D4000 */

    ret = pcinit ( port_addr, dpram_addr ) ; /* SYSMAC Board addresses set */
    if ( ret != 0 ) {
        printf ("PLC setting error (Return value: %d) \n", ret ) ;
        exit () ;
    }
    ret = pccopen () ;
    /* Library opened */
    if ( ret != 0 ) {
        printf ("Error opening library (Return value: %d) \n", ret ) ;
        exit () ;
    }

    printf ("PLC operating mode.\n") ;
    printf ("0: PROGRAM\n") ;
    printf ("2: MONITOR\n") ;
    printf ("3: RUN\n") ;
    printf ("Specify a mode: ") ;
    scanf ("%d", &mode) ;

    switch( pcmode ( ne, no, mode ) ) {
        case 0:
            printf ("Mode changed successfully\n") ;
            break ;
        case 1:
            printf ("Communications not open.\n") ;
            break ;
        case 5:
            printf ("Illegal parameter\n") ;
            break ;
        case 7:
            printf ("Error\n") ;
            break ;
        case 10:
            printf ("PLC communications timed out\n") ;
            break ;
    }
}

```

```

    }
    pcclose(); /* Library closed */
}

```

pcstat()

PLC STATUS

Purpose

Performs the following:

- Reads status
- Clears errors
- Reads error history
- Clears error history

Format

```

unsigned int pcstat(ne,no,mcmd,ch,val);
unsigned char ne;
unsigned char no;
unsigned char mcmd;
unsigned char far *ch;
unsigned int far *val;
or
unsigned char far *val;

```

Parameters

ne Remote network address. This parameter is not used; set it to 0.
no Remote node address. This parameter is not used; set it to 0.
mcmd Process to perform
ch No. of records to read from error history.
val Data buffer: contents depends on process being performed.

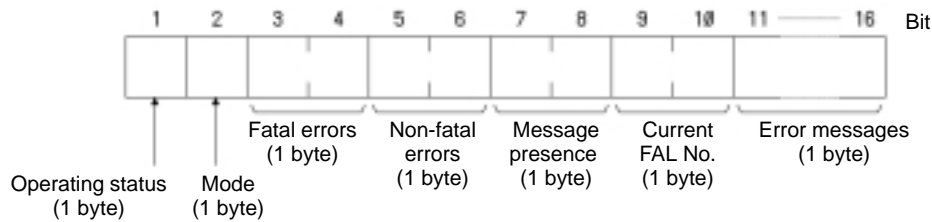
Parameter	Type	IN/OUT	Setting
ne	Integer	IN	Not used; set to 0.
no	Integer	IN	Not used; set to 0.
mcmd	Integer	IN	One of the following. 0: Read status 1: Clear error 2: Read error history 3: Clear error history
ch	Integer	IN/OUT	Used only when reading the error history. Set to 0 for all other processes. IN 1 to 10: Number of records to read. 0: Reads the number of records stored in the SYSMAC Board. OUT 1 to 10: The number of records read. 0: Indicates that the number of records present in the SYSMAC Board has been read.
val	Integer or text string	IN/OUT	Depends on the process being performed. Refer to the <i>Description</i> .

Description

Reading Status

When **mcmd** is set to 0, the status of the SYSMAC Board is read.

Set `ch` to 0 and set `val` to the beginning address of memory in which to store the status data. The status data will be stored beginning at the address specified for `val` after `pcstat` is executed.



Operating Status

Always 0.

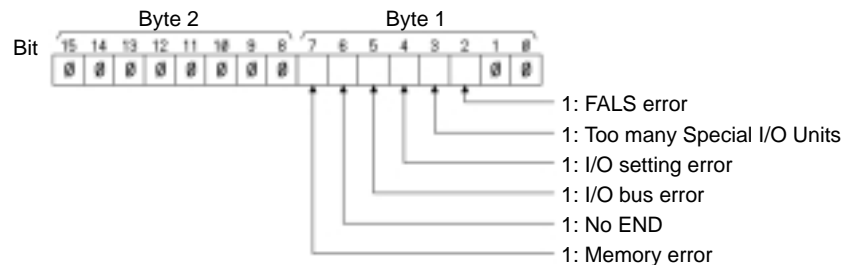
Mode (hex)

The operating mode of the SYSMAC Board as follows:

- 01: PROGRAM mode
- 02: MONITOR mode
- 04: RUN mode

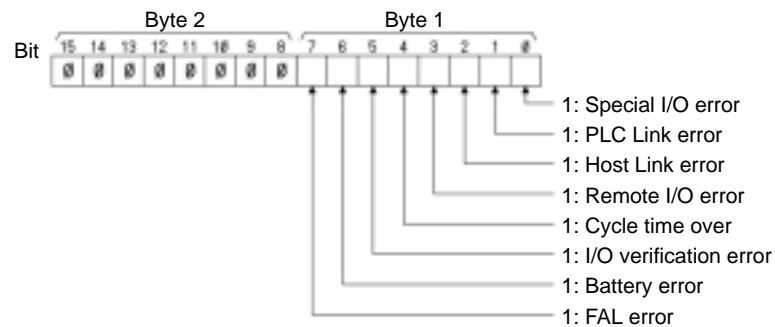
Fatal Errors

Any fatal errors in the SYSMAC Board will be indicated as follows:



Non-fatal Errors

Any non-fatal errors in the SYSMAC Board will be indicated as follows:



Message Presence

Messages cannot be attached to FAL and FALS errors with the SYSMAC Board. This bit will always be 0.

Current FAL No.

These two bytes will provide the FAL No. if a fatal (FALS) or non-fatal (FAL) error has occurred in the SYSMAC Board. They will be 0 if neither error has occurred.

Error Message

All bytes will be filled with spaces.

Communications must be opened with the SYSMAC Board before `pcstat` can be executed.

Clearing Errors

When `mcmd` is set to 1, any FAL or FALS errors that have occurred are cleared from the SYSMAC Board.

Set `ch` to 0 and set `val` to the address of memory location storing the FAL/FALS No. as a 1-byte hexadecimal number. Set `val` to specify the most recent FAL/FALS No.

The specified error will be cleared after `pccstat` is executed.

No errors will be cleared and the function will end normally even if the specified FAL/FALS No. does not exist.

Communications must be opened with the SYSMAC Board before `pccstat` can be executed.

Reading the Error History

When `mcmd` is set to 2, either the number of records currently stored in the error history or the contents of the error history will be read.

Communications must be opened with the SYSMAC Board before `pccstat` can be executed.

Reading the Number of Records

Set `ch` to the address of a memory location containing 00 (1 byte). The number of records currently in the error history (0 to 10) will be stored at the address specified for `ch` after `pccstat` is executed.

Set `val` to 0.

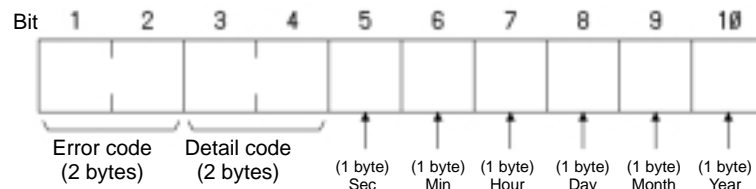
Reading the Contents of the Error History

Set `ch` to the address of a memory location containing the number of error history records to be read (1 byte, hex). Up to 10 records can be read at once. Set the value to between 1 and 10.

The contents of the specified number of records will be stored after `pccstat` is executed. Set `val` to the beginning address in which to store the contents read from the error history. The entire area beginning with the specified address will need to be 10 bytes times the number of records specified.

There is no specific order in which the records will be read out from the SYSMAC Board, e.g., there is no guarantee that the first record read out will be the newest. Always read out all 10 records whenever possible.

The structure of the 10 bytes read out for each record is as follows:



• Error code (hex)

The second byte classifies the error as follows:

- 80: Fatal error
- 00: Non-fatal error

The first byte provides the error code as follows:

Classification	Code	Error
Fatal error	C0 to C2 _H	I/O bus error
	E0 _H	I/O setting error
	E1 _H	Too many I/O Units
	F0 _H	No END
	F1 _H	Memory error
	01 to 99 _H , 9F _H	System error (FALS)
Non-fatal error	B0 to B1 _H	Remote I/O error
	D0 _H	Special I/O error
	E7 _H	I/O verification error
	F7 _H	Battery error
	F8 _H	Cycle time over
	01 to 99 _H , 8A _H , 8B _H , 9A to 9D _H	System error (FAL)

• **Detail Code**

Not used (always zeros).

• **Seconds, Minutes, Hour, Day**

The time and date the error occurred is stored in BCD, 2 bytes each.

• **Month, Year**

Not used (always zeros).

Note The error history is stored in SYSMAC Board memory in DM 6000 to MD 6030 as follows:

Words	Contents
DM 6000	Error history pointer
DM 6001 to DM 6030	Error history records

Each record is stored in three words as follows:

	15	8	7	0	Bit
DM n	Error class			Error code	
DM n+1	Minutes			Seconds	
DM n+2	Day			Hour	

The values in AR 18 and AR 19 are used for the seconds, minutes, hour, and day.

Clearing the Error History

When `mcmd` is set to 3, the error history in the SYSMAC Board is cleared. Set `ch` and `val` to 0.

`pcstat` will end normally even if there are no records in the error history.

Communications must be opened with the SYSMAC Board before `pcstat` can be executed.

Return Values

Value	Meaning
0	Ended normally.
1	Communications have not been opened; execution not possible
5	Illegal parameter.
7	Ended in an error (response error).
10	Communications timeout (no response within 10 s.

Example

Reading Status (`mcmd = 0`)

```

/*****
/* Reading SYSMAC Board Status */

```

```

/*****
extern unsigned int  pcinit() ;
extern unsigned int  pccopen() ;
extern unsigned int  pccclose() ;
extern unsigned int  pcstat() ;

void  main(void)
{
    int ret,i,j ;
    unsigned char  val[26] ;                                /* Location to store status */
    unsigned char  far *valp ;
    unsigned char  dmy ;                                    /* Dummy */
    unsigned char  far *dmyp ;
    unsigned char  ne=0 ;                                  /* Network address */
    unsigned char  no=0 ;                                  /* Node address */
    unsigned int   mcmd=0 ;                                  /* Main command: Status read */
    unsigned char  port_addr=1 ;                            /* I/O port base address = 0370 */
    unsigned char  dpram_addr=7 ;                          /* DPRAM base address = D4000 */

    valp = (unsigned char far *)&val[0];
    dmyp = (unsigned char far *)&dmy;

    ret = pcinit ( port_addr, dpram_addr ) ;    /* SYSMAC Board addresses set */
    if ( ret != 0 ) {
        printf ("PLC setting error (Return value: %d) \n", ret ) ;
        exit ( ) ;
    }

    ret = pccopen ( ) ;                                /* Library opened */
    if ( ret != 0 ) {
        printf ("Error opening library (Return value: %d) \n", ret ) ;
        exit ( ) ;
    }

    printf ("Status Read\n");
    printf ("The status of the SYSMAC Board will be read\n");

    switch( pcstat ( ne, no, mcmd, dmyp, valp ) ) {
        case 0 :
            printf ("PCSTAT successful\n") ;
            printf ("Status is as follows:\n");
            printf ("Operating status (1 byte): 0x%02x\n", val[0]);
            printf ("Mode (1 byte): 0x%02x\n", val[1]);
            printf ("Fatal errors (2 bytes): 0x%02x 0x%02x\n", val[2], val[3]);
            printf ("Non-fatal errors (2 bytes): 0x%02x 0x%02x\n", val[3], val[4]);
            printf ("Message presence (2 bytes): 0x%02x 0x%02x\n", val[5], val[6]);
            printf ("Current FAL No. (2 bytes): 0x%02x 0x%02x\n", val[7], val[8]);
            printf ("Error message (16 bytes): \n\t");
            for ( i = 10; i < 26; i ++ )
                printf ("%02x ", val[i]);
            printf ("\n\t");
            break ;
        case 1 :
            printf ("Communications not open..\n") ;
            break ;
        case 5 :
            printf ("Illegal parameter\n") ;
            break ;
        case 7 :
            printf ("Error\n") ;
            break ;
        case 10 :
            printf ("PLC communications timed out\n") ;
            break ;
    }
}

```

```

    pcclose(); /* Library closed */
}

```

Clearing an Error (mcmd = 1)

```

/*****/
/* Clearing an Error */
/*****/
extern unsigned int  pcinit() ;
extern unsigned int  pccopen() ;
extern unsigned int  pcclose() ;
extern unsigned int  pcstat() ;

void main(void)
{
    int    ret ;
    unsigned char  val ; /* Location to specify FAL No.*/
    unsigned char  far *valp ;
    unsigned char  dmy ; /* Dummy */
    unsigned char  far *dmyip ;
    unsigned char  ne=0 ; /* Network address */
    unsigned char  no=0 ; /* Node address */
    unsigned int   mcmd=1 ; /* Main command: Error clear */
    unsigned char  port_addr=1 ; /* I/O port base address=0370 */
    unsigned char  dpram_addr=7 ; /* DPRAM base address = D4000 */

    val = 0;
    valp = (unsigned char far *)&val;
    dmyip = (unsigned char far *)&dmy;

    ret = pcinit ( port_addr, dpram_addr ) ; /* SYSMAC Board addresses set */
    if ( ret != 0 ) {
        printf ("PLC setting error (Return value: %d) \n", ret ) ;
        exit ( ) ;
    }

    ret = pccopen ( ) ; /* Library opened */
    if ( ret != 0 ) {
        printf ("Error opening library (Return value: %d) \n", ret ) ;
        exit ( ) ;
    }

    printf ("Error will be cleared from SYSMAC Board.\n\n");
    printf ("Specify the FAL No.\n", val);

    switch ( pcstat ( ne, no, mcmd, dmyip, valp ) ) {
        case 0 :
            printf ("Error cleared\n") ;
            break ;
        case 1 :
            printf ("Communications not open.\n") ;
            break ;
        case 5 :
            printf ("Illegal parameter\n") ;
            break ;
        case 7 :
            printf ("Error\n") ;
            break ;
        case 10 :
            printf ("PLC communications timed out\n") ;
            break ;
    }

    pcclose ( ) ; /* Library closed */
}

```


Reading Error History Contents (mcmd = 2)

```

/*****
/* Reading the Error History */
*****/
extern unsigned int  pcinit() ;
extern unsigned int  pccopen() ;
extern unsigned int  pccclose() ;
extern unsigned int  pcstat() ;

void  main(void)
{
    int    ret, i, j ;
    unsigned char  val[10][10] ;                /* Location to store error history data */
    unsigned char  far *valp ;
    unsigned char  ch ;                        /* Dummy */
    unsigned char  far *chp ;
    unsigned char  ne=0 ;                      /* Network address */
    unsigned char  no=0 ;                      /* Node address */
    unsigned int   mcmd=2 ;                    /* Main command: Read history */
    unsigned char  port_addr=1 ;               /* I/O port base address=0370 */
    unsigned char  dpram_addr=7 ;              /* DPRAM base address = D4000 */

    ch = 10 ;                                /* Specify 10 records */
    valp = (unsigned char far *)&val[0][0] ;
    chp = (unsigned char far *)&ch ;

    ret = pcinit ( port_addr, dpram_addr ) ; /* SYSMAC Board addresses set */
    if ( ret != 0 ) {
        printf ("PLC setting error (Return value: %d) \n", ret ) ;
        exit () ;
    }

    ret = pccopen () ;                        /* Library opened */
    if ( ret != 0 ) {
        printf ("Error opening library (Return value: %d) \n", ret ) ;
        exit () ;
    }

    printf ("The SYSMAC Board error history will be read from the first
record.\n\n");
    printf ("A total of %d records will be read.\n", ch);

    switch ( pcstat ( ne, no, mcmd, chp, valp ) ) {
        case 0 :
            printf ("Error history was read successfully\n") ;
            printf ("A total of %d records were read.\n\n", ch);
            if ( ch > 0 ) {
                printf ("The following data was read.\n");
                for ( i = 0; i < ch; i++ ) {
                    printf ("Error History Record %d:", i+1);
                    for ( j = 0; j < 10; j++ )
                        printf ("0x0%02x ", val[i][j]);
                    printf ("\n");
                }
            } else
                printf ("There are no records in the error history.\n");
            printf ("\n");
            break ;
        case 1 :
            printf ("Communications not open.\n") ;
            break ;
        case 5 :
            printf ("Illegal parameter\n") ;
            break ;
        case 7 :

```

```

        printf ("Error\n") ;
        break ;
    case 10 :
        printf ("PLC communications timed out\n") ;
        break ;
}

pcclose () ;                                /* Library closed */
}

```

Clearing the Error History (mcmd = 3)

```

/*****
/* Clearing the Error History */
*****/
extern unsigned int  pcinit() ;
extern unsigned int  pccopen() ;
extern unsigned int  pcclose() ;
extern unsigned int  pcstat() ;

void main(void)
{
    int  ret, i, j ;
    unsigned char  dmy ;                                /* Dummy */
    unsigned char  far *dmyp ;
    unsigned char  ne=0 ;                                /* Network address */
    unsigned char  no=0 ;                                /* Node address */
    unsigned int  mcmd=3 ;                                /* Main command: Clear error history */
    unsigned char  port_addr=1 ;                          /* I/O port base address=0370 */
    unsigned char  dpram_addr=7 ;                          /* DPRAM base address = D4000 */

    dmyp = (unsigned char far *)&dmy;

    ret = pcinit ( port_addr, dpram_addr ) ; /* SYSMAC Board addresses set */
    if ( ret != 0 ) {
        printf ("PLC setting error (Return value: %d) \n", ret ) ;
        exit () ;
    }

    ret = pccopen () ;                                    /* Library opened */
    if ( ret != 0 ) {
        printf ("Error opening library (Return value: %d) \n", ret ) ;
        exit () ;
    }

    printf ("The error history will be cleared from the SYSMAC Board.\n\n");

    switch ( pcstat ( ne, no, mcmd, dmyp, dmyp ) ) {
        case 0 :
            printf ("Error history successfully cleared.\n") ;
            break ;
        case 1 :
            printf ("Communications not open.\n") ;
            break ;
        case 5 :
            printf ("Illegal parameter\n") ;
            break ;
        case 7 :
            printf ("Error\n") ;
            break ;
        case 10 :
            printf ("PLC communications timed out\n") ;
            break ;
    }
}

```

```

    pcclose ( ) ;                                /* Library closed */
}

```

pcread ()**PLC READ****Purpose**

Reads data from memory areas on the SYSMAC Board.

Format

```

unsigned int    pcread(ne,no,sub_format,val[,val...]);
unsigned char   ne;
unsigned char   no;
unsigned char   far *sub_format;
unsigned int    far *val;

```

or

```

unsigned char   far *val;

```

[val] can be omitted and parameters in brackets can be input more than once.

Parameters

ne Remote network address. This parameter is not used; set it to 0.
no Remote node address. This parameter is not used; set it to 0.
sub_format The sub-format.
val Response storage buffer.

Parameter	Type	IN/OUT	Setting
ne	Integer	IN	Not used; set to 0.
no	Integer	IN	Not used; set to 0.
sub_format	Text string	IN	Refer to the <i>Description</i> .
val	Integer/text string	OUT	Refer to the <i>Description</i> .

Description

pcread is used to read data from the memory areas of the SYSMAC Board. Data is read in word units, and up to 128 words can be read at once.

Set sub_format to the first address of the memory location holding the sub-format specification. The sub-format specifies the area to be read, the storage method for the data, and other parameters. The sub-format is specified as follows, with commas separating parameters:

```
sub,start,num,format[,format]
```

[format] can be omitted and parameters in brackets can be input more than once.

Note Use only uppercase letters in the sub-format.

sub: Subcommand

sub	Area	Words	Unit
@R	CIO Area	0 to 255	Word
	SR Area	256 to 511	Word
	LR Area	1000 to 1063 (to specify LR 00 to LR 63)	Word
	HR Area	1064 to 1163 (to specify HR 00 to HR 63)	Word
@A	AR Area	0 to 27	Word
@T	TIM Area, PV	0 to 511	Word
@C	CNT Area, PV	0 to 511	Word
@TF	TIM Area, Completion Flag (See note.)	0 to 511	Word
@CF	CNT Area, Completion Flag(See note.)	0 to 511	Word
@D	DM Area	0 to 6655	Word
	Expansion DM Area	7,000 to 9,999	Word
@E0	EM Area	0 to 6,143	Word

Note The Completion Flag is ON when the timer has timed out or the counter has counted out.

start: First Word to Read

Specify the address of the first word to read from memory. Use the address ranges given in the table, above.

num: Number of Words to Read

Specify the number of words to read between 1 and 128, but be careful not to specify more words than are in the area being read. If the end of the area is exceeded, a return value of 5 will be returned to indicate an illegal parameter. This is also true for the individual areas specified with a `sub` of `@R`, i.e. a write must be within only one of the areas.

format: Storage Format

The format used to store the read data in memory is specified with one of the codes in the following table. A different format can be specified for each of the words specified with `num`. If the contents of the data cannot be interpreted in the specified format, zeros will be stored.

"n" in the formats listed in the following table is replaced with the number of words to which the format is to be applied. "n" can be omitted, in which case a default value of 1 is used. The formats are applied in order to each word being read.

Note A format must be specified for each word being read, i.e., the total of all values for "n" must be equal to the value specified for `num`. The function may not execute properly if there is not a format specified for each word.

Format	Operation and storage method
%nl	Interprets read data as decimal, and stores it as numerals. Data that cannot be interpreted as decimal will be regarded as "0." A single word of data will be stored in a single <i>val</i> variable. If a numeral of 2 or greater is specified for <i>n</i> , then <i>n</i> number of <i>val</i> variables will be required. The variable <i>val</i> type will be numeral for <i>unsigned int far</i> *.
%nH	Interprets read data as hexadecimal, and stores it as numerals. Data that cannot be interpreted as hexadecimal will be regarded as "0." Specifications regarding <i>val</i> are the same as for %nl.
%nO	Interprets read data as octal, and stores it as numerals. Data that cannot be interpreted as octal will be regarded as "0." Specifications regarding <i>val</i> are the same as for %nl.
%SnI	Interprets read data as decimal, and stores it as numerals. Data that cannot be interpreted as decimal will be regarded as "0." Read data will be stored in array variable <i>val</i> []. If two or more numerals are specified for <i>n</i> , then <i>n</i> number of arrays will be required for the variable <i>val</i> []. The variable <i>val</i> [] type will be numeral for <i>unsigned int far</i> *.
%SnH	Interprets read data as hexadecimal, and stores it as numerals. Data that cannot be interpreted as hexadecimal will be regarded as "0." Specifications regarding <i>val</i> are the same as for %SnI.
%SnO	Interprets read data as hexadecimal, and stores it as numerals. Data that cannot be interpreted as hexadecimal will be regarded as "0." Specifications regarding <i>val</i> are the same as for %SnI.
\$nl	Interprets read data as decimal, and converts it to character strings for storage. Data that cannot be interpreted as decimal will be regarded as "0." A single word of data will be stored in a single <i>val</i> variable. If a numeral of 2 or greater is specified for <i>n</i> , then <i>n</i> number of <i>val</i> variables will be required. A single word (two bytes) of data will be converted to a 4-byte character string expressing a 4-digit numeral. Thus four bytes of data area will be required for a single variable <i>val</i> . The variable <i>val</i> type will be character for <i>unsigned char far</i> *.
\$nH	Interprets read data as hexadecimal, and stores it as numerals. Data that cannot be interpreted as hexadecimal will be regarded as "0." Specifications regarding <i>val</i> are the same as for \$nl.
\$nO	Interprets read data as hexadecimal, and stores it as numerals. Data that cannot be interpreted as octal will be regarded as "0." Specifications regarding <i>val</i> are the same as for \$nl.
\$nA	Interprets read data as ASCII, and converts it to character strings for storage. Data read into array variable <i>val</i> [] will be stored. A single word (two bytes) of data will be converted to a 2-byte character string. Thus a data area of <i>n</i> x 2 bytes will be required for an array variable <i>val</i> []. The variable <i>val</i> type will be character, for <i>unsigned char far</i> *.

Format	Operation and storage method
\$SnI	Interprets read data as decimal, and converts it to character strings for storage. Data that cannot be interpreted as decimal will be regarded as "0." Read data will be stored in array variable <i>val</i> []. If two or more numerals are specified for <i>n</i> , then <i>n</i> number of arrays will be required for the variable <i>val</i> []. A single word (two bytes) of data will be converted to a 4-byte character string expressing a 4-digit numeral. Thus a data area of <i>n</i> x 4 bytes will be required for an array variable <i>val</i> []. The variable <i>val</i> type will be character for <i>unsigned char far</i> *.
\$SnH	Interprets read data as hexadecimal, and converts it to character strings for storage. Data that cannot be interpreted as hexadecimal will be regarded as "0." Specifications regarding <i>val</i> are the same as for \$SnI.
\$SnO	Interprets read data as hexadecimal, and converts it to character strings for storage. Data that cannot be interpreted as octal will be regarded as "0." Specifications regarding <i>val</i> are the same as for \$SnI.
\$SnA	Interprets read data as ASCII, and converts it to character strings for storage. Data read into array variable <i>val</i> [] will be stored. A single word (two bytes) of data will be converted to a 2-byte character string. Thus a data area of <i>n</i> x 2 bytes will be required for an array variable <i>val</i> []. The variable <i>val</i> type will be character, for <i>unsigned char far</i> *.

Note Communications must be opened with the SYSMAC Board before *pcread* can be executed.

Conversion Examples

Examples are given below, according to the various storage formats, of converting data that has been read.

1, 2, 3... 1. I-type (Decimal) Format

Read data: 12345678

Numerals (%nI)

```
unsigned int val1, val2;
unsigned int far *p1, *p2;
p1 = (unsigned int far *)&val1;
p2 = (unsigned int far *)&val2;
pcread(...,"...",%2I",p1,p2);
```

Results

```
val1 = 1234 = 0x04d2
val2 = 5678 = 0x162e
```

Numeral Array (%SnI)

```
signed int val[2];
unsigned int far *p;
p = (unsigned int far *)&val;
pcread(...,"...",%S2I",p);
```

Results

```
val[0] = 1234 = 0x04d2
val[1] = 5678 = 0x162e
```

Characters (\$nI)

```
unsigned char val1[4],val2[4];
unsigned char far *p1,*p2;
p1 = (unsigned char far *)&val1;
p2 = (unsigned char far *)&val2;
pcread(...,"...",$2I",p1,p2);
```

Results

```
val1[0] = '1' = 0x31; val1[1] = '2' = 0x32;
val1[2] = '3' = 0x33; val1[3] = '4' = 0x34;
val2[0] = '5' = 0x35; val2[1] = '6' = 0x36;
val2[2] = '7' = 0x37; val2[3] = '8' = 0x38;
```

Character Array (\$SnI)

```
unsigned char val[8];
unsigned char far *p;
p = (unsigned char far *)&val;
pcread(...,"...",$S2I",p);
```

Results

```
val[0] = '1' = 0x31; val[1] = '2' = 0x32;
val[2] = '3' = 0x33; val[3] = '4' = 0x34;
val[4] = '5' = 0x35; val[5] = '6' = 0x36;
val[6] = '7' = 0x37; val[7] = '8' = 0x38;
```

2. H-type (Hexadecimal) Format**Read data: 789ABCDE**

Numerals (%nH)

```
unsigned int val1, val2;
unsigned int far *p1, *p2;
p1 = (unsigned int far *)&val1;
p2 = (unsigned int far *)&val2;
pcread(...,"...,%2H",p1,p2);
```

Results

```
val1 = 0x789a
val2 = 0xbcde
```

Numeral Array (%SnH)

```
unsigned int val[2];
unsigned int far *p;
p = (unsigned int far *)&val;
pcread(...,"...,%S2H",p);
```

Results

```
val[0] = 0x789a
val[1] = 0xbcde
```

Characters (\$nH)

```
unsigned char val1[4],val2[4];
unsigned char far *p1,*p2;
p1 = (unsigned char far *)&val1;
p2 = (unsigned char far *)&val2;
pcread(...,"...,$2H",p1,p2);
```

Results

```
val1[0] = '7' = 0x37; val1[1] = '8' = 0x38;
val1[2] = '9' = 0x39; val1[3] = 'A' = 0x41;
val2[0] = 'B' = 0x42; val2[1] = 'C' = 0x43;
val2[2] = 'D' = 0x44; val2[3] = 'E' = 0x45;
```

Character Array (\$SnH)

```
unsigned char val[8];
unsigned char far *p;
p = (unsigned char far *)&val;
pcread(...,"...,$S2H",p);
```

Results

```
val[0] = '7' = 0x37; val[1] = '8' = 0x38;
val[2] = '9' = 0x39; val[3] = 'A' = 0x41;
val[4] = 'B' = 0x42; val[5] = 'C' = 0x43;
val[6] = 'D' = 0x44; val[7] = 'E' = 0x45;
```

3. O-type (Octal) Format**Read data: 12345670**

Numerals (%nO)

```
unsigned int val1, val2;
unsigned int far *p1, *p2;
p1 = (unsigned int far *)&val1;
p2 = (unsigned int far *)&val2;
pcread(...,"...,%20",p1,p2);
```

Results

```

val1 = 01234 = 0x029c
val2 = 05678 = 0x0bb8

```

Numeral Array (%SnO)

```

unsigned int val[2];
unsigned int far *p;
p = (unsigned int far *)val;
pcread(...,"...,%S20",p);
Results
val[0] = 01234 = 0x029c
val[1] = 05670 = 0x0bb8

```

Characters (\$nO)

```

unsigned char val1[4],val2[4];
unsigned char far *p1,*p2;
p1 = (unsigned char far *)val1;
p2 = (unsigned char far *)val2;
pcread(...,"...,$20",p1,p2);
Results
val1[0] = '1' = 0x31; val1[1] = '2' = 0x32;
val1[2] = '3' = 0x33; val1[3] = '4' = 0x34;
val2[0] = '5' = 0x35; val2[1] = '6' = 0x36;
val2[2] = '7' = 0x37; val2[3] = '0' = 0x30;

```

Character Array (\$SnO)

```

unsigned char val1[8];
unsigned char far *p;
p = (unsigned char far *)val;
pcread(...,"...,$S20",p);
Results
val[0] = '1' = 0x31; val[1] = '2' = 0x32;
val[2] = '3' = 0x33; val[3] = '4' = 0x34;
val[4] = '5' = 0x35; val[5] = '6' = 0x36;
val[6] = '7' = 0x37; val[7] = '0' = 0x30;

```

4. A-type (ASCII Code) Format

Read data: 51525354

Characters (\$nA)

```

unsigned char val1[4];
unsigned char far *p;
p = (unsigned char far *)val;
pcread(...,"...,$2A",p);
Results
val1[0] = 'Q' = 0x51; val1[1] = 'R' = 0x52;
val1[2] = 'S' = 0x53; val1[3] = 'T' = 0x54;

```

Character Array (\$SnA)

```

unsigned char val[4];
unsigned char far *p;
p = (unsigned char far *)val;
pcread(...,"...,$S2A",p);
Results
val[0] = 'Q' = 0x51; val[1] = 'R' = 0x52;
val[2] = 'S' = 0x53; val[3] = 'T' = 0x54;

```

Return Values

Value	Meaning
0	Ended normally.
1	Communications have not been opened; execution not possible
5	Illegal parameter.
7	Ended in an error (response error).
10	Communications timeout (no response within 10 s.

Example

```

/*****
/* Reading from a SYSMAC Board Memory Area */
*****/
extern unsigned int  pcinit() ;
extern unsigned int  pcopen() ;
extern unsigned int  pcclose() ;
extern unsigned int  pcread() ;

#define      INT_SIZE2          /* % */
#define      CHANEL  3         /* Channel */

void  main(void)
{
    int  ret;
    unsigned char  sub[80];
    unsigned char  far *subp;
    unsigned int  buf[CHANEL][INT_SIZE];    /* 3 words */
    unsigned int  far *bufp;
    unsigned char  ne=0;
    unsigned char  no=0;
    unsigned char  port_addr=1 ;            /* I/O port base address = 0370 */
    unsigned char  dpram_addr=7 ;           /* DPRAM base address = D4000 */

    bufp = (unsigned int far *)&buf[0][0];

    ret = pcinit ( port_addr, dpram_addr ) ; /* SYSMAC Board addresses set */
    if ( ret != 0 ) {
        printf ("PLC setting error (Return value: %d) \n", ret ) ;
        exit () ;
    }

    printf("ret = %d\n",ret = pcopen());
    if ( ret != 0 ) {
        printf ("Error opening library (Return value: %d) \n", ret ) ;
        exit () ;
    }

                                /* Setting sub-format: @D,0,3,%S3H */
    subp = (unsigned char far *)&sub[0];
    strcpy(sub, "@D,0,3,%S3H");

    printf ("Memory Area Read\n");

    printf ("Data will be read from an SYSMAC Board memory area.\n\n");
    printf ("The specified read command is \"%s\"\n", sub);

                                /* Reading memory area */
    switch ( pcread(ne,no,subp,bufp) ) {
    case 0:
        printf ("Read was successful.\n");
        printf ("The data that was read is as follows:\n\t");
        for( i = 0; i < 3; i++ )
            printf("0x%04x ", *bufp++);
        break;
    case 1:

```



```

        printf ("Communications not open.\n");
        break;
case 5:
    printf ("Illegal parameter\n");
    break;
case 7:
    printf ("Error\n");
    break;
case 10:
    printf ("PLC communications timed out\n");
    break;
}

pcclose();
}

```

pcwrite()

PLC WRITE

Purpose Writes data to memory areas on the SYSMAC Board.

Format

```

unsigned int    pcwrite(ne,no,sub_format,val[,val...]);
unsigned char   ne;
unsigned char   no;
unsigned char   far    *sub_format;
unsigned int    far    *val;
or
unsigned char   far    *val;

```

[val] can be omitted and parameters in brackets can be input more than once.

Parameters

ne	Remote network address. This parameter is not used; set it to 0.
no	Remote node address. This parameter is not used; set it to 0.
sub_format	The sub-format.
val	Buffer storing the write data.

Parameter	Type	IN/OUT	Setting
ne	Integer	IN	Not used; set to 0.
no	Integer	IN	Not used; set to 0.
sub_format	Text string	IN	Refer to the <i>Description</i> .
val	Integer/text string	OUT	Refer to the <i>Description</i> .

Description `pcwrite` is used to write data to the memory areas of the SYSMAC Board. Data is written in word units, and up to 128 words can be written at once.

Set `sub_format` to the first address of the memory location holding the sub-format specification. The sub-format specifies the area to be written, the conversion method for the data, and other parameters. The sub-format is specified as follows, with commas separating parameters:

```
sub,start,num,format[,format]
```

[format] can be omitted and parameters in brackets can be input more than once.

Note Use only uppercase letters in the sub-format.

sub: Subcommand

sub	Area	Words	Unit
@R	CIO Area	0 to 255	Word
	SR Area	256 to 511	Word
	LR Area	1000 to 1063 (to specify LR 00 to LR 63)	Word
	HR Area	1064 to 1163 (to specify HR 00 to HR 63)	Word
@A	AR Area	0 to 27	Word
@T	TIM Area, PV	0 to 511	Word
@C	CNT Area, PV	0 to 511	Word
@D	DM Area	0 to 6655	Word
	Expansion DM Area	7,000 to 9,999	Word
@E0	EM Area	0 to 6,143	Word

start: First Word to Write

Specify the address of the first word to write in memory. Use the address ranges given in the table, above.

num: Number of Words to Write

Specify the number of words to write between 1 and 128, but be careful not to specify more words than are in the area being written. If the end of the area is exceeded, a return value of 5 will be returned to indicate an illegal parameter. This is also true for the individual areas specified with a `sub` of @R, i.e. a write must be within only one of the areas.

format: Conversion Format

The format used to convert the data in memory is specified with one of the codes in the following table. A different format can be specified for each of the words specified with `num`.

“n” in the formats listed in the following table is replaced with the number of words to which the format is to be applied. “n” can be omitted, in which case a default value of 1 is used. The formats are applied in order to each word being written.

Note A format must be specified for each word being read, i.e., the total of all values for “n” must be equal to the value specified for `num`. The function may not execute properly if there is not a format specified for each word.

Format	Operation and storage method
%nl	Regards write data as numerals and expands it to decimal (BCD conversion) for writing. At the time of expansion to decimal, overflow portions will be ignored. Each variable from <i>val</i> will be converted to one word of data. If a number greater than 1 is specified for “n,” then that number of variables will be required from <i>val</i> . The variable <i>val</i> type will be numeral for <i>unsigned int far*</i> .
%nH	Regards write data as numerals and expands it to hexadecimal for writing. At the time of expansion to decimal, overflow portions will be ignored. Specifications regarding <i>val</i> are the same as for %nl.
%nO	Regards write data as numerals and expands it to octal for writing. At the time of expansion to octal, overflow portions will be ignored. Specifications regarding <i>val</i> are the same as for %nl.
%SnI	Regards write data as numerals and expands it to decimal (BCD conversion) for writing. At the time of expansion to decimal, overflow portions will be ignored. The write data is stored in <i>val</i> as an array variable. If a number greater than 1 is specified for “n,” then a single array variable containing that number of variables will be required for <i>val</i> . The variable <i>val</i> [] type will be numeral for <i>unsigned int far*</i> .
%SnH	Regards write data as numerals and expands it to hexadecimal for writing. At the time of expansion to decimal, overflow portions will be ignored. Specifications regarding <i>val</i> are the same as for %SnI.
%SnO	Regards write data as numerals and expands it to octal for writing. At the time of expansion to octal, overflow portions will be ignored. Specifications regarding <i>val</i> are the same as for %SnI.

Format	Operation and storage method
\$nl	Regards write data as characters and expands it to decimal (BCD conversion) for writing. At the time of expansion to decimal, overflow portions will be ignored. Each variable from <i>val</i> will be converted to one word of data. If a number greater than 1 is specified for "n," then that number of variables will be required from <i>val</i> . Four bytes of data will become a single word (two bytes) of data. Thus four bytes of data area will be required for a single variable <i>val</i> . The variable <i>val</i> type will be character for <i>unsigned char far *</i> .
\$nH	Regards write data as character strings and expands it to hexadecimal for writing. At the time of expansion to decimal, overflow portions will be ignored. Specifications regarding <i>val</i> are the same as for <i>\$nl</i> .
\$nO	Regards write data as character strings and expands it to octal for writing. At the time of expansion to octal, overflow portions will be ignored. Specifications regarding <i>val</i> are the same as for <i>\$nl</i> .
\$nA	Regards the write data as ASCII and writes it without alteration. Data from n words is stored in a single variable <i>val</i> . Two bytes of data will become a single word (two bytes) of data. Thus variable <i>val</i> will require a data area of n x 2 bytes. The variable <i>val</i> type will be character for <i>unsigned char far *</i> .
\$SnI	Regards write data as characters and expands it to decimal (BCD conversion) for writing. At the time of expansion to decimal, overflow portions will be ignored. The write data is stored in an array variable. Four bytes of data will become a single word (two bytes) of data. Thus it will be necessary to specify a character array variable <i>val[]</i> which has n x 4 arrays. The variable <i>val</i> type will be character for <i>unsigned char far *</i> .
\$SnH	Regards write data as characters and expands it to hexadecimal for writing. At the time of expansion to decimal, overflow portions will be ignored. Specifications regarding <i>[]</i> are the same as for <i>\$SnI</i> .
\$SnO	Regards write data as characters and expands it to octal for writing. At the time of expansion to octal, overflow portions will be ignored. Specifications regarding <i>[]</i> are the same as for <i>\$SnI</i> .
\$SnA	Regards the write data as ASCII and writes it without alteration. The write data is stored in an array variable. Two bytes of data will become a single word (two bytes) of data. Thus it will be necessary to specify a character array variable <i>val[]</i> which has a data area of n x 2 bytes. The variable <i>val</i> type will be character for <i>unsigned char far *</i> .

Note Communications must be opened with the SYSMAC Board before `pcwrite` can be executed.

Conversion Examples

Examples are given below, according to the various storage formats, of converting data that has been read.

1, 2, 3... 1. I-type (Decimal) Format

Data to write to SYSMAC Board: 12345678

Numerals (%nl)

```
unsigned int val1, val2;
unsigned int far *p1, *p2;

val1 = 1234;
val2 = 5678;
p1 = (unsigned int far *)&val1;
p2 = (unsigned int far *)&val2;
pcwrite(..."%2I",p1,p2);
```

Numeral Array (%SnI)

```
unsigned int val[2];
unsigned int far *p;

val[0] = 1234;
val[1] = 5678;
p = (unsigned int far *)val;
pcwrite(...,"%S2I",p);
```

Characters (\$nl)

```
unsigned char val1[4],val2[4];
unsigned char far *p1,*p2;

val1[0] = '1' = 0x31; val1[1] = '2' = 0x32;
```

```

vall[2] = '3' = 0x33; vall[3] = '4' = 0x34;
val2[0] = '5' = 0x35; val2[1] = '6' = 0x36;
val2[2] = '7' = 0x37; val2[3] = '8' = 0x38;
p1 = (unsigned char far *)vall;
p2 = (unsigned char far *)val2;
pcwrite(...,"...,$2I",p1,p2);

```

Character Array (\$SnI)

```

unsigned char val[8];
unsigned char far *p;

val[0] = '1' = 0x31; val[1] = '2' = 0x32;
val[2] = '3' = 0x33; val[3] = '4' = 0x34;
val[4] = '5' = 0x35; val[5] = '6' = 0x36;
val[6] = '7' = 0x37; val[7] = '8' = 0x38;
p = (unsigned char far *)val;
pcwrite(...,"...,$S2I",p);

```

2. H-type (Hexadecimal) Format

Data to write to SYSMAC Board: 789ABCDE

Numerals (%nH)

```

unsigned int vall, val2;
unsigned int far *p1, *p2;

vall = 0x789a
val2 = 0xbcde
p1 = (unsigned int far *)&vall;
p2 = (unsigned int far *)&val2;
pcwrite(...,"...,%2H",p1,p2);

```

Numerical Array (%SnH)

```

unsigned int val[2];
unsigned int far *p;

val[0] = 0x789a
val[1] = 0xbcde
p = (unsigned int far *)val;
pcwrite(...,"...,%S2H",p);

```

Characters (\$nH)

```

unsigned char vall[4],val2[4];
unsigned char far *p1,*p2;

vall[0] = '7' = 0x37; vall[1] = '8' = 0x38;
vall[2] = '9' = 0x39; vall[3] = 'A' = 0x41;
val2[0] = 'B' = 0x41; val2[1] = 'C' = 0x43;
val2[2] = 'D' = 0x44; val2[3] = 'E' = 0x45;
p1 = (unsigned char far *)vall;
p2 = (unsigned char far *)val2;
pcwrite(...,"...,$2H",p1,p2);

```

Character Array (\$SnH)

```

unsigned char val[8];
unsigned char far *p;

val[0] = '7' = 0x37; val[1] = '8' = 0x38;
val[2] = '9' = 0x39; val[3] = 'A' = 0x41;
val[4] = 'B' = 0x42; val[5] = 'C' = 0x43;
val[6] = 'D' = 0x44; val[7] = 'E' = 0x45;

```

```
p = (unsigned char far *)val;
pcwrite(...,"...,$S2H",p);
```

3. O-type (Octal) Format

Data to write to SYSMAC Board: 12345670

Numerals (%nO)

```
unsigned int val1, val2;
unsigned int far *p1, *p2;

val1 = 01234;
val2 = 05678;
p1 = (unsigned int far *)&val1;
p2 = (unsigned int far *)&val2;
pcwrite(...,"...,%20",p1,p2);
```

Numeral Array (%SnO)

```
unsigned int val[2];
unsigned int far *p;

val[0] = 01234;
val[1] = 05670;
p = (unsigned int far *)val;
pcwrite(...,"...,%S20",p);
```

Characters (\$nO)

```
unsigned char val1[4],val2[4];
unsigned char far *p1,*p2;

val1[0] = '1' = 0x31; val1[1] = '2' = 0x32;
val1[2] = '3' = 0x33; val1[3] = '4' = 0x34;
val2[0] = '5' = 0x35; val2[1] = '6' = 0x36;
val2[2] = '7' = 0x37; val2[3] = '0' = 0x30;
p1 = (unsigned char far *)val1;
p2 = (unsigned char far *)val2;
pcwrite(...,"...,$20",p1,p2);
```

Character Array (\$SnO)

```
unsigned char val[8];
unsigned char far *p;

val[0] = '1' = 0x31; val[1] = '2' = 0x32;
val[2] = '3' = 0x33; val[3] = '4' = 0x34;
val[4] = '5' = 0x35; val[5] = '6' = 0x36;
val[6] = '7' = 0x37; val[7] = '0' = 0x30;
p = (unsigned char far *)val;
pcwrite(...,"...,$S20",p);
```

4. A-type (ASCII) Format

Data to write to SYSMAC Board: 51525354

Characters (\$nA)

```
unsigned char val[4];
unsigned char far *p;

val[0] = 'Q' = 0x51; val[1] = 'R' = 0x52;
val[2] = 'S' = 0x53; val[3] = 'T' = 0x54;
p = (unsigned char far *)val;
pcwrite(...,"...,$2A",p);
```

Character Array (\$SnA)

```
unsigned char val[4];
unsigned char far *p;
```

```
val[0] = 'Q' = 0x51; val[1] = 'R' = 0x52;  
val[2] = 'S' = 0x53; val[3] = 'T' = 0x54;  
p = (unsigned char far *)val;  
pcwrite(..., "...,$S2A",p);
```

Return Values

Value	Meaning
0	Ended normally.
1	Communications have not been opened; execution not possible
5	Illegal parameter.
7	Ended in an error (response error).
10	Communications timeout (no response within 10 s.

Example

```

/*****
/* Writing to a SYSMAC Board Memory Area */
*****/
extern unsigned int  pcinit() ;
extern unsigned int  pcopen();
extern unsigned int  pcclose();
extern unsigned int  pcwrite();

void  main(void)
{
    int  ret, i;
    unsigned char  sub[20];
    static unsigned char buf[8] = { 0x31,0x32,0x33,0x34,0x35,0x36,0x37,0x38 };
    unsigned char  far *subp;
    unsigned int   far *bufp;
    unsigned char  ne=0;
    unsigned char  no=0;
    unsigned char  port_addr=1 ;           /* I/O port base address = 0370 */
    unsigned char  dpram_addr=7 ;          /* DPRAM base address = D4000 */

    ret = pcinit ( port_addr, dpram_addr ) ; /* SYSMAC Board addresses set */
    if ( ret != 0 ) {
        printf ("PLC setting error (Return value: %d) \n", ret ) ;
        exit () ;
    }

    printf("ret = %d\n",ret = pcopen());
    if ( ret != 0 ) {
        printf ("Error opening library (Return value: %d) \n", ret ) ;
        exit () ;
    }

                                /* Setting sub-format: @D,0,3,%S3H */
    subp = (unsigned char far *)sub;
    bufp = (unsigned char far *)buf;
    strcpy(sub, "@D,0,3,%S3H");

    printf ("Memory Area Write\n");
    printf ("Data will be written to an SYSMAC Board memory area.\n\n");
    printf ("The specified write command is \"%s\"\n", sub);
    printf ("The following data will be written:");
    for ( i = 0; i < 8; i++ )
        printf("%02x ", buf[i]);
    printf("\n\n");

                                /* Writing memory area */
    switch ( pcwrite(ne,no,subp,bufp) ) {
    case 0:
        printf("The data was successfully written.\n");
        break;
    case 1:
        printf("Communications not open.\n");
        break;
    case 5:
        printf("Illegal parameter\n");
        break;
    case 7:
        printf("Error\n");
        break;
    case 10:
        printf("PLC communications timed out\n");
        break;
    }
    pcclose();
}

```

SECTION 4

Read/Write Performance

This section shows the average times required to execute the `pcread` and `pcwrite` functions under various conditions as a characteristic of the SYSMAC Board.

4-1	Measurement Program	90
4-2	Measurement Results	92
4-2-1	Measurement Conditions	92
4-2-2	Results	92

4-1 Measurement Program

This section describes the program used to measure read and write execution times. The read times will vary for the computer being used and the operating environment it is used under. Use the following program as a reference to write a program for your computer to confirm execution times.

The following program executes the read and write 10,000 times to find the average read/write time for the following.

- Reading 1 AR word
- Reading 128 AR words
- Writing 1 AR word
- Writing 128 AR words

The areas and numbers of words are indicated in the following program in the lines with the "Measured word specification" comments. Change these lines to the required areas and numbers of words.

Program

```
#include <stdio.h>
#include <time.h>

void main()
{
    void r_main(), w_main();
    pcinit(2,5);                /* I/O port: 0380, Memory: D0000 */

    r_main();
    w_main();
}

void r_main(void)
{
    unsigned char    ne, no;
    unsigned int     i, rc, num, data[128]={0};
    unsigned int     far *bufp;
    time_t           t1, t2;
    unsigned char    far *subp;

    rc=pcopen();

    printf("-> Start 1-word read loop.\n");

    bufp = (unsigned int far *)data;
    subp = "@R,0,1,%SI";        /* Measured word specification: 1 word */

    time(&t1);                  /* Getting start time */

                                /* Calling pcread 10,000 times */
    for(i=0;i<10,000;i++)
        rc=pcread(ne,no,subp,bufp);

    time(&t2);                  /* Get end time */

    printf("%4,1f\n", difftime(t2,t1)) ;

    printf("-> End 1-word read loop.\n");
    printf("-> Start 128-word read loop.\n");

    subp = "@R,0,128,%S128I" ; /* Measured word specification: 128 words */

    time(&t1);                  /* Getting start time */
                                /* Calling pcread 10,000 times */
    for(i=0;i<10000;i++)
```

```

        rc=pcread(ne,no,subp,bufp);
        time(&t2); /* Get end time */

        printf("%4.1f\n", difftime(t2,t1)) ;
        printf("-> End 128-word read loop.\n");

        rc=pcclose();
    }

void w_main(void)
{
    unsigned char    ne, no;
    unsigned int     i, rc, num, data[128]={0};
    time_t          t1, t2;
    unsigned char    far *subp;
    unsigned int     far *bufp;

    rc=pcopen();

    printf("-> Start 1-word write loop.\n");

    bufp = (unsigned int far *)data;
    subp = "@R,0,1,%SI" ; /* Measured word specification: 1 word */

    time(&t1); /* Getting start time */

    /* Calling pcwrite 10,000 times */
    for(i=0;i<10000;i++)
        rc=pcwrite(ne,no,subp,bufp);

    time(&t2); /* Get end time */

    printf("%4.1f\n", difftime(t2,t1)) ;

    printf("-> End 1-word write loop.\n");
    printf("-> Start 128-word write loop.\n");

    supb = "@R,0,128,%S128I" ; /*Measured word specification: 128 words*/

    time(&t1); /* Getting start time */

    /* Calling pcwrite 10,000 times */
    for(i=0;i<10000;i++)
        rc=pcwrite(ne,no,subp,bufp);

    time(&t2); /* Get end time */

    printf("%4.1f\n", difftime(t2,t1)) ;
    printf("-> End 128-word write loop.\n");

    rc=pcclose();
}

```

4-2 Measurement Results

The section provides the average times required for each execution of a read or write (i.e., for a function call) using the program presented above.

4-2-1 Measurement Conditions

System configuration: No Slave Racks or Expansion I/O Racks
 Memory model: Large model
 SYSMAC Board: C200PC-ISA01-E
 C200PC-ISA02-DRM-E, C200PC-ISA12-DRM-E
 C200PC-ISA02-SRM-E, C200PC-ISA12-SRM-E
 Board operating mode: MONITOR
 Board cycle time: 1.2 ms (ISA01), 3.1 ms (-DRM), 1.8 ms (-SRM),
 20 ms (all), 50 ms (all)
 Computer used: IBM PC/AT or compatible computer
 120-MHz Pentium (32 Mbytes of memory, executed by
 Windows 95 MS-DOS prompt)
 400-MHz Pentium II (128 Mbytes of memory, executed
 by Windows 98 MS-DOS prompt)
 Areas: All writeable areas
 Numbers of words: Max. No. of words for each area (AR: 28 words,
 LR: 64 words, HR: 100 words, other: 128 words)

4-2-2 Results

DOS/V (120 MHz) and C200PC-ISA01-E

pcread

Cycle time	PLC mode	Memory model	Average time per call (ms)											
			CIO		SR		AR		LR		HR		Timer PV	
			1 word	128 words	1 word	128 words	1 word	28 words	1 word	64 words	1 word	100 words	1 word	128 words
1.2 ms	MONITOR	Large	2.10	4.10	2.00	4.20	2.10	3.70	2.00	3.40	2.00	3.90	2.10	4.10
20 ms	MONITOR	Large	20.00	20.30	20.00	20.30	20.00	20.00	20.00	20.10	20.00	20.00	20.00	20.30
50 ms	MONITOR	Large	50.00	50.00	50.00	50.00	50.00	50.00	50.00	50.00	50.00	50.00	50.00	50.00

Cycle time	PLC mode	Memory model	Average time per call (ms)											
			Counter PV		Timer Flag		Counter Flag		DM		Extended DM		EM (0)	
			1 word	128 words	1 word	128 words	1 word	128 words	1 word	128 words	1 word	128 words	1 word	128 words
1.2 ms	MONITOR	Large	2.00	4.20	2.10	4.40	2.00	4.40	2.10	4.20	2.10	4.10	2.30	4.20
20 ms	MONITOR	Large	20.00	20.30	20.00	23.90	20.00	24.00	20.00	20.30	20.00	28.30	20.00	20.80
50 ms	MONITOR	Large	50.00	50.10	50.00	50.10	49.90	50.10	50.00	50.00	50.10	50.00	50.00	50.00

pcwrite

Cycle time	PLC mode	Memory model	Average time per call (ms)											
			CIO		SR		AR		LR		HR		Timer PV	
			1 word	128 words	1 word	128 words	1 word	28 words	1 word	64 words	1 word	100 words	1 word	128 words
1.2 ms	MONITOR	Large	2.50	4.50	2.50	4.50	2.50	3.50	2.40	3.80	2.40	4.50	2.60	5.00
20 ms	MONITOR	Large	20.00	40.00	20.00	40.00	20.00	20.00	20.00	20.00	20.00	40.00	20.00	40.00
50 ms	MONITOR	Large	50.00	50.00	50.00	50.00	50.00	50.00	50.10	50.00	50.00	50.00	50.00	50.00

Cycle time	PLC mode	Memory model	Average time per call (ms)							
			Counter PV		DM		Extended DM		EM (0)	
			1 word	128 words	1 word	128 words	1 word	128 words	1 word	128 words
1.2 ms	MONITOR	Large	2.50	5.00	2.60	4.70	2.60	4.70	2.70	5.10
20 ms	MONITOR	Large	20.00	40.00	20.10	40.00	20.00	40.00	20.00	40.10
50 ms	MONITOR	Large	50.00	50.00	50.00	50.00	50.00	50.00	50.00	50.00

DOS/V (120 MHz) and C200PC-ISA02-DRM-E**pcread**

Cycle time	PLC mode	Memory model	Average time per call (ms)											
			CIO		SR		AR		LR		HR		Timer PV	
			1 word	128 words	1 word	128 words	1 word	28 words	1 word	64 words	1 word	100 words	1 word	128 words
3.1 ms	MONITOR	Large	6.90	10.10	6.90	10.20	6.90	9.50	6.90	10.50	6.90	11.40	6.90	10.10
20 ms	MONITOR	Large	20.00	20.30	20.00	20.30	20.00	20.00	20.00	20.00	20.00	20.00	20.00	20.30
50 ms	MONITOR	Large	50.00	50.00	50.00	50.00	50.00	50.00	50.00	50.00	50.00	50.00	50.00	50.00

Cycle time	PLC mode	Memory model	Average time per call (ms)											
			Counter PV		Timer Flag		Counter Flag		DM		Extended DM		EM (0)	
			1 word	128 words	1 word	128 words	1 word	128 words	1 word	128 words	1 word	128 words	1 word	128 words
3.1 ms	MONITOR	Large	6.90	10.20	6.90	10.90	7.00	10.90	6.90	10.20	7.00	11.60	7.10	10.30
20 ms	MONITOR	Large	20.00	20.30	20.00	24.10	20.00	24.00	20.00	20.30	20.00	28.30	20.00	20.90
50 ms	MONITOR	Large	50.00	50.00	50.00	50.00	50.00	50.00	50.00	50.10	50.00	50.00	50.00	50.00

pcwrite

Cycle time	PLC mode	Memory model	Average time per call (ms)											
			CIO		SR		AR		LR		HR		Timer PV	
			1 word	128 words	1 word	128 words	1 word	28 words	1 word	64 words	1 word	100 words	1 word	128 words
3.1 ms	MONITOR	Large	7.20	12.20	7.20	12.40	7.30	11.30	7.30	11.90	7.30	10.70	7.40	11.00
20 ms	MONITOR	Large	20.00	40.00	20.00	40.00	20.00	20.00	20.10	20.00	20.00	40.00	20.00	40.10
50 ms	MONITOR	Large	50.00	50.00	50.00	50.00	50.00	50.00	50.00	50.00	50.00	50.00	50.00	50.00

Cycle time	PLC mode	Memory model	Average time per call (ms)							
			Counter PV		DM		Extended DM		EM (0)	
			1 word	128 words	1 word	128 words	1 word	128 words	1 word	128 words
3.1 ms	MONITOR	Large	7.30	11.00	7.50	12.30	7.20	10.90	7.60	11.10
20 ms	MONITOR	Large	20.00	40.00	20.00	40.00	20.00	40.00	20.00	40.00
50 ms	MONITOR	Large	50.00	50.00	50.00	50.00	50.00	50.00	50.00	50.00

DOS/V (120 MHz) and C200PC-ISA12-DRM-E**pcread**

Cycle time	PLC mode	Memory model	Average time per call (ms)											
			CIO		SR		AR		LR		HR		Timer PV	
			1 word	128 words	1 word	128 words	1 word	28 words	1 word	64 words	1 word	100 words	1 word	128 words
3.1 ms	MONITOR	Large	6.50	10.00	6.50	9.90	6.50	9.00	6.50	9.40	6.60	9.10	6.60	9.80
20 ms	MONITOR	Large	20.00	20.30	20.00	20.30	20.20	20.00	20.00	20.00	20.00	20.00	20.00	20.30
50 ms	MONITOR	Large	50.00	50.00	50.00	50.10	50.40	50.00	50.30	50.10	50.00	50.00	50.00	50.00

Cycle time	PLC mode	Memory model	Average time per call (ms)											
			Counter PV		Timer Flag		Counter Flag		DM		Extended DM		EM (0)	
			1 word	128 words	1 word	128 words	1 word	128 words	1 word	128 words	1 word	128 words	1 word	128 words
3.1 ms	MONITOR	Large	6.50	9.80	6.50	9.60	6.60	9.50	6.50	9.80	6.40	9.30	6.40	9.80
20 ms	MONITOR	Large	20.00	20.30	20.00	24.00	20.00	24.00	20.00	20.30	20.00	28.20	20.00	20.90
50 ms	MONITOR	Large	50.00	50.60	50.00	50.00	50.00	50.00	50.00	50.00	50.00	50.00	50.00	50.00

Cycle time	PLC mode	Memory model	Average time per call (ms)			
			Counter PV		Timer Flag	
			1 word	128 words	1 word	128 words
3.1 ms	MONITOR	Large	6.30	9.80	6.30	9.80
20 ms	MONITOR	Large	20.00	20.90	20.00	21.00
50 ms	MONITOR	Large	50.00	50.60	50.10	50.20

pcwrite

Cycle time	PLC mode	Memory model	Average time per call (ms)											
			CIO		SR		AR		LR		HR		Timer PV	
			1 word	128 words	1 word	128 words	1 word	28 words	1 word	64 words	1 word	100 words	1 word	128 words
3.1 ms	MONITOR	Large	6.60	9.90	6.80	10.40	6.70	9.10	6.60	9.00	6.70	10.50	6.80	11.10
20 ms	MONITOR	Large	20.00	40.00	20.00	40.00	20.00	20.10	20.00	20.00	20.00	39.70	20.00	40.00
50 ms	MONITOR	Large	50.00	50.00	50.80	50.00	50.00	50.60	50.20	50.10	50.00	50.00	50.00	50.20

Cycle time	PLC mode	Memory model	Average time per call (ms)											
			Counter PV		DM		Extended DM		EM (0)		EM (1)		EM (2)	
			1 word	128 words	1 word	128 words	1 word	128 words	1 word	128 words	1 word	128 words	1 word	128 words
3.1 ms	MONITOR	Large	6.80	11.20	6.90	10.00	6.80	9.90	6.90	10.90	6.90	10.80	6.90	10.90
20 ms	MONITOR	Large	20.00	40.00	20.00	40.00	20.00	40.00	20.00	40.00	20.00	40.00	20.00	40.00
50 ms	MONITOR	Large	50.10	50.00	50.00	50.00	50.00	50.00	50.00	50.00	50.00	50.00	50.00	50.00

DOS/V (120 MHz) and C200PC-ISA02-SRM-E**pcread**

Cycle time	PLC mode	Memory model	Average time per call (ms)											
			CIO		SR		AR		LR		HR		Timer PV	
			1 word	128 words	1 word	128 words	1 word	28 words	1 word	64 words	1 word	100 words	1 word	128 words
1.8 ms	MONITOR	Large	4.10	5.70	4.30	5.60	4.10	4.90	4.10	5.70	4.10	5.40	4.10	5.60
20 ms	MONITOR	Large	20.00	20.30	20.00	20.60	20.10	20.00	20.00	20.00	20.00	20.00	20.00	20.30
50 ms	MONITOR	Large	50.00	50.00	50.00	50.00	50.10	50.00	50.00	50.00	50.00	50.00	50.00	50.00

Cycle time	PLC mode	Memory model	Average time per call (ms)											
			Counter PV		Timer Flag		Counter Flag		DM		Extended DM		EM (0)	
			1 word	128 words	1 word	128 words	1 word	128 words	1 word	128 words	1 word	128 words	1 word	128 words
1.8 ms	MONITOR	Large	4.10	5.70	4.20	5.00	4.10	5.10	4.10	5.70	4.30	5.70	4.40	5.50
20 ms	MONITOR	Large	20.00	20.40	20.00	24.30	20.00	24.30	20.00	20.40	20.00	27.90	20.10	21.30
50 ms	MONITOR	Large	50.90	50.00	50.00	50.00	50.00	50.00	50.00	50.00	50.00	50.00	50.10	50.00

pcwrite

Cycle time	PLC mode	Memory model	Average time per call (ms)											
			CIO		SR		AR		LR		HR		Timer PV	
			1 word	128 words	1 word	128 words	1 word	28 words	1 word	64 words	1 word	100 words	1 word	128 words
1.8 ms	MONITOR	Large	4.60	6.10	4.70	6.80	4.70	5.60	4.70	5.70	4.70	5.80	4.80	7.10
20 ms	MONITOR	Large	20.00	40.00	20.10	40.40	20.00	20.00	20.00	20.00	20.00	40.00	20.00	40.00
50 ms	MONITOR	Large	50.00	50.00	50.00	50.30	50.00	50.00	50.00	50.00	50.00	50.00	50.00	50.30

Cycle time	PLC mode	Memory model	Average time per call (ms)							
			Counter PV		DM		Extended DM		EM (0)	
			1 word	128 words	1 word	128 words	1 word	128 words	1 word	128 words
1.8 ms	MONITOR	Large	4.70	7.10	4.90	6.30	4.70	5.50	5.00	6.30
20 ms	MONITOR	Large	20.00	40.00	20.00	40.00	20.00	40.50	20.10	40.10
50 ms	MONITOR	Large	50.00	50.00	50.00	50.00	50.00	50.30	50.00	50.00

DOS/V (120 MHz) and C200PC-ISA12-SRM-E

pcread

Cycle time	PLC mode	Memory model	Average time per call (ms)											
			CIO		SR		AR		LR		HR		Timer PV	
			1 word	128 words	1 word	128 words	1 word	28 words	1 word	64 words	1 word	100 words	1 word	128 words
1.8 ms	MONITOR	Large	3.30	5.80	3.30	5.70	3.30	5.20	3.30	4.80	3.30	9.90	3.30	5.70
20 ms	MONITOR	Large	20.10	20.30	20.00	20.30	20.00	20.00	20.00	20.00	20.00	20.00	20.00	20.40
50 ms	MONITOR	Large	50.00	50.00	50.10	50.00	50.00	50.00	50.00	50.00	50.00	50.00	50.00	50.00

Cycle time	PLC mode	Memory model	Average time per call (ms)											
			Counter PV		Timer Flag		Counter Flag		DM		Extended DM		EM (0)	
			1 word	128 words	1 word	128 words	1 word	128 words	1 word	128 words	1 word	128 words	1 word	128 words
1.8 ms	MONITOR	Large	3.30	5.70	3.30	5.50	3.30	5.50	3.30	5.70	3.50	5.10	3.60	5.60
20 ms	MONITOR	Large	20.00	20.30	20.00	24.20	20.00	24.20	20.00	20.40	20.00	27.60	20.00	20.90
50 ms	MONITOR	Large	50.00	50.00	50.00	50.00	50.00	50.00	50.00	50.00	50.00	50.00	50.00	50.00

Cycle time	PLC mode	Memory model	Average time per call (ms)			
			Counter PV		Timer Flag	
			1 word	128 words	1 word	128 words
1.8 ms	MONITOR	Large	3.60	5.70	3.60	5.70
20 ms	MONITOR	Large	20.00	20.90	20.00	20.80
50 ms	MONITOR	Large	50.00	50.00	50.00	50.00

pcwrite

Cycle time	PLC mode	Memory model	Average time per call (ms)											
			CIO		SR		AR		LR		HR		Timer PV	
			1 word	128 words	1 word	128 words	1 word	28 words	1 word	64 words	1 word	100 words	1 word	128 words
1.8 ms	MONITOR	Large	3.80	5.70	3.90	5.70	3.90	4.50	3.90	4.80	3.80	6.00	3.90	6.60
20 ms	MONITOR	Large	20.00	40.00	20.00	40.10	20.00	20.00	20.00	20.00	20.00	39.70	20.00	40.00
50 ms	MONITOR	Large	50.00	50.00	50.00	50.00	50.00	50.00	50.00	50.00	50.00	50.00	50.00	50.00

Cycle time	PLC mode	Memory model	Average time per call (ms)											
			Counter PV		DM		Extended DM		EM (0)		EM (1)		EM (2)	
			1 word	128 words	1 word	128 words	1 word	128 words	1 word	128 words	1 word	128 words	1 word	128 words
1.8 ms	MONITOR	Large	3.90	6.60	4.00	5.80	3.90	6.00	4.20	6.40	4.20	6.40	4.20	6.40
20 ms	MONITOR	Large	20.00	40.10	20.00	40.00	20.00	40.00	20.00	40.00	20.00	40.00	20.00	40.00
50 ms	MONITOR	Large	50.00	50.00	50.00	50.00	50.00	50.00	50.00	50.00	50.00	50.00	50.00	50.00

DOS/V (400 MHz) and C200PC-ISA01-E**pcread**

Cycle time	PLC mode	Memory model	Average time per call (ms)											
			CIO		SR		AR		LR		HR		Timer PV	
			1 word	128 words	1 word	128 words	1 word	28 words	1 word	64 words	1 word	100 words	1 word	128 words
1.2 ms	MONITOR	Large	2.10	4.10	2.20	4.00	2.10	3.60	2.20	3.20	2.10	4.30	2.20	4.00
20 ms	MONITOR	Large	20.00	20.20	20.10	20.20	20.10	20.00	20.00	20.00	20.00	20.00	20.00	20.20
50 ms	MONITOR	Large	50.00	50.00	50.10	50.00	50.00	50.00	50.00	50.00	50.00	50.10	50.00	50.00

Cycle time	PLC mode	Memory model	Average time per call (ms)											
			Counter PV		Timer Flag		Counter Flag		DM		Extended DM		EM (0)	
			1 word	128 words	1 word	128 words	1 word	128 words	1 word	128 words	1 word	128 words	1 word	128 words
1.2 ms	MONITOR	Large	2.20	4.10	2.20	4.40	2.20	4.30	2.20	4.10	2.10	4.50	1.90	4.00
20 ms	MONITOR	Large	20.00	20.20	20.00	23.60	20.00	23.40	20.00	20.30	20.00	24.80	20.00	20.30
50 ms	MONITOR	Large	50.00	50.10	50.00	50.00	50.00	50.00	50.00	50.00	50.00	50.00	50.00	50.00

pcwrite

Cycle time	PLC mode	Memory model	Average time per call (ms)											
			CIO		SR		AR		LR		HR		Timer PV	
			1 word	128 words	1 word	128 words	1 word	28 words	1 word	64 words	1 word	100 words	1 word	128 words
1.2 ms	MONITOR	Large	2.00	4.90	2.00	4.70	2.00	3.70	2.00	4.20	2.00	4.40	2.00	4.80
20 ms	MONITOR	Large	20.00	40.00	20.00	40.00	20.00	20.00	20.00	20.00	20.00	38.10	20.00	40.10
50 ms	MONITOR	Large	50.00	50.00	50.00	50.00	50.00	50.00	50.00	50.00	50.00	50.00	50.00	50.00

Cycle time	PLC mode	Memory model	Average time per call (ms)							
			Counter PV		DM		Extended DM		EM (0)	
			1 word	128 words	1 word	128 words	1 word	128 words	1 word	128 words
1.2 ms	MONITOR	Large	2.00	4.80	2.10	5.20	2.00	4.80	2.30	5.00
20 ms	MONITOR	Large	20.00	40.00	20.00	40.00	20.00	39.90	20.00	40.00
50 ms	MONITOR	Large	50.00	50.10	50.00	50.00	50.00	50.00	50.00	50.00

DOS/V (400 MHz) and C200PC-ISA02-DRM-E**pcread**

Cycle time	PLC mode	Memory model	Average time per call (ms)											
			CIO		SR		AR		LR		HR		Timer PV	
			1 word	128 words	1 word	128 words	1 word	28 words	1 word	64 words	1 word	100 words	1 word	128 words
3.1 ms	MONITOR	Large	5.70	8.70	5.60	8.80	5.70	7.30	5.60	8.40	5.60	8.40	5.70	8.70
20 ms	MONITOR	Large	20.00	20.30	20.00	20.20	20.00	20.00	20.00	20.00	20.00	20.10	20.00	20.20
50 ms	MONITOR	Large	50.10	50.00	50.00	50.00	50.10	50.00	50.00	50.00	50.00	50.00	50.00	50.00

Cycle time	PLC mode	Memory model	Average time per call (ms)											
			Counter PV		Timer Flag		Counter Flag		DM		Extended DM		EM (0)	
			1 word	128 words	1 word	128 words	1 word	128 words	1 word	128 words	1 word	128 words	1 word	128 words
3.1 ms	MONITOR	Large	5.70	8.70	5.70	8.80	5.70	8.80	5.70	8.80	5.70	8.70	5.80	8.50
20 ms	MONITOR	Large	20.00	20.30	20.00	23.40	20.00	23.40	20.00	20.30	20.00	24.90	20.00	20.20
50 ms	MONITOR	Large	50.00	50.00	50.00	50.00	50.00	50.00	50.00	50.00	50.00	50.00	50.00	50.00

pcwrite

Cycle time	PLC mode	Memory model	Average time per call (ms)											
			CIO		SR		AR		LR		HR		Timer PV	
			1 word	128 words	1 word	128 words	1 word	28 words	1 word	64 words	1 word	100 words	1 word	128 words
3.1 ms	MONITOR	Large	5.70	9.10	5.70	9.30	5.70	8.30	5.70	8.50	5.70	8.90	5.80	9.90
20 ms	MONITOR	Large	20.00	40.00	20.10	40.00	20.00	20.00	20.00	20.00	20.00	38.00	20.00	40.10
50 ms	MONITOR	Large	50.00	50.00	50.00	50.00	50.00	50.00	50.10	50.00	50.00	50.00	50.00	50.10

Cycle time	PLC mode	Memory model	Average time per call (ms)							
			Counter PV		DM		Extended DM		EM (0)	
			1 word	128 words	1 word	128 words	1 word	128 words	1 word	128 words
3.1 ms	MONITOR	Large	5.80	9.90	5.80	9.30	5.70	9.10	5.70	9.50
20 ms	MONITOR	Large	20.00	40.00	20.00	40.00	20.00	39.90	20.00	40.10
50 ms	MONITOR	Large	50.10	50.00	50.00	50.10	50.00	50.00	50.00	50.00

DOS/V (400 MHz) and C200PC-ISA12-DRM-E**pcread**

Cycle time	PLC mode	Memory model	Average time per call (ms)											
			CIO		SR		AR		LR		HR		Timer PV	
			1 word	128 words	1 word	128 words	1 word	28 words	1 word	64 words	1 word	100 words	1 word	128 words
3.1 ms	MONITOR	Large	5.40	8.40	5.40	8.50	5.70	7.80	5.30	7.90	5.20	8.20	5.30	8.50
20 ms	MONITOR	Large	20.00	20.20	20.00	20.20	20.00	20.00	20.00	20.10	20.00	20.00	20.00	20.20
50 ms	MONITOR	Large	50.10	50.00	50.00	50.00	50.00	50.00	50.00	50.00	50.00	50.00	50.00	50.10

Cycle time	PLC mode	Memory model	Average time per call (ms)											
			Counter PV		Timer Flag		Counter Flag		DM		Extended DM		EM (0)	
			1 word	128 words	1 word	128 words	1 word	128 words	1 word	128 words	1 word	128 words	1 word	128 words
3.1 ms	MONITOR	Large	5.20	8.50	5.30	8.20	5.40	8.20	5.20	8.50	5.40	8.50	5.80	8.40
20 ms	MONITOR	Large	20.00	20.30	20.00	23.40	20.00	23.40	20.00	20.20	20.00	25.00	20.00	20.20
50 ms	MONITOR	Large	50.00	50.00	50.00	50.00	50.00	50.10	50.00	50.00	50.00	50.00	50.00	50.00

Cycle time	PLC mode	Memory model	Average time per call (ms)			
			Counter PV		Timer Flag	
			1 word	128 words	1 word	128 words
3.1 ms	MONITOR	Large	5.80	8.40	5.70	8.40
20 ms	MONITOR	Large	20.00	20.30	20.00	20.30
50 ms	MONITOR	Large	50.00	50.00	50.00	50.00

pcwrite

Cycle time	PLC mode	Memory model	Average time per call (ms)											
			CIO		SR		AR		LR		HR		Timer PV	
			1 word	128 words	1 word	128 words	1 word	28 words	1 word	64 words	1 word	100 words	1 word	128 words
3.1 ms	MONITOR	Large	5.80	8.80	5.90	9.00	5.80	8.20	5.70	8.30	5.70	8.90	5.80	9.60
20 ms	MONITOR	Large	20.00	40.00	20.00	40.00	20.00	20.00	20.00	20.00	20.00	37.10	20.10	40.00
50 ms	MONITOR	Large	50.00	50.00	50.00	50.00	50.00	50.00	50.00	50.10	50.00	50.00	50.00	50.00

Cycle time	PLC mode	Memory model	Average time per call (ms)											
			Counter PV		DM		Extended DM		EM (0)		EM (1)		EM (2)	
			1 word	128 words	1 word	128 words	1 word	128 words	1 word	128 words	1 word	128 words	1 word	128 words
3.1 ms	MONITOR	Large	5.80	9.70	5.90	9.20	5.80	8.90	6.00	9.30	6.00	9.40	6.00	9.40
20 ms	MONITOR	Large	20.00	40.00	20.00	40.00	20.00	39.90	20.10	40.00	20.00	40.00	20.00	40.00
50 ms	MONITOR	Large	50.00	50.00	50.00	50.00	50.10	50.00	50.00	50.00	50.00	50.00	50.00	50.00

DOS/V (400 MHz) and C200PC-ISA02-SRM-E

pcread

Cycle time	PLC mode	Memory model	Average time per call (ms)											
			CIO		SR		AR		LR		HR		Timer PV	
			1 word	128 words	1 word	128 words	1 word	28 words	1 word	64 words	1 word	100 words	1 word	128 words
1.8 ms	MONITOR	Large	3.60	5.50	3.60	5.40	3.50	4.80	3.60	5.30	3.60	5.00	3.60	5.40
20 ms	MONITOR	Large	20.00	20.20	20.00	20.30	20.00	20.00	20.00	20.00	20.00	20.00	20.00	20.30
50 ms	MONITOR	Large	50.00	50.00	50.00	50.00	50.00	50.00	50.00	50.00	50.00	50.00	50.00	50.00

Cycle time	PLC mode	Memory model	Average time per call (ms)											
			Counter PV		Timer Flag		Counter Flag		DM		Extended DM		EM (0)	
			1 word	128 words	1 word	128 words	1 word	128 words	1 word	128 words	1 word	128 words	1 word	128 words
1.8 ms	MONITOR	Large	3.60	5.40	3.70	4.90	3.70	4.90	3.60	5.40	3.70	5.20	3.80	5.40
20 ms	MONITOR	Large	20.00	20.20	20.00	23.30	20.00	23.20	20.10	20.20	20.10	25.50	20.00	20.30
50 ms	MONITOR	Large	50.00	50.00	50.00	50.10	50.00	50.00	50.00	50.00	50.00	50.00	50.00	50.10

pcwrite

Cycle time	PLC mode	Memory model	Average time per call (ms)											
			CIO		SR		AR		LR		HR		Timer PV	
			1 word	128 words	1 word	128 words	1 word	28 words	1 word	64 words	1 word	100 words	1 word	128 words
1.8 ms	MONITOR	Large	3.80	5.50	4.00	6.30	3.90	5.00	3.90	5.10	3.90	5.80	4.00	6.50
20 ms	MONITOR	Large	20.00	40.00	20.00	40.00	20.00	20.00	20.00	20.10	20.00	38.50	20.00	40.00
50 ms	MONITOR	Large	50.00	50.00	50.10	50.00	50.00	50.00	50.00	50.00	50.10	50.00	50.00	50.00

Cycle time	PLC mode	Memory model	Average time per call (ms)									
			Counter PV		DM		Extended DM		EM (0)			
			1 word	128 words	1 word	128 words	1 word	128 words	1 word	128 words		
1.8 ms	MONITOR	Large	4.00	6.50	4.10	5.60	3.90	5.60	4.10	6.30		
20 ms	MONITOR	Large	20.00	40.10	20.00	40.00	20.00	39.90	20.00	40.00		
50 ms	MONITOR	Large	50.00	50.00	50.00	50.00	50.00	50.00	50.00	50.00		

DOS/V (400 MHz) and C200PC-ISA12-SRM-E**pcread**

Cycle time	PLC mode	Memory model	Average time per call (ms)											
			CIO		SR		AR		LR		HR		Timer PV	
			1 word	128 words	1 word	128 words	1 word	28 words	1 word	64 words	1 word	100 words	1 word	128 words
1.8 ms	MONITOR	Large	2.80	5.30	2.80	5.40	2.80	4.90	2.80	4.60	2.90	4.80	2.90	5.30
20 ms	MONITOR	Large	20.00	20.30	20.00	20.20	20.00	20.00	20.00	20.00	20.00	20.00	20.00	20.30
50 ms	MONITOR	Large	50.00	50.00	50.10	50.00	50.00	50.00	50.00	50.00	50.00	50.00	50.00	50.00

Cycle time	PLC mode	Memory model	Average time per call (ms)											
			Counter PV		Timer Flag		Counter Flag		DM		Extended DM		EM (0)	
			1 word	128 words	1 word	128 words	1 word	128 words	1 word	128 words	1 word	128 words	1 word	128 words
1.8 ms	MONITOR	Large	2.80	5.30	2.90	5.40	2.80	5.40	2.90	5.30	2.90	5.10	3.10	5.30
20 ms	MONITOR	Large	20.10	20.20	20.00	23.30	20.00	23.20	20.00	20.20	20.00	25.40	20.00	20.30
50 ms	MONITOR	Large	50.00	50.00	50.00	50.10	50.00	50.00	50.00	50.00	50.00	50.00	50.00	50.00

Cycle time	PLC mode	Memory model	Average time per call (ms)			
			Counter PV		Timer Flag	
			1 word	128 words	1 word	128 words
1.8 ms	MONITOR	Large	3.10	5.30	3.10	5.30
20 ms	MONITOR	Large	20.00	20.20	20.00	20.30
50 ms	MONITOR	Large	50.00	50.10	50.00	50.00

pcwrite

Cycle time	PLC mode	Memory model	Average time per call (ms)											
			CIO		SR		AR		LR		HR		Timer PV	
			1 word	128 words	1 word	128 words	1 word	28 words	1 word	64 words	1 word	100 words	1 word	128 words
1.8 ms	MONITOR	Large	3.10	5.60	3.20	5.20	3.20	4.40	3.00	4.80	3.10	5.50	3.20	6.20
20 ms	MONITOR	Large	20.00	40.00	20.10	40.00	20.00	20.00	20.00	20.00	20.00	37.30	20.00	40.00
50 ms	MONITOR	Large	50.00	50.00	50.00	50.00	50.00	50.00	50.00	50.00	50.00	50.10	50.00	50.10

Cycle time	PLC mode	Memory model	Average time per call (ms)											
			Counter PV		DM		Extended DM		EM (0)		EM (1)		EM (2)	
			1 word	128 words	1 word	128 words	1 word	128 words	1 word	128 words	1 word	128 words	1 word	128 words
1.8 ms	MONITOR	Large	3.20	6.20	3.30	5.60	3.20	5.70	3.50	6.20	3.50	6.10	3.50	6.20
20 ms	MONITOR	Large	20.00	40.00	20.00	40.00	20.00	40.00	20.00	40.00	20.10	40.00	20.00	40.00
50 ms	MONITOR	Large	50.00	50.00	50.00	50.00	50.00	50.00	50.00	50.00	50.00	50.00	50.00	50.00

SECTION 5

Troubleshooting

This section provides a table of problems that can occur with the SYSMAC Board along with possible corrections. Refer to the *C200HX/HG/HE (-E/-ZE) Installation Guide* (W302) for information on troubleshooting other PLC problems and to the *CompoBus/D Operation Manual* (W267) for CompoBus/D problems.

Problem	Probable cause	Possible correction
Communications time out (return value = 5) and the PLC communications library cannot be used.	The same I/O port addresses or shared memory address are being used by another device.	Use addresses that are not being used by another device. Change the DIP switch setting of the I/O port base address first before declaring it with the <code>pcinit</code> function.
	The I/O port base address in the <code>pcinit</code> function is not the same as the one set on the DIP switch.	Change either the setting of the DIP switch or the I/O port base address declared with the <code>pcinit</code> function so that both use the same address.
The PLC communications library is open (return value = 9) and the I/O port base address and the shared memory base address cannot be changed.	<code>pcclose</code> has not been executed since the PLC communications library was opened.	The addresses can be changed only when the PLC communications library is closed. Execute <code>pcclose</code> and then set the addresses with the <code>pcinit</code> function.
A response error occurs (return value = 7) and a normal response cannot be received.	An attempt was made to clear an error when the SYSMAC Board was not in PROGRAM mode.	Change the SYSMAC Board to PROGRAM or MONITOR mode and then execute the function again.
	An attempt was made to clear an error in RUN mode.	
	An attempt was made to write data to a memory area when the SYSMAC Board was in RUN mode.	
Illegal parameters are detected (return value = 5)	There was a mistake in the parameters given to the PLC communications library.	Check the values, ranges, and text strings for the functions and then execute the functions again after correcting the bad parameter.
The PLC communications library is not open (return value = 1) and the PLC communications library cannot be used.	<code>pcopen</code> has not been executed yet or <code>pcclose</code> has been executed.	Execute <code>pcopen</code> before executing any other functions except <code>pcinit</code> .
The PLC communications library is already open (return value = 2).	An attempt was made to open the PLC communications library when it was already open.	Do not execute <code>pcopen</code> when the PLC communications library is already open.

Appendix A

Replacing the Battery

The SYSMAC Board contains a battery that backs up memory data when power is turned off. If the voltage of the battery drops, HR area bits, DM area words, the PC Setup, and other data will not be maintained properly. Change the battery as soon as the voltage begins to drop.

The SYSMAC Board contains the following battery.
OMRON Battery Set: C200H-BAT09

⚠ WARNING Never touch the Board or the interior of any other Unit or attempt to disassemble a Unit while power is being supplied. Doing so may result in serious electrical shock or electrocution.

Note You may not be able to replace the battery in time to save the data unless you have a spare battery on hand. Prepare a spare battery in advance so that it is ready when you need it.

Battery Life

The life of the battery is 5 years at 25°C. The battery life will be shortened if it is used at higher temperatures. A battery error (non-fatal) will occur when the voltage of the battery drops and the following indications will be made.

- The ER/ALARM indicator will flash.
 - SR 25308 will turn ON.
 - A battery error will be registered in the SYSMAC Board and can be read out from a Programming Device.
- Detection of a low battery voltage can, however, be disabled in the PC Setup (DM 6655). If detection is disabled, none of the above indications will be made, i.e., a battery error will not occur regardless of the voltage of the battery.

- Note**
1. Replace the battery within one week after the battery error occurs. Program and other data will be lost if the battery is not replaced.
 2. The battery must be replaced every 5 years even if a battery error does not occur. Never use a battery longer than 5 years.

Replacement Procedure

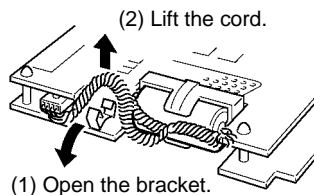
Use the following procedure to replace the battery.

- Note**
1. Before starting to replace the battery, take appropriate anti-static measures, such as grounding, to prevent computer circuits from being destroyed.
 2. There are shape points and edges on the SYSMAC Board. Be careful not to touch the components on either side when handling the SYSMAC Board.
 3. The battery may leak if the terminals are shorted or if the battery is charged, disassembled, heated, or incinerated. Never do any of these; handle and dispose of the battery correctly.

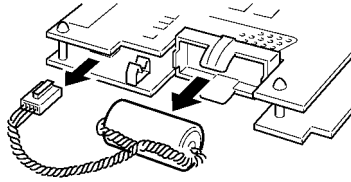
- 1, 2, 3...**
1. If the computer is off, turn it on for at least one minute to let the internal capacitor charge.

Note The internal capacitor is designed to back up data for 5 minutes while replacing the battery. The computer must be on for at least one minute immediately before replacing the battery to ensure that the capacitor is completely charged. Data will be lost if the capacitor is not sufficiently charged.

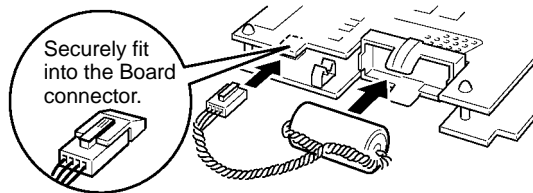
2. Open the chassis of the computer and remove the SYSMAC Board. Refer to your computer's user's manual for the procedure for your computer.
3. Remove the battery cord from the bracket. Open the bracket (1) and lift the cord (2) as shown in the following illustration.



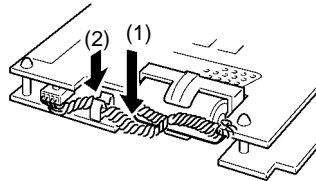
4. Disconnect the battery and connector. Before disconnecting the old battery connector, position the Board so that it is ready for connecting a new battery connector and proceed to the next step.



5. Connect the new battery connector and mount the new battery. While being sure that the connector is correctly orientated, push it in until it securely fits into the connector on the Board side. Be careful not to push the cord.



6. Secure the new battery cord to the SYSMAC Board using the bracket. Push the cord through the opening of the bracket (1) and secure the bracket to the Board so that it will not lift from the Board (2).



7. Replace the SYSMAC Board into the computer and reassemble the chassis, again referring to the user's manual for your computer.
8. After replacing the battery, use the error clear operation to clear the battery error.

Appendix B

Handling Power Interruptions

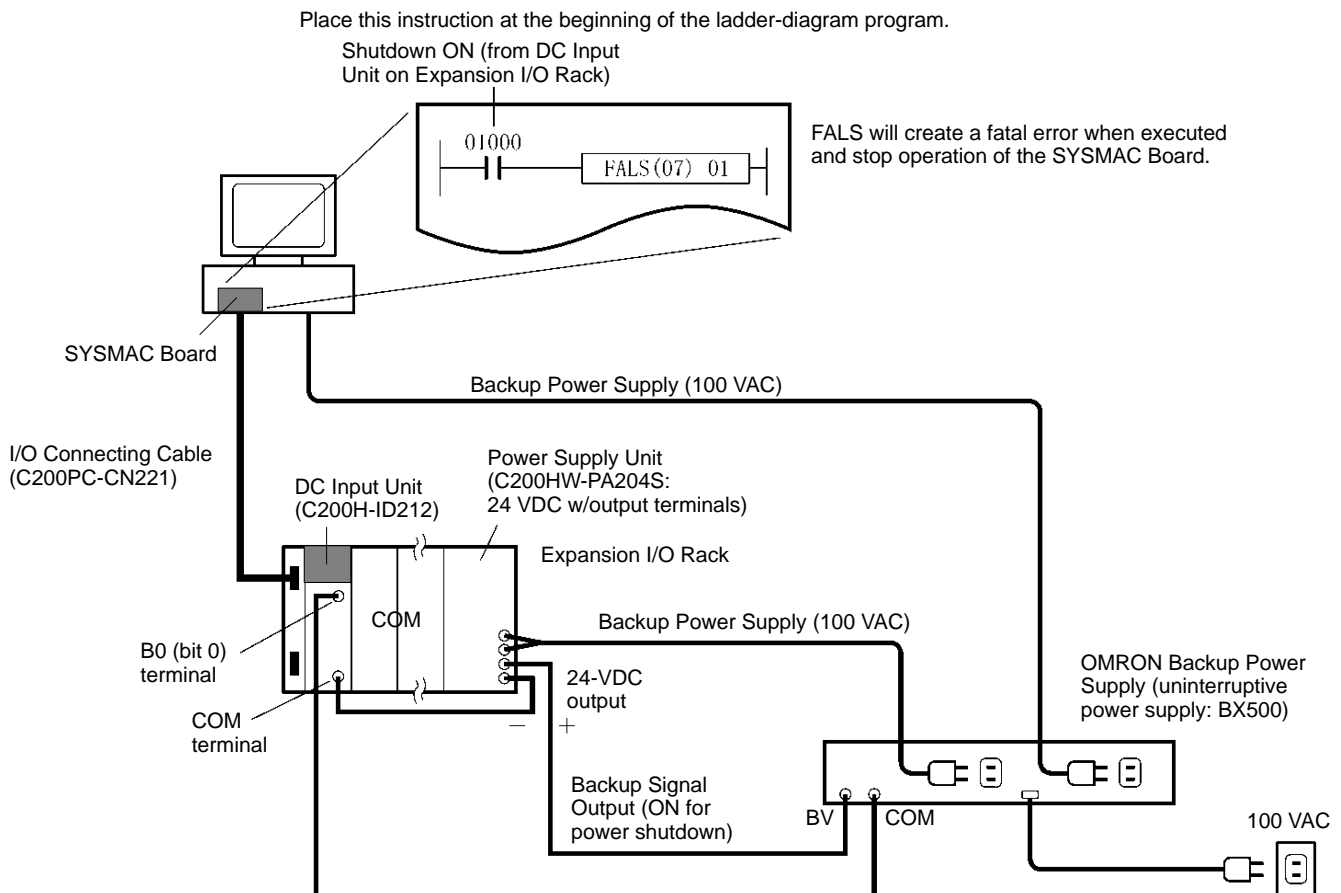
The SYSMAC Board will not detect power interruptions, and thus cannot stop program execution or execute shut-down processing. If the I/O memory is being updated when a power interruption occurs, the data in the DM, HR, and other memory areas may not be saved correctly. The following steps must thus be taken:

- Always switch to PROGRAM mode before turning off power to the personal computer. Never turn off power when the SYSMAC Board is in MONITOR or RUN mode.
- Do one of the following: 1) Configure the system to detect power interruptions, such as those caused by power outages, and stop program execution in the SYSMAC Board before allowing power to be turned off or 2) Restore the desired default values to the DM area any other required areas before starting operation.

The following configuration can be used as reference in configuring the system to implement item 1), above. In this configuration, the shutdown signal from an uninterruptive power supply is input to an Input Unit on an Expansion I/O Rack. This input is used to cause an FALS instruction (fatal system error) to be executed and thus stop SYSMAC Board operation.

In this configuration, an OMRON BX500 Backup (uninterruptive) Power Supply is used to input a signal to the SYSMAC Board to stop operation when a power interruption occurs. The Backup Signal Output (BV) from the Backup Power Supply turns ON when a power interruption occurs. This signal is input to an Input Unit on an Expansion I/O Rack. When the signal turns ON, an FALS instruction is executed to create a fatal system error and stop execution of the ladder-diagram program in the SYSMAC Board.

The BX500 Backup Power Supply will provide a maximum of 300 W backup power for 3.5 minutes. The Backup Power Signal can thus be used to in the ladder-diagram program to execute system shutdown processing before FALS is executed, as required by the system. (After the power is reset, clear the FALS and then resume operation.)



Index

B

B7A Interface Unit (group 2), 3
Backup Power Supply, 105
battery
 average life, 103
 replacing, 103
baud rate, CompoBus/D, 16
baud rate, 47

C

C language library, 2, 23
cable, RS-232C port cable specifications, 28
cables
 connecting communications cables, 53
 connecting transmission cables, 49
communications, 23
 closing communications with the Board, 66
 opening communications with the Board, 65
Communications Boards, 24
communications cables, connecting, 53
communications distance, CompoBus/D, 16
communications errors, 48
communications library, 62
 functions, 61
 required files, 59
communications parameters, 45
communications settings, 64
CompoBus/D communications cables, connecting, 53
CompoBus/D Communications DIP Switch, 47, 49
CompoBus/D Configurator, 16
CompoBus/D Master Unit, functionality, 15
CompoBus/D Node Address DIP Switch, 46
CompoBus/D Slaves, 14, 16
CompoBus/D Unit Number Switch, 47, 48
components, Board components, 35
Configurator. *See* CompoBus/D Configurator

D

DeviceNet, 4
Dimensions, 39

E

EEPROM memory transfer, 45
EEPROM Write Protect Switch, 46
error history
 clearing, 68
 reading, 68
errors
 clearing, 68
 processing, 101
Expansion I/O Cables, 12
Expansion I/O Racks, 2, 12
 connecting, 53
expansion instructions, 45

F

FINS commands, 17
floppy disk drive, requirements, 9
functions, SYSMAC Board functions, 22

H

hard disk, requirements, 9
hardware setup, 30
hardware setup, 33
High-density I/O Unit (group 2), 3
Host Link commands, 12
Host Link Unit, 3

I

I/O Allocation, 14
 default allocations, 16
 user-set allocations, 16
I/O port address, setting, 46, 64
I/O ports, 34
Indicators, 43
installation, 30
installation, 33
 Support Board, 49

L

language library, C language library, 2, 23
LED indicators, 43

M

- measurement program, 90
- memory addresses, 34
 - setting, 64
- memory areas, 12
 - CPU, 25
 - reading data, 76
 - reading/writing, 23
 - retained data areas, 24
 - writing data, 82
- message communications, 17
- Mountable Units, 13
- MS (Master status) indicator, 43

N

- node address, setting, 46
- NS (Node status) indicator, 43

O

- operating mode, 24
 - changing, 66
- operating system, requirements, 9
- OS, requirements, 9

P

- pcclose(), 23
- pcclose(), 62, 66
- pcinit(), 23
- pcinit(), 62, 64
- pcmode(), 23, 62, 66
- pcopen(), 23
- pcopen(), 62, 65
- pcread(), 23, 62, 76
- pcstat(), 23, 62, 68
- pcwrite(), 23, 62, 82
- performance
 - measuring, 89
 - read/write performance, 89
- Peripheral Device, connecting, 59

- Personal computer, requirements, 9
- PLC communications library, 62
 - functions, 61
 - required files, 59
- PLC DIP Switch, settings, 44
- PLC ERR/ALARM indicator, 43
- PLC Link Unit, 3
- PLC RUN indicator, 43
- power interruptions, handling, 105
- precautions
 - for Expansion I/O Racks, 53
 - general, xi
- programming, 30
- Programming Devices, 13
- programming instructions, 12

R

- remote I/O communications, 16
 - communications errors, 48
- Remote I/O Master Unit, 3
- requirements, PC system requirements, 9
- RS-232C port, 24
 - cable specifications, 28

S—U

- setup procedures, 30
- Special I/O Units
 - connecting, 2
 - I/O allocation, 15
- specifications, 25
 - Board specifications, 28
 - CompoBus/D communications, 27
 - CPU, 25
 - general, 25
 - RS-232C port cable, 28
- status
 - reading PLC status, 68
 - SYSMAC Board status, 23
- switches, 35
- system configuration, 5
- troubleshooting, 101
- user memory area, enabling, 45

Revision History

A manual revision code appears as a suffix to the catalog number on the front cover of the manual.

Cat. No. V201-E1-1

↑
Revision code

The following table outlines the changes made to the manual during each revision. Page numbers refer to the previous version.

Revision code	Date	Revised content
1	December 1998	Original production