SYSMAC CV-series
ISA Control System

Operation Manual

Produced May 1996

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 the product.

⚠ DANGER! Indicates information that, if not heeded, is likely to result in loss of life or serious injury.

⚠ WARNING Indicates information that, if not heeded, could possibly result in loss of life or serious injury.

⚠ Caution Indicates information that, if not heeded, could result in relatively serious or minor injury, damage to the product, or faulty operation.

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.

The abbreviation “PC” means Programmable Controller and is not used as an abbreviation for anything else.

Unless otherwise indicated, “personal computers” refer to IBM PC/AT or compatible personal computers.

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.

© OMRON, 1996

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form, or by any means, mechanical, electronic, photocopying, recording, or otherwise, without the prior written permission of 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

TABLE OF CONTENTS

About this Manual:

This manual describes the installation and operation of the CV-series ISA Control System and includes the sections described below.

Please read this manual carefully and be sure you understand the information provided before attempting to install and operate the CV-series ISA Control System.

Section 1 provides general precautions for using the ISA Control System.

Section 2 explains the features and system configuration of the ISA Control System.

Section 3 provides basic explanations of the individual system components and their functions. Open each of the boxes and check to be sure that nothing is missing.

Section 4 outlines the steps involved from Developer's Kit, ISA Control Unit, and peripheral device settings on up to the point of operation.

Section 5 explains the hardware and software setup for the Developer's Kit.

Section 6 explains the hardware and software setup for ISA Control Units.

Section 7 introduces the communications and control functions that can be used with the ISA Control Unit, and provides instructions on how to use each of the functions. Use this section to determine the functions needed for your system configuration.

Section 8 introduces the CPU Bus Driver and describes the FINS commands used in the CPU Bus Driver.

Section 9 explains how to process errors and change batteries.

The ***Appendices*** provide information on standard models, specifications, connector pin arrangements, error codes, FINS commands, troubleshooting with FINS commands, and PC and ISA Control Unit memory configurations.

⚠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.

SECTION 1
Precautions

This section provides general precautions for using the ISA Control System.

1-1 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.

1-2 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 using the ISA Control System. Be sure to read this manual before attempting to use the Units and keep this manual close at hand for reference during operation.

1-3 Safety Precautions

⚠ **WARNING** Never attempt to disassemble any Units while power is being supplied. Doing so may result in serious electrical shock.

⚠ **WARNING** Never touch any of the terminals while power is being supplied. Doing so may result in serious electrical shock.

1-4 Operating Environment Precautions

Do not operate the control system in the following places.

- Where the PC is exposed to direct sunlight.
- Where the ambient temperature is below 0°C or over 55°C.
- Where the PC may be affected by condensation due to radical temperature changes.
- Where the ambient humidity is below 10% or over 90%.
- Where there is any corrosive or inflammable gas.
- Where there is excessive dust, saline air, or metal powder.
- Where the PC is affected by vibration or shock.
- Where any water, oil, or chemical may splash on the PC.

1-5 Application Precautions

Observe the following precautions when using the ISA Control System or the Programmable Controller (PC).

- Provide double safety mechanisms to handle incorrect signals that can be generated by broken signal lines or momentary power interruptions.
- Provide external interlock circuits, limit circuits, and other safety circuits in addition to any provided within the PC to ensure safety.
- Use the Units only with the power supplies and voltages specified in the operation manuals. Other power supplies and voltages may damage the Units.
- Take measures to stabilize the power supply to conform to the rated supply if it is not stable.
- Provide circuit breakers and other safety measures to provide protection against shorts in external wiring.
- Do not apply voltages exceeding the rated input voltage to Input Units. The Input Units may be destroyed.
- Do not apply voltages exceeding the maximum switching capacity to Output Units. The Output Units may be destroyed.
- Always disconnect the LG terminal when performing withstand voltage tests.
- Always ground the system to 100 Ω or less when installing the system to protect against electrical shock.
- Install all Units according to instructions in the operation manuals. Improper installation may cause faulty operation.
- Provide proper shielding when installing in the following locations:
    - Locations subject to static electricity or other sources of noise.
    - Locations subject to strong electromagnetic fields.
    - Locations subject to possible exposure to radiation.
    - Locations near to power supply lines.
- Always turn off the power supply to the PC before attempting any of the following. Performing any of the following with the power supply turned on may lead to electrical shock:
    - Mounting or removing any Units (e.g., I/O Units, CPU Unit, etc.) or memory cassettes.
    - Assembling any devices or racks.
    - Connecting or disconnecting any cables or wiring.
- Be sure to tighten Backplane screws, terminal screws, and cable connector screws securely.
- Do not attempt to take any Units apart, to repair any Units, or to modify any Units in any way.

SECTION 2
Features and System Configuration

This section explains the features and system configuration of the ISA Control System.

2-1 Features

The ISA Control Unit is a CPU Special I/O Unit with personal computer functions. It can be connected to CV-series Programmable Controllers and to a wide selection of commercially available peripheral devices for personal computers to configure a highly advanced system. The ISA Control Unit offers the following features.

ISA Board Connectability to Standard Backplanes

The ISA Control Unit can be mounted to the same CV-series CPU Backplane that is currently being used. Moreover, by using an ISA Bus Expander, it is possible to connect up to two half-size commercially available ISA boards for communications, data files, I/O, and so on. This allows the user to set up a system such as that shown in *2-2 System Configuration*.

PCMCIA 2.1 Usability

The ISA Control Unit has two built-in PC card interface slots in conformity with the standards of PCMCIA 2.1 (except for 3.3V cards). Either two type I/II PC cards or one type III PC card can be installed. A wide range of functions can be supported by using commercially available PC cards.

Support for CPU Bus Interface Equivalent to Personal Computer Unit

The ISA Control Unit supports a CPU bus interface equivalent to the CV500-VP□□□ Personal Computer Unit. Because the Unit communicates with the PC via the CPU bus, this facilitates high-speed communications. All of the libraries and drivers needed for using the CPU bus are provided, so programming is made easy. For explanations of libraries, refer to the *CV-series Personal Computer Unit Technical Manual (W252)*.

High Reliability

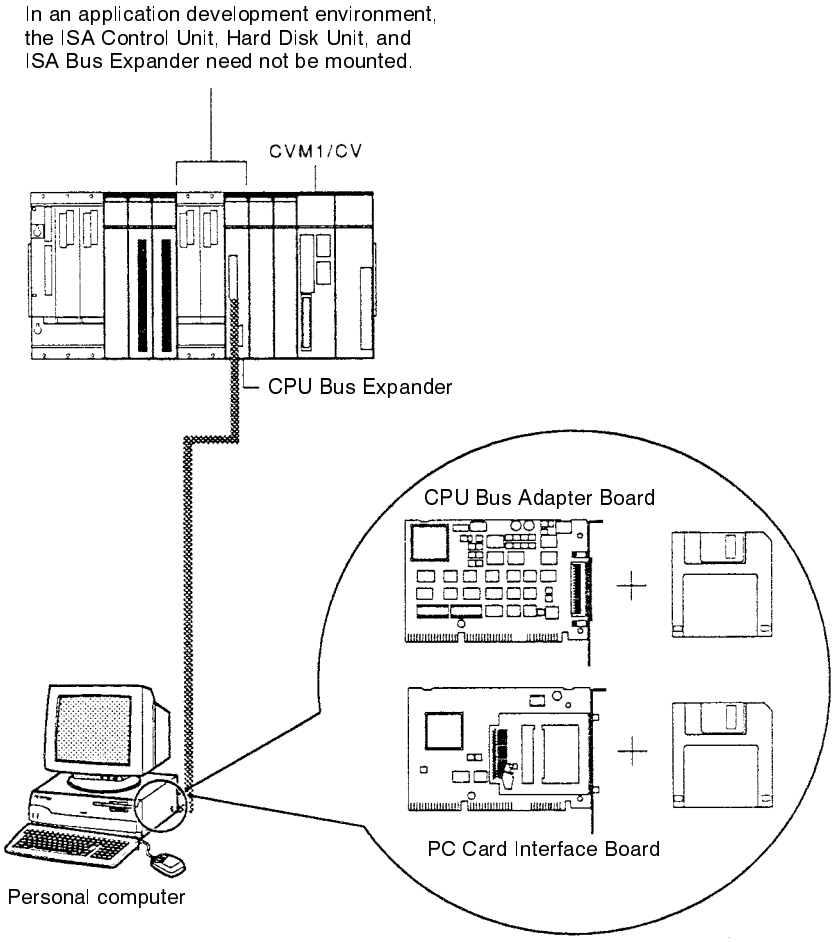
The ISA Control Unit's built-in ROM contains MS-DOS Ver. 5, and the Unit also has a built-in flash ROM (1M bytes). This enables communications without a hard disk and provides a high level of reliability in FA environments.

2-2 System Configuration

The ISA Control Unit was designed for use in both an application development environment and a working environment, thereby providing cost savings covering both development and operations. Examples of system configurations for each of these environments are shown below.

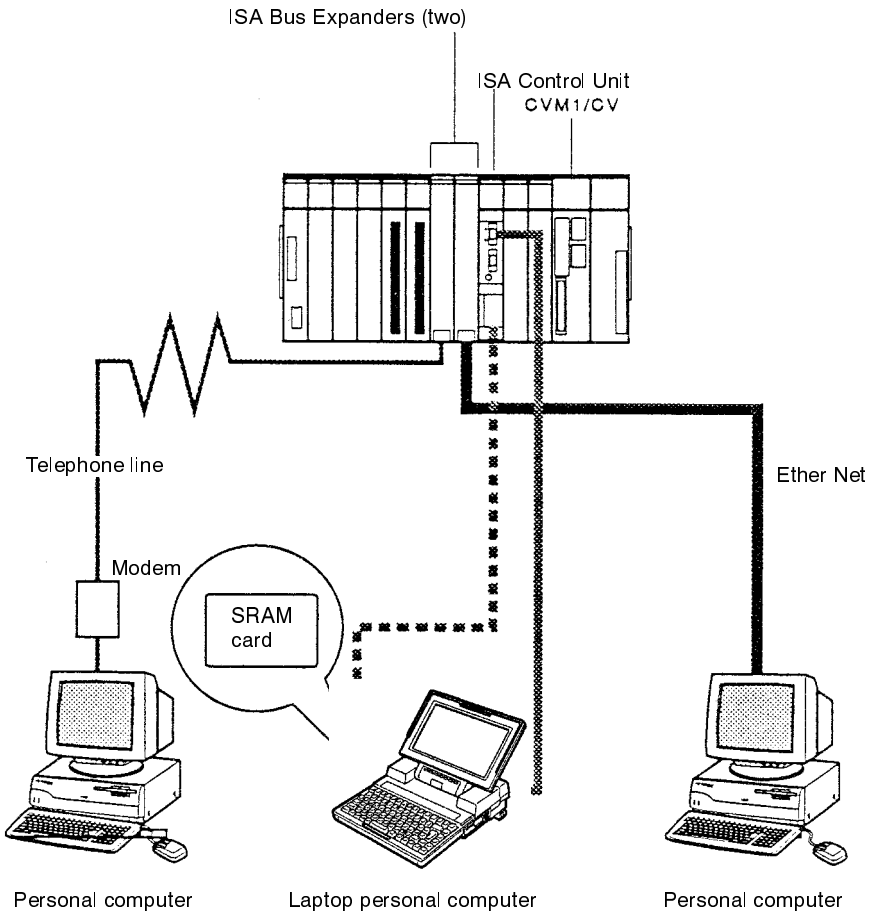
2-2-1 Development Environment System Configuration Example

The following illustration shows an example of an environment for application development.



2-2-2 Working Environment System Configuration Example

The following illustration shows an example of a system configuration for operations.



SECTION 3
Components and Their Functions

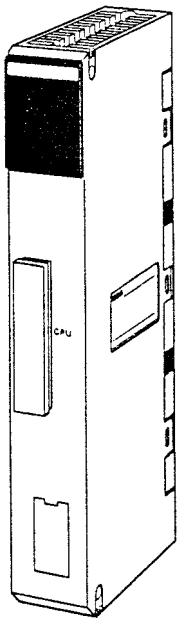
This section provides basic explanations of the individual system components and their functions. Open each of the boxes and check to be sure that nothing is missing.

3-1 CV500-IDK01-E Developer's Kit

The Developer's Kit, which offers an application development environment, consists of the four components described below. Check to be sure that the items you have match the ones shown in the illustrations.

3-1-1 CV500-BEX01 CPU Bus Expander

This Unit extends the CPU bus from the CPU Backplane. It is mounted to the CPU Backplane.

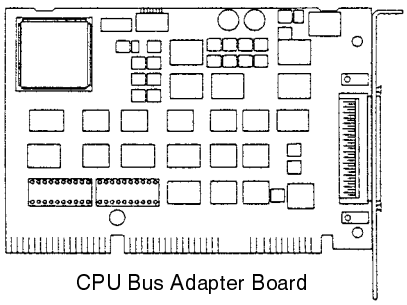


CPU Bus Expander

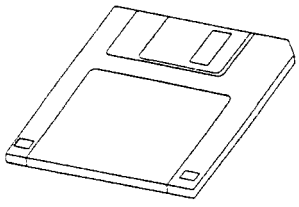
3-1-2 3G8F5-BP001 CPU Bus Adapter Board

This is an ISA board with a CPU bus interface. It is mounted in the ISA bus slot of the personal computer.

A floppy disk called "System Disk" is included. It contains the CPU bus driver and the various libraries.



CPU Bus Adapter Board

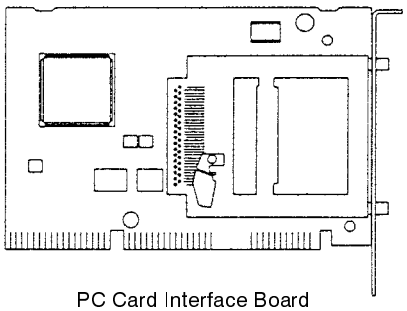


System Disk (one)

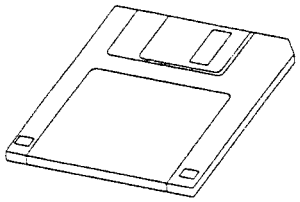
3-1-3 3G8F5-PCM01 PC Card Interface Board

This is an ISA board with a PC card interface. It is mounted in the ISA bus slot of the personal computer.

A floppy disk called "System Disk" is included. It contains the software for the PC card interface.



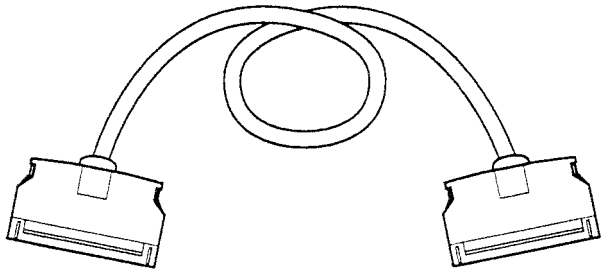
PC Card Interface Board



System Disk (one)

3-1-4 CV500-CN611 CPU Bus Extension Cable

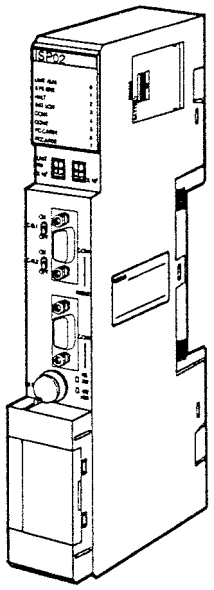
This cable connects the CPU Bus Adapter Board and the CPU Bus Expander. It is the same as the cable that connects the CV500-IC101 I/O Control Unit and the CV500-II101 I/O Interface Unit.



CPU Bus Extension Cable

3-2 CV500-ISP01/ISP02 ISA Control Unit

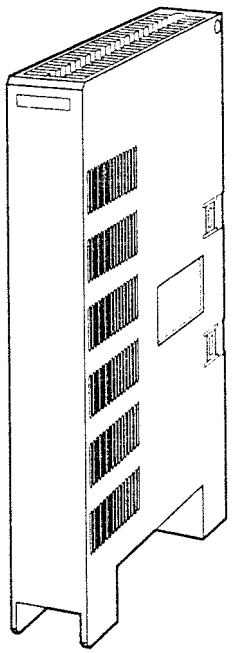
The CV500-ISP01/ISP02 ISA Control Unit is shown below. Check to be sure that the item you have matches the one shown in the illustration. The CV500-ISP02 has an NPU (numerical processor unit), while the CV500-ISP01 does not.



ISA Control Unit

3-3 CV500-ISX01 ISA Bus Expander

This Unit is used for connecting an ISA board to the ISA Control Unit. It is the same as the ISA Bus Expander for the CV500-VP□□□. An ISA Sub-backplane is required in order to mount the ISA Bus Expander to the CPU Backplane.



ISA Bus Expander

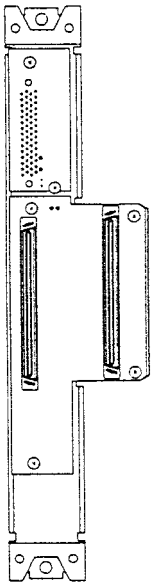
3-4 CV500-ISB01/ISB02 ISA Sub-backplane

The ISA Sub-backplane is used for connecting the ISA Control Unit to the ISA Bus Expander or the CV500-HDD11 Hard Disk Unit.

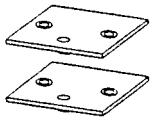
3-4-1 CV500-ISB01 ISA Sub-backplane

Either an ISA Bus Expander or a Hard Disk Unit can be mounted in addition to the ISA Control Unit.

The set includes two plates, two binding screws, and four flat-head screws. These are used to secure the CV500-ISB01 ISA Sub-backplane to the CPU Backplane.



ISA Sub-backplane



Plates (2)

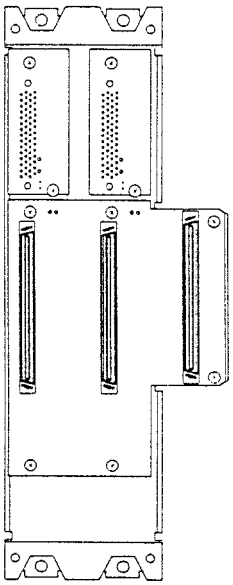
Binding screws (2)

Flat-head screws (2)

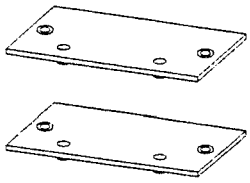
3-4-2 CV500-ISB02 ISA Sub-backplane

Either ISA Bus Expanders or Hard Disk Units (2 Units max.) can be mounted in addition to the ISA Control Unit.

The set includes two plates, four binding screws, and four flat-head screws. These are used to secure the CV500-ISB02 ISA Sub-backplane to the CPU Backplane.



ISA Sub-backplane



Plates (2)

Binding screws (4)

Flat-head screws (4)

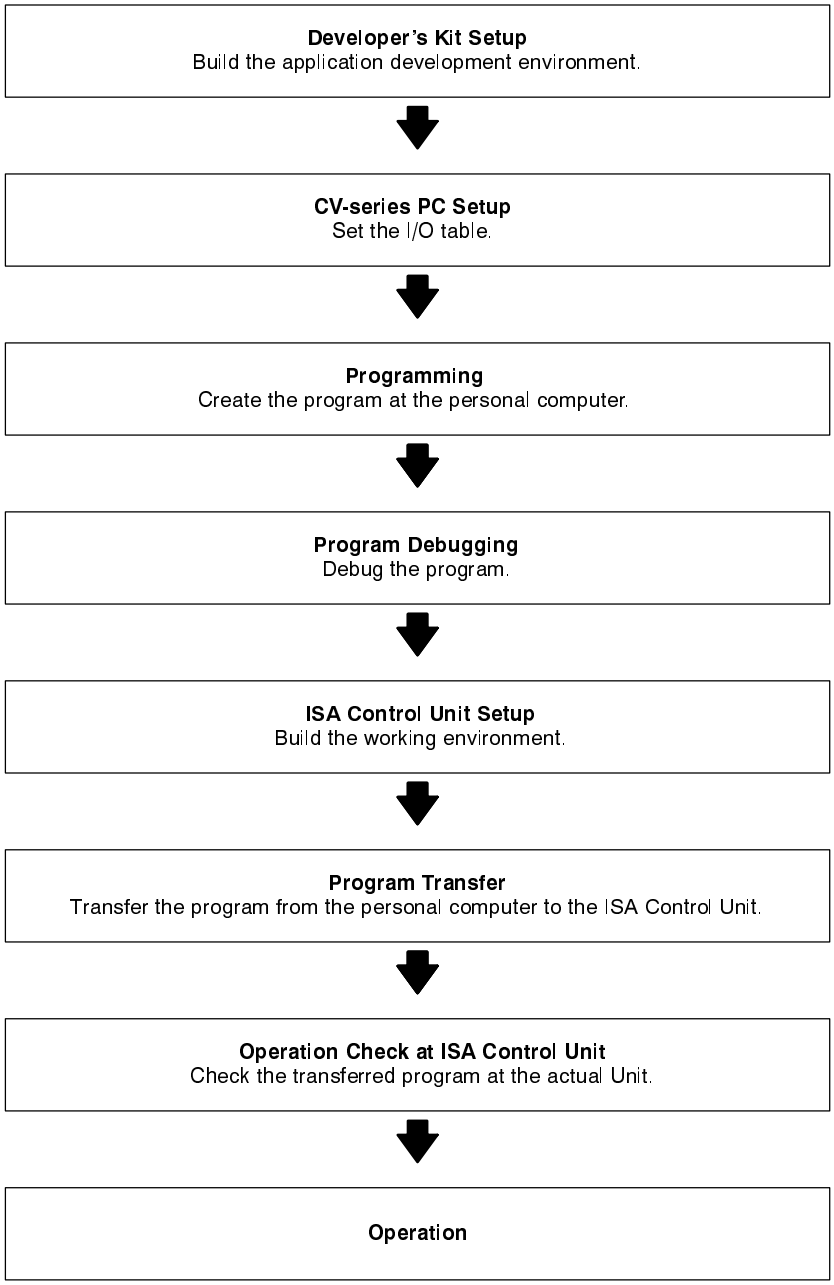
SECTION 4
Before Operation

This section outlines the steps involved from Developer's Kit, ISA Control Unit, and peripheral device settings on up to the point of operation. Be sure you have read this section and understood all of the procedures before attempting to actually make the settings or do any programming.

4-1 Outline

The procedures up to the point of operation are outlined below. Be sure to familiarize yourself with this basic flow of operations.

These procedures are explained in more detail in *4-2 Procedures and Reference Items*.



4-2 Procedures and Reference Items

This section outlines the steps involved from Developer's Kit, ISA Control Unit, and peripheral device settings on up to the point of application development and operation. Be sure that you thoroughly understand all of these procedures. References for further reading are provided for each procedure.

2. Developer's Kit Setup

Build the application development environment by installing the personal computer hardware and software and the CPU Expander. For details regarding the hardware and software setup, refer to *Section 5 Developer's Kit Setup*.

- CPU Bus Adapter Setup *5-2*

  Make the settings for connecting the CPU bus from the CPU Bus Expander to the personal computer.
- CPU Bus Expander Setup *5-4*

  Make the settings for extending the CPU bus from the CPU Backplane.
- PC Card Interface Board/Driver Setup *5-3, 5-7*

  This is the setup for using a PC card at the personal computer.
- CPU Bus Driver Installation *5-5*

  Set CONFIG.SYS to install the CPU bus driver. EMM386.EXE must also be set.
- Compiler Installation *Manual attached to compiler*

  Install the compiler that is to be used.

3. CV-series PC Setup

- CV-series PC I/O Table Settings *5-8*

  Use a tool such as SYSMAC Support Software to make the settings so that the ISA Control Unit will be recognized by the CV-series PC.

4. Programming

Programming *Chapter 7, 8*

Create the applications.

5. Program Debugging

Debug the program using the debugger that comes with the compiler. *Manual attached to compiler*

6. ISA Control Unit Setup

- Switch Settings *6-2*

  Make the unit number, Console, communications port, and system settings.
- Mounting the ISA Control Unit and Peripheral Devices *6-3*

  Mount the ISA Control Unit, ISA Expander, etc., to the PC.
- ISA Control Unit and Software Setup *6-4*

  Make the settings for the system software.

7. Transferring the Program

- Transfer the program from the personal computer to the ISA Control Unit, using either a PC card (memory card) or RS-232C. *6-3, 6-5*

8. Checking Operation at the ISA Control Unit

Observing the indicator status, connect the terminals and check for the following points:

- CPU speed errors
- Timing errors
- Disk configuration errors

9. Operation

SECTION 5
Developer's Kit Setup

This section explains the hardware and software setup for the Developer's Kit.

5-1 Developer's Kit

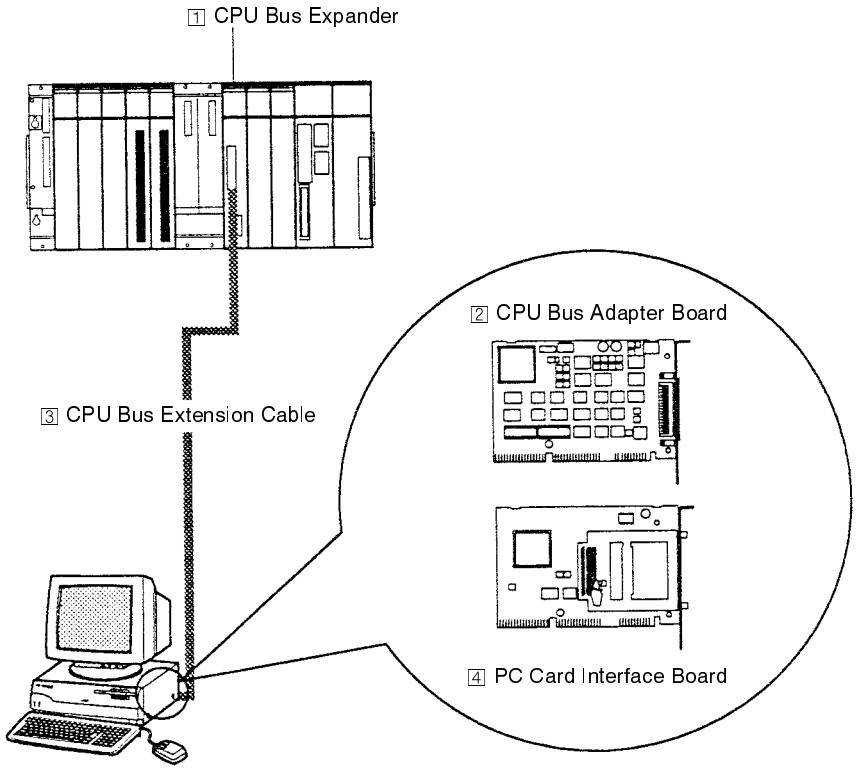
This section provides a general summary of the Developer's Kit.

The Developer's Kit is a hardware and software set for developing programs to be used with ISA control systems.

Note The purpose of the Developer's Kit is program development, and it is not intended to be built into a working environment. For instructions on building a working environment, refer to *Section 6 ISA Control Unit Setup*.

5-1-1 Developer's Kit Hardware Configuration

The Developer's Kit includes the hardware shown in the following illustration.



1 CV500-BEX01 CPU Bus Expander
The CPU Bus Expander mounts to the CPU Backplane and extends the CPU bus.

2 3G8F5-BP001 CPU Bus Adapter Board
The CPU Bus Adapter Board is an ISA board with a CPU bus interface. It is mounted in the ISA bus slot of the personal computer.

3 CV500-CN611 CPU Bus Extension Cable
This cable connects the CPU Bus Adapter Board and the CPU Bus Expander.

4 3G8F5-PCM01 PC Card Interface Board
The PC Card Interface Board is an ISA board with a PC card interface. It is mounted in the ISA bus slot of the personal computer.

5-1-2 Developer's Kit Software Configuration

The Developer's Kit includes the software shown in the following illustration.

CPU Bus Adapter Board System Disk Contents Chart

```
\
 ├ PRG   <DIR>
 FINS driver library
 │   │     DGIOX      COM          34,554      94-06-23    17:33
 │   │     PCDGSUB   LIB           4,655      94-05-19     2:00
 │   │     PCDGSUB   OLB           9,758      94-05-19     2:00
 │   │     T042      BAT             112      94-05-19     2:00
 │   └──   SAMPLE          <DIR>
 │         SMP         ASM           1,927      92-08-27     1:01
 │         OB_RECV     BAS           3,116      92-08-27     1:01
 │         OB_SEND     BAS           3,223      92-08-27     1:01
 │         BUSLINK     C             3,452      92-08-27     1:01
 │         CYCLIC      C             3,456      92-08-27     1:01
 │         DMRDWT      C             4,875      92-08-27     1:01
 │         EVENT       C             3,863      92-08-27     1:01
 └──DEV <DIR>
Device driver
 │   SBUS              SYS          26,637      95-12-20     8:13
 └──CVLIB          <DIR>
CPU bus library
     ├──BASIC      <DIR>
     │     BASLIB     LIB          21,133      94-05-19     2:00
     │     BASLIB     QLB          25,471      94-05-19     2:00
     │     T042       BAT             213      94-05-19     2:00
     │     └──SAMPLE  <DIR>
     │           SQ_MSRD   BAS          1,056      92-08-27     1:01
     │           SQ_MSWR   BAS            968      92-08-27     1:01
     │           SQ_OPEN   BAS            503      92-08-27     1:01
     │           SQ_READ   BAS          1,436      92-08-27     1:01
     │           SQ_STAT   BAS          1,544      92-08-27     1:01
     │           SQ_WRITE  BAS          1,384      92-08-27     1:01
     └──C          <DIR>
           CLIBC      LIB          14,379      94-05-19     2:00
           CLIBL      LIB          14,379      94-05-19     2:00
           CLIBM      LIB          12,843      94-05-19     2:00
           CLIBS      LIB          12,843      94-05-19     2:00
           └──SAMPLE        <DIR>
                 MODE    C          1,222      92-08-27     1:01
                 MSRD    C          1,442      92-08-27     1:01
                 MSWR    C          1,198      92-08-27     1:01
                 OPEN    C            483      92-08-27     1:01
                 PCRD    C          1,527      92-08-27     1:01
                 PCWR    C          1,473      92-08-27     1:01
```

PC Card Interface Board System Disk Contents Chart

\

PCMVCD	386	13,670	95-12-18	10:40
PCMCS	EXE	59,472	95-12-18	10:38
PCMSSVD	EXE	22,221	95-12-18	10:32
PCM	NEW	1,649	94-11-02	9:04
PCMRMAN	SYS	34,509	95-12-18	10:39
DEINST	TXT	27,643	95-08-11	14:57
PCMFDISK	EXE	44,655	95-12-18	10:35
CNFIGNAM	EXE	1,724	95-12-18	10:32
PCMRMAN	EXE	78,329	95-12-18	10:39
PCMWIN	HLP	14,447	95-12-18	10:39
INSTALL	EXE	60,538	95-12-18	10:41
PCMWARN	EXE	8,192	95-12-18	10:39
PCMSSIT	EXE	22,263	95-12-18	10:32
PCMMTD	EXE	19,529	95-12-18	10:34
PCMWIND	DLL	5,023	95-12-18	10:39
DEINSTAL	EXE	78,738	95-12-18	10:42
PCMWIN	EXE	79,360	95-12-18	10:40
PCMATA	SYS	21,267	95-12-18	10:33
PCMSSDB	EXE	22,585	95-12-18	10:33
PCMSCD	EXE	34,671	95-12-18	10:36
PCMSSRC	EXE	22,325	95-12-18	10:32
PCMCSFUL	EXE	69,710	95-12-18	10:37
PCMGROUP	NEW	1,298	94-11-05	12:41
PCMFFCS	EXE	21,111	95-12-18	10:34
PCMSETUP	EXE	122,321	95-12-18	10:41
PCMINST	TXT	13,039	95-07-14	13:38
PCMSSPP	EXE	23,413	95-12-18	10:33
PCMSSCL	EXE	23,049	95-12-18	10:32
PCMSSCL9	EXE	23,461	95-12-18	10:33
PCMSSVLS	EXE	22,431	95-12-18	10:33
PCM	EXE	95,805	95-12-18	10:38
DPMS	EXE	47,375	94-03-01	1:00
MS-FLASH	SYS	53,838	94-06-30	2:00
FLASHCMP	EXE	15,940	94-04-30	2:00
MEMCARD	EXE	76,716	94-04-30	2:00

There are no subdirectories.

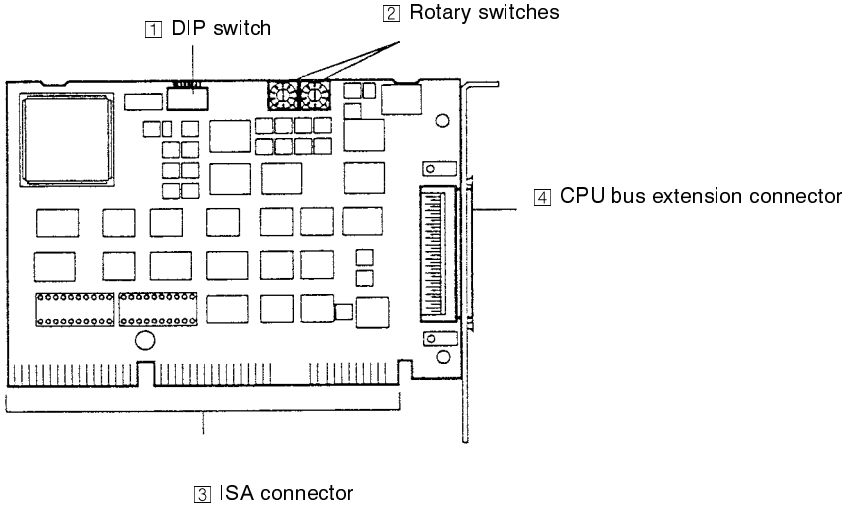
5-2 CPU Bus Adapter Board

The CPU Bus Adapter Board is an ISA board with a CPU bus interface. It is mounted in the ISA bus slot of the personal computer.

This section explains how to set up, mount, and remove the CPU Bus Adapter Board.

5-2-1 External Parts Nomenclature

CPU Bus Adapter Board



1 DIP Switch
Sets the CPU bus memory address, interrupt level, and the data bus width for the ISA bus.

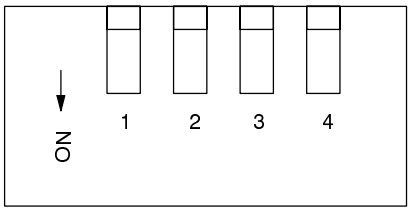
2 Rotary Switches
Sets the unit number.

3 ISA Connector
Inserts into the personal computer's ISA bus slot.

4 CPU Bus Extension Connector
Connects CPU bus extension cable.

5-2-2 DIP Switch Settings

The DIP switch shown below is used for setting the CPU bus memory address, interrupt level, and the data bus width for the ISA bus.



Pin	Setting
1, 2	CPU bus memory address
3	Interrupt level for CPU bus
4	Data bus width for ISA bus

Note
1. The pins are all set to OFF as the default.
2. The I/O addresses are fixed at 390h to 394h.

Setting the Memory Address DIP switch pins 1 and 2 set the memory address for the CPU Bus Adapter Board.

⚠ Caution Do not use a memory address that is already in use for another ISA board, and do not use an address that is being used for a memory manager such as EMM386.EXE.

The following table shows the settings and address allocations.

Pin 1	Pin 2	Address allocations
OFF	OFF	0DC000h to 0DFFFFh
OFF	ON	0D8000h to 0DBFFFh
ON	OFF	0D4000h to 0D7FFFh
ON	ON	0D0000h to 0D3FFFh

Setting the Interrupt Level Pin 3 sets the interrupt level.

⚠ Caution Do not set an overlapping interrupt level. Be particularly careful not to use the same interrupt level as that being used by the PC Card Interface Board.

The pin 3 settings are as follows:

Pin setting	Interrupt level
ON	IRQ10
OFF	IRQ11

Data Bus Width for ISA Bus Pin 4 sets the data bus width for the ISA bus.

Note The data bus width should normally be set to 16 bits (the default). Set it to 8 bits if the SA signal is output later than the LA signal at the personal computer that is being used.

The pin 4 settings are as follows:

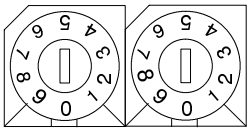
Pin setting	Data bus width
ON	8 bits
OFF	16 bits

5-2-3 Setting the Rotary Switches

The two rotary switches set the unit number within a range of 00 to 15.

Set the rotary switches by turning them with a small flat-blade screwdriver.

The switch on the left (facing the Unit) sets the “10” digit, and the switch on the right sets the “1” digit.



⚠ Caution Do not set a unit number that is already being used for another Special I/O Unit.

5-2-4 Installing and Removing Boards

This section provides examples of installing and removing the CPU Bus Adapter Board in a personal computer bus slot. The procedures are the same for the PC Card Interface Board.

For instructions regarding covers, slot covers, card folders, etc., for the personal computer itself, be sure to refer to the manual for the particular computer that is being used.

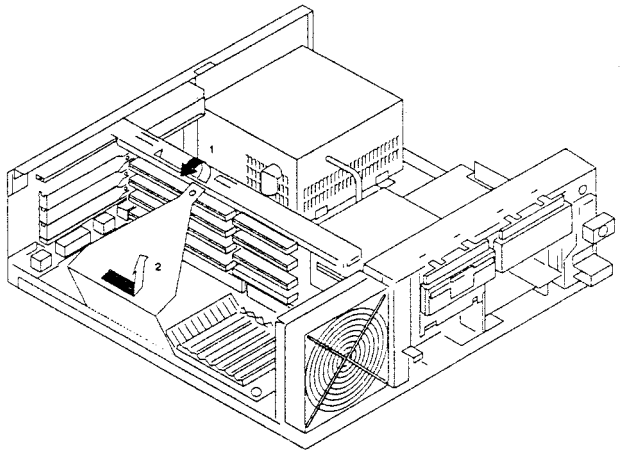
⚠ **Caution** Be sure to turn OFF the power supply to the personal computer before installing or removing a board and be careful to discharge any static electricity from your body before handling a board.
Failure to do so can result in damage to components on the board or in the computer.
When installing or removing a board, be careful not to damage the personal computer's internal memory or other components.

Installing a Board

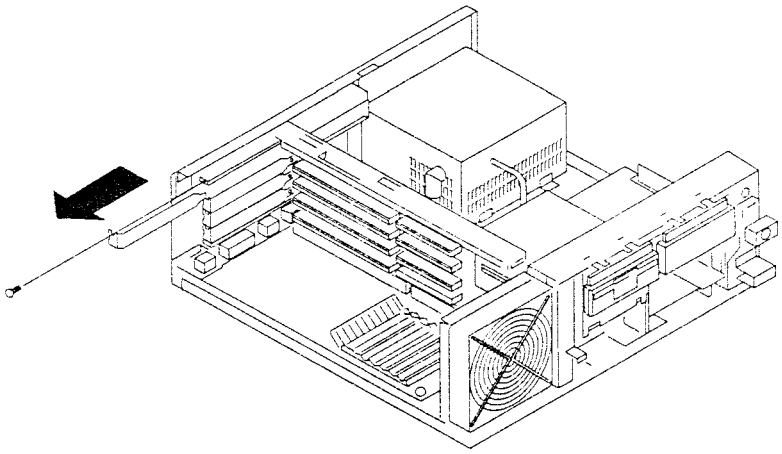
Follow the procedure outlined below to install a board.

1, 2, 3...

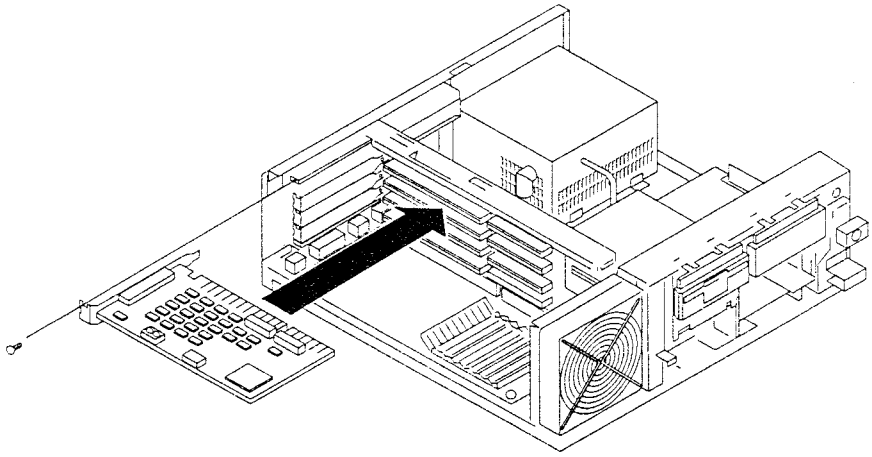
1. Turn OFF the power supply to the personal computer.
2. Remove the cover from the personal computer.
3. Remove the card folder.



4. Remove the screws and take out the slot cover as shown in the following illustration.



5. Insert the board into the ISA bus slot as shown below. Then fasten the screws. If there is another board to be installed, then insert it into an open slot and fasten the screws.



6. Replace the card folder.
7. Replace the personal computer's cover.

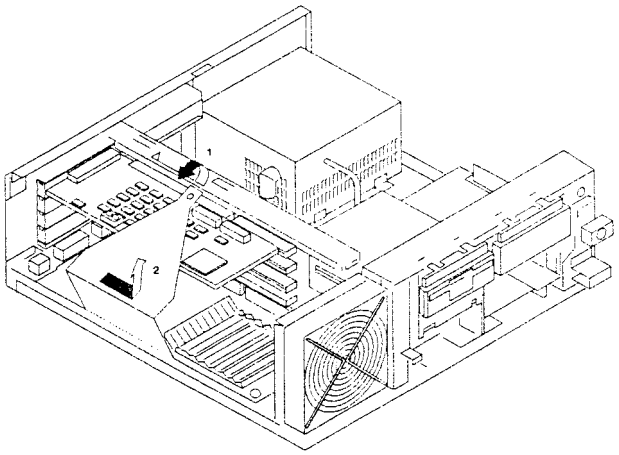
   The installation is now complete.

Removing a Board

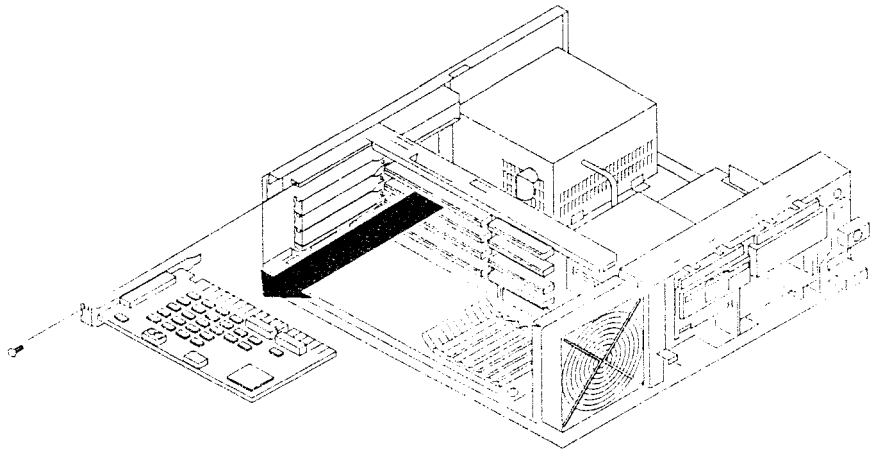
Follow the procedure outlined below to remove a board.

1, 2, 3...

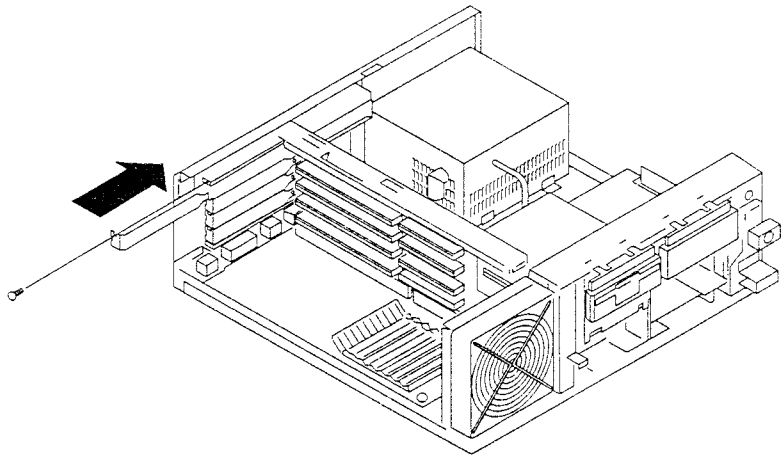
1. Turn OFF the power supply to the personal computer.
2. Remove the cover from the personal computer.
3. Remove the card folder.



4. Remove the screws and pull the board out from the ISA bus slot as shown in the following illustration.



5. Replace the card folder.
6. Replace the slot cover.



7. Replace the personal computer's cover.

The board removal procedure is now complete.

5-3 PC Card Interface Boards

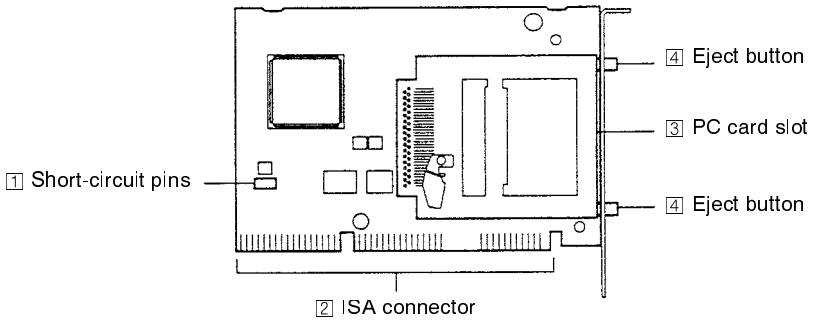
The PC Card Interface Board is an ISA board that enables PC cards to be used. It is installed in the ISA bus slot of the personal computer. The procedure for setting it up is explained below.

The interrupt level for the PC Card Interface Board is set by the software. For details regarding driver setup, refer to *5-6 PC Card Interface Driver Setup*.

For instructions on installing and removing PC Card Interface Boards, refer to *5-2 CPU Bus Adapter Boards*.

5-3-1 Nomenclature and Functions

The following illustration shows the names and functions of the PC Card Interface Board's components.



1 Short-circuit Pins
These set the I/O address.

2 ISA Connector
Inserts into the computer's ISA bus slot.

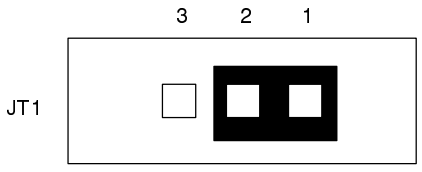
3 PC Card Slot
Slot for installing the PC card.

4 Eject Button
Ejects PC card when pressed.

5-3-2 Short-circuit Pin Setting

Set the I/O address with the short-circuit pin in the following way.

When using only one PC Card Interface Board, be sure to set the pins so that they are short-circuited across 1 and 2. When using two PC Card Interface Boards, the short-circuit pin settings must be different for the two Boards.



Setting	Index base value	I/O address
Short across 1 and 2	00h	3E0h/3E1h
Short across 2 and 3	80h	3E0h/3E1h

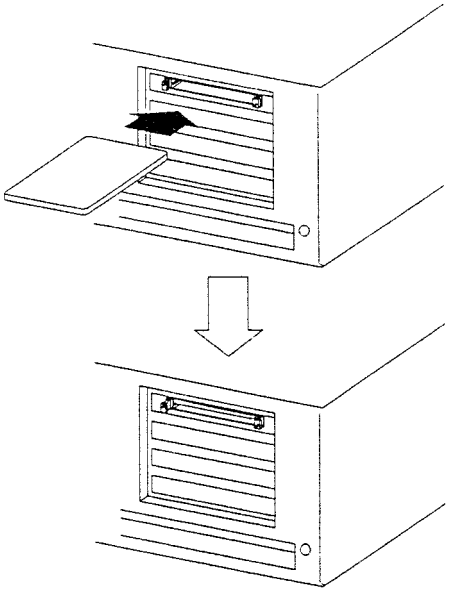
Note The default setting is for shorting across pins 1 and 2.

5-3-3 Installing and Removing PC Cards

PC cards can be installed and removed once a PC Card Interface Board has been installed in the personal computer. In this example, the lower slot is slot no. 1, and the upper slot is slot no. 2.

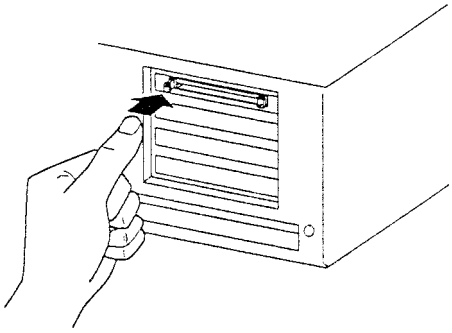
Installing PC Cards

Insert the PC card into the PC card slot as shown in the following illustration. Push the card in until it locks securely into place.



Removing PC Cards

To remove the PC card, press the eject button as shown in the following illustration. Pressing the button on the left ejects the PC card from the upper slot, and pressing the button on the right ejects the PC card from the lower slot.



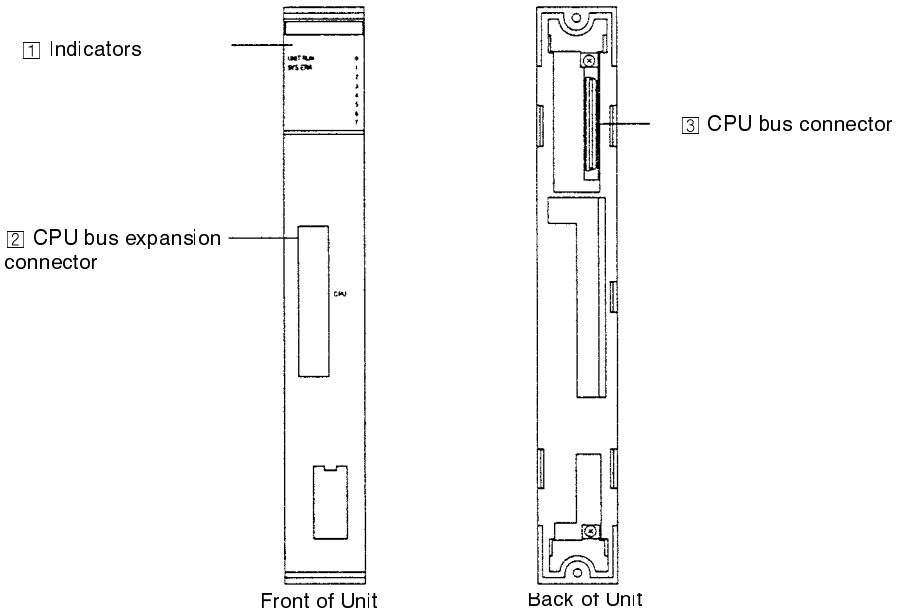
5-4 CPU Bus Expanders

This section describes CPU Bus Expander components and their functions, and explains how to mount an Expander and connect the cables.

For details regarding driver setup, refer to *5-5 CPU Bus Driver Installation and Startup.*

5-4-1 Nomenclature and Functions

The following illustration shows the names and functions of the CPU Bus Expander's components.



1 Indicators
The meanings of the indicators are shown in the following table.

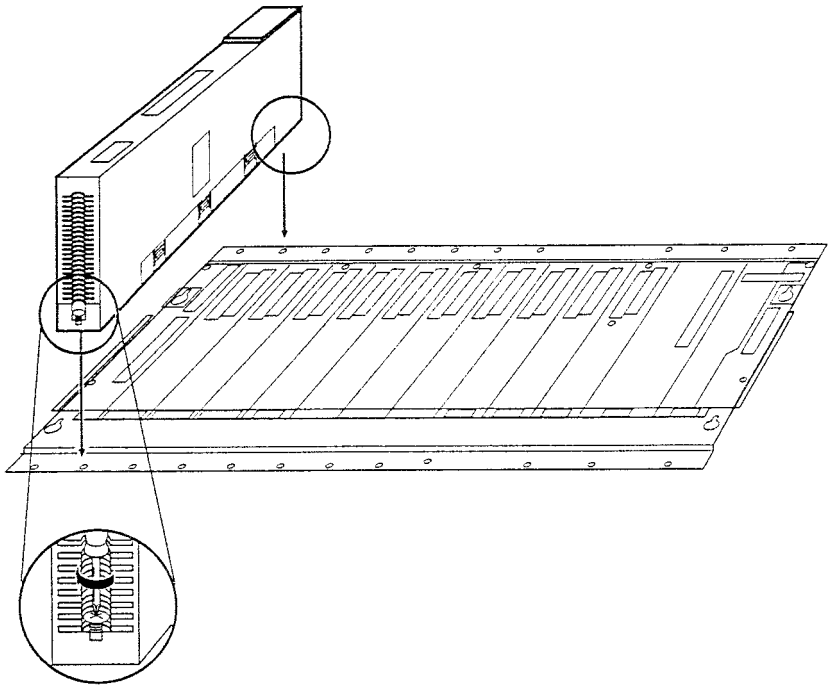
Display	Color	Lit condition	Comments
UNIT RUN	Green	Unit operating.	Displays UNIT RUN signal status.
SYS ERR	Red	Error in CPU bus.	
0	Orange	User defined.	Can be freely used as program I/O from host system.
1	Orange	User defined.	
2	Orange	User defined.	
3	Orange	User defined.	
4	Orange	User defined / Error	
5	Orange	User defined / Error	
6	Orange	User defined / Error	
7	Orange	User defined.	

2 CPU Bus Expansion Connector
Connects the CPU bus expansion cable.

3 CPU Bus Connector
Connects to the CPU bus connector on the CPU Backplane.

5-4-2 Mounting to the Backplane

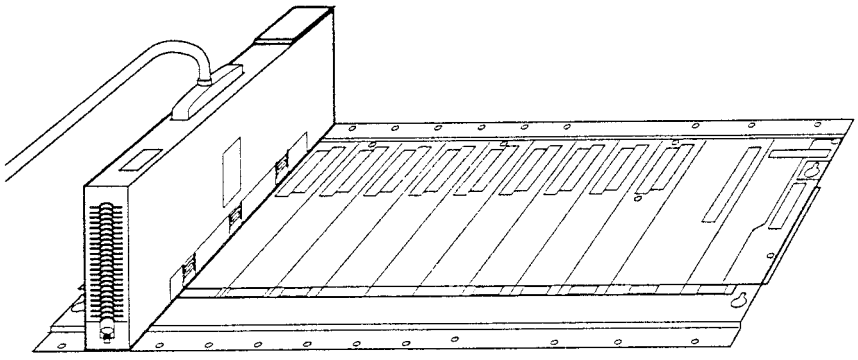
1, 2, 3... 1. Line up the CPU Bus Expander's CPU bus connector as shown in the following illustration, and push the Unit into place. The CPU Bus Expander can be mounted to any slot position on the CPU Backplane or on an Expansion CPU Backplane.



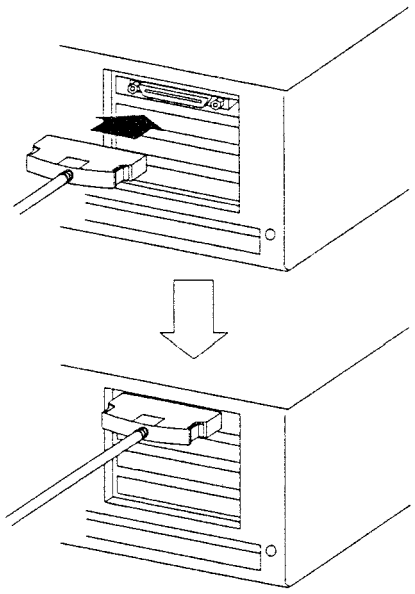
2. Secure the Unit by tightening the screws to a torque of 1.2 N • m.

5-4-3 Connecting CPU Bus Expansion Cable

1, 2, 3... 1. Connect the end of the CPU bus expansion cable with the proper connector type to the CPU Bus Expander's CPU bus expansion connector.



2. Connect the other end of the CPU bus expansion cable to the CPU Bus Adapter's CPU bus expansion connector.



5-5 CPU Bus Driver Installation and Startup

This section explains how to install the CPU Bus Driver and how to run the start-up procedures for the personal computer and CV-series PC.

5-5-1 Copying the CPU Bus Driver

Copy the CPU Bus Driver (SBUS.SYS) to the hard disk of the personal computer from the system disk that comes with the CPU Bus Adapter Board. Copy it to the root directory of the drive for starting the system. (SBUS.SYS can be found in \DEV on the system disk.

5-5-2 Installing the CPU Bus Driver

Add the following line to CONFIG.SYS to install the CPU Bus Driver.

```
DEVICE = n:\SBUS.SYS/V65/C3/B12/W5
(n: Driver name)                  Option
```

SBUS.SYS must be described after the keyboard driver, or it will not be possible to use the Escape Key to exit.

Options

The following options can be designated for SBUS.SYS:

/Vnn

To make a direct request, designate an interrupt number "nn" from 60 to 65 (60H to 65H). Direct request processing will not be executed if any other numbers are designated. For details regarding direct requests, refer to *8-2 Using the CPU Bus Driver*. Designate "65" when using a FINS Driver, FINS Library, or CPU Bus Driver.

/Cnn

This specifies the number of retries when a CPU bus error occurs. Any number from 0 to 10 can be set. If a number outside of this range is specified, the driver will recognize it as "10." If this option is not specified, no retries will be carried out. Increase this value when quitting with PC busy.

/Bnn

This specifies the number of event service reception buffers. Any number from 1 to 14 can be set. If no number is specified, the default is "14." Setting a smaller number decreases the amount of memory occupied by the CPU Bus Driver. Make the setting according to the system that is being used.

/Wnn

This specifies the waiting time when the CV-series Power Supply is turned ON, in units of approximately one second. If this option is not specified, the default is a wait of 30 seconds.

Note

1. If an EMS Driver is installed, do not overlap the memory space used by the CPU Bus Adapter Board.

   Example: DEVICE = C:\DOS\EMM386.EXE RAM X=DC00–DFFF

2. When using shadow memory, do not overlap the memory space used by the CPU Bus Adapter Board.
3. Install the CPU Bus Driver before the PC Card Driver. If installing it afterwards cannot be avoided, then make sure that the hardware series (memory, IRQ, I/O) used by the CPU Bus Adapter Board is not controlled by the PC card's software. For details, refer to *5-10-2 PCMRMAN.EXE*.

5-5-3 Personal Computer and CV-series Startup Procedure

Follow the procedure outlined below to start up the personal computer and CV-series PC.

1, 2, 3...

1. Turn ON the power supply to the personal computer. If the driver is correctly installed, the following messages will appear. (The messages enclosed in parentheses are not actually displayed.)

   ```
   3G8F5-BP001 Memory address DC00-DFFF     IR011
   ```

   (CPU Bus Adapter memory address and interrupt number)

   ```
   Turn on PLC within 30 seconds
   ```

   (Message prompting user to turn ON power supply)

2. Turn ON the power supply to the CVM1/CV when the above message appears. If the PC's communications are normal, the following message will appear.

   ```
   SBUS.SYS: CV CPU I/F DRIVER Ver. 1.00
   ```

Troubleshooting

If the PC's communications are not normal, the following message will appear.

```
SBUS.SYS: CV CPU I/F DRIVER can't be installed.
```

If this message appears, look at the SYS ERR indicator on the front of the CPU Bus Expander and at user indicators 4 to 6 to determine the content of the error, and then execute the error processing. For details regarding error processing and the meanings of indicators, refer to *9-1 Indicators*. For details regarding I/O table settings, refer to *5-14 CV-series PC Setup*.

The following message will then appear:

```
Turn off PLC. And re-load SBUS.SYS
```

First turn OFF the power supply to the PC, and then repeat the startup procedure outlined above.

⚠ Caution Be sure to turn ON the power supply to the personal computer before the PC. If the PC is already powered up when the computer is turned ON, the computer will not operate properly.

5-6 PC Card Interface Driver Setup

This section explains how to install the driver required for using PC cards.

5-6-1 Terminology

PCM Plus or PCM+ This is a trademark of Phoenix Technologies, Ltd. which indicates a group of files for using PC cards.

PC Card Memory or I/O card conforming to PCMCIA standards.

Client All programs using Card Services (drivers, utility programs).

PCMCIA Acronym for “Personal Computer Memory Card International Association.”

Window An area in Personal Computer Unit memory, or on an I/O board, that is allocated for PC cards.

5-6-2 Usable PC Cards

Type I, II, or III PC cards conforming to PCMCIA2.1 can be used. (Only one type III PC card can be used.) Any PC cards that are used must support PCM Plus. The following PC cards can be used:

- SRAM cards
- FLASH cards
- ATA interface cards

5-6-3 Minimum Hardware Requirements

The personal computer must meet the following conditions in order to use PCM Plus. Be sure that these conditions are met before installing PCM Plus.

- At least 2M bytes of RAM.
- At least 784K bytes of available space on the hard disk.
- At least an 80386 CPU.

Note When a different type of PC card is inserted into the same slot, an error may occur. If an error occurs, interrupt access to the new card, replace it with the previous card, access the previous card using the DIR command, and replace the previous card with the new card and check its operation.

5-6-4 Installing Drivers

In a normal system, the required files and are copied and the drivers are copied and installed using the installer (INSTALL.EXE) and setup program (PCMSETUP.EXE). When the installer is executed, the setup program is then executed automatically.

The files necessary for the PC card are copied to the hard disk by the installer, and the drivers required by the PC cards that are being used are automatically installed by the setup program. This is done simply by means of an on-screen specification, thereby avoiding the need for complicated settings.

In an environment in which the installer and setup program cannot be used, such as when no display or keyboard are connected to the ISA Control Unit, or when a special PC card is used, install the various drivers without using the installer.

5-7 Copying Files for PC Cards

Use the installer to copy PC card files from the system disk to the hard disk of the personal computer. The driver installation will begin after the files have been copied.

⚠ **Caution** MS-FLASH.SYS must be installed in order to use flash cards when the Microsoft Flash File System is used. Before the install operation is begun by the installer, first copy MS-FLASH.SYS, MEMCARD.EXE, and FLASHCMP.EXE from the system disk.

⚠ **Caution** The installer can only operate properly in English mode. Be sure that the system is in English mode before beginning the installation.

Note When a DOS 6.0 or later multi-boot system is used, the installation will be done in the current configuration

5-7-1 Installation Procedure

1, 2, 3...

1. If the system is not already in English mode, then switch it to English mode. The method for switching the mode will depend on the system that is being used, so please refer to the appropriate manual for instructions.
2. Insert the system disk into drive A. To make drive A the current drive, enter "A:" at the DOS prompt.
3. Type "Y\INSTALL" at the DOS prompt. The following screen will be displayed.

Phoenix CARD Manager Installation

PhoenixCARD Manager Version 3.20

PCMCIA Installation Utility
Revision 2.00

Copyright (c) 1993–1995, Phoenix Technologies Ltd.
All Rights Reserved.

Press any key to continue

4. Press any key to continue. A warning will be displayed if the system is a multi-configuration system. If PCM Plus is to be installed in the current configuration, press any key.

   If it is not a multi-configuration system, the warning will not be displayed and the following screen will appear.

Phoenix CARD Manager Installation

PCMCIA Controller Found

The following controller has been detected in your system:

Vadem VG-365, VG-465 or VG-468

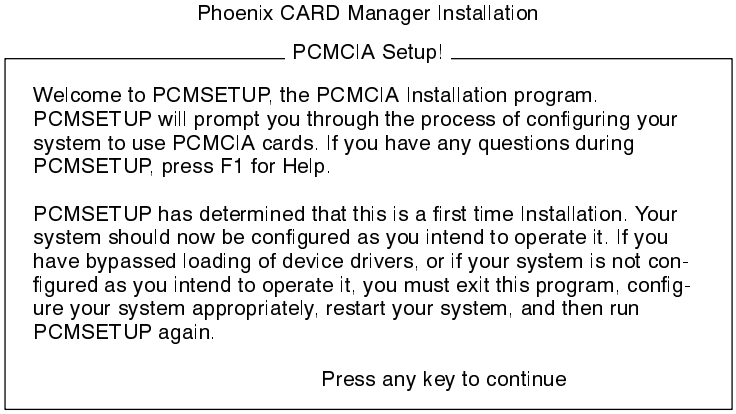
Only the files needed for this controller will be installed to your system.

Press any key to continue

   If this screen does not appear, it may mean that the board settings are wrong. Check the hardware settings.

5. Press any key to continue. The screen for setting the install directory (Destination Drive and Path) will be displayed.
6. The default directory can be specified here by simply pressing the Enter Key. To specify another directory, enter the drive name and directory name and then press the Enter Key.

   The files will begin to be copied. (Wait until the display shows 100%.) A message will then be displayed to indicate that the copying is complete.
7. Press any key to continue. The setup program (PCMSETUP.EXE) will start up and the following screen will be displayed.



The method for installing drivers using the setup program is described in *5-8 Driver Installation by Setup Program*.

5-8 Driver Installation by Setup Program

The method for installing drivers using the setup program is explained here. When the setup program is used, the drivers required for the PC cards being used are installed and set up automatically.

The setup program has both a Quick Mode option and an Advanced Mode option. The Quick Mode procedure will be explained first. PC cards can normally be used with Quick Mode installation. For details regarding Advanced Mode, refer to *5-8-2 Advanced Mode*.

Note The setup program can only operate properly in English mode. Be sure that the system is in English mode before executing the setup program.

5-8-1 Quick Mode

After the procedure explained in *5-7 Copying Files for PC Cards* has been completed, start up the setup program (PCMSETUP.EXE) to change the driver settings. Typing PCMSETUP and pressing the Enter Key in the directory that was copied in that procedure starts the setup program. The PCMCIA Setup screen is then displayed.

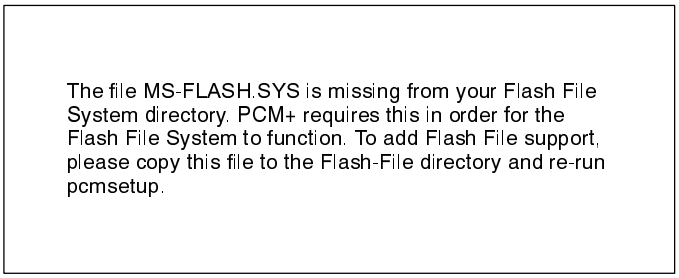
1, 2, 3...

1. Press any key while the PCMCIA Setup screen is displayed. The Installation Mode Screen screen will then be displayed for selecting either the Quick Mode or the Advanced Mode.
2. The default selection is Quick Mode, so press the Enter Key. The Socket Information screen will then be displayed for specifying the number of sockets located on the computer.
3. The program automatically checks the PCMCIA hardware and shows the correct number of sockets, so ordinarily only the Enter Key needs to be pressed. The Microsoft screen will then be displayed for specifying whether or not the PC cards are to be used under Windows 3.1.

4. In this case, press the Enter Key to specify “No.” The Flash File System screen will then be displayed for specifying whether or not the Microsoft Flash File System is to be used.
5. If a PC card that utilizes the Microsoft Flash File System is to be used, then select “Yes” and press the Enter Key. If not, then select “No” and press the Enter Key.

   Most unformatted flash cards on the market support the Microsoft Flash File System. If your PC cards use another format and file management utility, then select “No.”

   If “Yes” is selected, then the Flash File System screen will be displayed for setting the path to the the Flash File System. If “No” is selected, then skip to step number 7.
6. Specify the path for the directory where the Flash File System (MS-FLASH.SYS) is stored, and press the Enter Key. The Communication Ports screen will then be displayed for selecting whether a modem or fax/modem is to be used. If MS-FLASH.SYS cannot be found in the specified directory, the following message will be displayed.



   Copy MS-FLASH.SYS to the specified directory and then restart PCMSETUP.
7. If a modem or fax/modem is to be used, select “Yes” and press the Enter Key. If not, select “No” and press the Enter Key.

   If “Yes” is selected, the Communication Ports screen will be displayed for selecting the port. If “No” is selected, then skip to step number 9.
8. Select the communication port that is to be used, and press the Enter Key. Do not select the port that is currently in use.

   The ATA Socket screen will then be displayed for selecting whether or not ATA-type cards will be used.

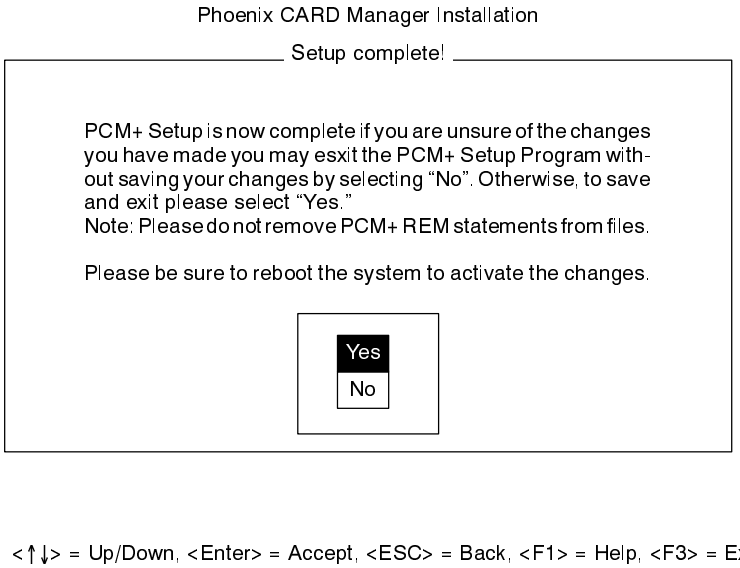
Note If a communication port already in use is selected, an alarm message will be displayed and the system will wait until a port not in use is input.

9. If an ATA card will be used, then select “Yes” and press the Enter Key. If not, the select “No” and press the Enter Key.

   The PCMCIA Card Selection screen will then be displayed for selecting the PCMCIA network card.
10. The program automatically determines whether a network card is installed, so ordinarily only the Enter Key needs to be pressed.

Note A warning screen will appear at this point, even though there is nothing wrong. Simply press the Enter Key. The screen will then reappear, so just press the Enter Key again.

The Setup Complete screen will then appear to inform you that the setup program has been completed.



11. To quit the program, select “Yes” and press the Enter Key. The PCM.INI, PROGRAM.INI, CONFIG.SYS, and SYSTEM.INI files will be refreshed and the system will return to DOS after the PCM Plus path has been written to AUTOEXEC.BAT.
12. Take out the system disk.
13. Restart the personal computer to validate the settings that have been made. Once it has been reset, the PC card can be used. The lower slot will be slot no. 1 and the upper slot will be slot no. 2.

Note

1. When new items such as SRAM cards, flash cards, or hard disk cards are used, they must be formatted. For instructions on how to format PC cards, refer to *5-9 Formatting PC Cards*.
2. Do not delete PCM Plus-related REM statements written to CONFIG.SYS.

5-8-2 Advanced Mode

Using the setup program’s Advanced Mode allows for more complex settings to be made. The following settings can be made, aside from the Quick Mode settings. Set these items by first starting the setup program and selecting the Advanced Mode, and then making the settings following the instructions on the screen.

Option	Settings changeable in Advanced Mode
PCM Plus resources	IRQ setting
	CIS window address
Card interaction	Token ring speed
	Card settle time
Flash File System	Memory-mapped window base address
	Window size
	Number of partitions
	Number of erase queues
ATA cards	Spin down time
	ATA socket number
	Number of partitions

5-9 Formatting PC Cards

There are two ways to format PC cards, depending on the type of PC card that is used. One way is to format by means of FORMAT.COM, and the other way is to format by means of MEMCARD.EXE. Both of these ways are explained here.

5-9-1 Formatting With FORMAT.COM

Use the DOS command FORMAT.COM to format ATA-type PC cards and SRAM cards. Type the following after the DOS prompt and press the Enter Key.

```
C:\>FORMAT G:
```

Use PCMFDISK to set partitions.

If the formatting is unsuccessful, use the /U option tor forcibly execute the formatting. With initial formatting such as for SRAM cards, a DOS error message may be displayed even if the /U option is specified. In that case, specify “F” (failure).

5-9-2 Formatting With MEMCARD.EXE

MEMCARD.EXE can be used for processing related to formatting and partitioning of standard SRAM cards and Intel Series I/II flash cards (or compatible cards). The status of memory cards can also be displayed.

At the MS-DOS prompt, enter “MEMCARD” and press the Enter Key.

```
>MEMCARD
```

The following menu will then be displayed.

```
Memcard option
Current memory card slot:1 of 2 slot.
Choose one of the following:

1. Create and format a new partition
2. Format an existing partition
3. Delete a partition
4. Display partition information
5. Erase entire memory card
6. Check Memory card and fix errors
7. Select next memory card slot

Enter your choice [4]
```

Selecting Memory Card Slot

If no memory card slot is selected for processing, enter “7” and press the Enter Key. Then select the slot. (With the ISA Control Unit, the built-in flash memory drive F can be selected.)

Formatting Flash Cards

Follow the procedure outlined below to format flash cards.

1, 2, 3...

1. Type “1” while in the main menu, and press the Enter Key. The following screen will be displayed.

```
Current memory card slot: 1

Letter    Type        Status      Size
------------------------------------------
G         MS-Flash    Formatted   4096K


Do you want to use the entire card for MS-Flash? (Y/N)
```

2. Type “Y” and press the Enter Key. If an MS-Flash partition has already been created on the card, a message will be displayed warning that the card has already been formatted and that creating a partition will destroy all the data on the card, and asking if you want to continue. Make sure that all essential data has been backed up, and then enter “Y” and press the Enter Key.

3. Specify the number of spare blocks (i.e., the number of blocks that must be reserved for clearing memory cards) and then press the Enter Key. (The default setting of "1" is recommended.)

Note At least one block must be reserved on the flash card for clearing files that have been deleted. Blocks reserved for this purpose are called "spare blocks." The setting range is "1" to "n–1" (with "n" representing the total number of blocks on the memory card). For example, if the total number of blocks on the card is four, then anywhere from one to three spare blocks can be set.

The default setting is "1." Increasing the number of spare blocks speeds up some memory cards, but the default setting is recommended because spare blocks cannot be used as write areas for files.

4. Input the volume name for the partition and press the Enter Key. If no volume name setting is required, then just press the Enter Key.

   A message will be displayed indicating that the operation may take some time. Do not remove the card from the slot while the message is displayed. When the formatting has been completed, the display will return to the main menu screen.

Formatting SRAM Cards

Follow the procedure outlined below to create new partitions and format SRAM cards.

1, 2, 3...

1. Carry out the first step as described in *Formatting Flash Cards* above. The procedure is the same for formatting SRAM cards.
2. If a partition has already been created on the card, carry out the second step as described in *Formatting Flash Cards* above. The procedure is the same.
3. Specify the maximum total number of files and subdirectories to be reserved for the root directory, and press the Enter Key. Any number from 16 to 512 can be set. The default value displayed on the screen depends on the size of the partition that is created. It is recommended that the default value be set.
4. Input the volume name for the partition and press the Enter Key. If no volume name setting is required, then just press the Enter Key.

Displaying Partition Information

1, 2, 3...

1. Type "4" while in the main menu, and press the Enter Key. The following screen will be displayed.

Display partition information

Current memory card slot : 1

Device Information:

Memory Type Size

Flash 4096K

Partition Information:

Letter	Start Addr.	End Addr	Type	Status	Size
G	0	4194303	MS-Flash	Formatted	4096K

To return to the main menu, press ESC.

If there is more than one page of information, press any key to display the next page. To return to the main menu, press the Escape Key.

The information displayed on the screen depends on the number of partitions in the memory card, as well as on the sizes and types. The following table explains the information displayed on the screen.

Item	Explanation
Memory type	Shows the type of card in the slot. The following memory types may be displayed: ROM — Read-only memory OTPROM — One-time programmable ROM EEPROM or Flash — ROM erasable electrically SRAM — Static random-access memory DRAM — Dynamic random access memory I/O — An input/output card
Size (device)	Shows the size (in kilobytes) of each device type.
Letter	Shows the name of the drive set for each partition.
Start address	Shows the start address for each partition.
End address	Shows the end address for each partition.
Type	Shows the type of partition. Partitions created by the following can be recognized: MS-DOS — MS-DOS operating system MS-Flash — MS-FFS2 Old FFS — MS-FFS1 Vendor — Memory card manufacturer Areas where no partitions are created are shown as “free.”
Status	Shows whether or not the partition is formatted. If the partition was created by the card manufacturer, it is shown as “Unrecognized.”
Size (partition)	Shows the size (in kilobytes) of each partition.

5-10 Other Utilities

The following PCM Plus utilities are explained here.

- PCM.EXE — Reads the status of PC cards.
- PCMRMAN.EXE — Displays and changes resources.
- PCMFDISK.EXE — Sets partitions for ATA-conforming PC cards.
- DEINSTAL.EXE — Deletes or invalidates information set by the setup program (PCMSETUP.EXE).

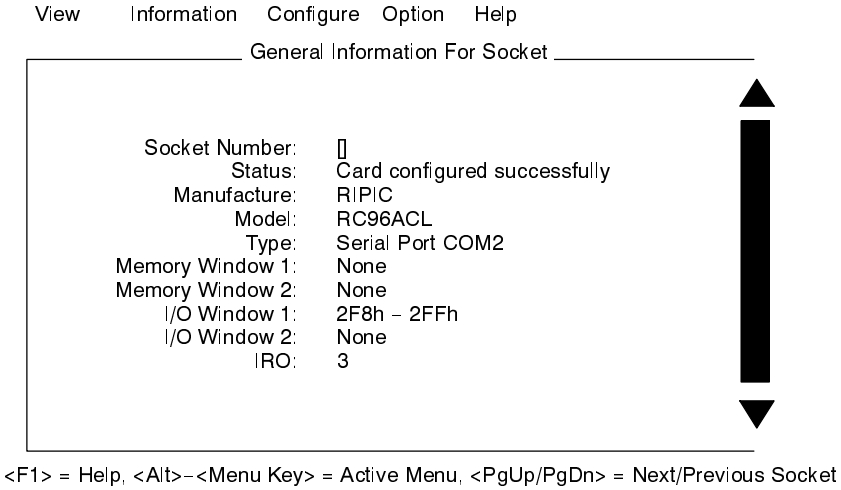
5-10-1 PCM.EXE

PCM.EXE is a utility for displaying the status of PC cards. The following table shows the information that is read by this utility. The display contents will vary, however, depending on the PC card that is used.

Displayed item	Explanation
Socket Number	The socket number that is currently being used by the PC card is shown. If the personal computer has two or more PCMCIA sockets, and if two or more PC cards are currently installed, the display can be changed by means of the PageUp and PageDown Keys.
Manufacturer	The name of the manufacturer is shown.
Model	The model type or serial number is shown. For example, with a LAN card this would show whether it was Ethernet or token ring, and it could also show the serial number.
Type	The card type (e.g., network LAN adapter, flash memory, modem/fax, etc.) is shown.
Compliance	The PCMCIA standards version that the PC card conforms to is shown.
Configuration Loaded	The configuration set for using the PC card is shown.
Configuration Registers	The status of registers storing configuration information is shown in hexadecimal.
Memory-mapped Window	The range of addresses used by the memory-mapped window for PC cards is shown. It is also shown here whether the access type is 16 bits or 8 bits.
I/O Window	The I/O window range for PC cards is shown. It is also shown here whether the access type is 16 bits or 8 bits.
IRQ	Interrupt request levels for PC cards are shown here.

PCM.EXE Main Screen

To display the PCM.EXE startup and main screen, type PCM and press the Enter Key.



5-10-2 PCMRMAN.EXE

PCMRMAN.EXE is a resource management utility. It can be used to display and change resources (e.g., memory, I/O addresses, IRQ levels) used by PCM Plus.

⚠ Caution The distribution of system resources is a highly technical procedure. Use PCMRMAN.EXE with extreme caution. If the wrong settings are made the computer may not be able to recognize PC cards.

The example procedure provided here shows how to use PCMRMAN.EXE to change the IRQ level. In this example, the setting is changed to "Excluded" so that the PC card cannot use IRQ10.

1, 2, 3...

1. Take out the PC card.
2. At the PCM Plus directory prompt, type "PCMRMAN" and press the Enter Key. A warning message will then be displayed. Be sure to read it carefully.
3. Press any key to continue. The main screen (Upper Memory Usage) will be displayed.

File Display Modify Help

Upper Memory Usage

Memory Range	System Status	User Override
A000–A0FF	Available	None
A100–A1FF	Available	None
A200–A2FF	Available	None
A300–A3FF	Available	None
A400–A4FF	Available	None
A500–A5FF	Available	None
A600–A6FF	Available	None
A700–A7FF	Available	None
A800–A8FF	Available	None
A900–A8FF	Available	None
AA00–AAFF	Available	None
AB00–ABFF	Available	None
AC00–ACFF	Available	None
AD00–ADFF	Available	None
AE00–AEFF	Available	None
AF00–AFFF	Available	None

<Alt>–<Menu Key> = Active Menu, <PgUp/PgDn> = Next/Prev Page, <F1> = Help

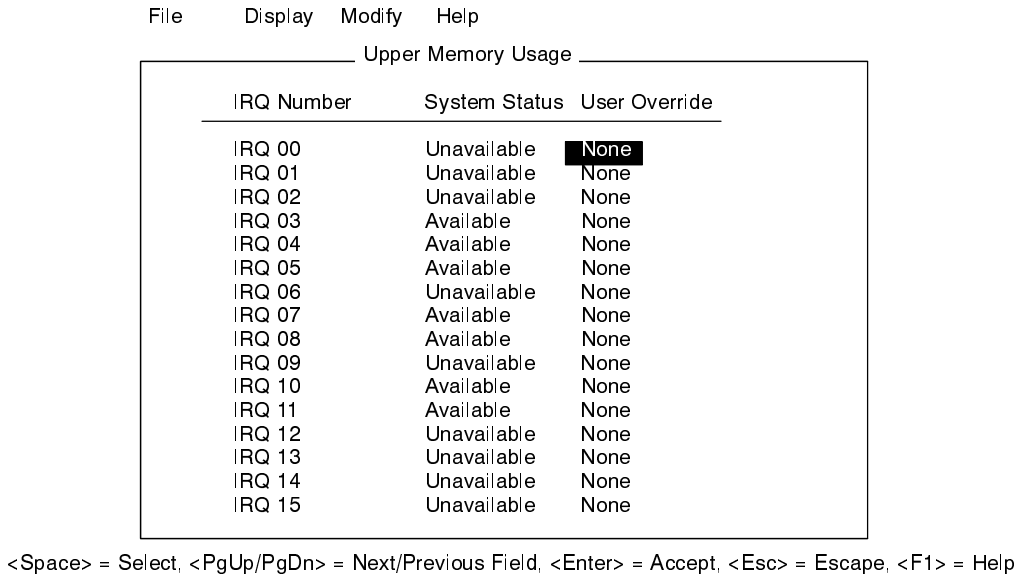
4. While in the main screen, press the Alt and M Keys simultaneously. The "Modify" submenu will be displayed.

Memory Ranges

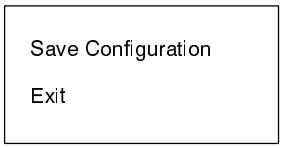
I/O Ranges

IRQ Number

5. Select the IRQ number and press the Enter Key. The following screen will be displayed.



6. Move the cursor to the IRQ10 field.
7. Press the Space Key twice to specify "Excluded."
8. After all the required settings have been made, press the Enter Key.
9. While in the main screen, press the Alt and F Keys simultaneously. The following submenu will be displayed.



10. Select "Save Configuration" and press the Enter Key. A message will be displayed for confirmation.
11. Select "Yes" and press the Enter Key. The changes will be saved to the PCM.INI file, and the new settings will go into effect when the computer is reset.

    If you select "No," the changes will not be saved and you will be returned to the main menu.

    To exit the PCMRMAN utility and return to the DOS prompt, select "Exit" under "File" in the main menu.

5-10-3 PCMFDISK.EXE

PCMFDISK.EXE is a utility for setting partitions for ATA-conforming PC cards. The settings are made in the same way as with the DOS FDISK command.

⚠ Caution When new partitions are created with PCMFDISK.EXE, all data on the PC card will be destroyed. Be sure to back up all essential data before creating partitions. Do not use PCMFDISK.EXE on SRAM PC cards formatted for use with PCMATA (the Phoenix ATA-IDE device driver) because PCMATA treats SRAM cards more like floppy disk drives than hard disk drives.

Note Before creating partitions with PCMFDISK.EXE, first use the PCMSETUP utility's Advanced Mode to set the number of partitions.

5-10-4 DEINSTAL.EXE

DEINSTAL.EXE deletes or invalidates PCM Plus information set by the setup program (PCMSETUP.EXE). This utility is automatically started up if PCM Plus information already exists in the system when PCMSETUP.EXE is executed. PCMSETUP.EXE cannot be executed if PCM Plus information exists in the system. Use this utility to delete or invalidate those settings.

This utility can also be started by typing DEINSTAL.EXE at the PCM Plus directory prompt and then pressing the Enter Key.

5-11 Drivers and Memory Management

In addition to utilities and programs, PCM Plus also includes drivers that provide essential services for PC cards. The drivers are automatically loaded when the setup program (PCMSETUP.EXE) is used. In addition, parameters such as memory and slot allocation and adress sizes are automatically set. The driver settings may require some fine-tuning, however, in order to obtain maximum performance. The drivers that support PCM Plus are covered here.

⚠ Caution In the normal course of operation, the default settings for the drivers should be adequate for most systems. Editing these settings requires some knowledge and experience in memory optimization. For anyone without such knowledge and experience, it is recommended that the default settings be used.

5-12 Types of PC Cards and Associated Drivers

The following table shows the drivers that are required for each type of PC card.

File name	PC card				
	ATA IDE	**Flash memory**	**LAN**	**Modem/Fax**	**SRAM**
PCMSS.EXE Socket Services	Required	Required	Required	Required	Required
PCMCSFUL.EXE Card Services	Required	Required	Required	Required	Required
PCMCS.EXE Card Services	Required		Required	Required	Required
PCMSCD.EXE LAN, fax/modem			Required	Required	
PCMATA.SYS IDE HDD emulation	Required				Required
PCMFFCS.EXE PC-FLASH support		Required			Required
PCMMTD.EXE PC-FLASH support		Required			Required

All PC cards require the installation of PCMSS.EXE and PCMCSFUL.EXE (or PCMCS.EXE), which must be loaded after CNFIGNAM.EXE in the CONFIG.SYS file.

PCM Plus supports multiple ATA cards.

The Microsoft Flash File System must be installed in order for MS-Flash to be supported.

When the ATA driver is used, SRAM Memory cards can be used as hard disk emulations or as MS-Flash cards under the Flash Memory driver.

5-13 PCM Plus Drivers

5-13-1 Summary of Driver Functions

The following table summarizes the basic functions of the various drivers.

Driver	Function
CNFIGNAM.EXE	Specifies which PCM Plus configuration parameters should be accessed from the PCM.INI file.
PCMATA.SYS	Uses SRAM cards, ATA-type hard disk cards, and ATA-type Flash memory cards for IDE hard disk emulation.
PCMCSFUL.EXE or PCMCS.EXE	Supports Card Services. Works at the operating system level to manage all PCMCIA cards. When flash memory cards are used, PCM Plus installs PCMCSFUL.EXE; for other cards it installs PCMCS.EXE.
PCMFFCS.EXE	Required when flash memory cards are used. Acts as high-level bridge between MS-FLASH.SYS and Card Services.
PCMMTD.EXE	Required when flash memory cards are used. Supports the Microsoft Flash File System (FFS) protocol.
PCMRMAN.SYS	Initializes Card Services resource database.
PCMSCD.EXE	Used with fax/modem and LAN cards.
PCMSS.EXE	Supports Socket Services. Works at the BIOS level as an interface between the PCMCIA card socket and the Card Services program.

Note When a different type of PC card is inserted into the same slot, an error may occur. If an error occurs, interrupt access to the new card, replace it with the previous card, access the previous card using the DIR command, and replace the previous card with the new card and check its operation.

5-13-2 MS-FLASH.SYS

MS-FLASH.SYS is a device driver for utilizing the Microsoft Flash File System. This driver is required in order to use flash memory cards.

Note Socket Services (PCMSS.EXE), Card Services (PCMCS.EXE), and PCMFFCS.EXE must all be installed in order to use MS-FLASH.SYS.

This driver is described by the following syntax:

`DEVICE=path\MS-FLASH.SYS` (option)

The options that can be specified are shown in the following table. The values in parentheses are the values that will be set if no option is specified.

Option	Meaning
/nocomp	Specify if FLASHCMP.EXE is loaded but files do not need to be compressed. (For details regarding FLASHCMP.EXE, refer to *5-13-3 FLASHCMP.EXE*.)
/cleanup=n	Specifies the number of cycles (from 1 to 10) for flash card cleanup. The higher the number, the quicker the cleanup cycle will be. (5)
/erase=n	Specifies the difference between the number of deletions in each block and its adjacent block in the flash card. Any number from 250 to 500 can be specified, and the difference will be kept at or below this value.
/cashe=e|x|n	Specifies the memory to be used for the Flash File System cache. e: EMS memory (or “x” if there is no EMS memory) x: Extended memory n: Cache not used.

Note If FLASHCMP.EXE is to be used, it must be installed before MS-FLASH.SYS.

CONFIG.SYS Example

```
DEVICE=path\dos\EMM386.EXE ram X=C800-C8FF X=C900-CCFF
DOS=HIGH umb
        •
        •
REM by PCM+ V3.20   X=C800-C8FF    X=C900-CCFF
DEVICE=path\PCM320\DPMS.EXE
DEVICEHIGH=path\PCM320\CNFIGNAM.EXE /DEFAULT
DEVICEHIGH=path\PCM320\PCMSS.EXE
DEVICEHIGH=path\PCM320\PCMCSFUL.EXE
DEVICEHIGH=path\PCM320\PCMRMAN.SYS
DEVICEHIGH=path\PCM320\PCMMTD.EXE
DEVICEHIGH=path\PCM320\PCMSCD.EXE
DEVICEHIGH=path\PCM320\PCMATA.SYS
DEVICEHIGH=path\PCM320\PCMFFCS.EXE
DEVICEHIGH=path\FLASHCMP.EXE
DEVICEHIGH=path\MS-FLASH.SYS
```

5-13-3 FLASHCMP.EXE

FLASHCMP.EXE is a device driver that compressing files in order to use flash card memory more efficiently. Even when this driver is installed, however, files will not be compressed if the "/nocomp" option is added to MS-FLASH.SYS. This driver cannot be used with an ISA Control Unit.

This driver is described by the following syntax:

`DEVICE=path\FLASHCMP.EXE` (option)

Note If FLASHCMP.EXE must be installed before MS-FLASH.SYS.

CONFIG.SYS Example

```
DEVICE=path\dos\EMM386.EXE ram X=C800-C8FF X=C900-CCFF
DOS=HIGH umb
        •
        •
REM by PCM+ V3.20   X=C800-C8FF    X=C900-CCFF
DEVICE=path\PCM320\DPMS.EXE
DEVICEHIGH=path\PCM320\CNFIGNAM.EXE /DEFAULT
DEVICEHIGH=path\PCM320\PCMSS.EXE
DEVICEHIGH=path\PCM320\PCMCSFUL.EXE
DEVICEHIGH=path\PCM320\PCMRMAN.SYS
DEVICEHIGH=path\PCM320\PCMMTD.EXE
DEVICEHIGH=path\PCM320\PCMSCD.EXE
DEVICEHIGH=path\PCM320\PCMATA.SYS
DEVICEHIGH=path\PCM320\PCMFFCS.EXE
DEVICEHIGH=path\FLASHCMP.EXE
DEVICEHIGH=path\MS-FLASH.SYS
```

5-13-4 CNFIGNAM.EXE

The CNFIGNAM.EXE device driver specifies which PCM Plus configuration parameters to use based on the selected boot configuration. The PCM Plus setup utility adds CNFIGNAM.EXE to the CONFIG.SYS file.

Note If the operating system is not running at least DOS 6.0 in multiple boot mode, only one configuration option exists. The operating system for ISA Control Units is DOS 5.0. The Developer's Kit can only be used with DOS 6.0 or later.

PCM.INI defines system resource requirements for all boot configurations for which PCM Plus has been installed. When a boot configuration is specified after startup, CNFIGNAM identifies the selection as CONFIG.SYS is loaded and removes the relevant configuration information from the PCM.INI file.

Syntax `CNFIGNAM.EXE/config_name/P=path`

Arguments

Option	Meaning
/config_name	Identifies the PCM Plus configuration to use. The PCM.INI file may contain multiple setup configurations for PCM Plus. Each of these configurations is identified and separated by square brackets: [ ].
/P=path	Specifies the path of the PCM.INI to use. If no path is specified, or if no PCM.INI exists in the specified path, the PCM.INI in the CNFIGNAM.EXE startup directory will be used. With ISA Control Units, G:\PCM.INI is referenced. If that does not exist, E:\PCM.INI is referenced as the default.

Example

```
CNFIGNAM.EXE/NEW/P=C:\
```

The boot configuration called “NEW” is selected from C:\PCM.INI at the system startup.

Discussion

When CNFIGNAM is input at the DOS prompt with no arguments, the name of the current boot configuration for PCM Plus is displayed.

5-13-5 PCMATA.SYS

PCMATA.SYS, the Phoenix ATA-IDE device driver, acts as an I/O client to Card Services. PCMATA.SYS enables the system to access ATA-configured PCMCIA cards as IDE hard drive devices using an IDE partition table, and maps the first socket (Socket 0) to the first available drive letter (usually D:), and the second socket (Socket 1) to the next available drive letter (usually E:), and so on.

The “PCMATA=” line appears in the PCM.INI file.

Syntax

```
PCMATA.SYS/Sn=x/SRAM/ADDR=nn/SD=mm
```

Arguments

Option	Meaning
/Sn=x	This should not ordinarily be specified. “n” represents the socket number and “x” represents the number of partitions allocated for that socket. Each socket requires its own argument. For example, /S1=2 indicates that socket 1 has two partitions. When no socket is specified, the program configures all installed sockets with one partition. Socket numbers start at zero (0).
/SRAM	Enables PCMATA to access SRAM cards.
/ADDR=nn	“*nn*” specifies the base address of the 16K memory-mapped window required to configure SRAM cards. When no address is specified, Card Services assigns an available address.
/SD=mm	Spins down the drive after *mm* minutes. When no time is specified, Card Services assigns no spin down.

Example

```
PCMATA.SYS/S0=1/SD=1/SRAM/ADDR=D0
```

Discussion

PCMCIA Plus enables both the ATA driver (PCMATA.SYS) and the Flash File driver (PCMMTD.EXE) to read SRAM cards. Adding the /SRAM option to PCMATA and loading PCMMTD permits access to one SRAM card with two drivers. As long as the cards have been formatted properly with DOS FORMAT or contain a FAT-compatible disk structure, PCMATA can access the following PCMCIA cards:

- PCMCIA ATA IDE fixed disks that have been properly configured. If necessary, use PCMFDISK to assign partitions.
- PCMCIA 1.0/2.0 compliant SRAM PC card (Read/Write).

5-13-6 PCMCSFUL.EXE

PCMCSFUL.EXE works as an OS expansion function by controlling access to PC cards and allocating system resources between client drivers. PCMCSFUL works as a driver loaded from CONFIG.SYS. PCMCSFUL must be loaded immediately after the socket driver. Card Services performs the following roles:

- Managing all usable resources for PC cards. When a PC card is inserted into the slot, Card Services determines whether or not the requested resources will be provided for that card.

- Managing client drivers written for special PC cards. Client drivers are registered along with Card Services during initialization (during system startup). When PC card events (such as insertion or removal of cards) occur, or when battery low, card ready, or card lock status changes, Card Services provides the relevant information to the registered client drivers.

The line "DEVICE=PCMCSFUL.EXE" is added to the CONFIG.SYS file. "PCMCSFUL=argument" is written to PCM.INI, and the options are transferred to PCMCSFUL.

Note
1. Be sure to use /RM for the ISA Control Unit.
2. When flash memory cards are not used, PCM Plus uses PCMCS.EXE instead of PCMCSFUL.EXE as the Card Services driver.

Syntax

```
PCMCSFUL.EXE/ADDR=nn/CLIENTS=nn/FLASH/NOBEEP/POLL
/REGIONS=n/WAIT=n/XIRQ=xxxx/RM
```

Arguments

Option	Meaning
/ADDR=nn	Sets internal memory window to address xx00:0000 (start address).
/CLIENTS=n	Sets "n" number of clients that can be registered.
/FLASH	Enables Flash File System compatibility.
/IRQ=n	Specifies IRQn for status change interruptions.
/NOBEEP	Eliminates the beep sound when cards are inserted or ejected.
/POLL	Specifies polling instead of interrupts for card events. Do not use this at the same time as /IRQ.
/REGIONS=n	Sets the maximum number of different memory regions to be used at one time.
/WAIT=n	Sets the time until I/O cards are usable, in units of approx. 55 ms.
/XIRQ=xxxx	This is the bit master for unusable interrupts.
/RM	Forces real mode and disables DOS Protect Modes Services (DPMS). As the default, PCMSFUL uses DPMS and loads most of its executable code into extended memory to maximize available conventional memory.

Discussion

"/ADDR=nn" specifies the Card Services start segment address. Specify the start address with a 2-line HEX segment address. The units for nn are 4K. If this argument is omitted, the first usable address beginning with C000 will be assigned as the default value. The range of addresses is /ADDR=C0h–E0h.

"/CLIENTS=n" specifies the number of client drivers that can be registered along with Card Services. The default setting is /CLIENTS=10.

"/IRQ=n" specifies the IRQ level for Card Services. Do not set this to overlap with any other IRQ. This option defines the IRQ resources to be used for PC card events (insertion or ejection of cards, and so on). This IRQ has no relation to the communication port (COM:).

Default: /IRQ=10

Range: /IRQ=8 to 15

Using "/POLL" enables Card Services polling. This option enables card events based on "sense" instead of interrupts. It is effective in cases such as when there is no IRQ availability. Use "/POLL" to avoid having to use multiple interrupts for monitoring card events when two or more adapter boards are used.

"/WAIT=n" specifies the card settle time in units of approximately 55 ms. This parameter uses a system time check. A PC card becomes usable when this delay time elapses after the card has been inserted. This option is effective when PC cards with long card settle times are used.

Default: /WAIT=1

Range: /WAIT=1 to 100

"/XIRQ=xxxx" specifies the system IRQ to be excluded. With the default, only IRQ10, IRQ11, IRQ12, and IRQ15 can be used as card events by Card Services. These IRQ levels are often used, however, with systems, such as multimedia systems, that stretch functions to their limits. In such cases, use /XIRQ to avoid duplication by excluding from Card Services the IRQ levels that are being used. The values of xxxx are are as follows:

/XIRQ=0400: Card Services does not use IRQ10.

/XIRQ=0800: Card Services does not use IRQ11.

/XIRQ=8A00: Card Services does not use IRQ10, IRQ11, or IRQ15. Only IRQ12 can be used.

"/RM" must be used with ISA Control Units.

5-13-7 PCMMTD.EXE

The PCMMTD.EXE driver enables PCM Plus to support industry-standard SRAM cards, as well as Intel Series I and II Flash Cards (and compatible cards). The setup program (PCMSETUP.EXE) automatically installs PCMMTD if Flash memory cards will be used.

"PCMMTD.EXE = argument" is written to the PCM.INI file.

Syntax

```
PCMMTD/BASE=n/SIZE=n/PART=n/QUEUE=n
```

Arguments

Option	Meaning
/BASE=n	Specifies the memory window base address.
/SIZE=n	Specifies memory window size in kilobytes. Any of the following sizes can be specified: 4, 8, 16, or 32.
/PART=n	Specifies the maximum number of partitions per card.
/QUEUE=n	Specifies concurrent erasures for cards supporting background erasure. Any number from 1 to 9 can be set.

5-13-8 PCMFFCS.EXE

PCMFFCS.EXE acts as a high-level bridge between Microsoft's Flash File System driver (MSFLASH.SYS) and Card Services. Phoenix's Flash File support driver, PCMMTD.EXE, mediates between Card Services and the actual flash card, and performs low-level functions such as read, write, copy, and erase. This division of high and low level tasks allows PCM Plus to read, and map to the same drive letter, both ATA and flash memory. Additionally, PCMFFCS allows PCM Plus to map the first socket (Socket 0) to the first available drive letter (usually D:) and the second socket (Socket 1) to the next available drive letter (usually E:), and so on. (These are H: and I: for ISA Control Units.)

5-13-9 PCMRMAN.EXE

PCMCSFUL.EXE cannot determine either the allocated system resources or to which I/O addresses, IRQs, and UMBs those resources are assigned. PCMRMAN.SYS loads after PCMCSFUL.EXE, and then scans I/O and RAM space (in the UMB from A000 to FFFF) as well as IRQ assignments. It then checks the PCM.INI file under INSTALL.Resources to determine the usable assignments. Once these are established, PCMRMAN.SYS passes all of the information to PCMCSFUL.EXE which then carries out the device assignments.

When the setup program (PCMSETUP.EXE) is used, PCMRMAN.SYS is written to the CONFIG.SYS file.

Syntax

```
PCMRMAN/NOSCAN
```

Argument

Option	Meaning
/NOSCAN	Disables the PCMRMAN.SYS scan which determines system resources. Before using /NOSCAN, make certain that every system resource is either included with or excluded from PCM Plus in the INSTALL.Resources section of the PCM.INI file. Otherwise, do not include this parameter.

5-13-10 PCMSCD.EXE

The Phoenix Super Client Driver, a Card Services client, increases the efficiency of resource acquisition by requesting system resources (memory, I/O, IRQ) from Card Services. When requested in this way, those resources appear to the system as resident for use by software.

PCMSCD.EXE must reside in the same directory as the active PCM.INI. PCMSCD.EXE reads PCM.INI to determine the boot options and the configuration information used for enabling PCMCIA cards.

When PCMSCD.EXE is booted, selected card configurations are loaded from the PCM.INI file into a list maintained by PCMSCD.EXE. When a PC card is inserted, PCMSCD identifies the card and looks for a possible configuration in that list. If it finds a configuration that matches the card, PCMSCD attempts the configuration. If it cannot find a matching configuration, PCMSCD examines the CIS configurations on the card. If the card is a generic modem/fax/serial card, PCMSCD tries to enable the card as such. If a configuration is successful, PCMSCD beeps once.

Note As a default, the program emits a high-toned beep when a card is inserted or successfully configured. When a card is extracted, PCMSCD emits a lower-pitched beep.

PCM Plus automatically loads PCMSCD.EXE into the CONFIG.SYS file and writes the "PCMSCD=argument" to the PCM.INI files.

Syntax

```
PCMSCD/NOBEEP/NOMS/NODB/NW/RM
```

Arguments

Option	Meaning
/NOBEEP	Disables beeps when a LAN or Fax/Modem PC card is configured.
/NOMS	Modem status information is not saved on power down.
/NODB	Disables internal card information database.
/NOMODEM	Disables the generic configuration for modems. System no longer recognizes fax/modem PC cards.
/NW	Disables wait on error/warning messages.
/RM	Forces real mode and disables DOS Protect Modes Services (DPMS). As the default, PCMSCD.EXE uses DPMS and loads most of its executable code into extended memory to maximize available conventional memory.

5-13-11 PCMSS.EXE

When installation is carried out using an installer, the Socket Services driver PCMSS.EXE is automatically written to the CONFIG.SYS file. To change this option for Socket Services, either use the PCMSETUP utility's Advanced Mode or change the PCM.INI's "PCMSS=argument."

Syntax

```
PCMSS/RS=n/RW=n
```

Argument

Option	Meaning
/RS=n	Reserves socket number n. If there is no argument, no reservation will be made.
/RW=n	Reserves socket controller window number n. If there is no argument, no reservation will be made.

Example

```
PCMSS/RW=3
```

Discussion

"/RS=n" specifies the socket to be reserved. This socket reservation option prevents Socket Services from conflicting with the special PCMCIA software that directly accesses socket hardware. Also, use this option to avoid accessing sockets that must not be accessed within the system.

"RW=n" specifies the socket controller window to be reserved. Reserving I/O or memory windows prevents conflicts with software not assigned from Card Services. "n" specifies the first socket's beginning window number, starting from 0. The quantity and type of windows are determined by the controller, so there is no difference between I/O and memory windows.

5-14 CV-series PC Setup

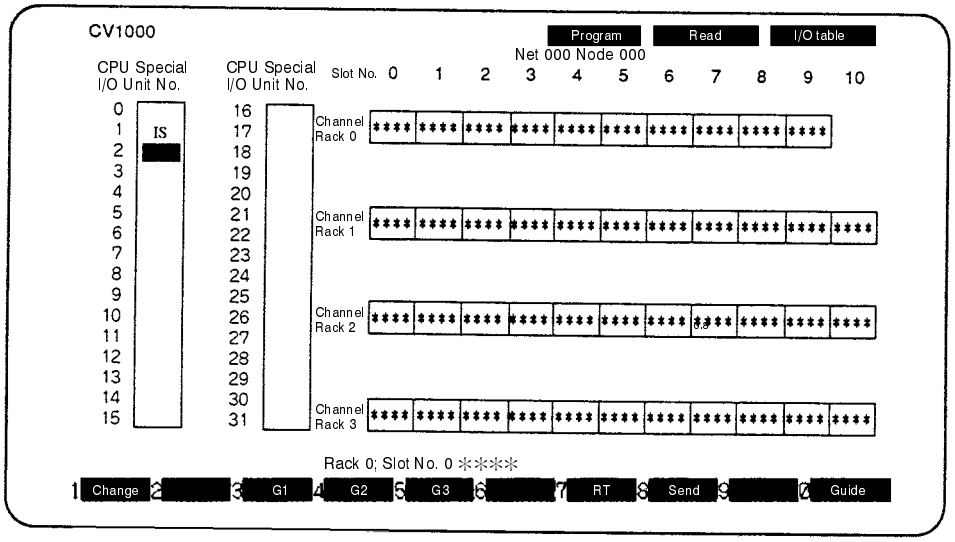
Before the CPU Bus Driver can be used, the I/O table must be set for the PC from a peripheral device as described below, to enable communications between the Developer's Kit and the PC. The method shown here uses CV Support Software.

1, 2, 3... 1. Turn ON the power supply to the personal computer and the PC.

⚠ Caution Be sure to turn ON the power supply to the personal computer before the PC. If the PC is already powered up when the computer is turned ON, the computer will not operate properly.

2. Connect the personal computer to the PC.
3. Boot up the CV Support Software.
4. Put the CV-series PC in Program Mode and create the I/O table.
5. Again power up the personal computer and the PC.

⚠ Caution Be sure to turn ON the power supply to the personal computer before the PC. If the PC is already powered up when the computer is turned ON, the computer will not operate properly.



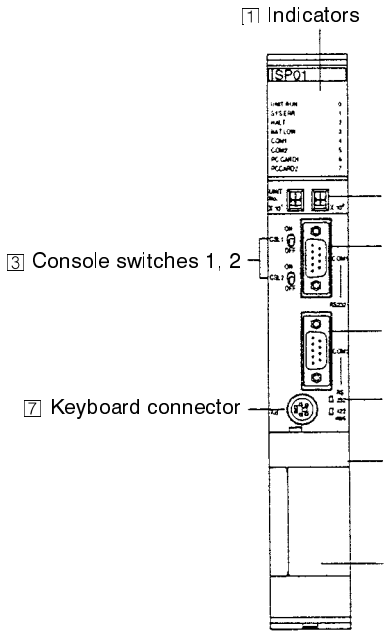
Note When "IS" is displayed for a set unit No., it means that the I/O table has been successfully created.

SECTION 6
ISA Control Unit Setup

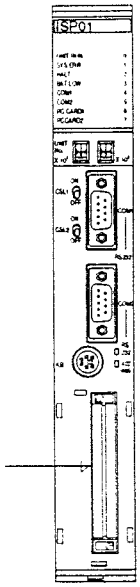
This section explains the hardware and software setup for ISA Control Units.

6-1 Nomenclature and Functions

6-1-1 Front of Unit



Front of Unit



Front of Unit with PC card cover removed

Functions

The functions of the items shown in the illustration are described below. (The references are to related operations covered in this manual.)

1 **Indicators**
Show the operating status of the Unit. (Refer to *6-1-2 Indicators*.)

2 **Unit Number Setting Switch**
Sets the unit number for the ISA Control Unit. (Refer to *6-2-1 Setting the Unit Number*.)

3 **Console Switches 1 and 2**
Set up the Console. (Refer to *6-2-2 Setting Up the Console*.)

4 **Serial Communications Connector (COM1)**
For connecting signal cable for devices using RS-232C. (Refer to *Connecting Cables* under *6-3-6 Installing MS-DOS Console Terminals*.)

5 **Serial Communications Connector (COM2)**
For connecting signal cable for devices using RS-232C or RS-422A/RS-485. (Refer to *Connecting Cables* under *6-3-6 Installing MS-Dos Console Terminals*.)

6 **Indicator for Serial Communications (COM2)**
Indicate the interface specifications (RS-232C or RS-422A/RS-485) used by COM2. (Refer to *Indicators for Serial Communications (COM2)* under *6-1-2 Indicators*.)

7 **Keyboard Connector**
For connecting the keyboard signal cable. (Refer to *6-3-7 Installing a VGA Board/Monitor and Keyboard*.)

8 **PC Card Slot Cover**
Remove this cover to install PC cards. (Refer to *6-3-5 Installing and Removing PC Cards*.)

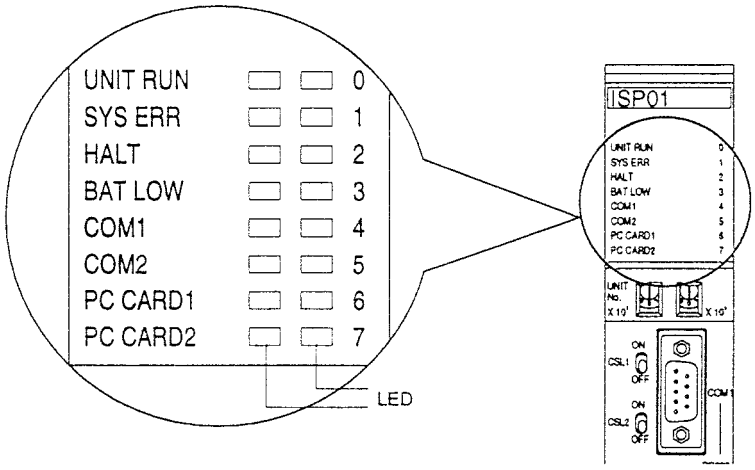
9 **PC Card Cable Pull-out Area**
Area where the cable for modem cards, LAN cards, and so on, is pulled out from. It should normally be closed during operation. (Refer to *6-3-5 Installing and Removing PC Cards*.)

⑩ PC Card Slots
Insert PC cards here. (Refer to *PC Card Slots* under *6-3-5 Installing and Removing PC Cards*.)

6-1-2 Indicators

The operating status of the ISA Control Unit and peripheral devices can be checked using these indicators. For details regarding error processing, refer to *9-1 Indicators*. For details regarding user indicators, refer to *Indicator Register* in *Appendix H ISA Control Unit Memory Configuration*.

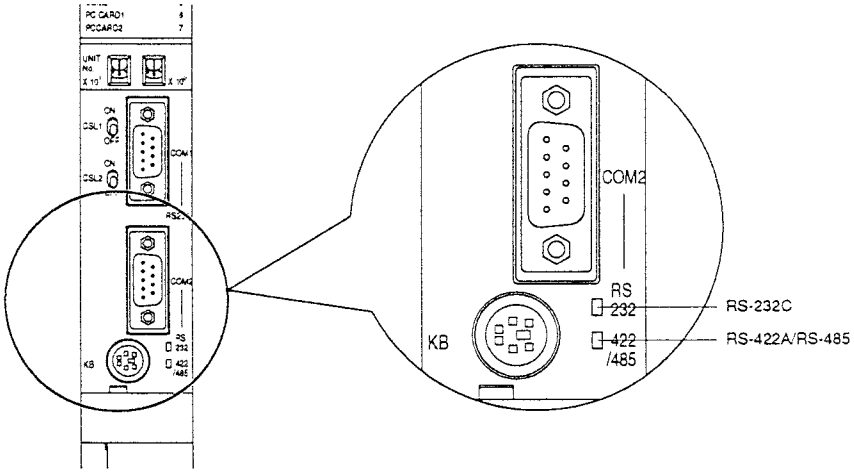
Meanings of Indicators



Indicator	Color	Meaning When Lit
UNIT RUN	Green	The Unit is running.
SYS ERR	Red	An error occurred in the BIOS diagnosis, or there is a CPU bus error.
HALT	Red	NMI (due to I/O check error or memory parity error).
BAT LOW	Red	The battery voltage is low.
COM1	Orange	Port 1 (COM1) is being accessed. (The indicator flashes only while the port is being accessed.)
COM2	Orange	Port 2 (COM2) is being accessed. (The indicator flashes only while the port is being accessed.)
PC CARD 1	Orange	PC card 1 is being accessed. (The indicator flashes only while the card is being accessed.)
PC CARD 2	Orange	PC card 2 is being accessed. (The indicator flashes only while the card is being accessed.)
0	Orange	User-defined/Error
1	Orange	User-defined/Error
2	Orange	User-defined/Error
3	Orange	User-defined
4	Orange	User-defined/Error
5	Orange	User-defined/Error
6	Orange	User-defined/Error
7	Orange	User-defined

Indicators for Serial Communications (COM2)

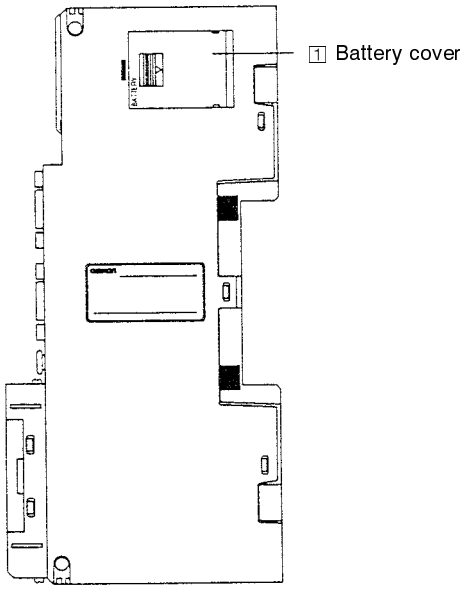
These indicators show the COM2 port communications interface specifications.



Indicator	Color	Meaning When Lit
RS-232C	Orange	The COM2 port is using RS-232C specifications.
RS-422A/ RS-485	Orange	The COM2 port is using RS-422A or RS-485 specifications.

Note If nothing lights, it may mean that the COM2 switch 1 and 2 settings are wrong. Refer to *6-2-4 Setting Serial Communications Port 2 (COM2)* to correctly set these switches.

6-1-3 Side of Unit

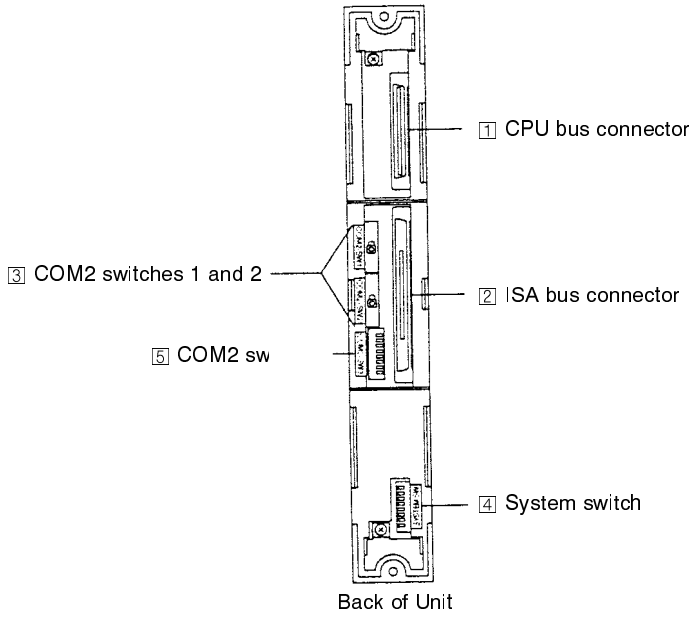


Side of Unit

1 Battery Cover

Open the battery cover to replace the backup battery. (Refer to *9-7 Changing the Battery*.)

6-1-4 Back of Unit



Functions

The functions of the items shown in the illustration are described below. (The references are to related operations covered in this manual.)

1 CPU Bus Connector
This is the connector for exchanging data with the Programmable Controller. (Refer to *6-3-2 Mounting ISA Control Units*.)

2 ISA Bus Connector
This is the connector for exchanging data with ISA Bus Expanders and Hard Disk Units. (Refer to *6-3-2 Mounting ISA Control Units*.)

3 COM2 Switches 1 and 2
These switches set the interface specifications (RS-232C or RS-422A/RS-485) for the serial communications port. (Refer to *6-2-4 Setting Serial Communications Port 2 (COM2)*).

4 System Switch
This switch sets the system (CPU bus memory address, CPU bus interrupt level, etc.). (Refer to *6-2-3 System Setup*.)

5 COM2 Switch 3
Set this switch when using RS-422A/RS-485 for the serial communications port (COM2). (Refer to *6-2-5 Setting Up RS-422A/RS-485 Communications Control*.)

6-2 Setting Switches

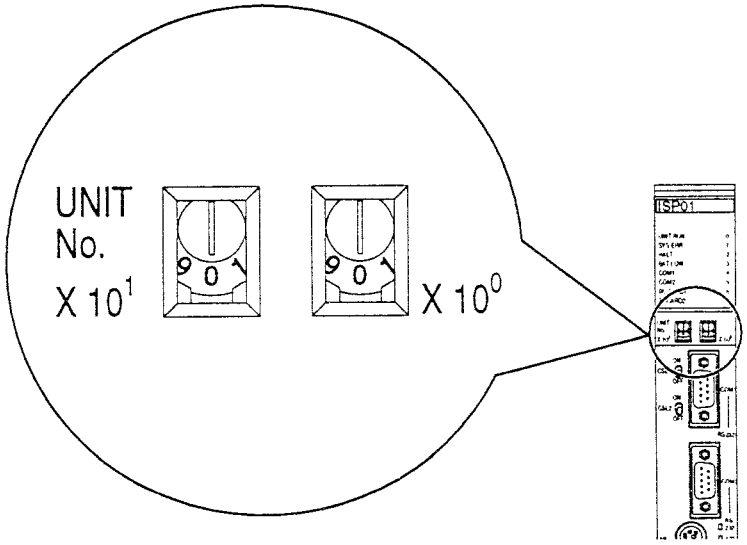
This section explains how to set the various switches for the ISA Control Unit.

6-2-1 Setting the Unit Number

Make this setting by turning the two rotary switches with a small flat-head screwdriver. Be careful not to damage the slots in the switches.

The switch on the left (as you face the Unit) sets the “10” digit, and the switch on the right sets the “1” digit. Set a unit number from 0 to 15.

Note Do not set a unit number that overlaps with any unit number already in use for another Special I/O Unit. Also do not set a number higher than 16.

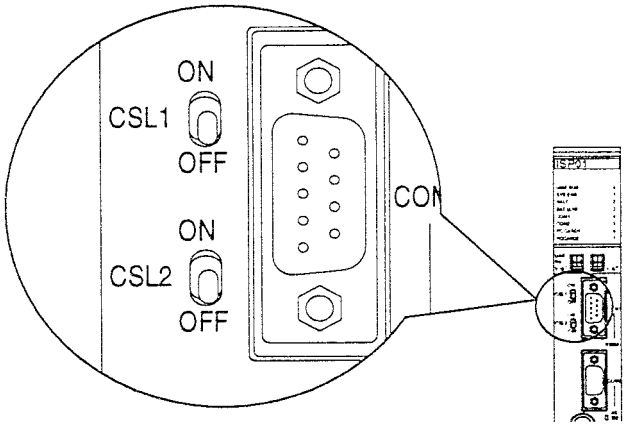


6-2-2 Setting Up the Console

Use console switches 1 and 2 to make the Console setting. Any of the following Console connection options can be selected: not connected, connected to COM1 port, connected to COM2 port, or connected to the VGA monitor and keyboard.

Note Using the VGA monitor + keyboard option requires a CV500-ISX01 ISA Bus Expander, a CV500-ISB01/02 ISA Sub-backplane, a 3G8F5-KB011 AT Keyboard, and a commercially available VGA board (half-size) for the ISA bus.

Make the settings as shown below.



Console switch 1	Console switch 2	Connection option
OFF	OFF	Not connected.
OFF	ON	Connected to COM2
ON	OFF	Connected to COM1
ON	ON	Connected to VGA and keyboard.

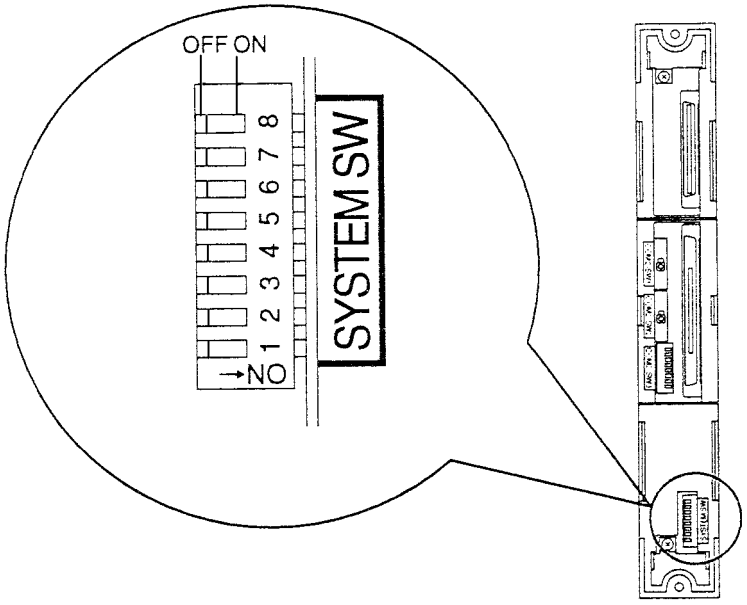
- When setting either COM1 or COM2, refer to *6-3-6 Installing MS-DOS Console Terminals* and set the device connections and software.
- When setting COM2, use RS-232C.

- When setting VGA monitor and keyboard, refer to *6-3-7 Installing a VGA Board/Monitor and Keyboard* and set the device connections and software.
- When no connection is set, the MS-DOS edit I/O will be null.

6-2-3 System Setup

The system settings are made using the system switch on the back of the Unit. The following settings can all be made by this switch: shared RAM address for the CPU bus, CPU bus interface interrupt level, serial communications port 1 (COM1) and serial communications port 2 (COM2) enable/disable, communications format for serial communications ports, CPU clock, and the startup system configuration.

Make the settings as shown below.



Note Make the settings as required. The default settings are all OFF. Only change the settings for the CPU bus shared RAM address and the CPU bus interface interrupt level (No. 1 to 3) if the settings overlap with another board.

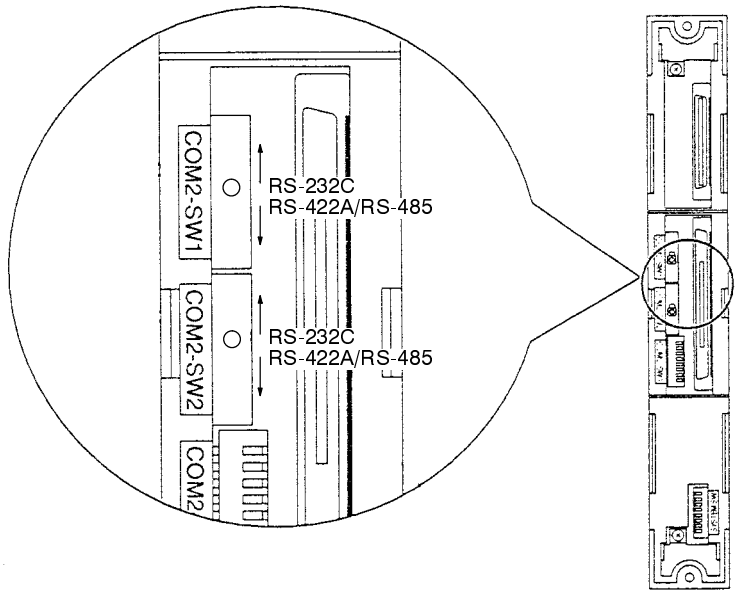
Pin	Item	Setting	
		OFF	ON
1	---	Leave this pin OFF. Do not set it to ON.	
2	Shared RAM address setting for CPU bus	0DC000h to 0DFFFFh	0D8000h to 0DBFFFh
3	CPU bus interface interrupt level setting	IRQ11	IRQ10
4	COM1 enable/disable (Note 1)	Enables usage of COM1.	Disables usage of COM1.
5	COM2 enable/disable (Note 1)	Enables usage of COM2.	Disables usage of COM2.
6	Communications format setting for serial communications port (Note 2)	9,600 bps, 8 bits, no parity, 1 stop bit (8N)	9,600 bps, 7 bits, even parity, 1 stop bit (7E)
7	CPU clock selection (Note 3)	25 MHz (external 50 MHz). ISA bus: 8.33 MHz.	20 MHz (external 40 MHz). ISA bus: 6.67 MHz.
8	Startup system configuration setting (Note 4)	Drive G system files (CONFIG.SYS, AUTOEXEC.BAT, PCM.INI) are searched.	Drive G system files are not searched.

Note 1. Set pins 4 and 5 when using modem cards for the COM1 and COM2 ports.

2. This format sets the conditions for communications when a COM port is used as a Console.
3. Set pin 7 to ON when using a CV500-HDD11 Hard Disk Unit. Also, set pin 7 to ON when using an ISA Board that does not have a bus clock of 8.33 MHz.
4. If the system does not start up normally when the drive G system file is changed, set pin 8 to ON and boot up the system with just the drive E's STARTUP.INI.

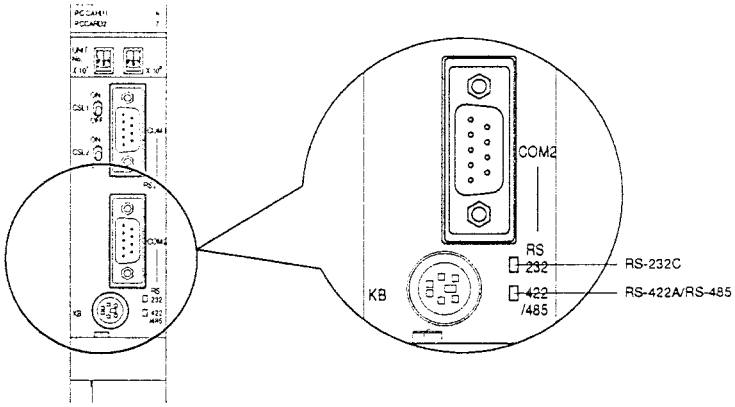
6-2-4 Setting Serial Communications Port 2 (COM2)

Serial communications port 2 (COM2) can be set for use with either RS-232C or RS-422A/RS-485 interface specifications. This setting is made with COM2 switches 1 and 2.



Note
1. Both switches must be set the same way.
2. The interface specifications indicated by the indicator must be used.
3. The Unit and peripheral devices may malfunction if the switches are set for RS-232C when RS-422A/RS-485 devices are connected, or vice versa.

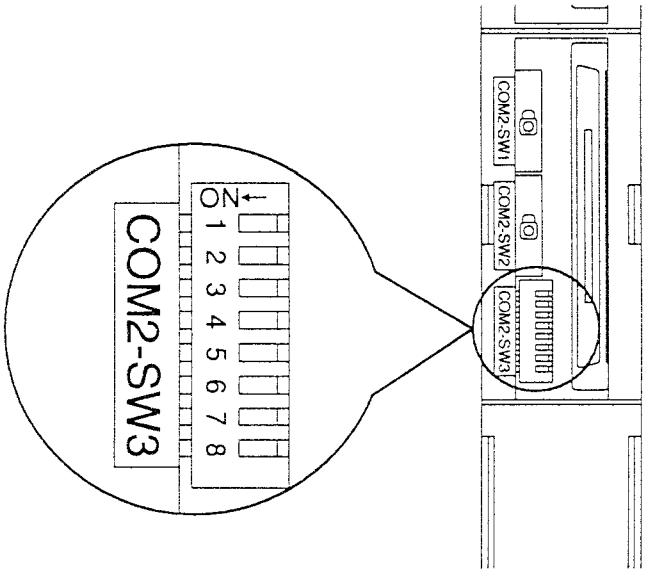
Check the setting at the indicators on the front of the Unit. If the indicators are not lit, it means that the switch settings are wrong.



6-2-5 Setting Up RS-422A/RS-485 Communications Control

Make this setting only when using serial communications port 2 (COM2) with RS-422A/RS-485 interface specifications. This setting is made by means of COM2 switch 3. This switch has no meaning when COM2 is used with RS-232C.

Make the settings as shown below.

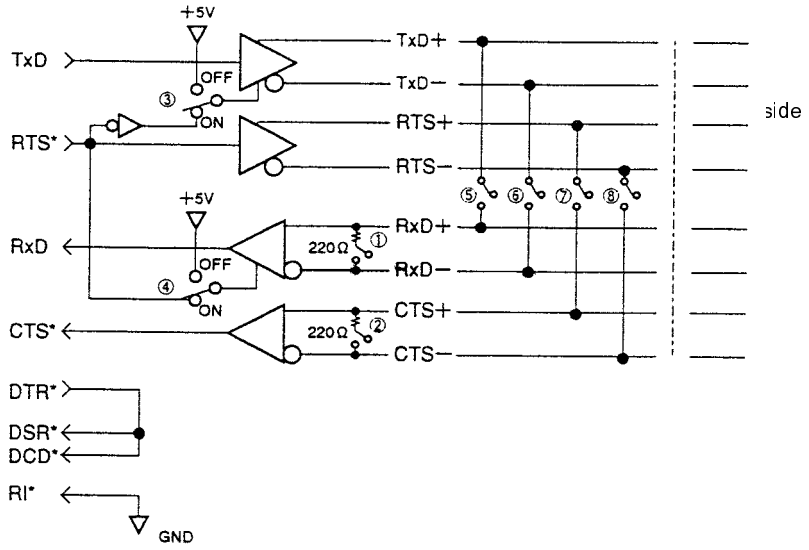


Pin No.	Item	Setting	
		OFF	ON
1	RxD terminator	Terminator not connected	Terminator connected
2	CTS terminator		
3*	TxD driver control	No RTS control (simultaneous enable)	Enable only when RTS active.
4*	RxD receiver control		Enable only when RTS inactive.
5	TxD+ to RxD+ loopback	Loopback not connected.	Loopback connected.
6	TxD– to RxD– loopback		
7	RTS+ to CTS+ loopback		
8	RTS– to CTS– loopback		

Note *For more details regarding pins 3 and 4, refer to the following *Connection Formats*.

Conceptual Diagram of Serial Communications Port 2 (COM2)

In the following diagram, numbers 1 through 8 represent pin numbers.



Connection Formats

Pins 3 and 4 set the connection formats shown in the following table.

		Pin 4	
		OFF	**ON**
Pin 3	**OFF**	RS-422A format • 1:1 • 1:1 (1 side)	---
	ON	RS-422A format • 1:N (N side)	RS-485 format • N:N

6-3 Mounting ISA Control Units and Peripheral Devices

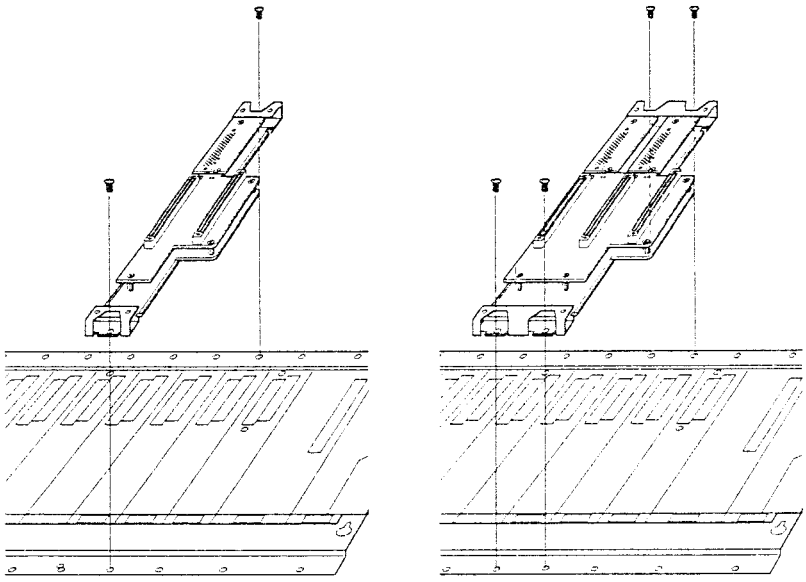
This section explains how to mount ISA Control Units and peripheral devices. In order to carry out the operations described here, an ISA Sub-backplane is required. The first step is to mount the Sub-backplane, and then the ISA Control Unit and peripheral devices can be mounted. If the ISA Control Unit is to be used alone, however, it can be connected directly to the CPU Backplane.

6-3-1 Mounting the ISA Sub-backplane

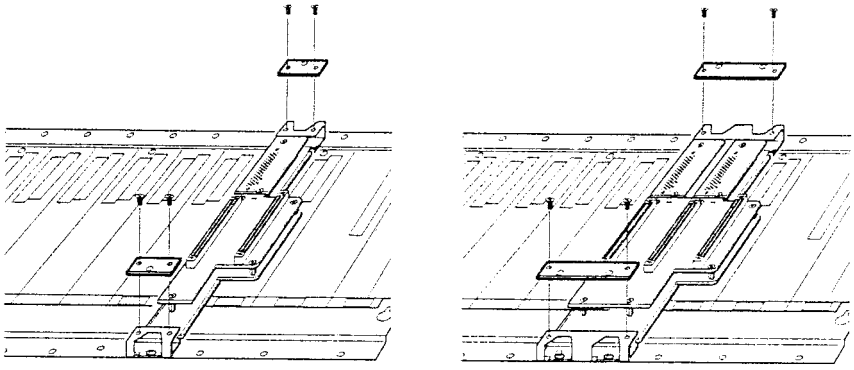
There are two operations involved here. The first is to mount the ISA Sub-backplane to the CPU Backplane, and the second is to mount the plate to the ISA Sub-backplane. The plate is required in order to mount an ISA Bus Expander or a Hard Disk Unit.

Procedure for Mounting the ISA Sub-backplane and Plate

1, 2, 3... 1. Line up the ISA Sub-backplane connectors with the CPU Backplane connectors as shown in the following diagram, and press the Sub-backplane firmly into place.



2. Tighten the binding screws to a torque of 1.2 N • m to secure the ISA Sub-backplane to the CPU Backplane.
3. Fasten the plate to the ISA Sub-backplane by tightening the flat-head screws to a torque of 0.3 N • m.



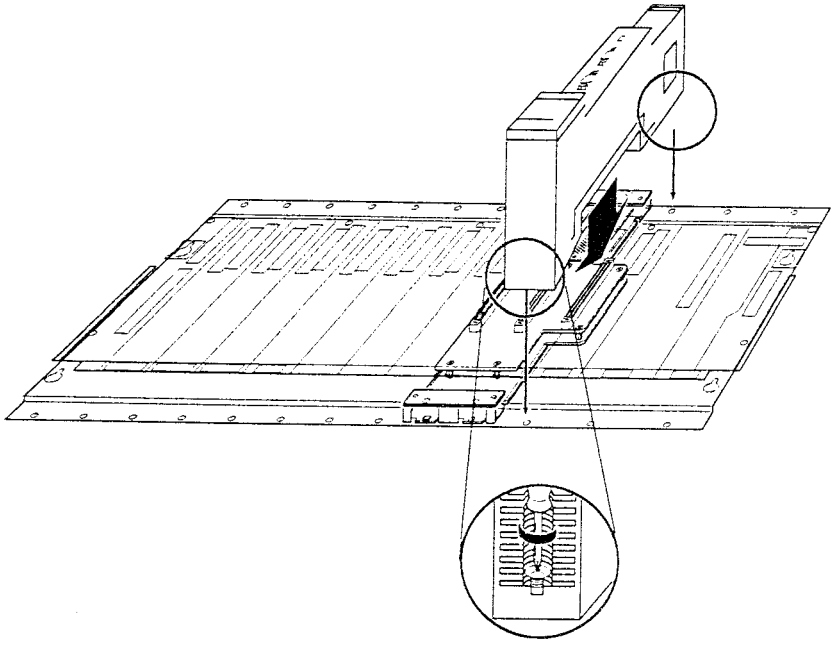
6-3-2 Mounting ISA Control Units

Before mounting an ISA Control Unit, first mount an ISA Sub-backplane to the CPU Backplane. The procedure outlined here uses a CV500-ISB02 ISA Sub-backplane as an example, but the procedure is the same when a CV500-ISB01 ISA Sub-backplane is used.

Note The ISA Control Unit must be mounted before the ISA Bus Expander or Hard Disk Unit. This is true even when a previously mounted ISA Control Unit is removed and subsequently re-mounted. In that case, any ISA Bus Expander or Hard Disk Unit that was previously mounted must also be removed and then re-mounted.

Procedure for Mounting the ISA Control Units

1, 2, 3... 1. Line up the ISA Control Unit connector with the ISA Sub-backplane and CPU Backplane connector as shown in the following diagram, and press the Unit firmly into place.



2. Tighten the binding screws to a torque of 1.2 N • m to secure the ISA Control Unit to the CPU Backplane.

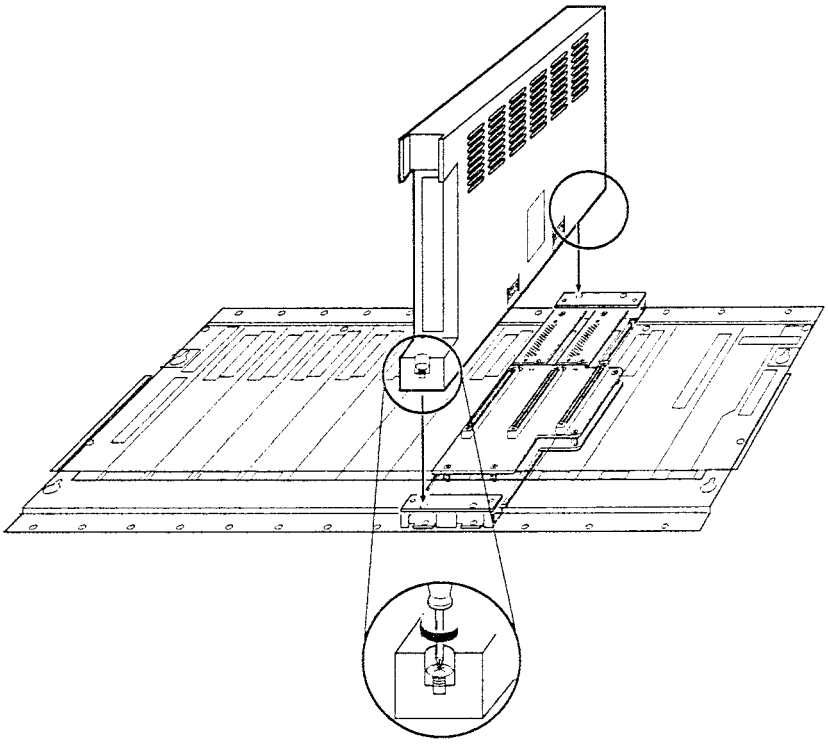
6-3-3 Mounting ISA Bus Expanders

Before mounting an ISA Bus Expander, first mount an ISA Sub-backplane and ISA Control Unit to the CPU Backplane. In addition, an ISA Board must be premounted to the ISA Bus Expander. For instructions on how to do this, refer to the *ISA Bus Expander Operation Manual (W261)* that comes with the ISA Bus Expander.

The procedure outlined here uses a CV500-ISB02 ISA Sub-backplane as an example, but the procedure is the same when a CV500-ISB01 ISA Sub-backplane is used.

Procedure for Mounting the ISA Bus Expanders

1, 2, 3... 1. Line up the ISA Bus Expander connector with the ISA Sub-backplane connector as shown in the following diagram, and press the Unit firmly into place.



Note: The ISA Control Unit is not shown here.

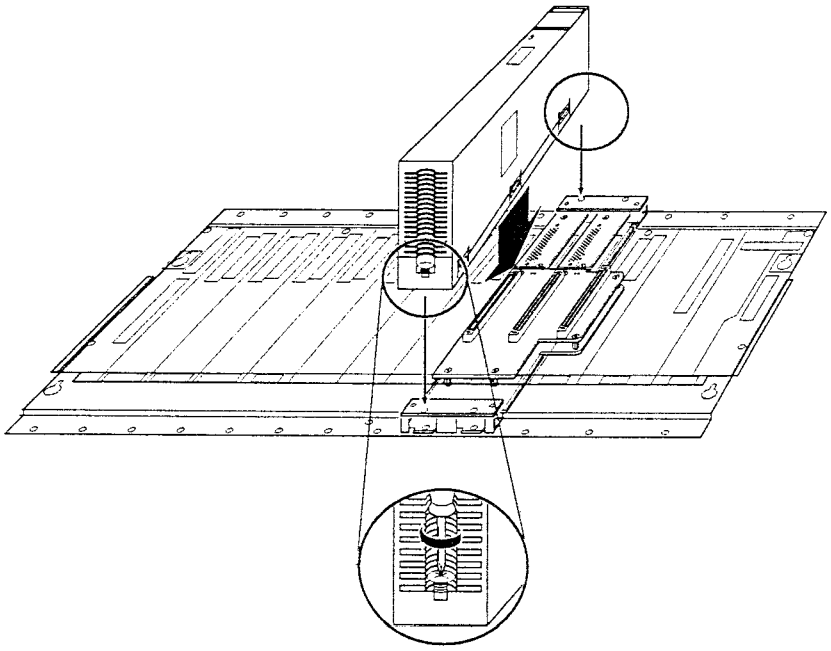
2. Tighten the binding screws to a torque of 1.2 N • m to secure the ISA Bus Expander to the ISA Sub-backplane.

6-3-4 Mounting Hard Disk Units

Before mounting a Hard Disk Unit, first mount an ISA Sub-backplane and ISA Control Unit to the CPU Backplane. Once that has been done, follow the procedure outlined below.

Procedure for Mounting the Hard Disk Units

1, 2, 3... 1. Line up the Hard Disk Unit connector with the ISA Sub-backplane connector as shown in the following diagram, and press the Unit firmly into place.



Note The ISA Control Unit is not shown here.

2. Tighten the binding screws to a torque of 1.2 N • m to secure the Hard Disk Unit to the ISA Sub-backplane.

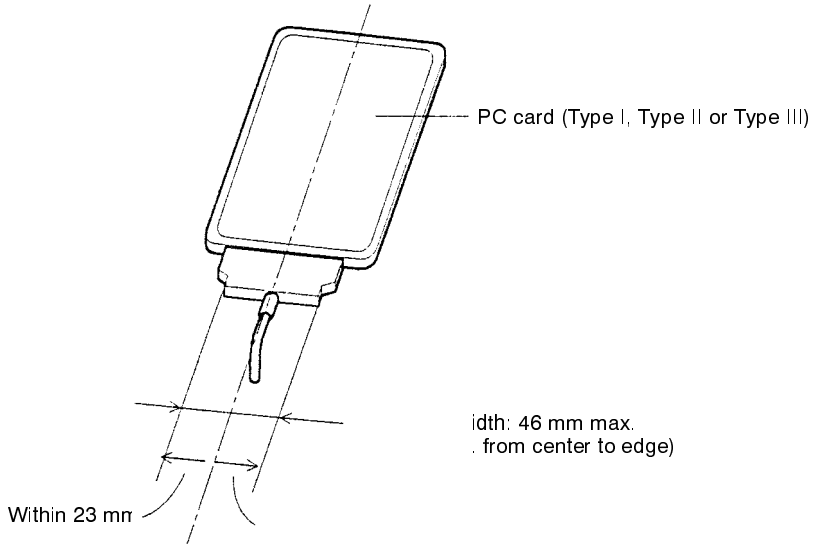
Note In order to use a Hard Disk Unit, the hardware configuration must be set and the hard disk must be formatted. For details, refer to *6-6-1 Setting Up the Hardware Configuration* and *6-6-3 Formatting the Hard Disk*.

6-3-5 Installing and Removing PC Cards

This section explains how to install PC cards in the ISA Control Unit PC card slot, and how to extract them.

PC Card Requirements

When a PC card with an I/O connectors is used, the connector must be no more than 46 mm in width (i.e., the distance from the center of the connector to the edge must be within 23 mm, as shown in the following diagram). If the connector width is any greater than that, the PC card slot cover will not fit.

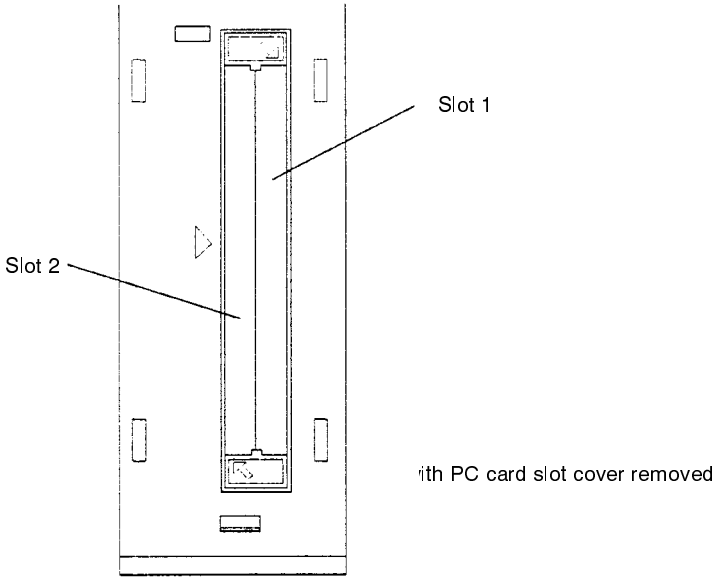


Note There are limitations to PC card current consumption. Use only PC cards that satisfy the following conditions:

I_{5V} (1 slot) ≦ 0.5 A, I_{12V} (1 slot) ≦ 0.1 A

Where I_{5V} (2 slots) + 3.4 × I_{12V} (2 slots) ≦ 1.0 A

PC Card Slots



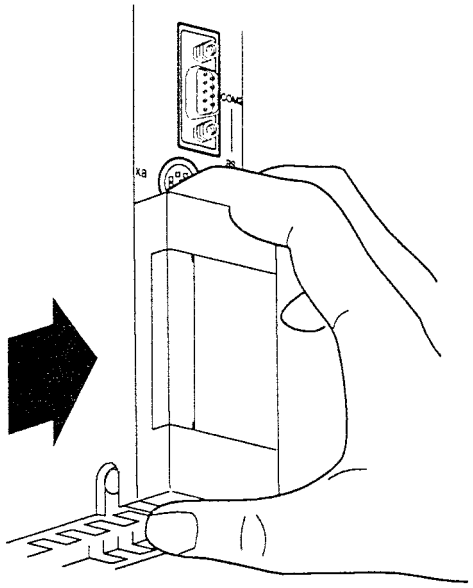
Slot 1: Any Type-I, -II or -III PC card can be installed.

Slot 2: Any Type-I or -II PC card can be installed, but this slot cannot be used when a Type-III PC card is installed in slot 1.

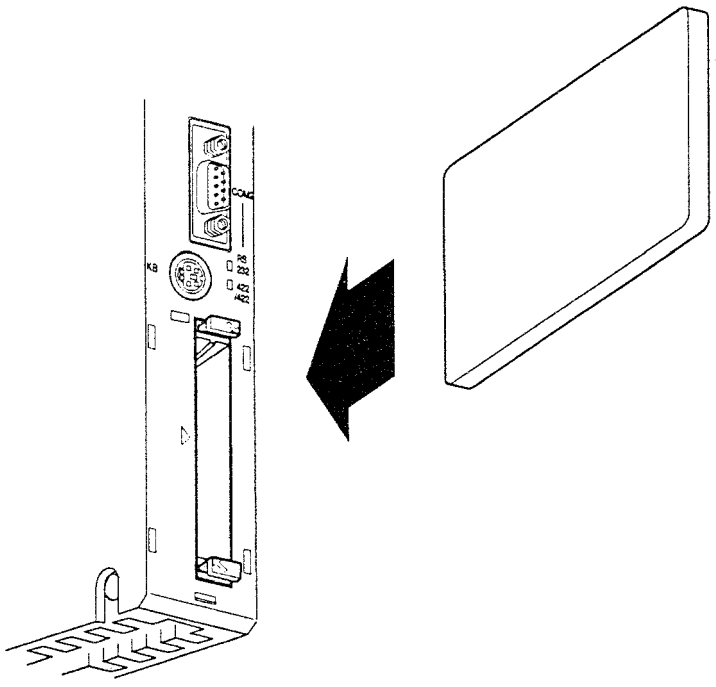
Note When a different type of PC card is inserted into the same slot, an error may occur. If an error occurs, interrupt access to the new card, replace it with the previous card, access the previous card using the DIR command, and replace the previous card with the new card and check its operation.

Installing PC Cards

1, 2, 3... 1. Remove the PC card slot cover as shown in the following diagram.



2. Insert the PC card into the PC card slot, pushing it in until is locks firmly in place.

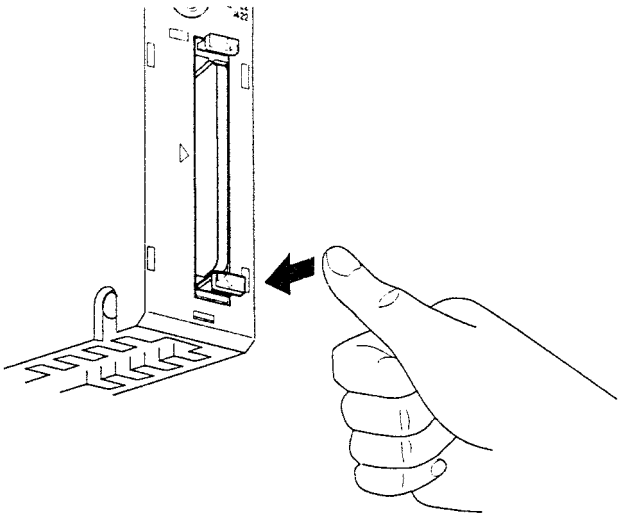


3. Replace the PC card slot cover.

Removing PC Cards

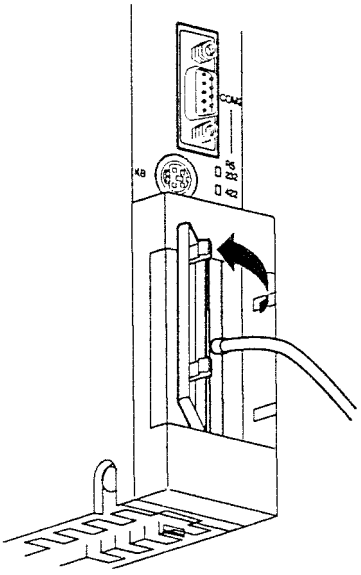
1, 2, 3...

1. Remove the PC card slot cover as shown in the previous procedure.
2. Press the eject button as shown in the following diagram.



3. Replace the PC card slot cover.

Note When memory cards or LAN cards are used, a cable must be connected to the PC card. To use a PC card in this way, pull the cable out of the PC card cable pull-out slot as shown in the following diagram.



6-3-6 Installing MS-DOS Console Terminals

When COM1 or COM2 is set using a Console setting, follow the procedures described below to install the required devices and to set the software.

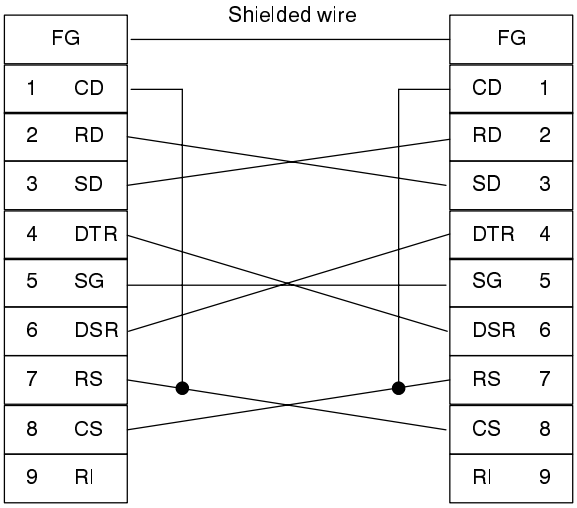
Usable Console Terminals

The following terminals and software can be used with either the COM1 port or the COM2 port.

- ANSI escape sequence display-capable terminals (such as VT100- or VT220-conforming displays).
- ANSI escape sequence display-capable terminal software (such as Windows 95 Hyper Terminal).

Connecting Cables

Use the cables indicated below for 9-pin wiring.



Applicable Connectors

Use the following products or their equivalents:

Socket: XM2D-0901 (OMRON)

Hood: XM2S-0913 (OMRON)

Recommended Cables

AWG28x5P: IFVV-SB (Fujikura Ltd.)

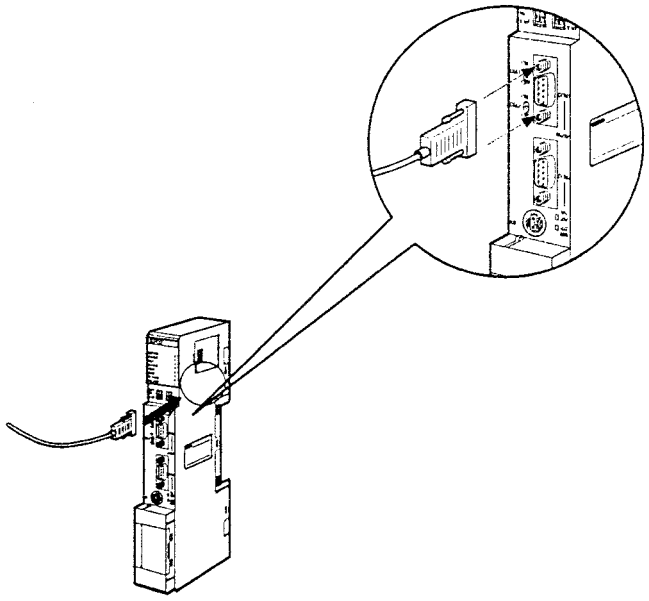
CO-MA-VV-SB: 5PxAWG28 (Hitachi Cable, Ltd.)

Connection Procedure

Make the connection as shown in the following procedure.

1, 2, 3...

1. Connect the cable to the connector selected by the Console setting.



2. Connect the other connector to the communications port of the terminal being used.

⚠ Caution Be sure to turn OFF the power supply to the Unit and also to any external devices before attaching or disconnecting connectors. Doing this with the power turned ON can damage the equipment.

Communications Settings

Make sure that the setting of system switch pin 6 matches the communications settings of the terminal and terminal software that are used. For details regarding communications settings, refer to the manuals for the particular terminal and terminal software that are used.

6-3-7 Installing a VGA Board/Monitor and Keyboard

When "VGA monitor + keyboard" has been specified by the Console setting, follow the procedure outlined below to install the devices and set the software.

Note Using the VGA monitor + keyboard option requires a CV500-ISX01 ISA Bus Expander, a CV500-ISB01/02 ISA Sub-backplane, a 3G8F5-KB011 AT Keyboard, and a commercially available VGA board (half-size) for the ISA bus.

1, 2, 3...

1. Install a VGA board in the ISA Bus Expander. For instructions on how to do this, refer to the *ISA Bus Expander Operation Manual (W261)*.
2. Mount the ISA Bus Expander to the ISA Sub-backplane. For instructions on how to do this, refer to *6-3-3 Mounting ISA Bus Expanders*.
3. Connect the VGA monitor cable to the connector of the VGA board in the ISA Bus Expander.
4. Connect the keyboard cable to the ISA Control Unit's keyboard connector.

⚠ **Caution** Be sure to turn OFF the power supply to the Unit before attaching or disconnecting connectors. Doing this with the power turned ON can damage both the keyboard and the Unit.

6-3-8 Mounting Other RS-232C and RS-422A/RS-485 Devices

In addition to terminals, it is possible to connect other peripheral devices with RS-232C, RS-422A, or RS-485 communications interfaces. The cables that are required depend on the peripheral devices. For details, refer to the specific peripheral device manuals.

⚠ **Caution** Be sure to turn OFF the power supply to the Unit and also to any external devices before attaching or disconnecting connectors. Doing this with the power turned ON can damage the equipment.

Applicable Connectors

Use the following products or their equivalents:

Socket:	XM2D-0901 (OMRON)
Hood:	XM2S-0913 (OMRON)

Recommended Cables

AWG28x5P:	IFVV-SB (Fujikura Ltd.)
CO-MA-VV-SB:	5PxAWG28 (Hitachi Cable, Ltd.)

6-4 ISA Control Unit Startup and System Software Setup

This section explains how to start up the ISA Control Unit and set up the system software.

6-4-1 Starting Up the ISA Control Unit

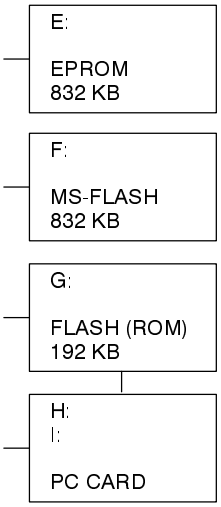
The ISA Control Unit automatically starts up the system when the PC is powered up. The CPU Bus Driver is also installed automatically, and communications with the PC are enabled. In addition, the PC card interface driver is installed automatically, so SRAM, ATA, and FLASH cards can be used.

If the unit number has been changed, perform the operations described in *5-14 CV-series PC Setup*.

When the system starts up normally, the UNIT RUN indicator on the front of the Unit lights. If this indicator does not light, it means that an error has occurred. In that case, refer to *9-1 Indicators* and carry out the specified countermeasures to correct the error.

6-4-2 Configuring the Drives

Configure the ISA Control Unit drives as shown in the following diagram.



6-4-3 Default File Configurations

The default file configurations are as shown below for drives E, F, and G, including the sizes of files and the dates and times when they were last changed.

Drive E Contents

CARDDRVX	EXE	37,104	95-10-26 17:46
CNFIGNAM	EXE	1,724	95-12-27 14:57
DEBUG	EXE	15,242	91-11-08 15:34
DGISKEY	SYS	3,896	95-12-20 9:52
DPMS	EXE	47,375	95-12-27 14:57
EMM386	EXE	91,742	91-11-08 15:45
ERRLOG	EXE	5,800	95-11-09 13:55
FCOPYX	EXE	6,063	95-11-09 14:56
FERASEX	EXE	2,532	95-11-07 0:31
FLASHCMP	EXE	15,940	94-04-30 2:00
FORMAT	COM	21,624	91-11-08 15:38
FTRANS	EXE	84,532	95-12-21 22:07
HDISK	DAT	512	95-08-14 11:08
HDISK	EXE	8,755	95-08-23 15:36
HIMEM	SYS	11,616	91-11-07 10:12
MEMCARD	EXE	46,115	94-04-30 2:00
MS-FLASH	SYS	53,838	94-06-30 2:00
PCM	INI	2,222	95-12-27 15:11
PCMATA	SYS	21,267	95-12-27 14:57
PCMCS	EXE	59,472	95-12-27 14:57
PCMCSFUL	EXE	69,710	95-12-27 14:57
PCMFFCS	EXE	21,111	95-12-27 14:57
PCMMTD	EXE	19,529	95-12-27 14:57
PCMRMAN	SYS	34,509	95-12-27 14:57
PCMSCD	EXE	34,671	95-12-27 14:57
PCMSS	EXE	22,221	95-12-27 14:57
RESET	EXE	5,348	94-05-23 2:00
ROMDISK	SYS	4,479	95-10-21 1:08
ROMUTY	EXE	22,952	95-12-13 12:35
SBUS	SYS	26,701	95-12-20 10:06
SETUP	EXE	9,090	95-09-24 21:12
STARTUP	INI	379	95-12-27 15:19

Drive F Contents

```
ANSI        SYS        9,029    91-11-08  15:43
ASSIGN      COM        6,399    91-11-08  15:31
BACKUP      EXE       36,092    91-10-31  10:38
CHKDSK      EXE       16,200    91-10-31  10:39
COUNTRY     SYS       17,069    91-10-31  11:34
DISKCOPY    COM       11,879    91-11-11  10:33
DISKIMG     EXE       88,581    95-12-24  17:01
DISPLAY     SYS       15,792    91-11-08  15:43
DOSKEY      COM        5,883    91-11-07  11:28
EDLIN       EXE       12,642    91-10-31  11:02
FC          EXE       18,650    91-11-08  15:34
INTERLINK   EXE       17,133    91-10-01  13:27
INTERSVR    EXE       37,266    91-10-21  14:11
JOIN        EXE       17,870    91-11-08  15:38
KEYB        COM       14,986    91-11-07  10:07
KEYBOARD    SYS       34,697    91-11-07  10:13
MEM         EXE       39,818    91-11-08  15:39
MODE        COM       23,537    91-11-08  15:41
MORE        COM        2,618    91-10-31  11:25
RESTORE     EXE       38,294    91-11-08  15:41
ROMINF      DAT           36    95-12-25  15:21
RSCOPY      EXE       38,701    95-10-16  13:36
SHARE       EXE       10,912    91-11-08  15:42
SUBST       EXE       18,478    91-11-08  15:42
XCOPY       EXE       15,820    91-11-08  15:43
```

Drive G Contents

```
AUTOEXEC    BAT           39    95-12-07  18:36
CONFIG      SYS           48    95-12-27  17:01
PCM         INI        2,179    95-12-04  18:03
```

Note Back-up and delete drive F and G files as required.

6-4-4 Default System Setup

The ISA Control Unit's default system setup files are stored in drive E and drive G as listed below.

E:\STARTUP.INI
G:\PCM.INI
G:\CONFIG.SYS
G:\AUTOEXEC.BAT

STARTUP.INI Contents

The contents of STARTUP.INI (which corresponds to CONFIG.SYS) are as follows, with their functions provided below.

```
DEVICE=E:\HIMEM.SYS                                  [1]
DEVICE=E:\EMM386.EXE NOEMS RAM X=C800-CCFF           [2]
X=D800-DFFF                                          [3]
DOS=UMB                                              [4]
DEVICE=E:\CARDDRVX.EXE                               [5]
DEVICE=E:\ROMDISK.SYS                                [6]
DEVICEHIGH=E:\SBUS.SYS /V65 /B3 /C10
DEVICE=E:\DPMS.EXE
DEVICE=E:\CNFIGNAM.EXE /DEFAULT /P=G:\
DEVICE=E:\PCMSS.EXE
DEVICE=E:\PCMCSFUL.EXE                               [7]
DEVICE=E:\PCMRMAN.SYS
DEVICE=E:\PCMMTD.EXE
DEVICE=E:\PCMATA.SYS
SEARCH_PATH=G:;
DEFAULT_DRIVE=G:                                     [8]
```

[1] Device driver that controls how the extended memory and high memory (HMS) are used.

[2] Device driver that emulates EMS memory during extended memory use, and loads programs and device drivers to the UMB area.

[3] Loads part of MS-DOS to the UMB area, thereby increasing the amount of conventional memory space that is available for using application programs.

4 Device driver for drive F

5 Device driver for drive G

6 CPU Bus Driver

7 Device driver for PC cards

8 Indicates that CONFIG.SYS and AUTOEXEC.BAT for user settings can be read from drive G.

PCM.INI Contents

Drive E and drive G have the same files.

```
; PCM.INI - Initialization File for PCM+ Version 3.20

[SETUP-DEFAULTS]

; Allow VAR the ability to specify working defaults and screen order

ResourceCheckMode=1

; Setup mode: 0 == Quick; 1 == Advanced
SetupModeType=1
SetupMode=

SocketCountType=1
SocketCount=

; Command line options
pcmss=
pcmcs=
pcmrman=
pcmscd=
pcmata=
usecards=

WindowsType=1
WindowsPath=c:\WINDOWS

FFSType=1
FFSPath=
FFSWindowBase=
FFSNoOfPartitions=1
FFSNoOfEraseQueues=1

ModemType=1
ComPort=

ATAType=1
ATAWindowBase=
ATASpinDownTime=5
ATANoOfPartitions=1

AdvancedInfoType=1
IRQ=
CISWindowBase=
TokenRingSpeed=4
CardSettleTiem=660

[INSTALL.Resources]
include_mem=
exclude_mem=0xa0-0xbf,0xe0-0xff
include_io=
exclude_io=
include_irq=
exclude_irq=0x00,0x01,0x02,0x08
```

```
[COM}
1=0x3f8,0x8,0x4,0x1
2=0x2f8,0x8,0x3,0x1
3=0x3e8,0x8,0x4,0x1
4=0x2e8,0x8,0x3,0x1

COMORDER=3
[NETWORK]
RS=4

; Card/Configuration List -

[PCMCARD_000]
ID_1=IBM
ID_2=TOKEN RING
MAJOR=0x4
MINOR=0x1
DESCRIPTOR=0x0010
NUMBERCONFIGS=2

[PCMCARD_000.0]
LOAD=YES
RESOURCES=0x3A3
MEMORY_0=0xD0,0x4,0x1,0xD0
MEMORY_1=0xD8,0x2,0x1,0xD8
IO_0=0xA20,0x4,0x0
IRQ=0x9,0x1
CONFIGREGS=0x61,0x20

[PCMCARD_000.1]
LOAD=NO
RESOURCES=0x3A3
MEMORY_0=0xD0,0x4,0x1,0xD0
MEMORY_1=0xD8,0x2,0x1,0xD8
IO_0=0xA24,0x4,0x0
IRQ=0x9,0x1
CONFIGREGS=0x61,0x20

[PCMCARD_001]
ID_1=IBM Corp.
ID_2=Ethernet
MAJOR=0x04
MINOR=0x01
DESCRIPTOR=0x0020
NUMBERCONFIGS=1

[PCMCARD_001.0]
LOAD=YES
RESOURCES=0x3E1
MEMORY_0=0xD4,0x4,0x1,0x4
IO_0=0x300,0x10,0x0
IO_1=0x310,0x10,0x1
IRQ=0x5,0x1
CONFIGREGS=0x41,0x00
```

```
[DEFAULT]
pcmss=/SO=2  /CMDevID=FFFFFFFF
pcmcs=/ADDR=C8  /IRQ=15   /WAIT=12/RM
pcmscd=/RM
pcmrman=
pcmata=/SD=5   /SO=1  /S1=1  /SRAM
UseCards=
SetupMode=1
ComPort=3
ATAWinddowBase=c900
ATASpinDownTime=5
ATANoOfParitions=1
IRQ=15
CISWindowBase=c800
TokenRingSpeed=4
CardSettleTime=660

pcmffcs=/QUEUE=1
FFSPath=E:
FFSWindowBase=c900
FFSWindowSize=16
FFSNoOfPartitions=1
FFSNoOfEraseQueues=1
[DEFAULT.Resources]
include_mem=0xc8-0xc8,0xc9,0xcc
exclude_mem=0xD8-0xDF
include_io=
exclude_io= 0x390-0x39F
include_irq=
exclude_irq= 0x0A-0x0B
```

CONFIG.SYS Contents

```
DEVICE=E:\PCMFFCS.EXE
DEVICE=E:\MS-FLASH.SYS
```

AUTOEXEC.BAT Contents

```
@ECHO OFF
PROMPT $P$G
PATH E:\;F:\
```

6-4-5 Changing the System Configuration

The ISA Control Unit system configuration can be changed by re-writing CONFIG.SYS, AUTOEXEC.BAT, and PCM.INI in drive G. Drive G emulates a ROM disk, so the following procedure must be carried out in order to change its contents.

1, 2, 3...

1. Edit the system-related files at the personal computer.
2. Save the system-related files as disk images at the personal computer.
3. Transfer the disk image files to the ISA Control Unit, and write them to drive G.
4. Restart the ISA Control Unit.

Place device drivers installed by G:\CONFIG.SYS in drive G or on a card.

6-4-6 Editing System-related Files

Use an editor to edit system-related files at the personal computer. For details regarding PC card interface settings, refer to *6-4-7 Setting Up the PC Card Interface Driver*.

Editing operations such as copying CONFIG.SYS, AUTOEXEC.BAT, and PCM.INI created by the Developers Kit, and changing the directory names, can provide greater efficiency.

Converting files to disk images and transferring them can be time-consuming, so it is recommended that suitable system-related files be created at the personal computer and then transferred to the ISA Control Unit.

Note The built-in drive F and flash cards cannot be used if the PCMFFSC.EXE and MS-FLASH.SYS lines are deleted from CONFIG.SYS in drive G. They can be deleted if there is not enough conventional memory or if drive F will not be used.

6-4-7 Setting Up the PC Card Interface Driver

Driver setup can be easily executed by first copying the CONFIG.SYS and PCM.INI created according to the explanation in *5-6 PC Card Interface Driver Setup*. Refer to that section for details regarding PC card interface drivers.

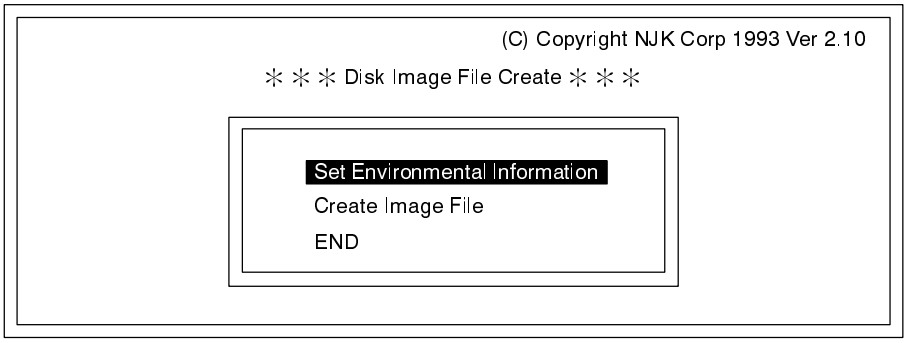
Copying Drivers to Disk For instructions on copying drivers to the Unit's disk, refer to *6-4-10 Transferring Disk Images to the ISA Control Unit and Writing Them to Drive G*.

CONFIG.SYS Example DEVICE=E:\PCMSCD.EXE

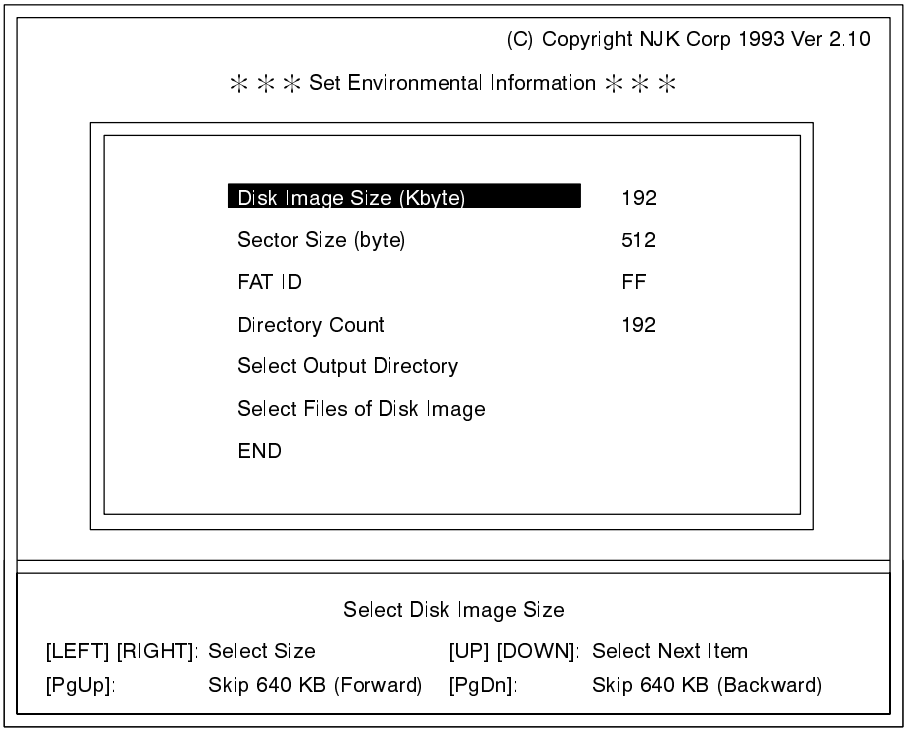
6-4-8 Saving System-related Files as Disk Images

Use the disk image creation utility to save system-related files as disk images at a personal computer with a VGA monitor and keyboard. Follow the procedure outlined below.

1, 2, 3... 1. Type DISKIMG at the DOS prompt, and press the Enter Key. The main menu for the disk image creation utility will be displayed.

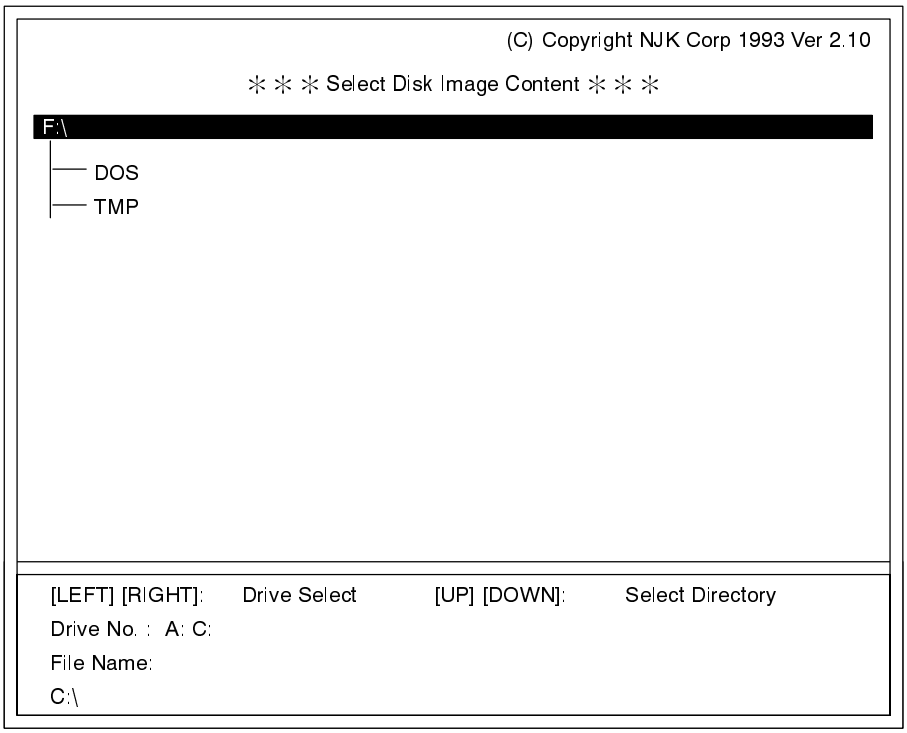


2. Select "Set Environmental Information," and press the Enter Key.



⚠ Caution The disk image size, sector size, FAT ID, and directory count must all use the default settings. The ISA Control Unit will not operate properly if these settings are changed.

3. Use the Up (↑) and Down (↓) Arrow Keys to move the cursor to “Select Files of Disk Image,” and press the Enter Key.



4. Move the cursor and press the Space Key to select the files that are to be saved as disk images. When all the required files have been selected, press the Enter Key. The Set Environmental Information menu will then return.

   The image files will be saved under the name “ROMDSK.BIN.” Move the cursor to “Select Output Directory” and specify the directory in which the images are to be saved.
5. The Set Environmental Information menu will then return.
6. Select “END” to return to the main menu.
7. Select “Create Image File.” The disk image file (ROMDSK.BIN) will be saved to the specified output destination directory. If the save operation is completed successfully, the following message will be displayed:

   ```
   Image Files Copy END
   Please Press Any Key . . .
   ```

8. Press any key to return to the main menu.
9. Select “END” to return to the MS-DOS prompt.

6-4-9 ROM Disk Creation Utility

This utility creates ROM disk image files at the ISA Control Unit, eliminating the need to transfer them from the personal computer.

At least 192K bytes of available space are required at drive F in order to use the ROM disk creation utility. This utility can also be used at ANSI terminals.

1, 2, 3... 1. Use an editor to describe the system-related file saved as a ROM disk image (ROMDSK.BIN). Use ROMINF.DAT as the file name. The ROM disk creation utility will reference this file to create a ROM disk image file.

ROMINF.DAT Example:

F:\CONFIG.SYS

F:\AUTOEXEC.BAT

F:\PCM.INI

2. Type “ROMUTY” at the DOS prompt and press the Enter Key.

F:\ROMUTY

The following menu will be displayed.

```
*****  ROM  DISK  MAKE  UTILITY  (Ver 1.00)  *****
1. Disk Size              192    KByte
2. Sector Size            512    Byte
3. FAT ID                 HFF (Hex)
4. Directory Count        192
5. Select Output Directory
          ROMDSK.BIN
6. Select ROM Image File
          ROMINF.DAT
7. Boot Base Address   F80000 (Hex)
G. Create Image File
E. END

Please Select a Head :
```

⚠ Caution The disk image size, sector size, FAT ID, and directory count must all use the default settings. The ISA Control Unit will not operate properly if these settings are changed.

3. Type “G” and press the Enter Key. The ROM disk image file will be created according to the ROMINF.DAT file created in step 1. When this is completed, the main menu will return.
4. In the main menu, type “E” and press the Enter Key to exit the ROM disk creation utility and return to the DOS prompt.

This ends the file creation operation. Use the FCOPYX command to copy ROMDSK.BIN to drive G.

6-4-10 Transferring Disk Images to the ISA Control Unit and Writing them to Drive G

There are three ways to transfer disk image files to the ISA Control Unit and transfer them to drive G.

- Use the FCOPYX command (at the ISA Control Unit) and the RSCOPY command (at the personal computer). The disk image files are transferred and copied to drive G at the same time.
- Use the FTRANS and FCOPYX commands. The FTRANS command transfers the disk image files to drive F, and then the FCOPYX command copies them to drive G.
- Save the disk images to a memory card and then use the FCOPYX command at the ISA Control Unit to copy them to drive G. For this method the ISA Control Unit must be able to use memory cards.

Note 1. If unable to write the files to drive G, set system switch pin 8 to ON and then restart the Unit using only E:\STARTUP.INI.

2. When disk image files are created at the ISA Control Unit with the ROMUTY program, there is no need to transfer them.

Using FCOPYX and RSCOPY

Connect the ISA Control Unit and the personal computer with RS-232C cable. Use the same kind of cable as is used for connecting the ISA Control Unit and terminals. (Refer to *6-3-6 Installing MS-DOS Console Terminals*.) Then follow the procedure outlined below.

1, 2, 3...

1. Start-up FCOPYX at the ISA Control Unit, and then enter the following line.

```
F:\>FCOPYX /R /COM1 (/COM2)
*** Start Flash Memory Copy Prog. V1.01
Start Erase
End Erase
Flash Memory Copy...
Get Receive data.

End of data receive.
*** End
```

2. Confirm that the message indicating that the Unit is beginning to receive data is displayed, and start-up RSCOPY at the personal computer. Then enter the following command.

```
C:\>RSCOPY /S ROMDSK.BIN
*** RS-232C File Copy Prog. V1.02
Start send data.
**
End of data send.
*** End file Copy Prog.
```

   The disk image file will be transferred to the ISA Control Unit and written to drive G.

3. Restart the ISA Control Unit.

Using FTRANS and FCOPYX

Connect the ISA Control Unit and the personal computer with RS-232C cable. Use the same kind of cable as is used for connecting the ISA Control Unit and terminals. (Refer to *6-3-6 Installing MS-DOS Console Terminals*.) Then follow the procedure outlined below.

1, 2, 3...

1. Use the FTRANS command to transfer the disk image file to drive F at the ISA Control Unit. (For instructions on using the FTRANS command, refer to *6-5 Transferring Files*.)
2. Start up FCOPYX at the ISA Control Unit's MS-DOS prompt, and then enter the following line.

```
F:\>FCOPYX /F=F:ROMDISK.BIN
*** Start Flash Memory Copy Prog. V1.01
Start Erase
End Erase
Flash Memory Copy...
*** End
```

   The disk image file will be written to drive G.

3. Restart the ISA Control Unit.

Copying From Memory Card

This method uses the FCOPYX command to copy files to drive G from a memory card. It will only be possible if the ISA Control Unit can use memory cards.

1, 2, 3...

1. At the personal computer, copy the disk image file (ROMDSK.BIN) to a memory card.
2. Insert the memory card at the ISA Control Unit.

3. Start up FCOPYX at the ISA Control Unit's MS-DOS prompt, and then enter the following line.

```
F:\FCOPYX /F=H:\ROMDISK.BIN
*** Start Flash Memory Copy Prog. V1.01
Start Erase
End Erase
Flash Memory Copy...
*** End
```

   The disk image file will be written to drive G.

4. Restart the ISA Control Unit.

6-5 Transferring Files

This section explains how to transfer files between the ISA Control Unit and the personal computer, using RS-232C. Use this method when not using memory cards.

6-5-1 Introduction

Use the FTRANS.EXE program for transferring files between the ISA Control Unit and the personal computer. This file transfer program uses RS-232C cable for sending and receiving files. It supports both the X-Modem (128-byte sum) and Z-Modem protocols.

6-5-2 Restrictions

The file transfer program has the following restrictions.

The X-Modem protocol is 128-byte sum. If the file size is not in multiples of 128, the remainder is padded by 1Ah.

The Z-Modem protocol cannot handle the following data.

- Audio data
- LZW compressed data
- RLE compressed data
- CRC-32 bits

Resume transfers and ASCII transfers are not supported.

Sequential files are the only valid file type.

6-5-3 Communications Settings

Make the following communications settings for the ISA Control Unit and the personal computer.

Baud rate:	300, 600, 1,200, 2,400, 4,800, 9,600, or 19,200 bps
Parity:	None
Stop bits:	1 bit
Data length:	8 bits

6-5-4 Using the File Transfer Program

Use the file transfer program as described below.

Transmission Syntax

FTRANS Filename /S /Function [/Comport] [/Flow] [/Baud Rate]

/**S:**
- Specifies "send."

/**Function:**
- "/X" specifies X-Modem, 128 sum.
- "/Z" specifies Z-Modem.

/**Comport:** (see note 1)
- "/COM1" specifies COM1 (the default).
- "/COM2" specifies COM2.

/**Flow:** (see note 2)
- "/F" specifies XON/XOFF flow control.
- If "/F" is not set (the default), XON/XOFF flow control is not specified.

/**Baud Rate:**
- /300 Sets baud rate to 300 bps.
- /600 Sets baud rate to 600 bps.
- /1200 Sets baud rate to 1,200 bps.
- /2400 Sets baud rate to 2,400 bps.
- /4800 Sets baud rate to 4,800 bps.
- /9600 Sets baud rate to 9,600 bps (the default).
- /19200 Sets baud rate to 19,200 bps.

Note
1. Communications cannot be executed by FTRANS.EXE if COM1 is specified by the DOS command CTTY. When COM1 is used as a Console, specify it with a switch setting and restart the system.
2. "/F" must be specified if XON/XOFF flow control is used by file transmission software such as terminal software.

Reception Syntax

FTRANS Filename /R /Function [/Comport] [/Flow] [/Baud Rate]

/**R:**
- Specifies "receive."

/**Function:** (see note 1)
- "/X" specifies X-Modem, 128 sum.
- "/Z" specifies Z-Modem.

/**Comport:** (see note 2)
- "/COM1" specifies COM1 (the default).
- "/COM2" specifies COM2.

/**Flow:** (see note 3)
- "/F" specifies XON/XOFF flow control.
- If "/F" is not set (the default), XON/XOFF flow control is not specified.

/**Baud Rate:**
- /300 Sets baud rate to 300 bps.
- /600 Sets baud rate to 600 bps.
- /1200 Sets baud rate to 1,200 bps.
- /2400 Sets baud rate to 2,400 bps.
- /4800 Sets baud rate to 4,800 bps.
- /9600 Sets baud rate to 9,600 bps (the default).
- /19200 Sets baud rate to 19,200 bps.

Note
1. With Z-Modem, the transmission filenames are automatically taken as the names of files received. Specify a directory name (or bus name) for storing files received by Z-Modem. If no directory is specified, the files will be stored in the current directory.
2. Communications cannot be executed by FTRANS.EXE if COM1 is specified by the DOS command CTTY. When COM1 is used as a Console, specify it with a switch setting and restart the system.
3. "/F" must be specified if XON/XOFF flow control is used by file transmission software such as terminal software.
4. To cancel receive waiting, type "C" and press the Enter Key.

6-5-5 Example: Using X-Modem

In this example procedure, a file called “TEST.DAT” is transferred using X-Modem.

1, 2, 3... 1. Enter the following syntax to start up the program for reception.

```
F:\>FTRANS TEST.DAT /R /F /X

FTRANS: File translate utility. Version 1.0
Copyright (c) 1995, 1996 by NJK. All right reserved.
File (TEST.DAT) Receives---

Normal End.
```

2. Enter the following syntax to start up the program for transmission.

```
C:\>FTRANS TEST.DAT /S /F /X

FTRANS : File translate utility. Version 1.0
Copyright (c) 1995,1996 by NJK. All right reserved.
File (TEST.DAT) Send_

Normal End.
```

With both reception and transmission, when “Normal End” is displayed it means that the operation was successful.

Note If the same filename already exists at the receiving end, a message will be displayed asking whether or not to overwrite the existing file. To overwrite the file, enter “Y.” To change the filename of the file being received, enter “N” and restart the reception program.

```
Same File Name! Overwrite ? (Y/N)
```

6-5-6 Example: Using Z-Modem

In this example procedure, a file called “TEST.DAT” is transferred using Z-Modem.

1, 2, 3... 1. Enter the following syntax to start up the program for reception.

```
C:\>FTRANS  F:\DATA1  /R /F /Z

FTRANS : File translate utility.  Version 1.0
Copyright (c) 1995,1996 by UJK. All right reserved.
File (TEST.DAT) Receives---

Nomal End.
```

2. Enter the following syntax to start up the program for transmission.

```
F:\>FTRANS TEST.DAT  /S /F /Z

FTRANS : File translate utility. Version 1.0
Copyright (c) 1995, 1996 by UJK. All right reserved.
File ( TEST.DAT ) Send_

Normal End.
```

In this example, the TEST.DAT file is stored in the DATA1 directory.

With both reception and transmission, when “Normal End” is displayed it means that the operation was successful.

6-5-7 Messages

The following table shows the meanings of the messages that may be displayed.

Message	Meaning
Normal End.	Operation was successfully completed.
Parameter Error	Parameter error occurred.
No File Open	File does not exist or cannot be opened.
Same File Name! Overwrite? (Y/N)	Same filename already exists. Overwrite the file? Enter Y (Yes) or N (No).
Nothing Reply, Abort.	No response, so abort operation.
RS-232C Init Error.	RS-232C initialization error occurred.
File Read Error.	File read error occurred.
File Write Error.	File write error occurred.
Line Error.	Communications error occurred.
Line Time Out.	Communications time-out
Check Sum Error.	Checksum error occurred.
Not File Open.	File cannot open.
Block No. Error.	Block number error occurred.
Data Error	Data error occurred.
Line Canceled	Communications were cancelled.

6-6 Using Hard Disk Units

This section explains how to make the settings for using a CV500-HDD11 Hard Disk Unit with the ISA Control Unit, and how to format the hard disk.

The following operations are required in order to use the Hard Disk Unit.

1, 2, 3...

1. Set the hardware configuration (SETUP.EXE).
2. Set the hard disk (HDISK.EXE).
3. Format the hard disk (FORMAT.COM).

In the example procedure provided here, a new Hard Disk Unit is connected to an ISA Control Unit.

Note

1. When using a Hard Disk Unit, set system switch pin 7 to ON at the back of the Unit.
2. Connect the Hard Disk Unit properly to the ISA Sub-backplane before setting the hardware configuration.
3. If the system is started up without connecting the Hard Disk Unit once the hardware configuration has been set for a Hard Disk Unit, the system's hardware check will take approximately five minutes. This is not a system error.

6-6-1 Setting Up the Hardware Configuration (SETUP.EXE)

Use the SETUP utility to set up the ISA Control Unit hardware configuration.

1, 2, 3...

1. Type "SETUP" at the MS-DOS prompt and press the Enter Key. The SETUP utility will be started up, and a screen will be displayed for selecting either FDD or HDD setup.

```
E:\>SETUP

FDD & HDD SETUP Ver 1.00
1 ..... FDD SETUP
2 ..... HDD SETUP
Input number :__
```

2. Type “2” and press the Enter Key. A screen will be displayed for selecting the drive.

```
Input number : 2

Select driver C: or D: ( Input C or D ) ?_
```

3. Type “C” and press the Enter Key. Enter “C” for the first Hard Disk Unit, and “D” for the second. A screen will be displayed for selecting the hard disk type.

```
Select drive C: or D: ( Input C or D ) ? C
Select Drive is 'C'

Select hard disk type  ( 0: not install. 48. 49 )
Input hard disk type : __
```

4. Type “48” and press the Enter Key. If “0” is entered, the hard disk will not be installed. A screen will be displayed for entering the number of cylinders, the number of heads, and the number of sectors. Enter them as shown below.

```
Input hard disk type : 48

New cylinder input  ( 0 – 65535 ) : 988
New head input ( 0 – 255 ) : 10
New sector input ( 0 – 255 ) : 17
```

   The hardware configuration is now set, and the MS-DOS prompt will return.

5. Restart the system to activate the new hardware configuration settings.

6-6-2 Setting Up the Hard Disk (HDISK.EXE)

Use the HDISK utility to set up the Hard Disk Unit’s hard disk.

The HDISK utility sets the entire hard disk to one partition. It is not possible to set multiple partitions.

1, 2, 3...

1. Type “HDISK” at the MS-DOS prompt, and press the Enter Key. The HDISK utility will start up, and a screen will be displayed for selecting the drive.

```
E:\>HDISK

HDISK ( HDD TYPE SET ) Ver 1.00
input drive  1=C:  2=D:
Input disk number  :_
```

2. Type “1” to select drive C (or “2” to select drive D) and press the Enter Key. The hard disk will be set automatically according to the hardware configuration settings.

```
input disk number : 1
Set HDD type
Cylinders are 988 ( 3DCh )
Heads are 10  ( Ah )
Sectors are 17 ( 11h )

Start cylinder is 0 ( 0h )
Start head is  1 ( 1h )
Start sector is  1 (1h )

End cylinder is 219 ( DBh )
End head is 9 ( 9h )
End sector  is 209 ( D1h )

Start logical sector is 17 ( 11h )
Logical sector length is 167943 ( 29007h )
```

   The hard disk is now set, and the MS-DOS prompt will return.

3. Restart the system to activate the new hard disk setting.

6-6-3 Formatting the Hard Disk (FORMAT.COM)

Use the FORMAT command to format the hard disk.

1, 2, 3... 1. Type “FORMAT C:” at the MS-DOS prompt, and press the Enter Key. The following warning message will be displayed.

```
E:\>FORMAT  C:

WARNING, ALL DATA ON NON-REMOVABLE DISK
DRIVE C: WILL BE LOST !
Proceed with Format ( Y/N ) ? Y
```

2. Type “Y” and press the Enter Key. The formatting will begin. When the formatting has been successfully completed, the following message will be displayed.

```
Format complete.

Volume label ( 11 characters, ENTER for none ) ?__
```

3. Enter the volume label and press the Enter Key. If the volume label is not required, just press the Enter Key. The following message will be displayed.

```
85800960 bytes total disk space
85800960 bytes available on disk

   2048 bytes in each allocation unit.
  41895 allocation units available on disk.

Volume Serial number is 1E15-15ED
```

The hard disk is now formatted, and the MS-DOS prompt will return. The Hard Disk Unit can be used as drive C.

Carry out the following operation to confirm that the hard disk is usable.

```
E:\>C:

C:\>DIR

 Volume in drive C is XXXXXX          ←The volume label that is set is displayed here.
 Volume Serial number is 1E15-15ED
  Directory of  C:\

File not found
```

6-6-4 Setting Up the Floppy Disk Drive

This section explains how to set up the floppy disk drive connections. In order to connect a floppy disk drive, a commercially available I/O port that can use a floppy disk drive is required. Install the I/O port in the ISA Bus Expander and connect the floppy disk drive and the ISA Control Unit. Then carry out the following procedure to make the settings.

In this example procedure, a 1.44M-byte-format floppy disk drive is connected to the ISA Control Unit

1, 2, 3... 1. Type “SETUP” at the MS-DOS prompt and press the Enter Key. The SETUP utility will be started up, and a screen will be displayed for selecting either FDD or HDD setup.

```
E:\>SETUP

FDD & HDD SETUP Ver 1.00
1 ..... FDD SETUP
2 ..... HDD SETUP
INPUT NUMBER :__
```

2. Type “1” and press the Enter Key. A screen will be displayed showing the floppy disk configuration that is currently set, along with the options for the drive type.

```
Input number : 1

Current FDD type
Disk-A : No drive
Disk-B : No drive

0 ..... No drive
1 ..... 720KB
2 ..... 1.44MB
Input drive new type ( Return only is no change )
A : ?__
```

3. Type “2” and press the Enter Key to select 1.44M bytes. If “0” is selected, no floppy disk drive will be connected.

```
A : ? 2
B : ?__
```

4. Enter “0” and press the Enter Key. The hardware configuration is now set and the MS-DOS prompt will return.
5. Restart the system to activate the new hardware configuration settings.

SECTION 7
Communications and Control Functions

This section introduces the communications and control functions that can be used with the ISA Control Unit, and provides instructions on how to use each of the functions. Use this section to determine the functions needed for your system configuration.

7-1 Features

The ISA Control Unit's communications/control functions have the following features.

High-speed Communication

The ISA Control Unit communicates with the PC through the CPU bus, providing high-speed communications compared to an RS-232C or other interfaces. Almost 8K-bytes of data can be transferred at one time depending on the type of communications servicing being used.

Communications with other Hierarchies

The ISA Control Unit can communicate through the network up to three hierarchies.

Communications with all CPU Bus Units

The ISA Control Unit can communicate easily with other ISA Control Units, BASIC Units, and any other CPU Bus Unit within a three-hierarchy network.

7-2 Communications/Control Services

The following table shows the three communications/control services that can be used through the CPU bus interface.

No.	Service	Explanation
1	Event service	Communication possible with Programmable Controller and other Units in the local node, as well as with other Programmable Controllers and Units through the network.
2	Cyclic service	Communication possible with the Programmable Controller in the local node.
3	CPU Bus Link service	Communication possible with Programmable Controller and other CPU Bus Units in the local node.

These three services are described briefly below. Refer to *5-4 Communications/Control Service Details* for more details.

Event Service

Event service communication is accomplished with OMRON's FINS command/response protocol, and can be used to control other Units and obtain data from other Units.FINS is the communications protocol for the OMRON CVM1/CV Series. Refer to the *FINS Command Reference Manual (W227)* for details.

Cyclic Service

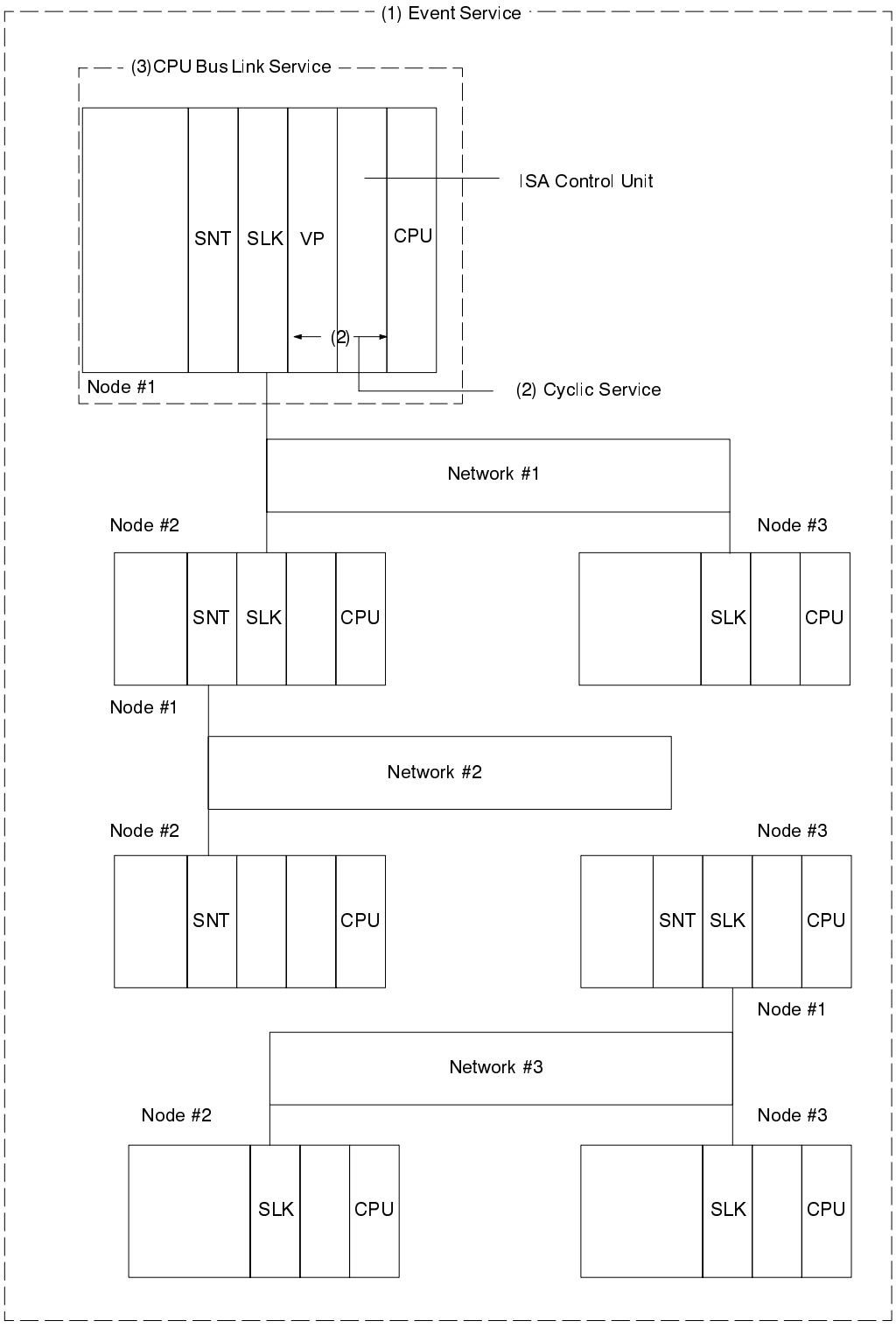
Use the cyclic service for communications (similar to a data link) between the ISA Control Unit and the local-node PC.

CPU Bus Link Service

This service periodically reads data from other CPU Bus Link Units in the local node; it is useful for monitoring the most recent data or operating status of the CPU Bus Link Units in the local node. The CPU Bus Driver must be used.

Effective Service Range

The effective ranges of the three types of communications and control service are as follows:

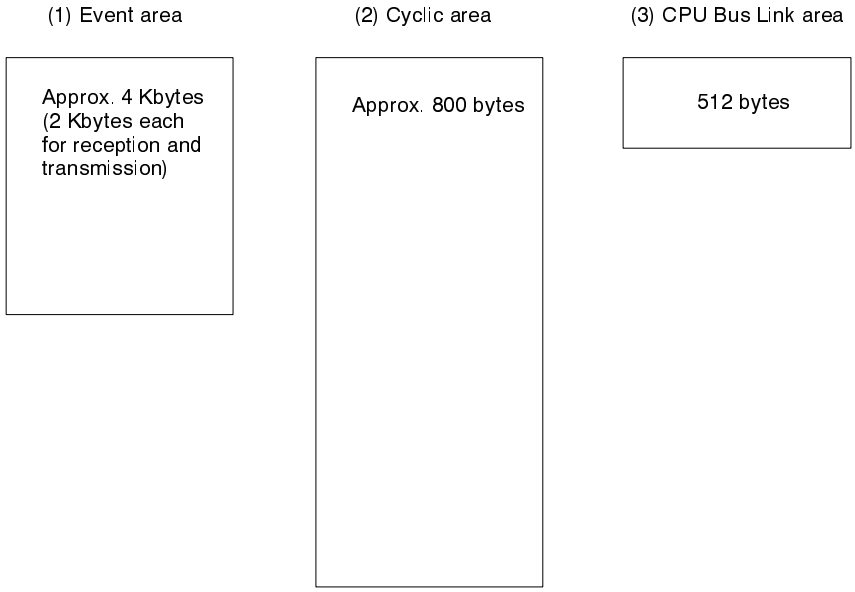


Each of these three types of service uses a dedicated data area within the ISA Control Unit. Therefore the amount of data that can be processed at one time for any of the services depends on the size of its respective data area. . For more detail, refer to *5-4 Communications/Control Service Details*.

The cyclic area is the largest, followed in order by the event area and the CPU bus link area. (Refer to the illustration below.) For example, when exchanging

large amounts of data with a Programmable Controller in the same node, efficient execution can be achieved by using cyclic service. Likewise, when exchanging data with a CPU Bus Link Unit in the same node, a greater volume of data can be handled by the event service than by the CPU Bus Link service.

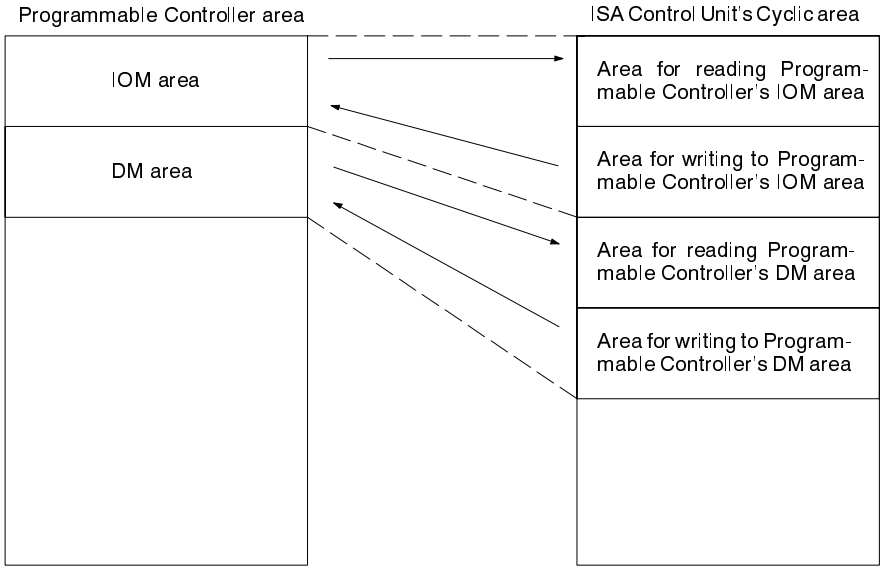
The following diagram shows the relative sizes of each service-dedicated data area of the ISA Control Unit.



7-3 Communications/Control Service Details

7-3-1 Cyclic Service

Cyclic service provides a data link-type connection between the ISA Control Unit and the local-node Programmable Controller. By means of this service, data from a Programmable Controller in the same node can be handled as if it were data being processed in the ISA Control Unit. Therefore this service is effective for monitoring and controlling data in the Programmable Controller (such as the contents of data areas).



This service can either read Programmable Controller data (in word units) into the cyclic area of the ISA Control Unit, or write data from the cyclic area of the ISA

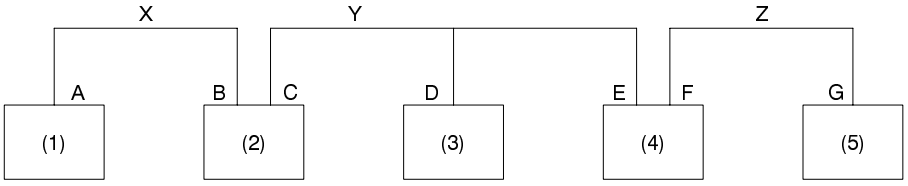
Control Unit into the Programmable Controller. Therefore, when using this service, it is necessary to specify the data area address of the Programmable Controller involved in the data transfer, as well as the direction of transfer. (The direction will be either reading or writing, as seen from the ISA Control Unit.)

Regardless of the direction of data transfer (i.e., Programmable Controller to ISA Control Unit or ISA Control Unit to Programmable Controller), the maximum number of data transfer locations that can be specified at one time with this service is six, and the maximum total length of data that can be specified for transfer is 403 words.

7-3-2 Event Service

Event service can execute communications with Programmable Controllers and other devices both within and outside of the node. It can control those devices and obtain information from them.

By means of setting the routing table, communications can be conducted through the network, and it is possible to control and exchange data with devices in other networks (up to a maximum of three hierarchical levels).



X to Z: Network address
A to G: Network node address

Given that the ISA Control Unit's node is (1), communications will be possible with devices in node (5) via networks X, Y, and Z.
(Communications will also be possible with devices in nodes (1) through (4).

In addition, even if the routing table is not set, communications can still be conducted within the same node. It is therefore possible, within the node, to reference present status such as connection information and to access the Programmable Controller's variable areas.

In general, when event service is used, devices are controlled and information is referenced in the following way. The event service executes a command with a request with respect to the device that it wants to control or reference information from. The response to the request (e.g., the referenced information, or whether the control was properly executed) is then received from the device.

In other words, the event service communications procedure consists of the following steps:

1, 2, 3...

1. Service request command is transmitted to a device that offers a service.
2. Response to the request is received from the device that offers the service.

This communications procedure is the same for other devices as well, so service request commands can also be transmitted from other devices to the ISA Control Unit. In such cases, the ISA Control Unit will transmit a response to the request to the device that sent the service request command.

Note Refer to *Appendix G PC Memory Configuration* for details on the PC data areas that can be specified by the user.

Service Request Command/Response

These conform to Programmable Controller FINS commands, and the contents of a command will vary according to the contents of the service requested. In addition, the contents of the response will vary according to the command. Refer to the *FINS Command Reference Manual (W227)* for details.

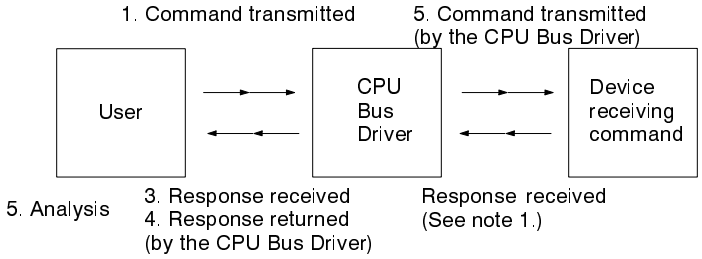
The system, however, will return responses with respect to the commands listed below. Thus there is no need to create responses for these commands.

Commands Eliciting System Response

No.	Command
1	Read Controller Information
2	Read Time Information
3	Write Time Information
4	Loopback Test
5	Read Error Log
6	Clear Error Log

We will now consider this communications processing procedure in more concrete terms.

Command from ISA Control Unit



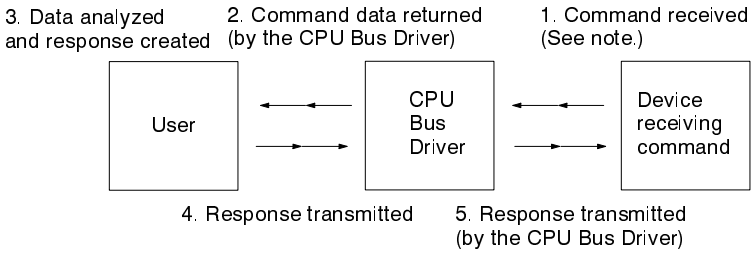
1, 2, 3...

1. The user creates a request command corresponding to the service requested, specifies the device offering the service, and requests the CPU Bus Driver to transmit the command.
   At this point, control is returned to the program.
2. The CPU Bus Driver receives the "transmit command" request from the user, and transmits the specified request command to the specified device.
3. In order to receive the response to the command that was transmitted, the user requests "receive response" of the CPU Bus Driver.
4. The CPU Bus Driver returns to the user the response data received from the device to which the command had been transmitted.
5. The user analyzes the contents of the response that is received.

Note

1. The response from the destination device is received in the CPU Bus Driver's reception buffer whether or not the user receives the response.
2. Steps 1, 3, and 5 are performed by the program.

Command from Another Device



1, 2, 3...

1. In order to receive the request command from the other device, the user requests "receive command" of the CPU Bus Driver.
2. In response to the "receive command" requested by the user, the CPU Bus Driver returns to the user the command data it has received from the other device.
3. The user analyzes the command data that is received, and creates corresponding response data.

4. Along with creating the response data, the user specifies the device which is the source of the request command, and requests "transmit response" of the CPU Bus Driver.
5. The CPU Bus Driver receives the "transmit response" from the user, and transmits the specified response data to the specified device.

Note The command reception from the device is executed regardless of the user request. This service request command/response can handle a maximum of 2,048 bytes of data. The system uses 36 bytes, so the maximum amount of data that can be utilized by the user is 2,012 bytes. The beginning of the 2,012 bytes is used for ICF data. Refer to *8-1 The FINS Format* for details.

Event service will be processed as outlined above, and the object of communications can be any device connected through the networks. In other words, the device that is to be the object of communications is determined by the user of the service.

The question, then, will be how to specify the location of that device. This is done by means of the three types of addresses explained below. (The address of the ISA Control Unit is indicated in the same way.)

Network Address

The network address is the address of the network to which the device belongs. (Network address $00 indicates the same network.)

`$00`	Same network address

Node Address

Within a given network, each device also has a node address. The following codes have special meanings.

`$00`	Same node address
`$FF`	Broadcast to all nodes on specified network (See note 2.)

Unit Address

Within a given node, each device has a unit address, specified by the absolute address.

Example: CPU Bus Unit #0 will have an address of $10.

The following codes have special meanings.

`$00`	Unit address of Programmable Controller
`$10 to $2F`	CPU Bus Units
`$FD`	Peripheral Tools (e.g., FIT)
`$FE`	Communications Units (See note 2.)

Note

1. The actual way in which these addresses are specified will vary according to the drivers and libraries that are used. Please refer to the sections covering drivers and libraries.
   (e.g., SYSMAC NET, SYSMAC LINK)
2. These codes can be used to specify a destination device, but can't be used to specify the same device, i.e., the device can't use these codes to specify itself.

Reception Processing

Reception of commands and responses is executed automatically by the CPU Bus Driver.

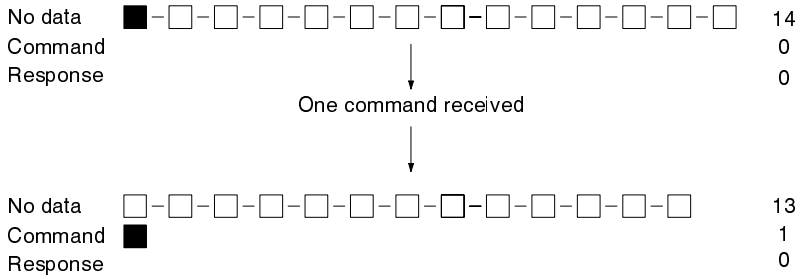
Number of Reception Buffers

The CPU Bus Driver has 14 internal reception buffers. (Fourteen is the default, but the number of reception buffers can be specified in the device driver options. Refer to *Section 5-5-2 Installing the CPU Bus Driver* for details.)

Among these reception buffers, there is no distinction in terms of whether they are used for commands or responses. Data is received and stored in one of the

buffers. For example, if the data received is a command (as shown in the illustration below), the buffer will be used as a command reception buffer.

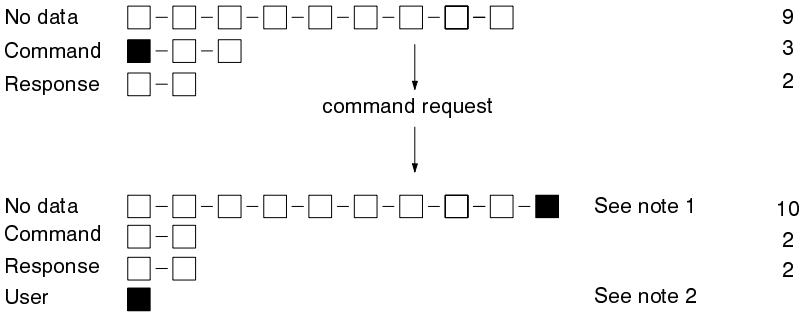
The same will apply if the data received happens to be a response. Therefore, a total of up to 14 commands and responses can be saved internally. If a command or response is received after that point, it cannot be saved and an error will occur. The CPU Bus ERR indicator will light, and the error will be logged.



Managing Reception Buffers

The CPU Bus Driver classifies received data into commands and responses, and saves the data internally until a request is executed for either “receive command” or “receive response.” Thus, for example, when a “receive command” request is executed (as shown in the illustration below), the leading data stored internally as command data will be returned to the user.

When even a portion of the received data is read, that data will be discarded. Then when the next request is executed, the next data will be returned. Therefore, when it is preferable that the received data not be lost, it is recommended that the maximum length reception buffer be secured, and that the maximum number of bytes be requested.

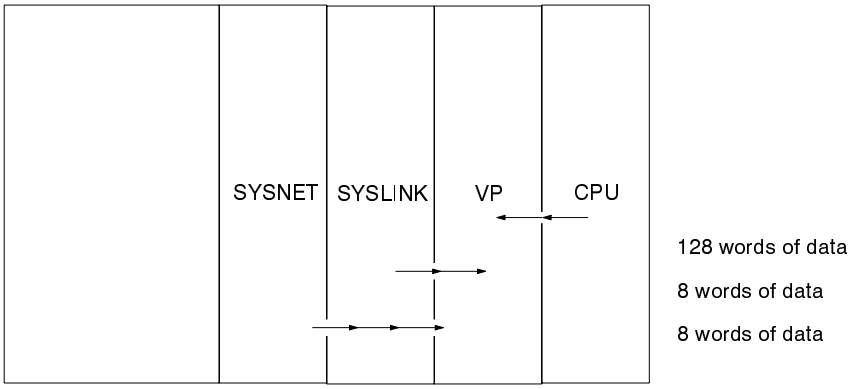


Note

1. After data is copied to the user’s specified area, that buffer will be made a “no data” buffer.
2. In response to the command reception request, the leading data of the command will be carried over.

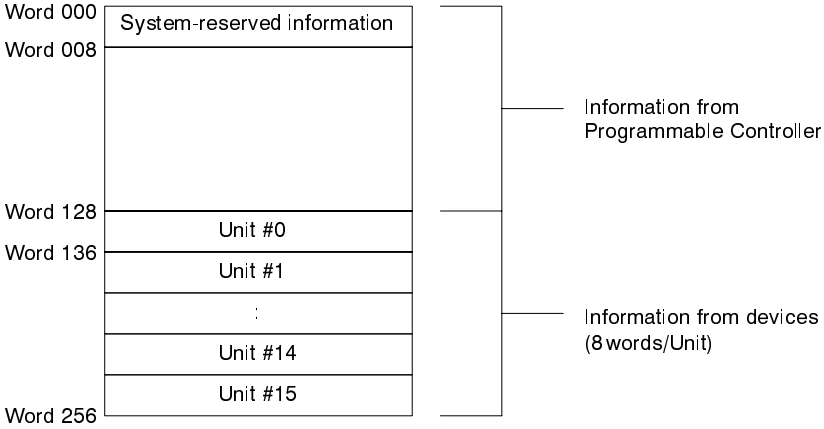
7-3-3 CPU Bus Link Service

CPU bus link service periodically reads data from every device within the same node, so it is effective for regular monitoring of new data and operating status of all the devices in the node. The data read by this service is data from the Programmable Controller and from all the devices within the same node. Altogether, 256 words of data (in word units) can be read.



Within this, a maximum of 128 words (including information reserved for the system) from the Programmable Controller and eight words each for Unit numbers 0 through 15 can be read.

CPU Bus Link Area Contents:



Therefore, when transmitting ISA Control Unit information to another device, a maximum of eight words can be written into the CPU bus link area and information can be provided.

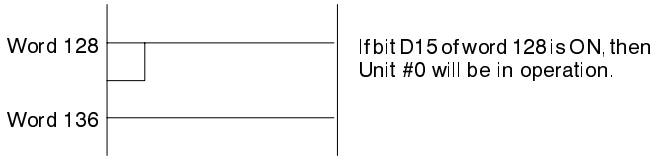
Data can be written only to the portion of the CPU Bus Link Area assigned to the ISA Control Unit's unit number. Also, the CPU Bus Link must be enabled in the PC Setup (letter I) in order to use this service.

The information obtained from the Programmable Controller (excluding that portion reserved for system use) or from other devices by means of this service depends on the contents of the CPU and devices. Therefore, when using this service to exchange information with other devices, the data contents must be settled in advance.

Regardless of the content of the information from a device, it can be recognized based on that information whether or not that device is currently participating in the CPU bus link service. The leading bit of the data area from each device can be checked, and if the bit is ON it indicates that the device is participating in the CPU bus link service.

The driver will automatically control the status of this flag when the user writes data to the CPU bus link area; the user shouldn't change the flag's status.

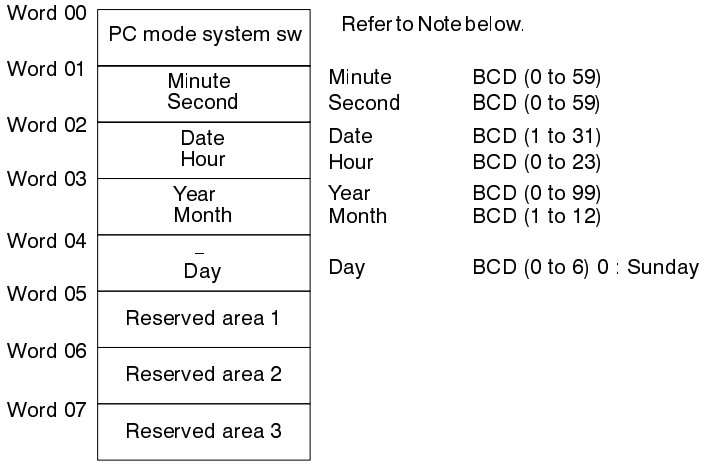
Example: Unit #0



Information reserved for the system is also received from the Programmable Controller. This information is configured as eight words of data. By reading this data, the user can obtain the status, time, etc., of the Programmable Controller at the time the data is read.

Note Even when the CPU bus link service is stopped, it is still possible to read the system reserved information.

The information reserved for system use is as shown below.



Note Programmable Controller Mode and System Switch Contents:

15	14	13	12	11	10 to 8	7	6	5	4	3	2	1	0
15	–	–	12	11	to	7	6	–	4	3	2	1	0

The meaning when each bit is turned ON is as follows:

15: KEY-SW system protected
12: Memory card write-protected
11: UM read-protected
07: Fatal error occurred
06: Non-fatal error occurred
04: PC is operating
03: RUN mode
02: MONITOR mode
01: DEBUG mode
00: PROGRAM mode

SECTION 8
CPU Bus Driver

The CPU Bus Driver provides efficient communications through the CPU bus. This section introduces the CPU Bus Driver and describes the FINS commands used in the CPU Bus Driver.

8-1 The FINS Format

This section explains the FINS command/response format used for event servicing when the CPU Bus Driver is used. Event service can communicate with other devices on a network. To communicate with a particular device, it is necessary to specify that device's network address, node address, and Unit number address.

In addition, when receiving a command from another device, it will not be possible to return a response unless it is known where the command was transmitted from. A FINS (Factory Interface Network System) format is thus used with this driver to specify the transmission source and destination. Refer to the *FINS Commands Reference Manual (W227)* for more details.

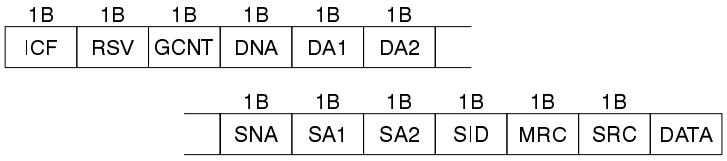
This format incorporates the information essential for communications in front of the command or response data that is to be actually transmitted. Based on this information, the transmission destination can be determined and the transmission source can be confirmed.

Note When using event service with this driver, edit the data in FINS format and leave the processing to the driver. The maximum length of data that can be processed by event service is the total length of the FINS format.

From this point on, the explanation of the FINS format, and particularly the designation of the transmission destination, will be divided into command format (FINS command) and response format (FINS response).

8-1-1 FINS Commands

The FINS commands have the data format shown below, and the transmission destination and data to be transmitted are specified according to this format. Command transmission and reception are always executed in this format.



ICF Information Control Field:

Specifies contents for controlling FINS command. This field indicates whether a response should be sent from the destination device. A "0" requests a response; a "1" indicates a response is not required.

RSV Reserve:

Reserved area. As a rule, the contents will be $00.
This area is sometimes used in communications with SYSMAC BUS/2 and BASIC Units.

GCNT Gateway Count (Bridge):

The gateway cannot be passed through more than this number of times.

Note When a command is transmitted from the Personal Computer Unit, this field is set by the driver and cannot be set by the user.

DNA Destination Network Address:

This specifies the network address of the transmission destination. This address is the final target location address. (A setting of $00 indicates that the destination network is the same as the source network.)

$00: Same network address

DA1 Destination Node Address:

This specifies the node address of the transmission destination. This address is the final target location address. The following codes have special meanings.

$00: Same node address
$FF: Broadcast to all nodes on specified network

DA2 Destination Unit Address:

The unit address for the transmitted data is specified by the absolute address. (Add $10 to the unit number to calculate the absolute address.)

Example: Special I/O Unit #0 will have an address of $10.

This address is the final target location address. The following codes have special meanings.

$00:	Unit address of PC
$10 to $2F:	CPU Bus Units
$FD:	Peripheral Tools (e.g., FIT)
$FE:	Communications Units (e.g., SYSMAC NET, SYSMAC LINK)

SNA Transmission Source Network Address:

This specifies the network address of the source of the transmitted data.

Note When a command is transmitted from the Personal Computer Unit, this field is set by the driver and cannot be set by the user.

SA1 Transmission Source Node Address:

This specifies the node address of the source of the transmitted data.

Note When a command is transmitted from the Personal Computer Unit, this field is set by the driver and cannot be set by the user.

SA2 Transmission Source Unit Address:

This specifies the unit address of the source of the transmitted data.

Note When a command is transmitted from the Personal Computer Unit, this field is set by the driver and cannot be set by the user.

SID Service ID:

The same service ID number is used in a command and the response to that command. Use the service ID to distinguish between several commands and responses.

MRC Main Request Class:

This classifies the service. Refer to the *FINS Commands Reference Manual (W227)* for details.

SRC Sub-request Class:

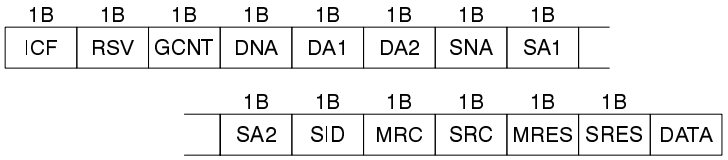
This specifies the service details. Refer to the *FINS Commands Reference Manual (W227)* for details.

DATA Data:

This is the data determined by MRC and SRC. Refer to the *FINS Commands Reference Manual (W227)* for details.

8-1-2 FINS Response

When a FINS command is issued to another device, a corresponding response can be returned from that device. In addition, when a command requiring a response is transmitted to the Personal Computer Unit from another device, a response must be returned to that device from the Personal Computer Unit. The data format for the response is as shown below.



ICF

Information Control Field:

Specifies contents for controlling FINS command.

When a command is transmitted from the Personal Computer Unit, this field is set by the driver and cannot be set by the user.

RSV

Reserve:

Reserved area

Note When a response is transmitted from the Personal Computer Unit, specify the same contents as for the RSV of the FINS command that was received.

GCNT

Number of Times Through Gateway (Bridge):

The gateway cannot be passed through more than this number of times.

Note When a command is transmitted from the Personal Computer Unit, this field is set to 2 by the driver and cannot be set by the user.

DNA to SA2

Just as for the FINS command described above, this specifies the transmission destination and source addresses.

Note When a response is transmitted from the Personal Computer Unit, replace the following items of the FINS command that was received and specify the response data.

DNA ↔ SNA: Network address
DA1 ↔ SA1: Node address
DA2 ↔ SA2: Unit address

SID

Service ID:

This specifies an identifier for recognizing request source processes.

Note When a response is transmitted from the Personal Computer Unit, specify the same contents as for the SID of the FINS command that was received.

MRC

Main Request Class:

This specifies the service classification.

Note When a response is transmitted from the Personal Computer Unit, specify the same contents as for the MRC of the FINS command that was received.

SRC

Sub-request Class:

This specifies the service details.

Note When a response is transmitted from the Personal Computer Unit, specify the same contents as for the SRC of the FINS command that was received.

MRES

Main Response Code:

This code indicates the response (i.e., normal, error, error details) to a request. Refer to the *FINS Commands Reference Manual (W227)* for details. When returning a response from the Personal Computer Unit, enter the result of the command execution.

SRES Sub-response Code:

This code indicates detailed information that cannot be adequately expressed by the main response code. Refer to the *FINS Commands Reference Manual (W227)* for details. When returning a response from the Personal Computer Unit, enter the result of the command execution.

DATA Data:

This is the part of the service indicating the results. Refer to the *FINS Commands Reference Manual (W227)* for details.

8-1-3 FINS Command/Response Example

Example: Reading PC Time Information in the Same Node

The following FINS command is transmitted to a PC address within the same node.

ICF	RSV	GCNT	DNA	DA1	DA2
$00	$00	$00	$00	$00	$00

SNA	SA1	SA2	SID	MRC	SRC
$00	$00	$00	$01	$07	$01

The following response is returned to the Personal Computer Unit from the PC in the same node. The PC time information can be recognized by analysis of these codes. In this case, the Personal Computer Unit's unit number is 0.

ICF	RSV	GCNT	DNA	DA1	DA2	SNA	SA1
$41	$00	$02	$00	$00	$10	$00	$00

SA2	SID	MRC	SRC	MRES	SRES
$00	$01	$07	$01	$00	$00

Year	Month	Date	Hour	Minute	Second	Day
$94	$10	$01	$12	$34	$00	$03

PC time information:
1994, October 1, Wednesday, 12:34:00

The DA2 value of $10 indicates that the response is addressed to the Personal Computer Unit, and the SID value of $01 indicates that the response corresponds to the command shown above. The MRES and SRES values of $00 are the normal response codes. Refer to the *FINS Command Reference Manual (W227)* for more details.

8-2 Using the CPU Bus Driver

There are two methods, described below, for using the CPU Bus Driver.

Opening and Accessing the Driver Through MS-DOS

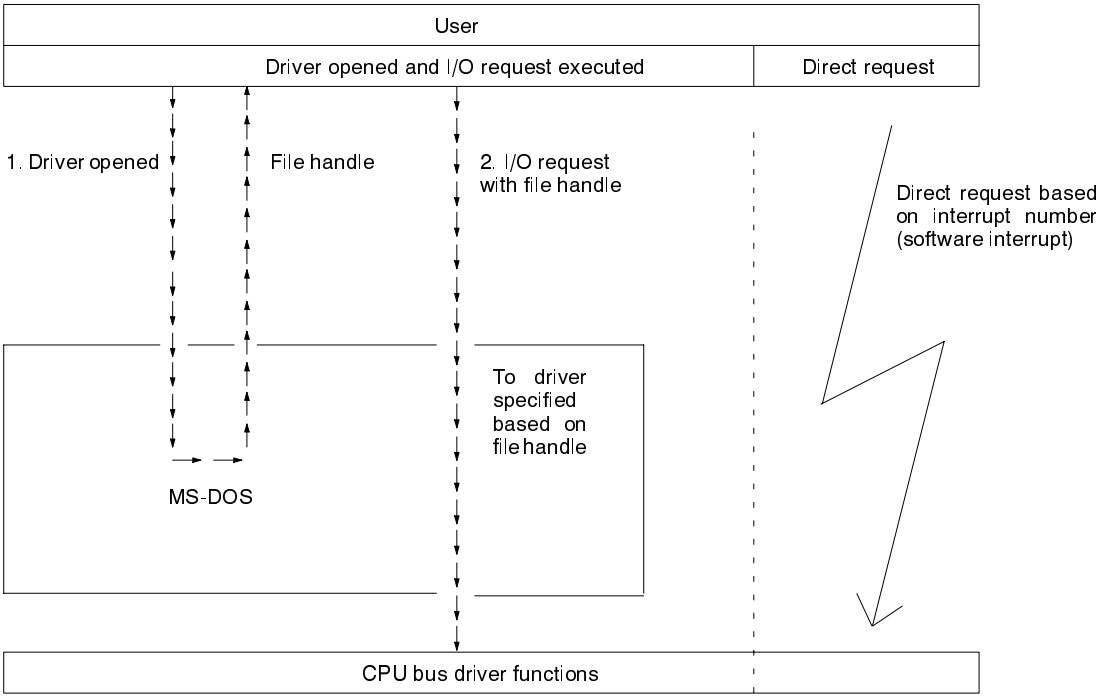
First of all, the driver is opened through MS-DOS, and the "file handle" file identifier is obtained.

When the driver services are employed, I/O requests are executed based on this file handle.

Direct Request Without Going Through MS-DOS

With this method, driver services are used without going through MS-DOS, but rather by executing system interrupts directly.

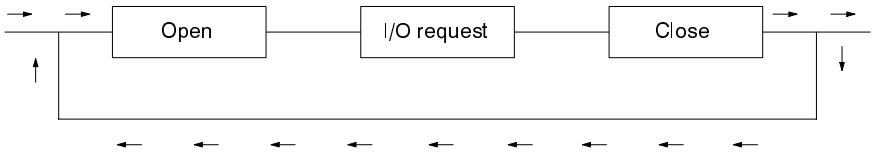
Differences Between I/O Requests by Opening Driver and Direct Requests



Additional Information — Opening and Closing

In our own lives, let us consider how we utilize a book. First of all, from among a large number of books, we must get the book that contains the material that we want to utilize. Once we have the book, we look it over and extract that portion of the contents that we need. If the material is not needed, we return the book to its original spot and the contents of the book then cannot be utilized.

The situation is much the same with a driver. If we want to utilize a driver through MS-DOS, first we must get (or, in other words, open) the driver. Then, by means of a file identifier that has been obtained, the driver is specified and utilized. When the operation is finished, the driver is closed. Once the driver has been closed, it can no longer be used and must be reopened in order to be used again.



These respective methods are described below.

8-2-1 Access by Opening a Driver

Opening the Driver

Finds the file handle.

Call Procedure:

AH = 3DH
AL = 02H (File access mode)
DS:DX = Leading address of path name (“CVIF”)
INT 21H

Return:

When Carry is Set (Error)
AX = 02H File does not exist.
AX = 03H Path name is invalid.
AX = 04H There are too many files open.
AX = 05H Access was denied.
AX = 0CH Access code is invalid.

When Carry is Not Set
AX = File handle (normal termination)

Closing the Driver

Closes the driver.

Call Procedure:

AH = 3EH
BX = File handle
INT 21H

Return:

When Carry is Set (Error)
AX = 06H File handle is invalid.

When Carry is Not Set
AX = 00H Normal termination

Requesting I/O

Prepares the prescribed I/O parameters, and makes the request using the file handle.

Call Procedure:

AH = 44H
AL = 03H
BX = File handle
CX = Number of bytes in data buffer
DS:DX = Data buffer segment: offset
(Refer to *8-4 CPU Bus Driver Operations* for details on the data buffer.)
INT 21H

Return:

When Carry is Set (Error)
AX = 01H Function is invalid.
AX = 05H Access was denied.
AX = 06H File handle is invalid.
AX = 0DH Data is invalid (data buffer error).

When Carry is Not Set
AX = 00H Normal termination
AX = FFH Improper command
AX = Other Error status (Refer to *8-4 CPU Bus Driver Operations* for details.)

8-2-2 Direct Request

Prepares the prescribed I/O parameters, and makes the request using the interrupt table.

Call Procedure:

AH = xxH Operation number
(Refer to *8-3 CPU Bus Driver Operations List* for details.)
CX = Number of bytes in data buffer
DS:DX = Data buffer segment: offset
(Refer to *8-4 CPU Bus Driver Operations* for details on the data buffer.)
INT xxH Empty interrupt table number (60H to 65H)
(The table to be used is specified by CONFIG.SYS.)

Return:

When Carry is Set (Error)
AX = 01H Function is invalid.
AX = 05H Access was denied.
AX = 0DH Data is invalid (data buffer error).

When Carry is Not Set
AX = 00H Normal termination
AX = Other Error status (Refer to *8-4 CPU Bus Driver Operations* for details.)

8-2-3 Limitations

Do not mask the interrupts (IRQ10 and IRQ11) when the event service's functions are used.

8-3 CPU Bus Driver Operations List

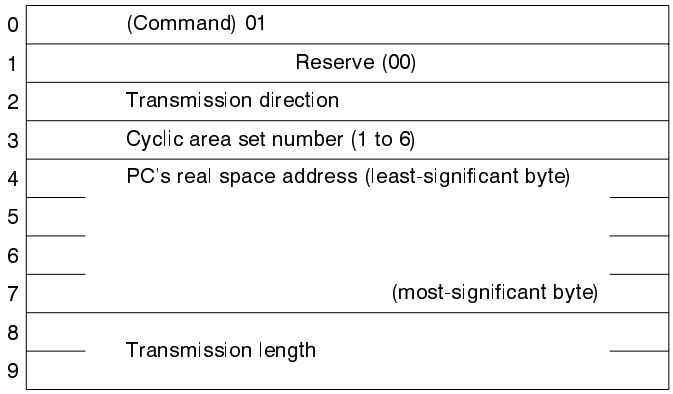
Number	Service	Summary	Page
01H	Cyclic service	Specifies the area to be read or written.	108
02H		Reads specified area.	109
03H		Writes to specified area.	110
04H	Event service	Transmits FINS command.	111
05H		Transmits FINS response.	112
06H		Transmits request to receive FINS command.	113
07H		Transmits request to receive FINS response.	114
08H		Saves branch entry address when transmission of FINS command is complete.	115
09H		Saves branch entry address when transmission of FINS response is complete.	115
0AH		Saves branch entry address when reception of FINS command is complete.	116
0BH		Saves branch entry address when reception of FINS response is complete.	116
0CH		Sets reception timeout value.	117
0DH		Flushes reception buffer.	117
0EH	CPU Bus Link service	Reads information reserved for system.	117
0FH		Reads link data.	118
10H		Writes link data.	119
11H	Other services	Branches to specified entry address after time specified by user has elapsed.	120
12H		Resets Personal Computer Unit.	120
13H		Inquires regarding address of Unit itself.	121
14H		Inquires regarding reception status of FINS commands and responses.	121

8-4 CPU Bus Driver Operations

This section will explain CPU Bus Driver operations, including contents of data buffers when these operations are used, as well as the return values, for each of the operations. The command numbers given in this section will match the operation numbers given in *8-3 CPU Bus Driver Operations List*.

8-4-1 Cyclic Service Transmission Address, Length, and Direction

Parameter Format



Transmission Direction:

This sets the direction in which data will be transmitted.

0: PC → Personal Computer Unit
Other than 0: Personal Computer Unit → PC

Cyclic Area Set Number:

This is the number for specifying the cyclic area that is to set the transmission status. It can be set within a range of 1 to 6.

PC's Real Space Address:

This is the real space address of the PC that is to execute cyclic service. It can be set within a range of $400000 to $4FFFFF. For the area, IOM, DM, or EM can be specified, but UM cannot. Only even-numbered addresses can be specified.

Transmission Length:

This is the cyclic area transmission length. It is specified in word units. An error will be generated if the sum of all the transmission lengths allocated to the set numbers exceeds 403 words. The default settings will be as follows:

Bit: Number 1
PC → Personal Computer Unit: 15 words ($400BB8)
Personal Computer Unit → PC: 10 words ($400BD6)

If "0" is set, the setting for the specified cyclic area will be cleared.

Return Status (AX)

00H: Normal termination
01H: Incorrect set number
02H: Incorrect real space address
03H: Total transmission length overflow (403 words max.)
0AH: Memory access error

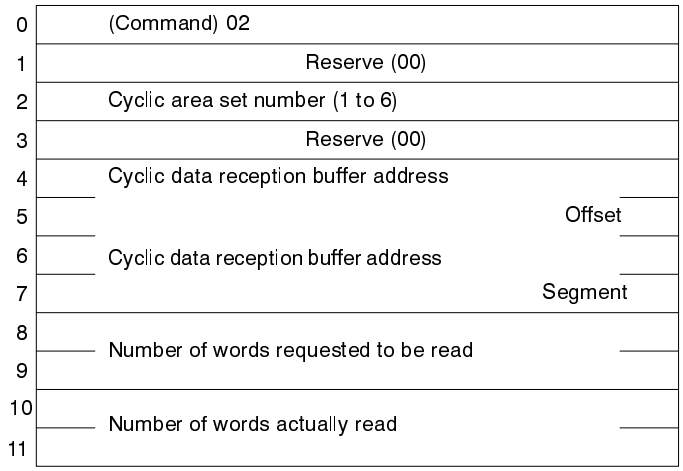
Operation

This operation sets the address and transmission length for the PC executing cyclic service. A total of 12 places can be set simultaneously, including six for transmission from the PC to the Personal Computer Unit and six for transmission from the Personal Computer Unit to the PC. When the setting is made, it will overwrite the previous status of the relevant specified bit number and transmission length.

This operation only sets the status of the cyclic service, and does not read or write data. Once these settings are made, they are valid until the Personal Computer Unit is reset; they aren't cleared by the closing processes.

8-4-2 Reading the Cyclic Area

Parameter Format



Cyclic Area Set Number:

This is the number for specifying the cyclic area that is to read data. It can be set within a range of 1 to 6.

Cyclic Data Reception Buffer Address:

This is the leading address of the buffer for storing cyclic data that is read.

Number of Words Requested to be Read:

This is the number of words of cyclic data requested to be read.

Number of Words Actually Read:

This is the area for storing the number of words that are actually read.

Return Status (AX)

00H: Normal termination
01H: Set number incorrect
02H: Reception buffer address incorrect
03H: Number of words requested incorrect
0AH: Memory access error
0BH: PC busy
0CH: Parity error

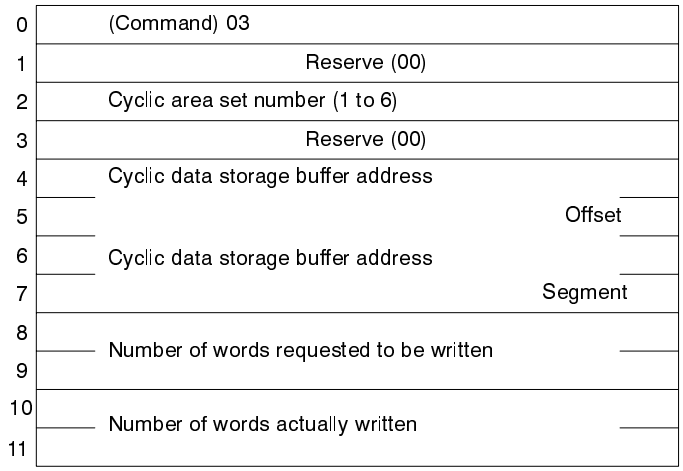
Operation

From the real space address of the PC set by the specified cyclic area set number, the number of words requested will be read and then stored in the specified reception buffer. At that time, the actual number of words read will be stored in the area specified for that purpose.

If the specified number of words to be read should exceed the number of words set for the cyclic area, then code 03H (number of words requested incorrect) will be generated. The only valid areas are ones that have had the transmission direction set to PC → Personal Computer Unit (0) with operation number 01.

8-4-3 Writing to the Cyclic Area

Parameter Format



Cyclic Area Set Number:

This is the number for specifying the cyclic area for writing data. It can be set within a range of 1 to 6.

Cyclic Data Reception Buffer Address:

This is the leading address of the buffer for storing cyclic data that is written.

Number of Words Requested to be Written:

This is the number of words of cyclic data requested to be written.

Number of Words Actually Written:

This is the area for storing the number of words that are actually written.

Return Status (AX)

00H: Normal termination
01H: Set number incorrect
02H: Data storage buffer address incorrect
03H: Number of words requested incorrect
0AH: Memory access error
0BH: PC busy

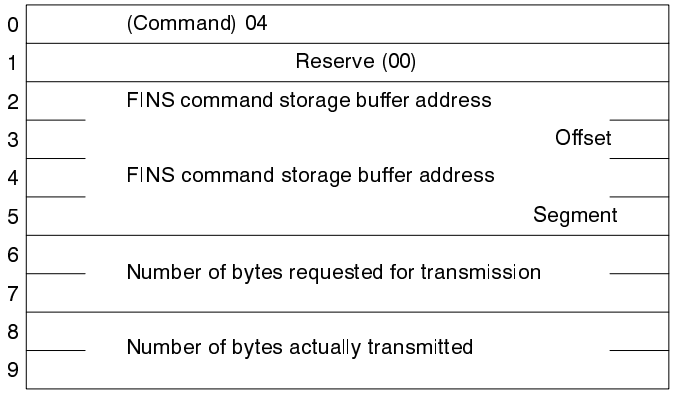
Operation

The data indicated by the data storage buffer address will be written to the real space address of the PC set by the specified cyclic area set number, for the number of words requested. At that time, the actual number of words written will be stored in the area specified for that purpose.

If the specified number of words to be written should exceed the number of words set for the cyclic area, then code 03H (number of words requested incorrect) will be generated. The only valid areas are ones that have had the transmission direction set to Personal Computer Unit → PC (non-zero) with operation number 01.

8-4-4 Transmitting FINS Commands

Parameter Format



FINS Command Storage Buffer Address:

This is the leading address of the buffer for storing transmitted FINS commands.

Number of Bytes Requested for Transmission:

This is the number of bytes requested for the FINS command. Specifies the total number of bytes from the ICF.

Number of Bytes Actually Transmitted:

This is the area for storing the number of bytes actually transmitted.

Return Status (AX)

00H: Normal termination
01H: Command storage buffer address incorrect
02H: Number of words requested incorrect
(i.e., fewer than 12 or more than 2,012 bytes)
03H: Transmission destination network address incorrect
0AH: Memory access error
0BH: PC busy error
Send again or increase the number of retries. (Refer to *5-5 CPU Bus Driver Installation and Startups* for details on the number of retries.)

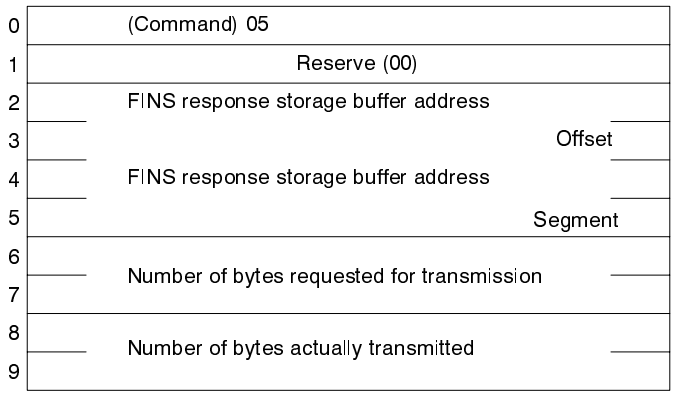
Operation

The data stored in the FINS command storage buffer address (in conformity with Programmable Controller FINS commands) is analyzed, the corresponding ICF, GCNT, SNA, SA1, and SA2 are set, and the number of bytes requested is transmitted to the specified address. At that time, the actual number of words written will be stored in the area specified for that purpose. A response to the transmitted FINS command can be received by means of a “receive FINS response” request.

If “branch entry address upon completion of command transmission” is set, then control will be transferred to the specified address after the transmission is complete.

8-4-5 Transmitting FINS Responses

Parameter Format



FINS Command Storage Buffer Address:

This is the leading address of the buffer for storing transmitted FINS responses.

Number of Bytes Requested for Transmission:

This is the number of bytes requested for the FINS response. Specifies the total number of bytes from the ICF.

Number of Bytes Actually Transmitted:

This is the area for storing the number of bytes actually transmitted.

Return Status (AX)

00H: Normal termination
01H: Command response storage buffer address incorrect
02H: Number of words requested incorrect
(i.e., fewer than 14 or more than 2,012 bytes)
03H: Transmission destination network address incorrect
0AH: Memory access error
0BH: PC busy error
Retry transmission or increase the number of retries. (Refer to *5-5 CPU Bus Driver Installation and Startups* for details on the number of retries.)

Operation

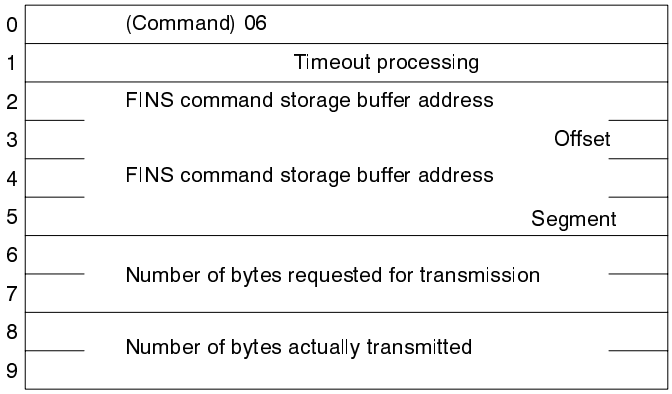
The data stored in the FINS command storage buffer address is analyzed, the corresponding ICF and GCNT are set, and the number of bytes requested is transmitted to the specified address. At that time, the actual number of words written will be stored in the area specified for that purpose.

This transmission request is used when a FINS command transmitted from another device is received and a response is transmitted to the source of the transmitted command. To receive the FINS command from the other device, use a “receive FINS command” request.

If “branch entry address upon completion of command transmission” is set, then control will be transferred to the specified address after the transmission is complete.

8-4-6 Receiving FINS Commands

Parameter Format



Timeout Processing:

0: Waits to receive until the timeout value has elapsed. (The timeout value is set with command 0CH.)

Other than 0: Waits to receive until Esc Key is pressed.

FINS Command Reception Buffer Address:

This is the leading address of the buffer for storing FINS commands that are received.

Number of Bytes Requested for Reception:

This is the number of bytes requested for receiving a FINS command.

Number of Bytes Actually Received:

This is the area for storing the number of actual bytes stored in the reception buffer.

Return Status (AX)

00H: Normal termination
01H: Command reception buffer address incorrect
02H: Number of bytes requested incorrect (i.e., fewer than 12 or more than 2,012 bytes)
0AH: Memory access error
0CH: Parity error
0EH: Forcibly ended by Escape Key
0FH: Timeout error

Operation

A FINS command transmitted from another device is received, and the number of bytes requested is stored in the specified FINS command reception buffer. At that time, the actual number of bytes stored in the buffer will be stored in the area specified for that purpose.

This reception request is used when a FINS command transmitted from another device is received. To transmit a response to the data received, use a “transmit FINS response” request.

If no command is received from any device when this command reception request is made, the Unit will wait to receive a command until the timeout value set with command 0CH has elapsed (if timeout processing has been specified) or until the Esc Key is pressed (if timeout processing hasn’t been specified).

Timeout Processing Specified

Waiting-to-receive status will remain in effect until the time period registered with command 0CH has elapsed. If a command is received during that period, the data that is received will be stored in the specified buffer. If no command is received, a timeout error will occur.

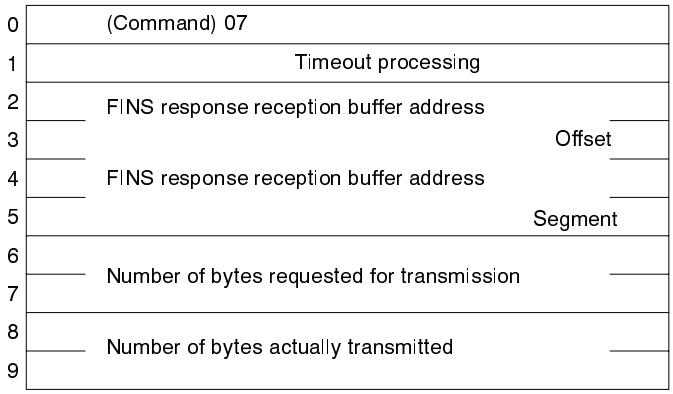
Timeout Processing Not Specified

Waiting-to-receive status will remain in effect regardless of the time period registered with command 0CH. If a command is received, the data that is received will be stored in the specified buffer. An exit can be forced during that period by pressing the Escape Key.

If even a portion of the command data that has been received is read, the command data will be cleared.

8-4-7 Receiving FINS Responses

Parameter Format



Timeout Processing:

0: Waits to receive until timeout value has elapsed. (The timeout value is set with command 0CH.)

Other than 0: Waits to receive until Esc Key is pressed.

FINS Response Reception Buffer Address:

This is the leading address of the buffer for storing FINS responses that are received.

Number of Bytes Requested for Reception:

This is the number of bytes requested for receiving a FINS response.

Number of Bytes Actually Received:

This is the area for storing the number of actual bytes stored in the reception buffer.

Return Status (AX)

00H: Normal termination
01H: Command response reception buffer address incorrect
02H: Number of bytes requested incorrect (i.e., fewer than 14 or more than 2,012 bytes)
0AH: Memory access error
0BH: Parity error
0EH: Forcibly ended by Escape Key
0FH: Timeout error

Operation

A FINS response to a service requested of another device is received, and the number of bytes requested for receiving the response is stored in the specified FINS response reception buffer. At that time, the actual number of bytes stored in the buffer will be stored in the area specified for that purpose.

This reception request is used when a FINS response to a service requested of another device is received. To request a service of another device, use a "transmit FINS command" request.

If no command is received from any device when this command reception request is made, the Unit will wait to receive a command until the timeout value set with command 0CH has elapsed (if timeout processing has been specified) or until the Esc Key is pressed (if timeout processing hasn't been specified).

Timeout Processing Specified

Waiting-to-receive status will remain in effect until the time period registered with command 0CH has elapsed. If a response is received during that period, the data that is received will be stored in the specified buffer. If no command is received, a timeout error will occur.

Timeout Processing Not Specified

Waiting-to-receive status will remain in effect regardless of the time period registered with command 0CH. If a response is received, the data that is received will be stored in the specified buffer. An exit can be forced during that period by pressing the Escape Key.

If even a portion of the command data that has been received is read, the command data will be cleared.

8-4-8 Branching Upon Completion of FINS Command Transmission

Parameter Format

0	(Command) 08	
1	Reserve (00)	
2	Entry address upon completion of command transmission	
3		Offset
4	Entry address upon completion of command transmission	
5		Segment

Entry Address Upon Completion of Command Transmission:

This is the leading address of the process that will be executed when the command transmission has been completed.

Return Status (AX)

00H: Normal termination

Operation

This operation registers the user entry address for branching when the FINS command transmission has been completed. If this address is set to "0", the registered user entry address will be cleared.

Note

1. For user routines, use the same stack area.
2. When user routine processing has been completed, return control with FAR RET.
3. In the process that is executed, the ds register points out the data segment of the driver. The register will be returned to its original status after the process is completed.

8-4-9 Branching Upon Completion of FINS Response Transmission

Parameter Format

0	(Command) 09	
1	Reserve (00)	
2	Entry address upon completion of response transmission	
3		Offset
4	Entry address upon completion of response transmission	
5		Segment

Entry Address Upon Completion of Response Transmission:

This is the leading address of the process that will be executed when the response transmission has been completed.

Return Status (AX)

00H: Normal termination

Operation This operation registers the user entry address for branching when the FINS response transmission has been completed. If this address is set to "0", the registered user entry address will be cleared.

Note
1. For user routines, use the same stack area.
2. When user routine processing has been completed, return control with FAR RET.
3. In the process that is executed, the ds register points out the data segment of the driver. The register will be returned to its original status after the process is completed.

8-4-10 Branching Upon Completion of FINS Command Reception

Parameter Format

0	(Command) 0A(H)
1	Reserve (00)
2	Entry address upon completion of command transmission
3	Offset
4	Entry address upon completion of command transmission
5	Segment

Entry Address Upon Completion of Command Reception:

This is the leading address of the process that is executed when the command reception has been completed.

Return Status (AX) 00H: Normal termination

Operation This operation registers the user entry address for branching when the FINS command reception has been completed. If this address is set to "0", the registered user entry address will be cleared.

Note
1. For user routines, use the same stack area.
2. When user routine processing has been completed, return control with FAR RET.
3. In the process that is executed, the ds register points out the data segment of the driver. The register will be returned to its original status after the process is completed.

8-4-11 Branching Upon Completion of FINS Response Reception

Parameter Format

0	(Command) 0B(H)
1	Reserve (00)
2	Entry address upon completion of response transmission
3	Offset
4	Entry address upon completion of response transmission
5	Segment

Entry Address Upon Completion of Response Reception:

This is the leading address of the process that is executed when the response reception has been completed.

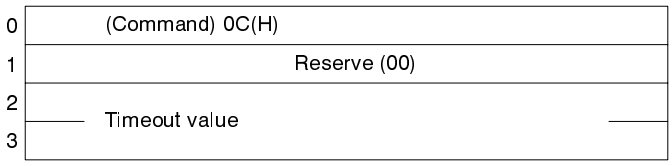
Return Status (AX) 00H: Normal termination

Operation This operation registers the user entry address for branching when the FINS response reception has been completed. If this address is set to "0", the registered user entry address will be cleared.

Note
1. For user routines, use the same stack area.
2. When user routine processing has been completed, return control with FAR RET.
3. In the process that is executed, the ds register points out the data segment of the driver. The register will be returned to its original status after the process is completed.

8-4-12 Setting Timeout Values

Parameter Format



Timeout Value:

This is the timeout value used when data reception is requested. The setting can be made within a range of 0 to 65,535 (in units of 110 ms).

Return Status (AX) 00H Normal termination

Operation This operation registers the time period for waiting-to-receive when a FINS command or a FINS response reception is requested. The timeout value is set in units of 110-ms. If "0" is registered, the Unit will not wait to receive the transmission. The default for the timeout value is 0 ms.

8-4-13 Flushing Reception Buffers

Parameter Format

Parameters:

Specifies which reception buffers are to be flushed. (The reception buffers are allocated by the driver.)

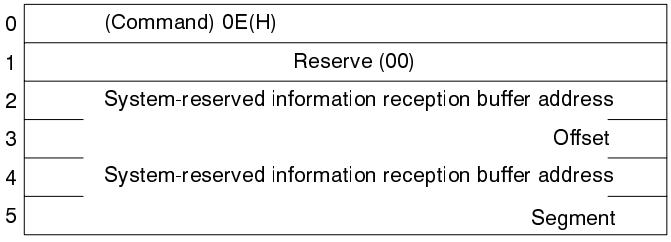
0: Reception buffer for command data
1: Reception buffer for response data
2: Reception buffer for both command and response data

Return Status (AX): 00H: Normal termination
01H: Parameter values incorrect

Operation This operation flushes (clears) the specified reception buffers, according to the specified parameters.

8-4-14 Reading Information Reserved for System

Parameter Format



System-reserved Information Reception Buffer Address:

This is the leading address of the buffer for storing system-reserved information that is read.

Return Status (AX)

00H: Normal termination
01H: Reception buffer address incorrect
0AH: Memory access error
0CH: Parity error

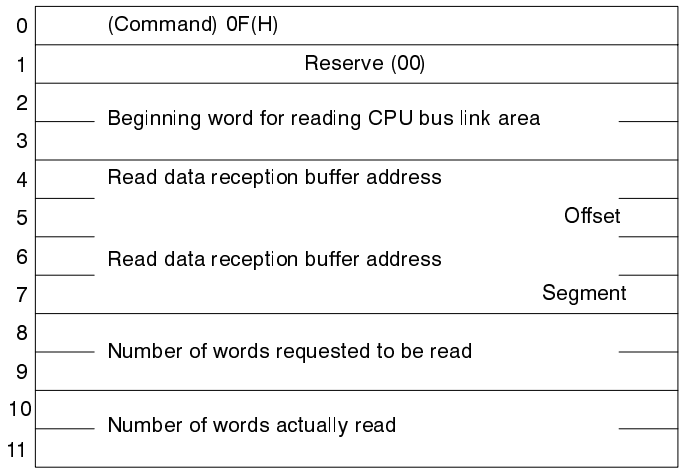
Operation

This operation reads information reserved for the system and stores it in the specified reception buffer. The system-reserved information is configured in eight words, so fixed 8-word blocks are read. This operation can be performed even when the CPU bus link service is stopped.

Refer to *7-3 Communications/Control Service Details* for details on the system-reserved information.

8-4-15 Reading the Link Area

Parameter Format



Beginning Word for Reading CPU Bus Link Area:

This is the beginning word of the CPU bus link area for reading data. (It can be set within a range of word 0 to word 255.)

Read Data Reception Buffer Address:

This is the leading address of the buffer for storing the link data that is read.

Number of Words Requested to be Read:

This is the number of words of link data that is requested to be read.

Number of Words Actually Read:

This is the area for storing the number of words that is actually read.

Return Status (AX)

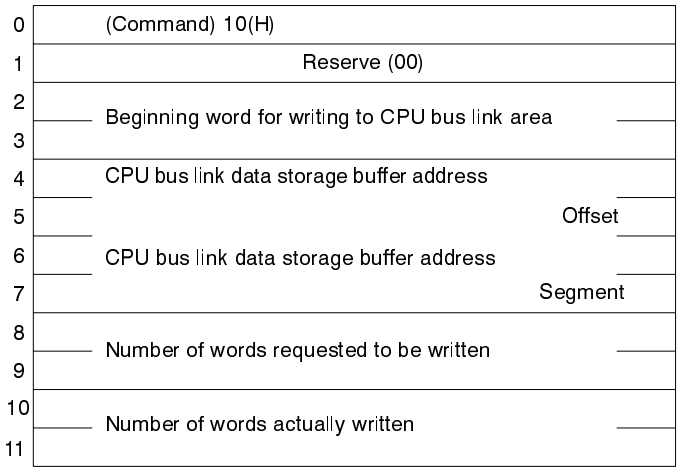
00H: Normal termination
01H: Beginning word incorrect
02H: Reception buffer address incorrect
03H: Number of words requested incorrect
0AH: Memory access error
0CH: Parity error

Operation

From the CPU bus link area address corresponding to the specified beginning word, the number of words requested will be read and then stored in the specified reception buffer. At that time, the actual number of words read will be stored in the area specified for that purpose. If the specified number of words to be read should exceed the CPU bus link area, then code 03H (number of words requested incorrect) will be generated.

8-4-16 Writing to the Link Area

Parameter Format



Beginning Word for Writing to CPU Bus Link Area:

This is the beginning word of the CPU bus link area for writing data. (It can be set within a range of word 0 to word 7.)

CPU Bus Link Data Storage Buffer Address:

This is the leading address of the buffer where the CPU bus link data that is to be written is stored.

Number of Words Requested to be Written:

This is the number of words of link data that is requested to be written.

Number of Words Actually Written:

This is the area for storing the number of words that is actually written.

Return Status (AX)

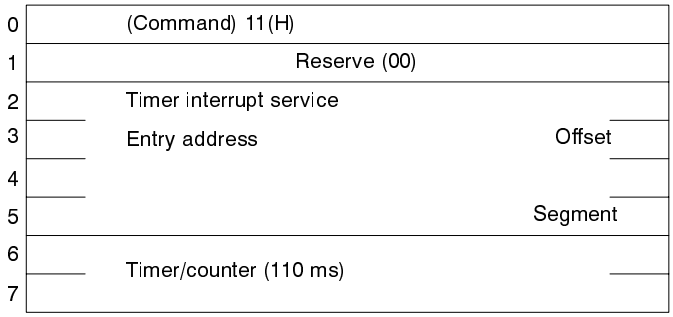
00H: Normal termination
01H: Beginning word incorrect
02H: Data storage buffer address incorrect
03H: Number of words requested incorrect
0AH: Memory access error

Operation

The data indicated by the data storage buffer address will be written to the CPU bus link area address corresponding to the specified beginning word, for the number of words requested. At that time, the actual number of words written will be stored in the area specified for that purpose. If the specified number of words to be written should exceed the CPU bus link area, then code 03H (number of words requested incorrect) will be generated.

8-4-17 User Timer Service Processing

Parameter Format



Timer Interrupt Service Entry Address:

This is the leading address of the process that is started when the time period specified by the timer/counter has elapsed.

Timer/Counter:

This is the time period until the timer is started. It can be set within a range of 0 to 65,535, in 110-ms units.

Return Status (AX)

00H: Normal termination

Operation

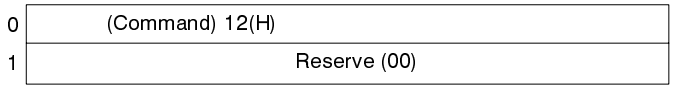
This operation registers the user entry address for starting after the time period specified by the timer/counter has elapsed. If this address is set to "0", the registered user entry address will be cleared. If the user entry address is registered, timer monitoring will begin immediately.

Note

1. For user routines, use the same stack area.
2. When user routine processing has been completed, return control with FAR RET.
3. In the process that is executed, the ds register points out the data segment of the driver. The register will be returned to its original status after the process is completed.

8-4-18 Resetting the Personal Computer Unit

Parameter Format



Return Status (AX)

00H: Normal termination
0AH: Memory access error

Operation

This operation resets the Personal Computer Unit and is just like performing an AR reset from the PC. Cyclic service settings and CPU bus link service contents are initialized, and the routing table is read again.

8-4-19 Unit Address Inquiry

Parameter Format

0	(Command) 13(H)
1	Reserve (00)
2	Local unit address
3	Reserve (00)

Local Unit Address:

The unit address of the Personal Computer Unit is stored in this area.

Return Status (AX) 00H: Normal termination

Operation This operation returns the Personal Computer Unit's (local unit's) unit address.

8-4-20 Reception Status Inquiry

Parameter Format

0	(Command) 14(H)
1	Timeout processing

Timeout Processing:

0: Waits to receive until timeout value has elapsed.
(The timeout value is set with command 0CH.)
Other than 0: Waits to receive until Esc Key is pressed.

Return Status (AX)

04H: Reception command data present
05H: Reception response data present
0EH: Forcibly ended by Esc Key
0FH: Timeout error

Operation This operation inquires regarding the present status of event data reception. When both command data and event data have been received, the response data reception status is given priority in being returned.

If neither command data nor response data have been received, the Unit will wait to receive a command until the timeout value set with command 0CH has elapsed (if timeout processing has been specified) or until the Esc Key is pressed (if timeout processing hasn't been specified).

Timeout Processing Specified
Waiting-to-receive status will remain in effect until the time period registered with command 0CH has elapsed. If a command or response is received during that period, then that will be reported. If no command is received, a timeout error will occur.

Timeout Processing Not Specified
Waiting-to-receive status will remain in effect regardless of the time period registered with command 0CH. If a command or response is received, then that will be reported. An exit can be forced during that period by pressing the Escape Key.

This operation only inquires regarding the reception status, and does not actually read the data. Therefore reception requests for commands or responses can be executed based on the returned values of this operation.

8-5 FINS Commands Serviced by Drivers

With event service, when a service is requested of the Personal Computer Unit by another device (i.e., when the Personal Computer Unit receives the command), the response processing will be executed in the CPU Bus Driver with respect to several commands, and a response will be returned to the source of the transmission.

Therefore, because the commands that the user can receive by means of a "command reception request" may be commands other than those processed in the CPU Bus Driver, it may be necessary for the user to execute the response processing and return the response to the source of the transmission.

The commands for which response processing is executed in the CPU Bus Driver are explained below, along with the response contents.

Note

1. The service request PDU is the format, from MRC onwards, of the FINS command created at the time of requesting a service.
2. The service response PDU is the format, from MRC onwards, of the FINS response when a response is transmitted or received with respect to a service request.

Commands for which response processing is executed by driver are as follows.

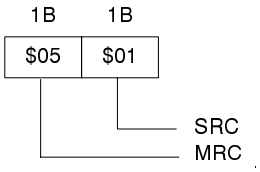
MRC	SRC	Command contents	Page
$05	$01	Read Controller Information	122
$07	$01	Read Time Information	123
$07	$02	Write Time Information	124
$08	$01	Loopback Test	125
$21	$02	Read Error Log	125
$21	$03	Clear Error Log	127

8-5-1 Read Controller Information (0501)

Operation

This operation inquires regarding the format and version of the Personal Computer Unit and the version of the driver.

Service Request PDU



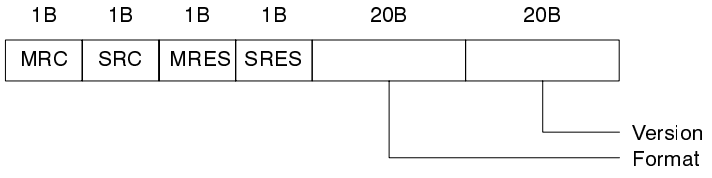
MRC (1 Byte):

Main Request Classification. Read Device Information is indicated by $05.

SRC (1 Byte):

Sub-request Classification. Read Controller Information is indicated by $01.

Service Response PDU



MRC (1 Byte):

Main Request Classification. The response to Read Device Information is indicated by $05.

SRC (1 Byte):

Sub-request Classification. The response to Read Controller Information is indicated by $01.

MRES (1 Byte):

Main Response Code. $00 is always returned.

SRES (1 Byte):

Sub-response Code. $00 is always returned.

Model (20 Bytes):

The Personal Computer Unit model is returned in ASCII code, with a maximum of 20 bytes. If the full 20 bytes is not needed, the remaining portion will be filled with spaces.

Version (20 Bytes):

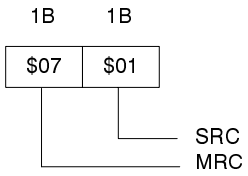
The Personal Computer Unit OS and the CPU Bus Driver version are returned in ASCII code, with a maximum of 20 bytes. If the full 20 bytes is not needed, the remaining portion will be filled with spaces.

8-5-2 Read Time Information (0701)

Operation

This operation inquires regarding Personal Computer Unit's time information.

Service Request PDU



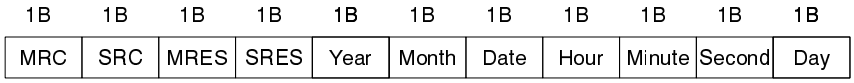
MRC (1 Byte):

Main Request Classification. Access Time Information is indicated by $07.

SRC (1 Byte):

Sub-request Classification. Read Time Information is indicated by $01.

Service Response PDU



MRC (1 Byte):

Main Request Classification. The response to Access Time Information is indicated by $07.

SRC (1 Byte):

Sub-request Classification. The response to Read Time Information is indicated by $01.

MRES (1 Byte):

Main Response Code. $00 is always returned.

SRES (1 Byte):

Sub-response Code. $00 is always returned.

Year (Two Rightmost Digits) to Second:

The time information that is read is returned as 1-byte pieces of BCD data.

Day:

Information on the day of the week is coded as follows:

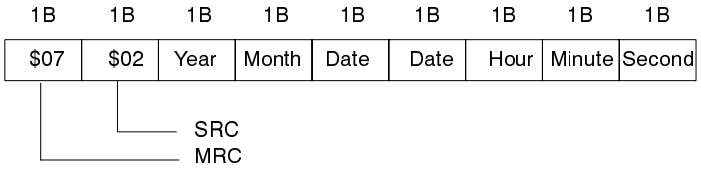
$00=Sunday, $01=Monday, $02=Tuesday, $03=Wednesday, $04=Thursday, $05=Friday, $06=Saturday

8-5-3 Write Time Information (0702)

Operation

This operation writes time information to the Personal Computer Unit.

Service Request PDU



MRC (1 Byte):

Main Request Classification. Access Time Information is indicated by $07.

SRC (1 Byte):

Sub-request Classification. Write Time Information is indicated by $02.

Year (Two Rightmost Digits) to Minute:

The time information to be written is indicated by 1-byte pieces of BCD data.

Second:

The second information to be written is indicated by 1-byte pieces of BCD data. This parameter is optional and can be omitted.

Day:

Information on the day of the week is coded as follows:
$00=Sunday, $01=Monday, $02=Tuesday, $03=Wednesday, $04=Thursday, $05=Friday, $06=Saturday
This parameter is optional and can be omitted.

Service Response PDU

MRC (1 Byte):

Main Request Classification. The response to Access Time Information is indicated by $07.

SRC (1 Byte):

Sub-request Classification. The response to Write Time Information is indicated by $02.

MRES (1 Byte):

Main Response Code. $00 is returned when correct, and $11 when there is an error.

SRES (1 Byte):

Sub-response Code. The cause of the error is reported.

$00: Normal termination
$01: Command format error
$02: Parameter error

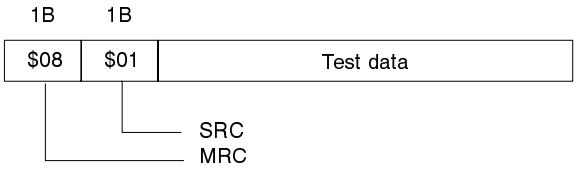
Note The ISA Control Unit reads the the PC's clock and sets the time automatically when the CPU driver is installed.

8-5-4 Loopback Test (0801)

Operation

This operation indicates the connected devices, and conducts a loopback test with each of the devices.

Service Request PDU



MRC (1 Byte):

Main Request Classification. Communications Tests are indicated by $08.

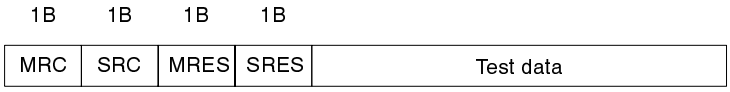
SRC (1 Byte):

Sub-request Classification. The Loopback Test is indicated by $01.

Test Data:

This is the arbitrary test code.

Service Response PDU



MRC (1 Byte):

Main Request Classification. The response to Communications Tests is indicated by $08.

SRC (1 Byte):

Sub-request Classification. The response to Loopback Test is indicated by $01.

MRES (1 Byte):

Main Response Code. $00 is returned when correct, and $11 when there is an error.

SRES (1 Byte):

Sub-response Code. The cause of the error is reported.

$00: Normal termination
$01: Command format error

Test Data:

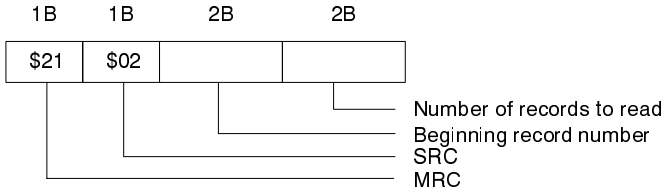
Data the same as that transmitted by service request is returned. The length of the test data is the same as the service request PDU.

8-5-5 Read Error Log (2102)

Operation

This operation inquires regarding error information generated at the ISA Control Unit. The error information is erased from the system after it is read.

Service Request PDU



MRC (1 Byte):

Main Request Classification. Error Logging Operations are indicated by $21.

SRC (1 Byte):

Sub-request Classification. Read Error Log is indicated by $02.

Beginning Record Number (2 Bytes):

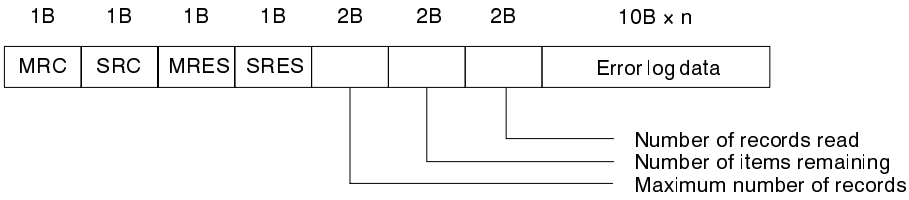
This indicates the first record number to be read. Each record consists of 10 bytes of data, and stores information on a single error.

The first record number is $0000. When the error history is read from a later record (greater than $0000), all earlier records will be erased.

Number of Records to Read (2 Bytes):

This indicates the number of records to be read.

Service Response PDU



MRC (1 Byte):

Main Request Classification. The response to Error Logging Operations is indicated by $21.

SRC (1 Byte):

Sub-request Classification. The response to Read Error Log is indicated by $02.

MRES (1 Byte):

Main Response Code. $00 is returned when correct, and $11 when there is an error.

SRES (1 Byte):

Sub-response Code. The cause of the error is reported.

$00: Normal termination
$01: Command format error
$02: Parameter error

Maximum Number of Records (2 Bytes):

This indicates the maximum number of error records that can be stored.

Number of Items Remaining (2 Bytes):

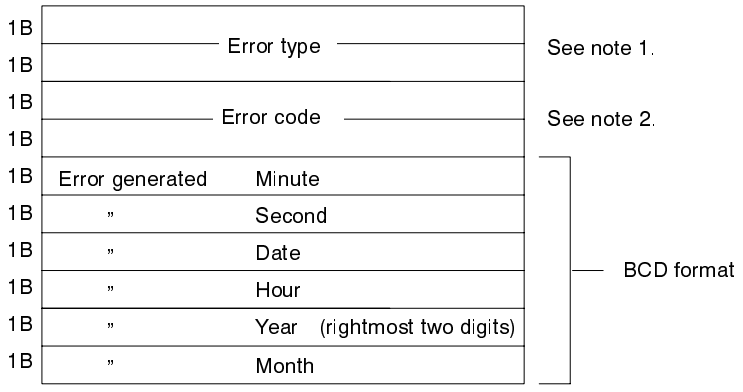
This indicates the number of items remaining after reading the error log.

Number of Records Read (2 Bytes):

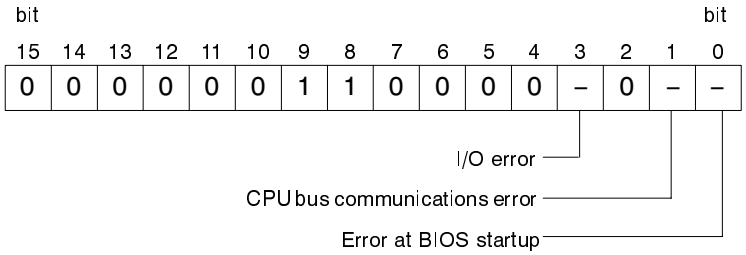
The number of records actually read is returned.

Error Log Data:

Error records are stored in the format shown below.



Note 1. Content and Meaning of Error Types



When the error log is read with the ISA Control Unit's ERRLOG.EXE program, bits 8 and 9 will be 0 (OFF).

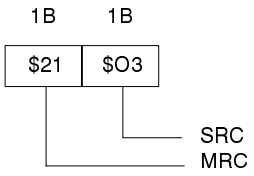
2. For a list of error codes, refer to the appendices.

8-5-6 Clear Error Log (2103)

Operation

This operation resets the error log generated at the ISA Control Unit.

Service Request PDU



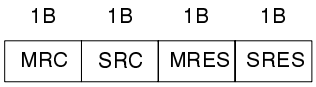
MRC (1 Byte):

Main Request Classification. Error Logging Operations are indicated by $21.

SRC (1 Byte):

Sub-request Classification. Clear Error Log is indicated by $03.

Service Response PDU



MRC (1 Byte):

Main Request Classification. The response to Error Logging Operations is indicated by $21.

SRC (1 Byte):

Sub-request Classification. The response to Clear Error Log is indicated by $03.

MRES (1 Byte):

Main Response Code. $00 is always returned.

SRES (1 Byte):

Sub-response Code. $00 is always returned.

8-6 Sample Programs

Sample programs using the CPU Bus Driver are provided below.

```
1    /*********************************************/
2    /* CPU Bus Driver                            */
3    /*                                           */
4    /* Cyclic Service Sample Program             */
5    /*********************************************/
6    #include        <dos.h>
7    #include        <stdio.h>
8
9    typedef void far        *farptr;
10
11   union    REGS   inregs,outregs;        /*I/O register structure*/
12   int      fd;                           /*File handle*/
13   char     length;                       /*Number of bytes for IOCTL transmission*/
14   static   char   buf[16], databuf[512]; /*Data buffer for IOCTL*/
15
16   static   char   driver_id[] = "CVIF";          /*Driver name*/
17   static   char   d_addrset[] = { 10, 0x01, 0x00 };  /*Sets transmission status*/
18   static   char   d_datread[] = { 12, 0x02, 0x00 };  /*Data reading*/
19   static   char   d_datwrit[] = { 12, 0x03, 0x00 };  /*Data writing*/
20
21   void     errclose();
22
23   void
24   main()
25   {
26     int   i;
27     register char *bufp;
28     register short *sp;
29
30     inregs.h.ah = 0x3d;        /*Opens driver.*/
31     inregs.h.al = 0x02;        /*Read/write mode*/
32     inregs.x.dx = (short)driver_id;
33     intdos(&inregs, &outregs); /*int 21h*/
34     if (outregs.x.cflag !=0){
35           printf("Driver not loaded./n");
36           printf("/tError code = 0x%x/n",outregs.x.ax);
37           exit(1);
38     }
39     fd = outregs.x.ax;         /*Acquires file handle.*/
40
41     strncpy (buf,&d_addrset[1],2);          /*Sets cyclic service status*/
42     length = d_addrset[0];
43     bufp = &buf[2];
44     *bufp++ = 0;          /*Transmission direction, bus driver to Unit*/
45     *bufp++ = 1;          /*Set number*/
46     *(long*)bufp = (long)0x00404000;          /*Sets real address of bus driver*/
47     bufp += 4;
48     *(short*)bufp = 10;          /*Sets transmission length*/
49     if (ioctl())
50           errclose();
51
52     strncpy (buf,&d_addrset[1],2);          /*Sets cyclic service status*/
53     length = d_addrset[0];
54     bufp = &buf[2];
```

```
55      *bufp++ = 1;          /*Transmission direction, Unit to bus driver*/
56      *bufp++ = 1;          /*Set number*/
57      *(long*)bufp = (long)0x00404000;          /*Sets real address of bus driver*/
58      bufp += 4;
59      *(short*)bufp = 10;          /*Sets transmission length*/
60      if(ioctl())
61      errclose();
62
63      strncpy(buf,&d_datwrit[1],2);          /*Writing cyclic area*/
64      length = d_datwrit[0];
65      bufp = &buf[2];
66      *bufp++ = 1;          /*Set number*/
67      *bufp++ = 0;          /*Reserve*/
68      strcpy (databuf,"1234567890abcdefghij");
69      *(long*)bufp = (long)(farptr)databuf;              /*Sets writing data buffer*/
70      bufp += 4;
71      *(short*)bufp = 10;            /*Sets number of words requested for writing*/
72      if (ioctl())
73            errclose();
74      printf ("The number of words actually written is %d.
/n",*(short*)&buf[10]);
75
76      strncpy(buf,&d_datread[1],2);          /*Reading cyclic area*/
77      length = d_datread[0];
78      bufp = &buf[2];
79      *bufp++ = 1;          /*Set number*/
80      *bufp++ = 0;          /*Reserve*/
81      *(long*)bufp = (long)(farptr)databuf;          /*Sets receiving buffer address*/
82      bufp += 4;
83      *(short*)bufp = 10;            /*Sets number of words requested for reading*/
84      if (ioctl())
85            errclose();
86
87      printf ("Data received in the DM area /n");
88      sp = (short*)databuf;  /*Displays received data*/
89      for (i = *(short*)&buf[10];--i> = 0;)
90            printf("0x%x",*sp++);
91      printf("/n");
92
93      inregs.h.ah = 0x3e;         /*Closes driver.*/
94      inregs.x.bx = fd;
95      intdos(&inregs, &outregs); /*int 21h*/
96    }
97
98    int
99    ioctl()
100   {
101     inregs.x.ax = 0x4403;         /*I/O request*/
102     inregs.x.bx = fd;
103     inregs.x.cx = length;         /*Sets number of bytes for transmission.*/
104     inregs.x.dx = (short)buf;     /*Sets parameter buffer address.*/
105     intdos(&inregs,&outregs);
106     if (outregs.x.cflag||outregs.x.ax)
107           return(1);
108     return(0);
109   }
```

```
110
111  void
112  errclose()
113  {
114    printf("IOCTL error/n");
115    printf("/tcmd=0x%x carry=0x%x AX=0x%x/n",
116                       buf[0],outregs.x.cflag,outregs.x.ax);
117    inregs.h.ah = 0x3e;          /*Closes driver.*/
118    inregs.x.bx = fd;
119    intdos(&inregs, &outregs);   /*int 21h*/
120    exit(2);
121  }

1    /*********************************************/
2    /* CPU Bus Driver                            */
3    /*                                           */
4    /* Event Service Sample Program              */
5    /*********************************************/
6    #include        <dos.h>
7    #include        <stdio.h>
8
9    #define MAX      2012                  /*Event data MAX size*/
10
11   typedef  void far       *farptr;
12
13   union    REGS   inregs,outregs;        /*I/O register structure*/
14   int      fd;                           /*File handle*/
15   char     length;                       /*Number of bytes for IOCTL transmission*/
16   static   char   buf[16];               /*Data buffer for IOCTL*/
17   static   unsighned char  databuf[MAX] = {     /*Command data for transmission*/
18   /* ICF   RSV   GCNT   DNA   DA1   DA2   SNA   SA1   SA2   SID */
19      0x00,0x00,0x00,0x00,0x00,0x14,0x00,0x00,0x00,0x00,
20   /* MRC   SRC  */
21      0X07,0X01
22   };             */ Parameter area  (Reads time information) */
23
24   static   char   driver_id[] = "CVIF";              /*Driver name*/
25   static   char   d_cmdsend[] = {10,0x04,0x00};      /*Transmits  command.*/
26   static   char   d_rspread[] = {10,0x07,0x00};      /*Receives response.*/
27   static   char   d_rrcvjmp[] = {6,0x0b,0x00};   /*Branches at response reception.*/
28   static   char   d_timeout[] = {4,0x0c,0x00};       /*Sets timeout value.*/
29
30   void far jmpfunc();
31   void far *jmpaddr[] = {jmpfunc};       /*Response reception-complete branch
address*/
32   void errclose();
33
34    /*Assembler function declarations*/
35    extern   void   f_set();
36    extern   int    f_sense();
37    extern   void   f_cls();
38
39    void
40    main()
41    {
42       int      i,ans;
43       register unsigned char      *bufp;
```

```
44
45        inregs.h.ah = 0x3d;          /*Opens driver.*/
46        inregs.h.al = 0x02;          /*Read/write mode*/
47        inregs.x.dx = (short)driver_id;
48        intdos(&inregs, &outregs); /*int 21h*/
49        if (outregs.x.cflag !=0){
50             printf("Driver not loaded./n");
51             printf("/tError code = 0x%x/n",outregs.x.ax);
52             exit(1);
53     }
54     fd = outregs.x.ax;          /*Acquires file handle.*/
55      bufp = &buf[2];
56
57      strcpy (buf,&d_timeout[1]);    /*Sets timeout value*/
58      length = d_timeout[0];
59      *(short*)bufp = 100;           /*Timeout value*/
60      if(ioctl())
61      errclose();
62
63      strcpy(buf,&d_rrcvjmp[1]);    /*Registers branch address at completion of response
reception*/
64      length = d_rrcvjmp[0];
65      *(long*)bufp = *(long*)jmpaddr;    /*Sets branch address at completion of  reception*/
66      if(ioctl())
67             errclose();
68
69      f_cls();                              /*Clears reception flag.*/
70      strcpy(buf,&d_cmdsend[1]);           /*Transmits command.*/
71      length = d_cmdsend[0];
72      *(long*)bufp = (long)(farptr)databuf;       /*Sets transmission buffer address*/
73      bufp += 4;
74      *(short*)bufp = 12;      /*Sets number of bytes requested for transmission*/
75      if(ioctl())
76            errclose();
77
78      ans = i = 0;      /*Calculates the sum from 1*/
79      while (!f_sense()){
80           ans += (i++);
81           if(ans>30000)
82                break;
83      }
84
85      strcpy(buf,&d_rspread[1]);          /*Receives response.*/
86      length = d_rspread[0];
87      bufp = &buf[2];
88      *(long*)bufp = (long)(farptr)databuf;       /*Sets reception buffer address  */
89      bufp += 4;
90      *(short*)bufp = MAX      /*Sets  number of bytes requested for reception*/
91      printf ("Response reception start/n");
92      if(ioctl())
93           errclose();
94
95      printf ("Sum from 1 to %d is %d. /n",(i-1,ans);
96      printf ("Response data received (reads only the time data) /n");
97      bufp = databuf;      /*Displays data received*/
98      for (i = *(short*)&buf[8];--i> = 0;)
```

```
99          printf("0x%x",*bufp++);
100   printf("/n");
101
102   inregs.h.ah = 0x3e;         /* Closes driver*/
103   inregs.x.bx = fd;
104   intdos(&inregs, &outregs); /*int 21h*/
105 }
106
107 #pragma check_stack(off)
108 void     far
109    jmpfunc()
110 {
111    f_set();      /*Sets reception flag*/
112 }
113 #pragma check_stack(on)
114
115 int
116 ioctl()
117 {
118   inregs.x.ax = 0x4403;         /*I/O request*/
119   inregs.x.bx = fd;
120   inregs.x.cx = length;         /*Sets number of bytes to receive*/
121   inregs.x.dx = (short)buf;     /*Sets parameter buffer address*/
122   intdos(&inregs,&outregs);
123   if (outregs.x.cflag||outregs.x.ax)
124          return(1);
125   return(0);
126 }
127
128 void
129 errclose()
130 {
131   printf("IOCTL error/n");
132   printf("/tcmd=0x%x carry=0x%x AX=0x%x/n",
133                      buf[0],outregs.x.cflag,outregs.x.ax);
134   inregs.h.ah = 0x3e;           /*Closes driver.*/
135   inregs.x.bx = fd;
136   intdos(&inregs, &outregs);    /*int 21h*/
137   exit(2);
138 }
```

```
1   /*********************************************/
2   /* CPU Bus Driver                            */
3   /*                                           */
4   /* Event Transmission Sample Program         */
5   /*********************************************/
6   #include        <dos.h>
7   #include        <stdio.h>
8   #include        <time.h>
9
10  typedef unsigned char uchar;
11  typedef unsigned short ushort;
12  typedef void far *farptr;
13
14  union    REGS   inregs,outregs;        /*I/O register structure*/
15  union    REGS   inregs2,outregs2;      /*I/O register structure*/
16  int      fd;                           /*File handle*/
```

```
17   int      rcnt;                          /*Transmission counter*/
18   char     length;                        /*Number of ioctl transmission digits*/
19   uchar    mbuf[15],*buf;                 /*ioctl buffer*/
20   uchar    recvbuf[2048];                 /*Reception buffer*/
21   char     cmd04hdr[] = {0xa,0x04,0x00}; /*Transmits event/FINS command.*/
22   char     cmd07hdr[] = {0xa,0x07,0x00}; /*Receives event/FINS response.*/
23   char     cmd0bhdr[] = {0x6,0x0b,0x00}; /*Branches at completion of FINS response
reception.*/
24   char     cmd11hdr[] = {0x8,0x11,0x00}; /*Registers time-up branch entry address.*/
25
26   /*Assembler function declarations*/
27   extern   int    f-sense(),c-sense();
28   extern   void   f-set(),f-cls();
29   extern   void   c-inc(),c-dec(),c-cls()
30   void cmd04(),cmd07(),cmd0b(),cmd11(),
31   void far jmp0ba(),far jmp11a();
32   void disp_recv(),ioctl();
33
34   void far *jmp0badr[] = {jmp0ba};       /*Response reception-complete entry address*/
35   void far *jmp11adr[] = {jmp11a};       /*Time-up branch entry address*/
36   void     timetime(int);                /*WAIT function (54.9 ms)*/
37   void     moritime(double*);
38
39   void     main()
40   {
41
42   inregs.h.ah = 0x3d;         /*Opens file.*/
43   inregs.h.al = 0x02;         /*read/write*/
44   inregs.x.dx = (short)“CVIF”;
45   intdos(&inregs, &outregs); /*int 21h*/
46   if(outregs.x.cflag !=0)
47      printf(“\n Driver not loaded.errcode = %x/n”,outregs.x.ax);
48   else{
49      fd = outregs.x.ax;          /*Saves file handle.*/
50      rcnt = 0;                   /*SID counter*/
51      f_cls();                    /*Clears flag.*/
52      c_cls();                    /*Clears counter.*/
53      cmd0b();                /*Branches at completion of FINS response reception.*/
54      cmd11();                    /*Registers time-up branch entry address.*/
55      do{
56            if(!c_sense()){
57                  timetime(2);         /*110ms*/
58
59                  cmd04();             /*Transmits event/FINS command.*/
60                  }else{
61                        cmd07();       /*Receives event/FINS response.*/
62                        c_dec();
63                  }
64            }
65            while(!f_sense()||c_sense());
66
67            inregs.h.ah = 0x3e;         /*Closes file.*/
68            inregs.x.bx = fd;
69            intdos(&inregs, &outregs); /*int 21h*/
70      }
71   }
```

```
72
73   /*FINS command transmission processing*/
74   void     cmd04()
75   {
76      char *adr;
77      static char  finsdata[] = {
78   /*       ICF,RSV,GCNT,DNA,DA1,DA2,SNA,SA1,SA2,SID*/
79            0,  0,  0,   0,  0,  0x14,0, 0,  0,  0,
80   /*       MRC,SRC,DATA*/
81            0x11,0x12,1,2};
82
83      buf = mbuf;
84      length = *cmd04hdr;                    /*Number of ioctl transmission digits*/
85      adr = cmd04hdr;
86      strncpy(buf,++adr,2);                  /*Command*/
87      buf+=2;
88      (uchar)famsdata[9] = (uchar)rcnt++;    /*SID*/
89      *(long*)buf = (long)(farptr)finsdata;  /*Command buffer*/
90      buf += 4;
91     *((ushort*)buf) = 14;               /*Number of bytes requested for transmission*/
92      ioctl();
93      if(outregs.x.ax)
94            printf("Command transmission  Code error: %d/n",outregs.x.ax);
95   }
96
97   /*FINS response processing*/
98   void     cmd07()
99   {
100     char   *adr;
101
102     buf = mbuf;
103     length = *cmd07hdr;                     /*Number of ioctl transmission digits*/
104     adr = cmd07hdr;
105     strncpy(buf,++adr,2);                   /*Command*/
106     buf++;
107     *((uchar*)buf)++ = 1;                /*Timer monitoring: 0: Yes; Other: No*/
108     *((long*)buf)++ = (long)(farptr)recvbuf; /*Command buffer*/
109     *((ushort*)buf) = 1000;                 /*Number of bytes requested for reception*/
110     ioctl();
111     if(!outregs.x.ax)
112           disp_recv(*(short*)(mbuf+8));/*Displays data received.*/
113     else
114           printf("Response reception  Code error: %d/n",outregs.x.ax);
115  }
116
117  /*Branch processing at completion of FINS response reception*/
118  void     cmd0b()
119  {
120     char   *adr;
121
122     buf = mbuf;
123     length = *cmd0bhdr;                      /*Number of ioctl transmission digits*/
124     adr = cmd0bhdr;
125     strncpy(buf,++adr,2);                    /*Command*/
126     buf+=2;
127     *(long*)buf = (long)(farptr)imp0badr[0];  /*Reception-complete entry address*/
```

```
128     ioctl();
129     if(outregs.x.ax)
130           printf("Branch at completion of response reception  Code
error:%d/n",outregs.x.ax);
131  }
132  #pragma check_stack(off)
133  void     far jmp0ba()
134  {
135     c_inc();
136  }
137  #pragma check_stack(on)
138
139  /*User timer service setting processing*/
140  void     cmd11()
141  {
142     char   *adr;
143
144     buf = mbuf;
145     length = *cmd11hdr;                        /*Number of ioctl transmission digits*/
146     adr = cmd11hdr;
147     strncpy(buf,++adr,2);                      /*Command*/
148     buf+=2;
149     *(long*)buf = (long)(farptr)imp11adr[0];   /*Timer interrupt entry address*/
150     buf += 4;
151     *((ushort*)buf) = 200;                     /*Timer value*/
152     ioctl();
153     if(outregs.x.ax)
154           printf("User timer set Code error:%d/n",outregs.x.ax);
155  }
156  #pragma check_stack(off)
157  void     far jmp11a()
158  {
159     f_set();
160  }
161  #pragma check_stack(on)
162
163  void     ioctl()
164  {
165     inregs.x.ax = 0x4403;/*SEND*/
166     inregs.x.bx = fd;
167     inregs.x.cx = length;
168     inregs.x.dx = (short)mbuf;
169     intdos(&inregs, &outregs);
170     if(outregs.x.cflag!=0){
171           printf("MS-DOS system error*/n");
172           inregs.h.ah = 0x3e;    /*Closes file.*/
173           inregs.x.bx = fd;
174           intdos(&inregs, &outregs); /*int 21h*/
175           exit(2);
176    }
177  }
178
179  void     disp_recv(j)
180  intj;
181  {
182     register int  i,k;
```

```
183
184     printf("Number of bytes of real data received = %d/n/n",j);
185     printf("/t/t0 1 2 3 4 5 6 7 8 9 a b c d e f /n");
186     printf("Reception data = /t");
187     for(i=0,k=0;i<j;i++){
188           printf("%02x",recvbuf[i]);
189           k++;
190           if(k==16){
191                   printf("/n/t/t");
192                   k = 0;
193           }
194     }
195     printf("/n");
196   }
197   void timetime(c)
198   {
199     double nowtime;
200     double aftertime;
201     moritime(&nowtime);
202     moritime(&aftertime);
203     for(;(aftertime-nowtime)<=c;)
204           moritime(&aftertime);
205   }
206   void moritime(t)
207   double   *t;
208   {
209     inregs2.h.ah=0;
210     int86(0x1a,&inregs2,&outregs2);
211     *t=(double)outregs2.x.dx;
212   }
```

```
1     /*************************************************/
2     /* CPU Bus Driver                                */
3     /*                                               */
4     /* Event Reception Sample Program                */
5     /*************************************************/
6     #include        <dos.h>
7     #include        <stdio.h>
8
9     typedef unsigned char uchar;
10    typedef unsigned short ushort;
11    typedef void far *farptr;
12
13    union     REGS   inregs,outregs;          /*I/O register structure*/
14    int       fd;                             /*File handle*/
15    char      length;                         /*Number of ioctl transmission digits*/
16    uchar     mbuf[15],*buf;                  /*ioctl buffer*/
17    uchar     recvbuf[2048];                  /*Reception buffer*/
18    char      cmd05hdr[] = {0xa,0x05,0x00}; /*Transmits event/FINS response.*/
19    char      cmd06hdr[] = {0xa,0x06,0x00}; /*Receives event/FINS command.*/
20    char      cmd0ahdr[] = {0x6,0x0a,0x00}; /*Branches at completion of event/FINS
reception.*/
21    char      cmd0chdr[] = {0x4,0x0c,0x00}; /*Sets event timeout value.*/
22    char      cmd11hdr[] = {0x8,0x11,0x00}; /*Registers time-up branch entry address.*/
23
24    /*Assembler function declarations*/
25    extern    int     f-sense(),c-sense();
```

```
26   extern   void   f-set(),f-cls();
27   extern   void   c-inc(),c-dec(),c-cls()
28   void cmd05(),cmd06(),cmd0a(),cmd0c(),cmd11(),
29   void far jmp0aa(),far jmp11a();
30   void disp_recv(),ioctl();
31
32   void far *jmp0aadr[] = {jmp0aa};       /*Entry address at completion of reception*/
33   void far *jmp11adr[] = {jmp11a};       /*Time-up branch entry address*/
34
35   void     main()
36   {
37      inregs.h.ah = 0x3d;         /*Opens file.*/
38      inregs.h.al = 0x02;         /*read/write*/
39      inregs.x.dx = (short)"CVIF";
40      intdos(&inregs, &outregs); /*int 21h*/
41      if(outregs.x.cflag !=0){
42            printf("\n Driver not loaded.errcode = %x/n"outregs.x.ax);
43            exit(1);
44      }
45      else{
46            fd = outregs.x.ax;      /*Saves file handle.*/
47            c_cls();
48            f_cls();
49            cmd0a();                /*Branches at completion of event/FINS reception.*/
50      cmd0c();                      /*Sets event timeout value.*/
51            cmd11();                /*Registers time-up branch entry address.*/
52            do{
53                  if(c_sense()){
54                        c_dec();
55                        cmd06();          /*Receives event/FINS command.*/
56                        cmd05();          /*Transmits event/FINS response.*/
57                  }
58            }
59            while(!f_sense()||c_sense());
60
61            inregs.h.ah = 0x3e;         /*Closes file.*/
62            inregs.x.bx = fd;
63            intdos(&inregs, &outregs); /*int 21h*/
64      }
65   }
66
67   /*Event service and FINS response transmission processing*/
68   void     cmd05()
69   {
70      char   *adr;
71      static char  finsdata[] = {
72   /*       ICF,RSV,GCNT,DNA,DA1,DA2,SNA,SA1,SA2,SID*/
73            0,  0,  0,   0,  0,  0,  0,  0,  0,  0,
74   /*       MRC, SRC, MRES,SRES,RES-DATA*/
75            0x11,0x12,0,   0,   3,4};
76
77      buf = mbuf;
78      length = *cmd05hdr;             /*Number of ioctl transmission digits*/
79      adr = cmd05hdr;
80      strncpy(buf,++adr,2);           /*Command*/
81      buf+=2;
```

```
82      finsdata[0] = recvbuf[0];       /*ICF*/
83      finsdata[1] = recvbuf[6];       /*RSV*/
84      finsdata[3] = recvbuf[6];       /*DNA*/
85      finsdata[4] = recvbuf[7];       /*DA1*/
86      finsdata[5] = recvbuf[8];       /*DA2*/
87      finsdata[9] = recvbuf[9];       /*SID*/
88      finsdata[10] = recvbuf[10];     /*MRC*/
89      finsdata[11] = recvbuf[11];     /*SRC*/
90      *(long*)buf = (long)(farptr)finsdata;  /*Buffer address*/
91      buf += 4;
92     *((ushort*)buf) = 16;            /*Number of bytes requested for transmission*/
93      ioctl();
94      if(outregs.x.ax)
95            printf("Response transmission Code error:%d/n",outregs.x.ax);
96   }
97
98   /*FINS command reception request*/
99   void     cmd06()
100  {
101     char   *adr;
102
103     buf = mbuf;
104     length = *cmd06hdr;                  /*Number of ioctl transmission digits*/
105     adr = cmd06hdr;
106     strncpy(buf,++adr,2);                /*Command*/
107     buf+=2;
108     *(long*)buf = (long)(farptr)recvbuf;  /*Reception buffer address*/
109     buf += 4;
110    *((ushort*)buf) = 1000;              /*Number of bytes requested for reception*/
111     ioctl();
112     if(!outregs.x.ax)
113           disp_recv(*(short*)(mbuf+8));   /*Displays data received.*/
114     else
115           printf("Command reception  Code error:%d/n",outregs.x.ax);
116  }
117
118  /*Branch processing at completion of FINS command reception*/
119  void     cmd0a()
120  {
121     char   *adr;
122
123     buf = mbuf;
124     length = *cmd0ahdr;                     /*Number of ioctl transmission digits*/
125     adr = cmd0ahdr;
126     strncpy(buf,++adr,2);                   /*Command*/
127     buf+=2;
128     *(long*)buf = (long)(farptr)imp0aadr[0];  /*Reception-complete entry address*/
129     ioctl();
130     if(outregs.x.ax)
131           printf("Branch at completion of command reception  Code
error:%d/n",outregs.x.ax);
132  }
133  #pragma check_stack(off)
134  void far jmp0aa()
135  {
136  c_inc();
```

```
137  }
138  #pragma check_stack(on)
139
140  /*Event timer setting processing*/
141  void     cmd0c()
142  {
143     char   *adr;
144
145     buf = mbuf;
146     length = *cmd0chdr;                  /*ioctlNumber of ioctl transmission digits*/
147     adr = cmd0chdr;
148     strncpy(buf,++adr,2);                /*Command*/
149     buf+=2;
150     *((ushort*)buf) = 100;              /*Timeout value*/
151     ioctl();
152     if(outregs.x.ax)
153          printf("Timer value setting Code error:%d/n",outregs.x.ax);
154  }
155
156  /*User timer service setting processing*/
157  void     cmd11()
158  {
159     char   *adr;
160
161     buf = mbuf;
162     length = *cmd11hdr;                  /*Number of ioctl transmission digits*/
163     adr = cmd11hdr;
164     strncpy(buf,++adr,2);                /*Command*/
165     buf+=2;
166     *(long*)buf = (long)(farptr)imp11adr[0];   /*Timer interrupt entry address*/
167     buf += 4;
168     *((ushort*)buf) = 100;              /*Timer value*/
169     ioctl();
170     if(outregs.x.ax)
171          printf("User timer setting Code error:%d/n",outregs.x.ax);
172  }
173  #pragma check_stack(off)
174  void     far jmp11a()
175  {
176     f_set();
177  }
178  #pragma check_stack(on)
179
180  void     ioctl()
181  {
182     register int  i;
183
184     inregs.x.ax = 0x4403;            /*SEND*/
185     inregs.x.bx = fd;
186     inregs.x.cx = length;
187     inregs.x.dx = (short)mbuf;
188     intdos(&inregs, &outregs);
189     if(outregs.x.cflag!=0){
190          printf("MS-DOS system error/n");
191          inregs.h.ah = 0x3e;         /*Closes file*/
192          inregs.x.bx = fd;
```

```
193           intdos(&inregs, &outregs); /*int 21h*/
194           exit(2);
195     }
196   }
197
198   void      disp_recv(j)
199   intj;
200   {
201      register int  i,k;
202
203      printf("Number of bytes of real data received = %d/n/n",j);
204      printf("/t/t0 1 2 3 4 5 6 7 8 9 a b c d e f/n");
205      printf("Reception data = /t");
206      for(i=0,k=0;i<j;i++){
207           printf("%02x",recvbuf[i]);
208           k++;
209           if(k==16){
210                   printf("/n/t/t");
211                   k = 0;
212           }
213     }
214     printf("/n");
215   }
```

```
1     /*************************************************/
2     /* CPU Bus Driver                                */
3     /*                                               */
4     /* CPU Bus Link Service Sample Program           */
5     /*************************************************/
6     #include        <dos.h>
7     #include        <stdio.h>
8
9     typedef void far *farptr;
10
11    union     REGS    inregs,outregs;         /*I/O register structure*/
12    int       fd;                             /*File handle*/
13    char      length;                         /*Number of IOCTL transmission digits*/
14    static    char    buf[16],databuf[512];   /*Data buffer for IOCTL*/
15
16    static    char    driver_id[] = "CVIF";           /*Driver name*/
17    static    char    d_rsvread[] = {6,0x0e,0x00};    /*Reads reserved information.*/
18    static    char    d_lnkread[] = {12,0x0f,0x00};   /*Reads link area.*/
19    static    char    d_lnkwrit[] = {12,0x10,0x00};   /*Writes link area.*/
20    static    char    d_gokiadr[] = {4,0x13,0x00};    /*Inquires regarding unit address.*/
21
22    void      errclose();
23
24    void
25    main()
26    {
27       int     i;
28       char    goki;
29       register short *bufp;
30
31       inregs.h.ah = 0x3d;          /*Opens driver.*/
32       inregs.h.al = 0x02;          /*Read/write mode*/
33       inregs.x.dx = (short)driver_id;
```

```
34      intdos(&inregs, &outregs); /*int 21h*/
35      if(outregs.x.cflag){
36            printf("Driver not loaded./n");
37            printf("/tError code = 0x%x/n"outregs.x.ax);
38            exit(1);
39      }
40      fd = outregs.x.ax;          /*Acquires file handle.*/
41
42      strncpy(buf,&d_rsvread[1],2);           /*Reads information reserved for system.*/
43      length = d_rsvread[0];
44      bufp = (short*)&buf[2];
45       *(long*)bufp = (long)(farptr)databuf;   /*Sets reception buffer address.*/
46      if(ioct())
47            errclose();
48
49      printf("System-reserved information data/n");
50      bufp = (short*)databuf;                 /*Displays data received.*/
51      for(i=8;-i>=0;)
52            printf("0x%x",*bufp++);
53      printf("/n");
54
55      strncpy(buf,&d_lnkwrit[1],2);           /*Writes CPU bus link area.*/
56      length = d_lnkwrit[0];
57      bufp = (short*)&buf[2];
58      *bufp++=0;                              /*Sets beginning word for writing.*/
59      strcpy(databuf,"0123456789abcdef");
60       *(long*)bufp = (long)(farptr)databuf;   /*Sets buffer for written data.*/
61      bufp+=2;
62      *bufp=8;                              /*Sets number of words requested for writing.*/
63      if(ioct())
64            errclose();
65      printf("The number of words actually written is %d./n",
*(short*)&buf[10]);
66
67      strncpy(buf,&d_gokiadr[1],2);           /*Inquires regarding unit address.*/
68      length = d_gokiadr[0];
69      if(ioct())
70            errclose();
71      goki=buf[2];                            /*for Special I/O Unit*/
72      printf("The unit address is %d./n",goki);
73
74      strncpy(buf,&d_lnkread[1],2);           /*Reads CPU bus link area.*/
75      length = d_lnkread[0];
76      bufp = (short*)&buf[2];
77      *bufp++=goki*8+128;                     /*Sets beginning word for reading.*/
78      *(long*)bufp = (long)(farptr)databuf;   /*Sets reception buffer address.*/
79      bufp+=2;
80      *bufp=8;                             /*Sets number of words requested for reading.*/
81      if(ioct())
82            errclose();
83
84      printf("Unit number data read/n");
85      bufp = (short*)databuf;                 /*Displays data received.*/
86      for(i=*(short*)&buf[10];-i>=0;)
87            printf("0x%x",*bufp++);
88      printf("/n");
```

```
89
90      inregs.h.ah = 0x3e;         /*Closes driver.*/
91      inregs.x.bx = fd;
92      intdos(&inregs, &outregs); /*int 21h*/
93   }
94
95   int
96   ioctl()
97   {
98      inregs.x.ax = 0x4403;          /*I/O request*/
99      inregs.x.bx = fd;
100     inregs.x.cx = length;          /*Sets number of bytes to receive.*/
101     inregs.x.dx = (short)buf;      /*Sets parameter buffer address.*/
102     intdos(&inregs, &outregs);
103     if(outregs.x.cflag||outregs.x.ax)
104           return(1);
105     return(0);
106  }
107
108  void
109  errclose()
110  {
111     printf("IOCTL error/n");
112     printf("/tcmd=0x%x carry=0x%x AX=0x%x/n",
113                       buf[0],outregs.x.cflag,outregs.x.ax);
114     inregs.h.ah = 0x3e;         /*Closes driver.*/
115     inregs.x.bx = fd;
116     intdos(&inregs, &outregs); /*int 21h*/
117     exit(2);
118  }
```

```
;*************************************************************
;* S Bus Interface Communications Driver                     *
;*                                                           *
;* Sample program for branch at completion and user timer    *
;*************************************************************
_TEXT segment byte public 'code'
      assume cs:_TEXT
      assume ds:_TEXT
      public _f_set,_f_sense,_f_cls,_endflg
      public _c_inc,_c_dec,_c_sense,_c_cls,_cnt
;***********************************
;*     Flag set processing         *
;***********************************
_f_set proc near
      mov          cs:_endflg,1
      ret
_f_set endp
;***********************************
;*     Flag sense processing       *
;***********************************
_f_sense proc near
      mov          ax,cs:_endflg
      ret
_f_sense endp
;***********************************
;*     Flag clear processing       *
```

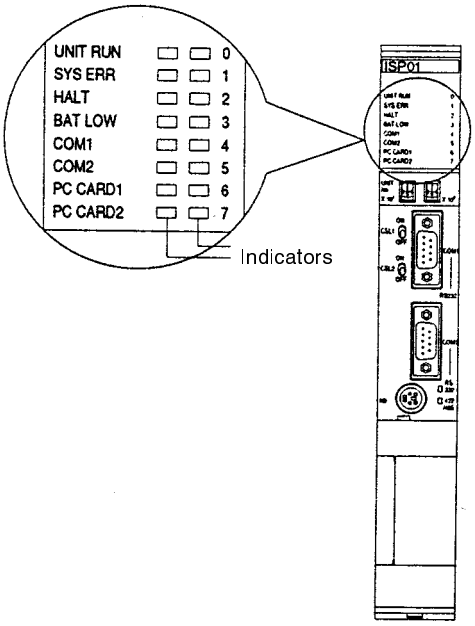
```
;************************************
_f_cls proc near
      xor          ax,ax
      mov          cs:_endflg, ax
      ret
_f_cls endp
;************************************
;*     Counter increment processing  *
;************************************
_c_inc proc near
      inc          cs:_cnt
      ret
_c_inc endp
;************************************
;*     Counter decrement processing  *
;************************************
_c_dec proc near
      dec          cs:_cnt
      ret
_c_dec endp
;************************************
;*     Counter sense processing      *
;************************************
_c_sense proc near
      mov          ax,cs:_cnt
      ret
_c_sense endp
;************************************
;*     Counter clear processing      *
;************************************
_c_cls proc near
      xor          ax,ax
      mov          cs:_cnt, ax
      ret
_c_cls endp
;************************************
;*     Flag and counter              *
;************************************
_endflg dw 0
_cnt dw 0

_TEXT ends
      end
```

SECTION 9
Error Processing and Maintenance

This section explains how to process errors and change batteries.

9-1 Indicators

The meanings of the various indicators on the front panel of the ISA Control Unit are explained here. Check these indicators during operation and take the appropriate countermeasures when an error occurs.



UNIT RUN

Status	Meaning and countermeasures
Lit	Normal operation as a CPU Bus Unit. (CPU Bus Driver installed.)
Not lit	If this indicator does not light within a short time after the Unit is powered up, it means that the Unit is not functioning as a Unit of the Programmable Controller. Carry out the following measures: • Check the SYS ERR indicator. • Check whether SBUS.SYS is installed in CONFIG.SYS. (Refer to *5-5 CPU Bus Driver Installation and Startup*.) • Check to be sure that the PC's I/O table has been set correctly. • Check to be sure that the Unit number is not overlapping with the Unit number of another CPU Special I/O Unit.

SYS ERR

Status	Meaning and countermeasures
Lit	A system error was discovered by the BIOS diagnosis or an error occurred in the CPU bus. Check the user indicators for the content of the error, or use the ERRLOG command to look at the error log. After correcting the cause of the error, restart by powering up the system again.
Not lit	The system is normal.

HALT

Status	Meaning and countermeasures
Lit	An I/O point error or memory parity error occurred. Replace (or remove) the board installed in the ISA Bus Expander and then restart the system. If the HALT indicator is still lit after that, replace the Unit.
Not lit	There is no hardware error.

BAT LOW

Status	Meaning and countermeasures
Lit	The voltage is low for the battery backing up the real time clock and CMOS memory. Remove the battery cover and replace the battery. (Refer to *9-7 Changing the Battery*.)
Not lit	The battery for the hardware is normal.

COM1, COM2

Status	Meaning and countermeasures
Blinking	Serial communications port 1 (COM1) or serial communications port 2 (COM2) is being accessed.
Not lit	The port is not being accessed.

PC CARD1, PC CARD2

Status	Meaning and countermeasures
Blinking	Card in PC card socket 1 or PC card socket 2 is being accessed, or the built-in drive F is being accessed.
Not lit	The card is not being accessed.

When SYS ERR is Lit

When SYS ERR is lit, the contents of the error can be understood from the number of the indicator that is lit.

Indicator lit	Meaning and countermeasures
0	Hardware error. Restart the system. If indicator is still lit after restarting, replace the Unit.
1	Interval timer error. (Not necessarily a problem in actual operation.) If this interferes with user program execution, replace the Unit.
2	CMOS check error. Replace the battery and then restart the system.
3	Not used.
4	Cyclic area memory error (parity error) or cyclic area access rights error. Restart the PC and the Unit.
5	Memory access error. Restart the system.
6	The actual Unit number does not match the Unit number set in the CPU Unit's I/O table, or Unit numbers overlap, or an illegal Unit (i.e., a number over 15) is used. Set the I/O table and Unit numbers correctly, and then restart the PC and the Unit.
7	Not used.

Note If a particular error continues to be generated even after the above countermeasures have been taken, it may mean that the hardware is malfunctioning. In that case, contact your OMRON representative.

9-2 Error Logs

The error log is the part of the built-in memory used to record any errors that occur when the Unit is started or during execution of the program.

9-2-1 Recording Error Records

Error records will be recorded in the following two cases.

BIOS Startup

The results of the self-diagnosis when BIOS is started will be automatically recorded as an error record. The following errors will be recorded.

- Battery errors
- System configuration errors
- Other hardware errors

Application Execution

From an application program, function calls can be used to record an error record. Consider recording the following kinds of errors.

- I/O errors
- CPU errors at the PC
- Other program execution errors

9-2-2 Error Log Recording Area

The error log is recorded in the CMOS memory area. Each error record takes up eight bytes and seven records can be logged.

9-2-3 Record Format

Error records are recorded in FIFO (First In, First Out) format in the CMOS memory area. If the memory area becomes full, the data that was recorded earliest is deleted as new data is added.

When more than seven errors are logged in the ISA Control Unit, the oldest records are deleted as new errors are logged.

9-2-4 Reading Error Records

You can display the error log by inputting the ERRLOG command at the MS-DOS prompt. The ERRLOG command is stored in the ROM disk. While the MS-DOS prompt is being displayed, enter `errlog` and then hit the Enter key. The following display will then appear on the screen.

	DATE	TIME	TYPE	CODE
No.00	11-04-94	14:20:56	02	03
No.01	11-21-94	15:10:58	02	06

Date of occurrence | Time of occurrence | Type of error | Error code

9-3 Writing Error Logs

This section will explain how to write error records from an application program.

9-3-1 Procedure

The procedure for writing an error record is as follows:

1, 2, 3...

1. Create in the memory a buffer for writing.
2. Store in the buffer the error record data that is to be written.
3. Execute a function call by means of a software interrupt, and write the error record.

9-3-2 Function Calls

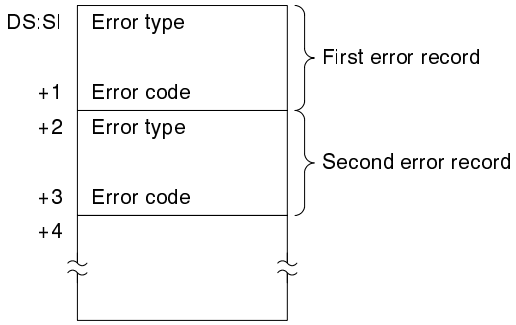
Error codes can be written by using software interrupt INT 15H to execute a function call.

- **Software Interrupt Number:** INT 15H
- **Input Parameters**
  AH = 21H: Function number
  AL = 01H: Write designation
  CX = Number of error records written
  DS:SI = Error record buffer address
- **Output Parameters**
  CF = Error flag

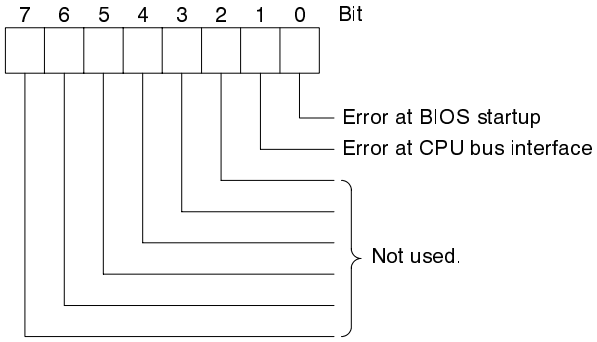
When a function call is executed with the above settings, the error record data stored in the designated error record buffer will be written, with date and time information added, in the specified number of error records.

9-3-3 Error Log Information Format

Error log contents can be written as shown below to the area designated as the error log buffer.



- **Type of Error**
  The code that distinguishes error types is recorded by turning ON and OFF the bits in one byte of data.



- **Error Code**
  The error code is recorded with one byte of data. It will differ according to the error type code. For details on the error code for error type 01H (when BIOS is started up), refer to *9-5 Errors at BIOS Startup*.

9-4 Reading Error Logs

This section will explain how to read error records from an application program.

9-4-1 Reading Procedure

The procedure for reading an error record is as follows:

1, 2, 3...

1. Create in memory the buffer for reading.
2. Execute a function call by means of a software interrupt, and store the error record data in the buffer.
3. Read the error record data stored in the buffer.

9-4-2 Function Calls

Error codes can be read by using software interrupt INT15H to execute a function call. The reading will be executed in order, beginning with the oldest error record. There are two methods for reading error records. One is to read new error records one by one, and the other is to repeatedly read the same error record.

Reading New Error Records

When this method is used, the error record pointer is refreshed moves from the newly read data to the next data written. Therefore, data read by this method is not read when the next function call is executed.

- **Software Interrupt Number:** INT15H
- **Input Parameters**
  AH = 21H: Function number
  AL = 00H: Read designation
  CX = Number of error records read
  ES : DI = Error record buffer address
  When this operation is used, a buffer of CX × 8 bytes must be protected.
- **Output Parameters**
  [When input parameter CX = 0]
  CX = Present number of error records
  [When input parameter CX ≠ 0]
  CX = Number of error records read
  CF = Error flag

When a function call is executed with the above settings, the error record information, with date and time information added, will be read for the designated number of error records and stored in the buffer.

Reading the Same Record

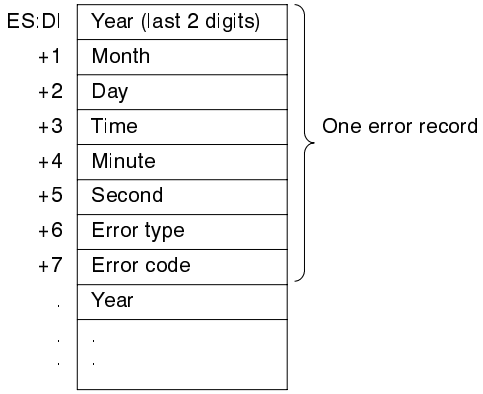
When this method is used, the error record pointer does not move. Therefore, data read by this method can also be read when the next read operation is executed.

- **Software Interrupt Number:** INT15H
- **Input Parameters**
  AH = 21H: Function number
  AL = 02H: Read designation
  CX = Number of error records read
  ES : DI = Error record buffer address
  When this operation is used, a buffer of CX × 8 bytes must be protected.
- **Output Parameters**
  [When input parameter CX = 0]
  CX = Present number of error records
  [When input parameter CX ≠ 0]
  CX = Number of error records read
  CF = Error flag

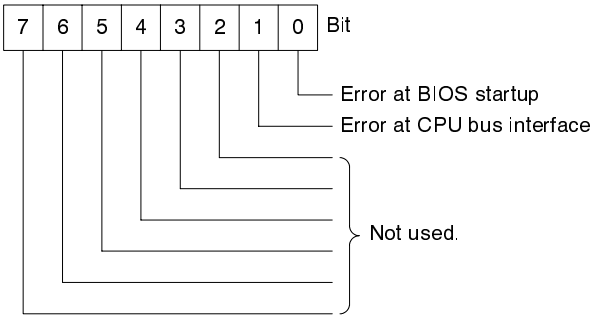
When a function call is executed with the above settings, the error record information, with date and time information added, will be read for the designated number of error records and stored in the buffer.

9-4-3 Error Log Information Format

The format of error record information stored in the area designated as the error record reading buffer is as shown below.



- **Type of Error**
  The code that distinguishes error types is recorded by turning ON and OFF the bits in one byte of data.



- **Error Code**
  The error code is recorded by one byte of data. It will differ according to the error type code. For details on the error code for error type O1H (when BIOS is started up), refer to *5-5 Errors at BIOS Startup*.

9-5 Errors at BIOS Startup

If an error should occur when the ISA Control Unit's BIOS is started, it will be recorded in an error record as an error code for error 01H. The meanings of error codes at that time are as shown in the following table. (For a list of CPU bus-related error types and error codes, refer to *Appendix D Error Codes*.)

Indicator 0: Hardware Error Restart the system. If this indicator still lights, then replace the Unit.

Indicator 1: Interval Timer Error This is not necessarily a problem in actual operation. If it interferes with user program execution, replace the Unit.

Indicator 2: CMOS Check Error Replace the battery and then restart the system.

Error code	Meaning	Indicator	Error log
00h	Exiting CSET_BFR_OPROM	Not used.	
01h	80286 register test in progress	––	No
02h	CMOS write/read test in progress or failure	0 (HALT)	No
03h	ROM BIOS checksum in progress or failure	0 (HALT)	No

Error code	Meaning	Indicator	Error log
04h	Programmable Interval Timer test in progress or failure	1	Yes
05h	DMA initialization in progress or failure	0 (HALT)	No
06h	DMA page register write/read test in progress or failure	0 (HALT)	No
07h	SUPERIO chip index register cannot be found or failure	Not used.	
08h	RAM refresh verification in progress or failure	0 (HALT)	No
09h	1st 64 K RAM test in progress	––	No
0Ah	1st 64 K RAM chip or data line failure, multi - bit	0 (HALT)	No
0Bh	1st 64 K RAM odd/even logic failure	0 (HALT)	No
0Ch	1st 64 K RAM address line failure	0 (HALT)	No
0Dh	1st 64 K RAM parity test in-progress or failure	0 (HALT)	No
0Eh	Fail-safe timer test in progress	0 (HALT)	No
0Fh	Software NMI port test in progress	0 (HALT)	No
10h	1st 64 K RAM chip or data line failure - bit 0	0 (HALT)	No
11h	1st 64 K RAM chip or data line failure - bit 1	0 (HALT)	No
12h	1st 64 K RAM chip or data line failure - bit 2	0 (HALT)	No
13h	1st 64 K RAM chip or data line failure - bit 3	0 (HALT)	No
14h	1st 64 K RAM chip or data line failure - bit 4	0 (HALT)	No
15h	1st 64 K RAM chip or data line failure - bit 5	0 (HALT)	No
16h	1st 64 K RAM chip or data line failure - bit 6	0 (HALT)	No
17h	1st 64 K RAM chip or data line failure - bit 7	0 (HALT)	No
18h	1st 64 K RAM chip or data line failure - bit 8	0 (HALT)	No
19h	1st 64 K RAM chip or data line failure - bit 9	0 (HALT)	No
1Ah	1st 64 K RAM chip or data line failure - bit A	0 (HALT)	No
1Bh	1st 64 K RAM chip or data line failure - bit B	0 (HALT)	No
1Ch	1st 64 K RAM chip or data line failure - bit C	0 (HALT)	No
1Dh	1st 64 K RAM chip or data line failure - bit D	0 (HALT)	No
1Eh	1st 64 K RAM chip or data line failure - bit E	0 (HALT)	No
1Fh	1st 64 K RAM chip or data line failure - bit F	0 (HALT)	No
20h	Slave DMA register in-progress or failure	0 (HALT)	No
21h	Master DMA register in-progress or failure	0 (HALT)	No
22h	Master interrupt mask register in-progress or failure	0 (HALT)	No
23h	Slave interrupt mask register in-progress or failure	0 (HALT)	No
25h	Interrupt vector loading in progress	––	No
27h	Keyboard controller test in-progress or failure	0 (HALT)	No
28h	CMOS power failure and checksum calculation in progress	2	No
29h	CMOS configuration validation in progress	––	No
2Bh	Screen memory test in-progress or failure	––	No
2Ch	Screen initialization in-progress or failure	––	No
2Dh	Screen retrace test in-progress or failure	––	No
2Eh	Search for video ROM in progress		
30h	Screen believed operable	––	No
30h	Screen believed running with video ROM	––	No
31h	Monochrome monitor believed operable	––	No
32h	Color monitor (40 column) believed operable	––	No
33h	Color monitor (80 column) believed operable	––	No

Error code	Meaning	Indicator	Error log
34h	Time tick interrupt test in-progress or failure	Not used.	
35h	Shutdown test in-progress or failure		
36h	Gate A20 failure		
37h	Unexpected interrupt in protected mode		
38h	RAM test in-progress or failure above address failure OFF		
3Ah	Interval timer channel 2 test in-progress or failure	Not used.	
3Bh	Time - of - Day clock test in-progress or failure		
3Ch	Serial port test in-progress or failure		
3Dh	Parallel port test in-progress or failure		
3Eh	Math coprocessor test in-progress or failure		
50h	Beginning of CSET_INIT	––	No
51h	Loading the RCM table	––	No
52h	Loading the FCM table, doing DMC	––	No
53h	Entering CSET_BFR_VDROM	––	No
54h	Entering CSET_BFR_SIZEMEM	––	No
55h	Entering CSET_ATF_MTEST	––	No
56h	Check CMOS config against actual	––	No
57h	Entering CSET_BFR_OPROM	Not used.	
59h	Base Memory Size error	––	No
70h	Interrupt failure	0 (HALT)	Yes

Note (HALT) indicates that the CPU stops by itself. The HALT indicator does not light in this case.

9-6 Troubleshooting

The following table shows the points to check when troubleshooting.

Symptom	Points to check
Nothing is displayed at the Console.	• Are the Console switch settings correct? (Refer to page 58) • Are the CRT and terminal software correctly connected? (Refer to page 69)
Erroneous characters appear on the Console display	• Are the communications settings correct? (Refer to page 59) • ANSI.SYS is required if a VGA board is used.
Memory card cannot be used.	• Is the driver installed for the card? (Refer to *5-6 PC Card Interface Driver Setup*) • Is the card formatted? (Refer to page 39) • Was an access made to the previous card when the card was replaced? (Refer to *5-6 PC Card Interface Driver Setup*)
UNIT RUN (indicator) is not lit.	• Is SBUS.SYS installed? (Refer to *5-5 CPU Bus Driver Installation and Startup*) • Is the system switch (interrupt CPU address) set correctly? (Refer to page 59)
Expansion board cannot be used.	• Does the address overlap? (Refer to page 59) • Do interrupts overlap? (Refer to page 59) • Does I/O overlap? (Refer to page 59) • Is the driver installed for the expansion board that is being used? (Refer to the expansion board manual.)
Hard Disk Unit cannot be used.	• Is the hardware configuration (SETUP) set correctly? (Refer to page 84) • Has the HDISK command been set? (Refer to page 85) • Has the hard disk been formatted? (Refer to page 86)

9-7 Changing the Battery

The ISA Control Unit has a built-in battery for backing up the real time clock and the CMOS memory. The method for changing this battery is explained here.

Battery Life

The maximum service life of the battery is five years, regardless of how long the Unit is powered up. When the power to the Unit is OFF, the battery service life is greatly affected by the ambient temperature. The service life is shortened as the temperature goes up.

Replacement Period

The BAT LOW indicator lights when the battery voltage gets low. Replace the battery within one week from when the indicator lights.

Battery Type

Replace the battery with a C500-BAT08 Battery Set (OMRON).

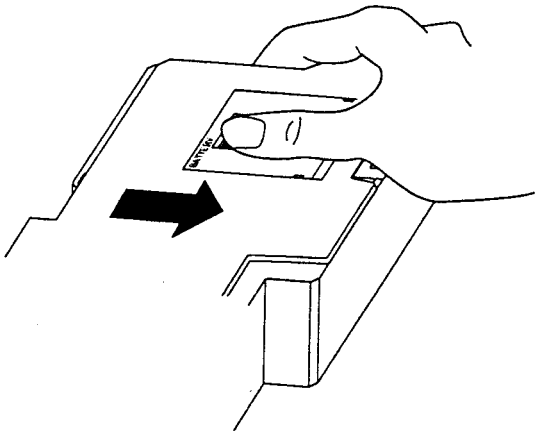
Procedure

Replace the battery according to the following procedure.

1, 2, 3...

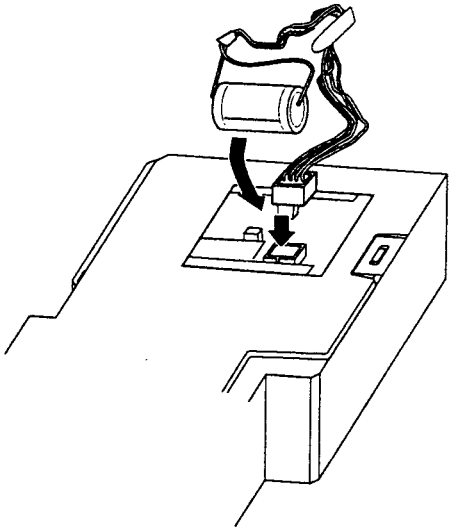
1. Turn OFF the power supply to the PC.

2. Remove the battery cover on the side of the Unit.



Note Once the old battery has been disconnected, the new battery must be connected within five minutes.

3. Remove the battery connectors and take the old battery out from the battery holder.
4. Connect the new battery to the connectors.
5. Insert the battery into the holder.



6. Put the battery cover back in place.

⚠ **WARNING** Batteries can leak or explode. Absolutely do not short across their plus and minus terminals, overcharge them, take them apart, heat them, or throw them into a fire.

Appendix A
Standard Models

Name	Specifications		Model number
ISA Bus Expander	For ISA board expansion.		CV500-ISX01
Hard Disk Unit	Capacity of 80M bytes.		CV500-HDD11
Extension Cable	Required for the Hard Disk Unit.		CV500-CN116
Memory Card	RAM type	64K bytes	HMC-ES641
		128K bytes	HMC-ES151
		256K bytes	HMC-ES251
		512K bytes	HMC-ES551
Memory Card Battery			HMC-BAT01
Battery Set			C500-BAT08

Appendix B
Specifications

ISA Control Unit

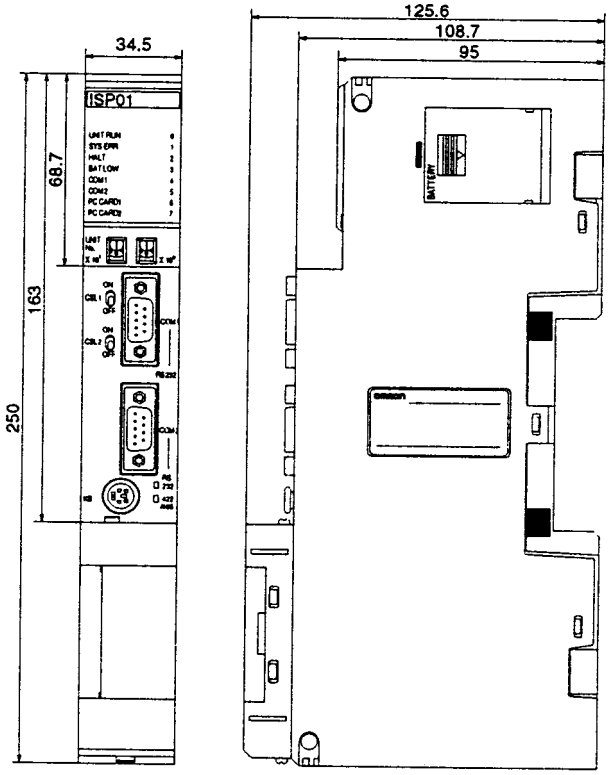
General Specifications

Item	Specifications
Current consumption	1.0 A at 5 VDC max. = Unit + PC card output current (Icard)
PC card output current (Icard)	I_{5V} (1 slot) ≦ 0.5 A, I_{12V} (1 slot) ≦ 0.1 A Where Icard = I_{5V} (2 slots)+3.4×I_{12V} (2 slots) ≦ 1.0A
Noise immunity	1,000 Vp-p; pulse width: 0.1 to 1 μs; rise time: 1-ns pulse
Vibration resistance	10 to 58 Hz; 0.075-mm amplitude, 58 to 150 Hz; acceleration: 9.8 m/s^2 in the X, Y, Z directions for 80 minutes each.
Shock resistance	147 m/s^2 in X, Y, Z directions for 3 times each.
Ambient temperature	0° to 55°C
Humidity	10% to 90% RH (without condensation)
Atmosphere	Must be free of corrosive gases, etc.
Storage temperature	–20° to 75 °C (without battery and PC card)
Weight	1 kg max.
Dimensions	CV Series, 1 slot (single size), 34.5 (W) x 250 (H) x 125.6 (D) mm

Note Specifications other than those shown here conform to the CV Series.

External Dimensions

All dimensions are in millimeters.



Performance Specifications

Item	Specifications
CPU	80386SX (25 MHz)
NPU (numeric value processor)	80387SX (25 MHz) (With CV500-ISP02 but not with CV500-ISP01)
Main memory	1M-byte DRAM (with parity check)
ROM	1M-byte EPROM (For storing BIOS, DOS, and system files)
Flash ROM	1M-byte Flash ROM (for storing user program); switching frequency: 1,000,000 (TYP.), 100,000 (min.)
RTC+CMOS RAM	Battery backup
Serial communications interface	16550-compatible x two channels RS-232C: D-sub, 9-pin, male, inch thread RS-232C or RS-422A/RS-485: D-sub, 9-pin, male, inch thread
PC card interface	PCMCIA 2.1 (+3.3-V card not supported). Type I or II × two slots or Type III × one slot. With ejector. (See note).
Keyboard interface	Conforms to IBM PC/AT; mini DIN; 6 pins.
ISA bus interface	Conforms to ISA bus; 100 pins (for ISA Sub-backplane); one slot.
CPU bus interface	Conforms to CV Series; 60 pins (for CV-series CPU Backplane); one slot.
Indicators	UNIT RUN, SYS ERR, HALT, BAT LOW, serial communications port being accessed, PC card being accessed, eight user-specified indicators, COM2 interface (RS-232C, RS-422A/485)
Switches	Two Console switches, three COM2 switches, system switch, two Unit number setting switches.
Battery	C500-BAT08 (OMRON)
Battery service life	5 years (at 25°C).
Diagnostic functions	• Battery voltage drop detection • Detection of 5-V power supply voltage drop in Unit • I/O point check, memory parity check • BIOS hardware check

Note When PC cards with I/O connectors are used, the connector width must be no more than 46 mm (i.e., 23 mm from the center of the PC card). If a connector is wider than that, the PC card slot cover will not fit.

Communications Specifications

RS-232C

Item	Specifications
UART specifications	16550-compatible (16 bytes, FIFO built-in)
Transfer format	Full-duplex
Synchronicity method	Start-stop
Connection method	1:1
Baud rate	110, 150, 300, 600, 1,200, 2,400, 4,800, 9,600, or 19,200 bps.
Data format	Start bits: 1/1.5/2 Data bits: 5/6/7/8 Parity bits: 1 (even or odd)/None Stop bits: 1/2
Transmission distance	15 m max.
Connector specifications	D-sub connector, 9 pins, inch thread

RS-422A/RS-485

Item	Specifications
UART specifications	16550-compatible (16 bytes, FIFO built-in)
Transfer format	Full-duplex (RS-422A); half-duplex (RS-485)
Synchronicity method	Start-stop
Connection method	1:1 or 1:N (RS-422A) N:N (RS-485)
Baud rate	110, 150, 300, 600, 1,200, 2,400, 4,800, 9,600, or 19,200 bps. (Supports 115.2 Kbps when 16,550 UART is operated independently.)
Data format	Start bits: 1/1.5/2 Data bits: 5/6/7/8 Parity bits: 1 (even or odd)/None Stop bits: 1/2
Transmission distance	500 m max.
Connector specifications	D-sub connector, 9 pins, inch thread
Applicable connectors	Socket: XM2D-0901 (OMRON) or equivalent Hood: XM2S-0913 (OMRON) or equivalent
Recommended cables	AWG28x5P IFVV-SB (Fujikura) CO-MA-VV-SB 5PxAWG28 (Hitachi)

Indicators

Indicator	Color	Meaning when lit
UNIT RUN	Green	Unit is operating.
SYS ERR	Red	Error occurred in BIOS diagnosis or CPU bus error occurred.
HALT	Red	NMI occurred (due to I/O point error or memory parity error).
BAT LOW	Red	Battery voltage is low.
COM1	Orange	Port 1 (COM1) is being accessed. (Indicator flashes during access.)
COM2	Orange	Port 2 (COM2) is being accessed. (Indicator flashes during access.)
PC CARD 1	Orange	PC card 1 is being accessed. (Indicator flashes during access.)
PC CARD 2	Orange	PC card 2 is being accessed. (Indicator flashes during access.)
0	Orange	User defined/Error occurred.
1	Orange	User defined/Error occurred.
2	Orange	User defined/Error occurred.
3	Orange	User defined
4	Orange	User defined/Error occurred.
5	Orange	User defined/Error occurred.
6	Orange	User defined/Error occurred.
7	Orange	User defined

Serial Communications Port (COM2) Indicator

Indicator	Color	Meaning when lit
RS-232C	Orange	Lit when COM2 SW1 and SW2 are both switched to RS-232C.
RS-422A/RS-485	Orange	Lit when COM2 SW1 and SW2 are both switched to RS-422A/RS-485.

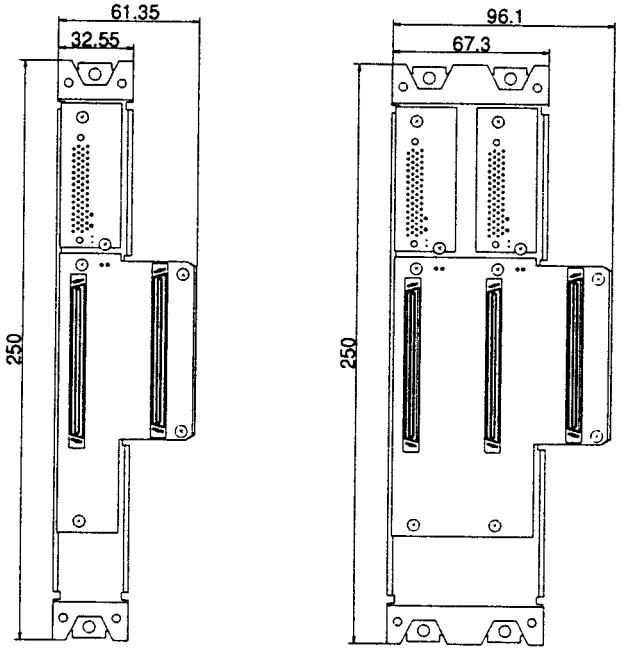
ISA Sub-backplane

Specifications

Item	Specifications
Noise immunity	1,000 Vp-p; pulse width: 0.1 to 1 μs; rise time: 1-ns pulse
Vibration resistance	10 to 58 Hz; 0.075-mm amplitude, 58 to 150 Hz; acceleration: 9.8 m/s^2 in the X, Y, Z directions for 80 minutes each.
Shock resistance	147 m/s^2 in X, Y, Z directions for 3 times each.
Ambient temperature	0° to 55°C
Humidity	10% to 90% RH (without condensation)
Atmosphere	Must be free of corrosive gases, etc.
Weight	CV500-ISB01: 250 g max.
	CV500-ISB02: 400 g max.
Dimensions	CV500-ISB01: 61.35 (W) x 250 (H) mm
	CV500-ISB02: 96.1 (W) x 250 (H) mm

External Dimensions

All dimensions are in millimeters.



Slot Specifications

Item	Specifications
ISA bus interface (except −5 V, ±12 V)	For connecting ISA Control Units: 1 slot
	For connecting ISA Bus Extenders and Hard Disk Units:
	CV500-ISB01: 1 slot
	CV500-ISB02: 2 slots
CPU bus interface (for +5 V)	CV500-ISB01: 1 slot
	CV500-ISB02: 2 slots

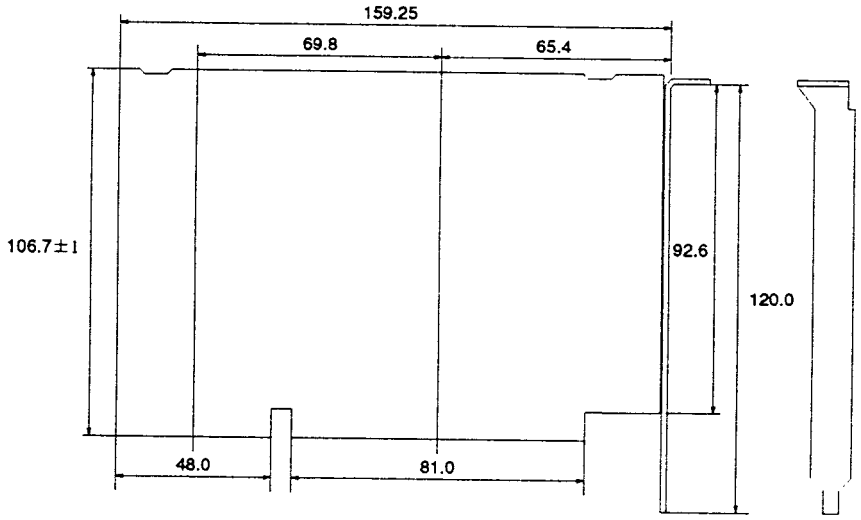
CPU Bus Adapter Board

Specifications

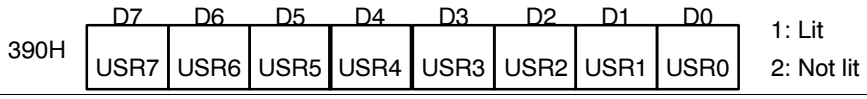
Item	Specifications
Power supply voltage	5 VDC
Power consumption	0.1 A
Ambient temperature	Operating: 5° to 45°C Storage: −20° to 60°C
Humidity	8% to 80% RH (without condensation)
Atmosphere	Must be free of corrosive gases, etc.
Weight	200 g max.
Dimensions	106.7 (H) x 159.25 (L) mm

External Dimensions

All dimensions are in millimeters.



Performance Specifications

<table>
<tr><th>Item</th><th>Specifications</th></tr>
<tr><td>CPU bus control</td><td>Cyclic service, event service, CPU bus link service</td></tr>
<tr><td>Rotary switches</td><td>Unit number setting switches (two rotary switches); setting range: 0 to 15</td></tr>
<tr><td>DIP switch</td><td>This sets the CPU bus shared RAM addresses, the interrupt level, and the CPU bus data path width.
CPU Bus Shared RAM Address Setting
<table>
<tr><th>Pin No. 1</th><th>Pin No. 2</th><th>Allocated addresses</th></tr>
<tr><td>OFF</td><td>OFF</td><td>0DC000h to 0DFFFFh</td></tr>
<tr><td>OFF</td><td>ON</td><td>0D8000h to 0DBFFFh</td></tr>
<tr><td>ON</td><td>OFF</td><td>0D4000h to 0D7FFFh</td></tr>
<tr><td>ON</td><td>ON</td><td>0D0000h to 0D3FFFh</td></tr>
</table>
Interrupt Level Setting
<table>
<tr><th>Pin No. 3</th><th>Interrupt level</th></tr>
<tr><td>ON</td><td>IRQ10</td></tr>
<tr><td>OFF</td><td>IRQ11</td></tr>
</table>
CPU Bus Data Path Width
<table>
<tr><th>Pin No. 4</th><th>Data path width</th></tr>
<tr><td>ON</td><td>8 bits</td></tr>
<tr><td>OFF</td><td>16 bits</td></tr>
</table></td></tr>
<tr><td>I/O register</td><td>Uses addresses 390 to 394 (HEX). Can light CPU Bus Expander indicators. (See Indicator Register in Appendix G Memory Configuration.)
 
</td></tr>
<tr><td>Interface</td><td>CV-series CPU expansion bus
ISA bus</td></tr>
<tr><td>Connectors</td><td>CV-series CPU expansion bus: Half-pitch, 68 pins
ISA bus: Metal-plated board edge, 98 pins</td></tr>
</table>

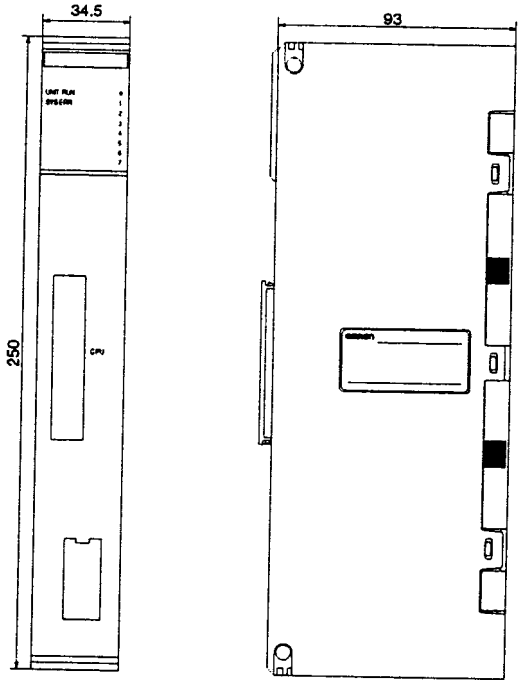
CPU Bus Expander

Specifications

Item	Specifications
Power supply voltage	5 VDC
Power consumption	0.1 A
Ambient temperature	Operating: 0° to 55°C Storage: −20° to 75°C
Humidity	10% to 90% RH (without condensation)
Atmosphere	Must be free of corrosive gases, etc.
Weight	400 g max.
Dimensions	34.5 (W) x 250 (H) x 93 (D) mm

External Dimensions of CPU Bus Expander

All dimensions are in millimeters.



Performance Specifications

Item	Specifications			
Display	**10 indicators**			
	Indicator	**Color**	**Meaning when lit**	**Remarks**
	UNIT RUN	Green	Unit is operating.	Shows UNIT RUN signal status.
	SYS ERR	Red	System error	Error occurred in CPU bus.
	0	Orange	User defined	Can be freely used as program I/O from host system. These use CPU Bus Adapter Board indicator register.
	1	Orange	User defined	
	2	Orange	User defined	
	3	Orange	User defined	
	4	Orange	User defined/Error	
	5	Orange	User defined/Error	
	6	Orange	User defined/Error	
	7	Orange	User defined	
Interface	CV-series CPU expansion bus			
	CV-series CPU bus			
Connectors	CV-series CPU expansion bus: Half-pitch, 68 pins			
	CV-series CPU bus: Half-pitch, 60 pins			

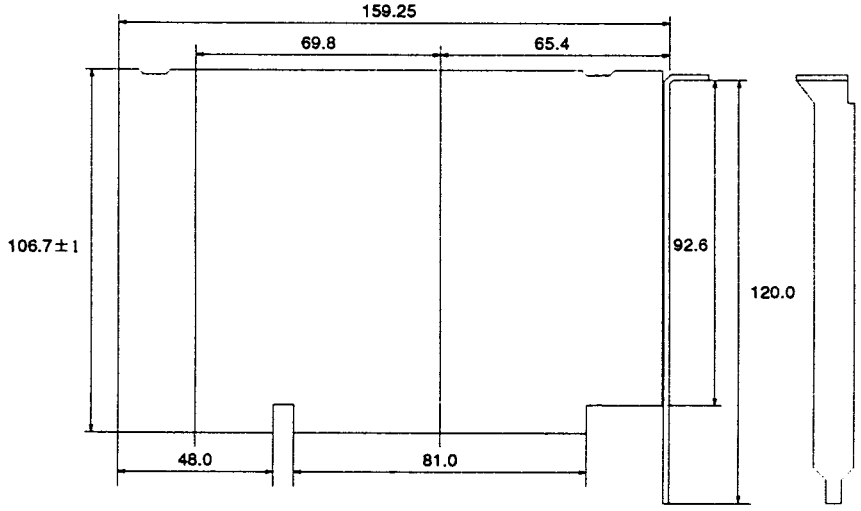
PC Card Interface Board

Specifications

Item	Specifications
PC card output current	Up to 0.75 A at 5 VDC/1 slot Up to 120 mA at 12 VDC/1 slot
Current consumption	0.02 A at 5 VDC+PC card output current 12 VDC (PC card output current)
Ambient temperature	Operating: 5° to 45°C Storage: –20° to 60°C
Humidity	8% to 80% RH (without condensation)
Atmosphere	Must be free of corrosive gases, etc.
Weight	200 g max.
Dimensions	106.7 (H) x 159.25 (L) mm

External Dimensions

All dimensions are in millimeters.



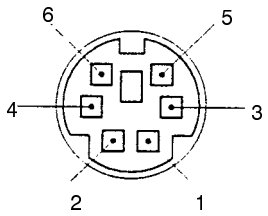
Performance Specifications

Item	Specifications
PC Card Controller	VADEM VG-468
Interface	PCMCIA 2.1 conforming (except 3.3-V cards) ISA bus
Connectors	PCMCIA: Two Type I/II or one Type III, 68 pins x 2 ISA bus: Metal-plated board edge, 98 pins

Appendix C
Connector Pin Arrangement

Keyboard Interface (PS/2 Type)

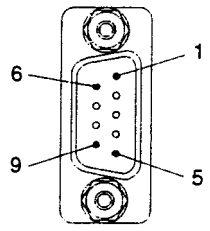
Connector Pins



Pin	Signal name
1	DATA
2	N.C.
3	GND
4	+5 VDC
5	CLK
6	N.C.

Serial Communications Interface

Connector Pins



D-sub, 9 pins (male)

Pin	Signal name (RS-232C)	Signal name (RS-422A/RS-485)
1	DCD (CD)	RxD+
2	RxD (RD)	CTS+
3	TxD (SD)	CTS–
4	DTR	RTS+
5	GND (SG)	TxD+
6	DSR	RxD–
7	RTS (RS)	N.C.
8	CTS (CS)	RTS–
9	RI	TxD–

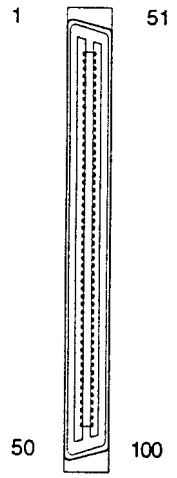
RS-232C signal pin arrangement: AT (OADG) specifications.

RS-422A/RS-485 signal pin arrangement: Original for this Unit.

ISA Expansion Bus Interface (At ISA Control Unit)

Note The asterisks (*) indicate negative logic.

Connector used: OMRON XH2B-0142



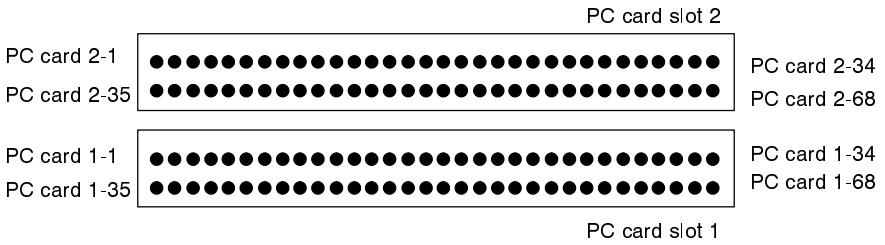
View at ISA Control Unit

Pin	Signal name	Pin	Signal name	Pin	Signal name	Pin	Signal name
1	*IOCHCK	26	SA5	51	GND	76	*DACK2
2	SD7	27	SA4	52	RESETDRV	77	T/C
3	SD6	28	SA3	53	+5 VDC	78	BALE
4	SD5	29	SA2	54	IRQ9	79	+5 VDC
5	SD4	30	SA1	55	N.C.	80	OSC
6	SD3	31	SA0	56	DRQ2	81	GND
7	SD2	32	GND	57	N.C.	82	N.C.
8	SD1	33	SBHE	58	OWS	83	*MEMCS16
9	SD0	34	LA23	59	N.C.	84	*I/O CS16
10	I/OCHRDY	35	LA22	60	GND	85	IRQ10
11	AEN	36	LA21	61	*SMEMW	86	IRQ11
12	SA19	37	LA20	62	*SMEMR	87	IRQ12
13	SA18	38	LA19	63	*IOW	88	IRQ15
14	SA17	39	LA18	64	*IOR	89	IRQ14
15	SA16	40	LA17	65	*DACK3	90	*DACK0
16	SA15	41	*MEMR	66	DRQ3	91	DRQ0
17	SA14	42	*MEMW	67	*DACK1	92	*DACK5
18	SA13	43	SD8	68	DRQ1	93	DRQ5
19	SA12	44	SD9	69	*REFRESH	94	*DACK6
20	SA11	45	SD10	70	CLK	95	DRQ6
21	SA10	46	SD11	71	IRQ7	96	*DACK7
22	SA9	47	SD12	72	IRQ6	97	DRQ7
23	SA8	48	SD13	73	IRQ5	98	+5 VDC
24	SA7	49	SD14	74	IRQ4	99	*MASTER
25	SA6	50	SD15	75	IRQ3	100	GND

PC Card Interface

Note The asterisks (*) indicate negative logic.

Connector Pins



Pin	Signal name	Pin	Signal name
1	GND	35	GND
2	D3	36	CD1*
3	D4	37	D11
4	D5	38	D12
5	D6	39	D13
6	D7	40	D14
7	CE1*	41	D15
8	A10	42	CE2*
9	E*	43	RFSH
10	A11	44	IORD*
11	A9	45	IOWR*
12	A8	46	A17
13	A13	47	A18
14	A14	48	A19
15	WE*/PGM	49	A20
16	RDY/BSY*/REQ*[I/O Card]	50	A21
17	VCC	51	VCC
18	VPP1	52	VPP2
19	A16	53	A22
20	A15	54	A23
21	A12	55	A24
22	A7	56	A25
23	A6	57	Reserved (N.C.)
24	A5	58	RESET
25	A4	59	WAIT*
26	A3	60	NPACK*/[I/O Card]
27	A2	61	REG*
28	A1	62	BVD2/SPKR*[I/O Card]
29	A0	63	BVD1/STSCHG*[I/O Card]
30	D0	64	D8
31	D1	65	D9
32	D2	66	D10
33	WP/IOCS16*[I/O Card]	67	CD2*
34	GND	68	GND

Appendix D
Error Codes

The table below lists the CPU bus-related error codes that might be read from the ISA Control Unit's error log when event servicing is used and "Error Log Read" is performed. Refer to *9-2 Error Logs* for more details.

Error type	Error code	Contents
02 or 32	01	Posted unit address does not match recognized unit address.
	02	Not used.
	03	Unit addresses duplicated.
	04	Improper unit address recognized.
	05	Hardware test unit recognized.
	06	Unit address could not be found in registered I/O table.
	07	Routing table error
	08	Routing table reading error
	09	Parity error: Cyclic service
	0A	Parity error: Event service
	0B	Parity error: CPU bus link service
	0C	Access rights return error: Cyclic service
	0D	Access rights return error: Event service
	0E	Access rights return error: CPU bus link service
	0F	CPU bus area memory access error
	10	Communications command discarded due to buffer overflow.
	11	Communications response discarded due to buffer overflow.
	12	Reception packet unit address error
	13	Programmable Controller routing process error (command)
	14	Programmable Controller routing process error (response)
	15	Reception packet size error
	16	Relay center routing process error
	17	Error in CPU bus of transmission destination unit.
	18	Transmission destination unit error
	19	Transmission destination unit could not be found.
	1A	Voltage drop: Battery no. 1 (clock/system RAM backup battery)
	1D	Improper interrupt occurred.
	1E	Buffer full: Could not transmit response.
	FF	POWER FAIL signal was received.

Appendix E
FINS Commands

FINS Commands for CV-series PCs

The following table shows the FINS commands addressable to CV-series PCs. Refer to the *FINS Commands Reference Manual (W227)* for more details.

Command code		PC mode				Name
		RUN	MONITOR	DEBUG	PROGRAM	
01	01	Valid	Valid	Valid	Valid	MEMORY AREA READ
	02	Valid	Valid	Valid	Valid	MEMORY AREA WRITE
	03	Valid	Valid	Valid	Valid	MEMORY AREA FILL
	04	Valid	Valid	Valid	Valid	MULTIPLE MEMORY AREA READ
	05	Valid	Valid	Valid	Valid	MEMORY AREA TRANSFER
02	01	Valid	Valid	Valid	Valid	PARAMETER AREA READ
	02	Valid	Valid	Valid	Valid	PARAMETER AREA WRITE
	03	Valid	Valid	Valid	Valid	PARAMETER AREA CLEAR
03	04	Valid	Valid	Valid	Valid	PROGRAM AREA PROTECT
	05	Valid	Valid	Valid	Valid	PROGRAM AREA PROTECT CLEAR
	06	Valid	Valid	Valid	Valid	PROGRAM AREA READ
	07	Not valid	Valid	Valid	Valid	PROGRAM AREA WRITE
	08	Not valid	Not valid	Not valid	Valid	PROGRAM AREA CLEAR
04	01	Valid	Valid	Valid	Valid	RUN
	02	Valid	Valid	Valid	Valid	STOP
05	01	Valid	Valid	Valid	Valid	CONTROLLER DATA READ
	02	Valid	Valid	Valid	Valid	CONNECTION DATA READ
06	01	Valid	Valid	Valid	Valid	CONTROLLER STATUS READ
	02	Valid	Valid	Not valid	Not valid	CYCLE TIME READ
07	01	Valid	Valid	Valid	Valid	CLOCK READ
	02	Valid	Valid	Valid	Valid	CLOCK WRITE
09	20	Valid	Valid	Valid	Valid	MESSAGE READ
						MESSAGE CLEAR
						FAL/FALS READ
0C	01	Valid	Valid	Valid	Valid	ACCESS RIGHT ACQUIRE
	02	Valid	Valid	Valid	Valid	ACCESS RIGHT FORCED ACQUIRE
	03	Valid	Valid	Valid	Valid	ACCESS RIGHT RELEASE
21	01	Valid	Valid	Valid	Valid	ERROR CLEAR
	02	Valid	Valid	Valid	Valid	ERROR LOG READ
	03	Valid	Valid	Valid	Valid	ERROR LOG CLEAR

Command code		PC mode				Name
		RUN	MONITOR	DEBUG	PROGRAM	
22	01	Valid	Valid	Valid	Valid	FILE NAME READ
	02	Valid	Valid	Valid	Valid	SINGLE FILE READ
	03	Valid	Valid	Valid	Valid	SINGLE FILE WRITE
	04	Valid	Valid	Valid	Valid	MEMORY CARD FORMAT
	05	Valid	Valid	Valid	Valid	FILE DELETE
	06	Valid	Valid	Valid	Valid	VOLUME LABEL CREATE/DELETE
	07	Valid	Valid	Valid	Valid	FILE COPY
	08	Valid	Valid	Valid	Valid	FILE NAME CHANGE
	09	Valid	Valid	Valid	Valid	FILE DATA CHECK
	0A	Valid	Valid	Valid	Valid	MEMORY AREA FILE TRANSFER
	0B	Valid	Valid	Valid	Valid	PARAMETER AREA FILE TRANSFER
	0C	(See note.)	Valid	Valid	Valid	PROGRAM AREA FILE TRANSFER
23	01	Not valid	Valid	Valid	Valid	FORCED SET/RESET
	02	Not valid	Valid	Valid	Valid	FORCED SET/RESET CANCEL

Note When the PC is in RUN mode, data transfers from files to the program area are not possible.

FINS Commands to the ISA Control Unit

The following table shows the FINS commands that are returned in response to service requests by the CPU Bus Driver using event service. When the ISA Control Unit receives these FINS commands, it automatically sends back a response. For details, refer to *8-5 FINS Commands Serviced by Drivers*.

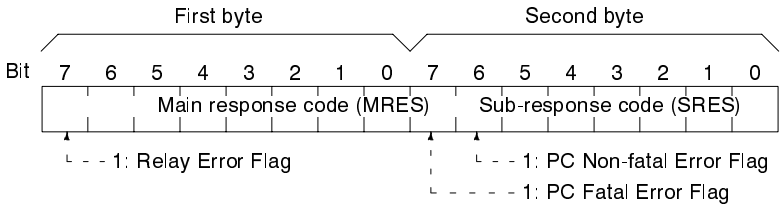
MRC	SRC	Contents of Service Request Commands
$05	$01	CONTROLLER DATA READ
$07	$01	TIME DATA READ
$07	$02	TIME DATA WRITE
$08	$01	LOOP BACK TEST
$21	$02	ERROR LOG READ
$21	$03	ERROR LOG CLEAR

Appendix F
Troubleshooting with FINS Response Codes

The following table lists the response codes (main and sub-codes) returned after execution of FINS commands, the probable cause of the errors, and recommended remedies.

Upon receipt of some commands, the destination node will issue a request to another node; the other node is referred to as the third node.

Response codes for FINS commands consist of two bytes that indicate the result of executing a command. The structure of the response codes is shown in the following diagram.



The main response code (MRES) in the first byte classifies the response and the sub-response code (SRES) in the second byte indicates details under the MRES classification.

If bit 7 of the first byte is ON, a network relay error has occurred. Refer to *Network Relay Errors* in the *FINS Commands Reference Manual (W227)* for details on troubleshooting the error.

If bit 6 or 7 of the second byte is ON, an error has occurred in the PC or computer returning the response. Refer to the operation manual for the device returning the response for details when troubleshooting the error.

Main code	Sub-code	Probable cause	Remedy
00: Normal completion	00	---	---
	01	Service was interrupted	Check the contents of the destination transmission area of third node.
			Check the data link status.
01: Local node error	01	Local node not part of Network	Add to Network.
	02	Token time-out, node number too large	Set the local node's node number below the maximum node number.
	03	Number of transmit retries exceeded	Check communications with internode echo test. If the test fails, check network.
	04	Maximum number of frames exceeded	Either check the execution of events in the network and reduce the number of events occurring in one cycle, or increase the maximum number of frames.
	05	Node number setting error (range)	Make sure the node number is within specified range and that there are no duplicate node numbers.
	06	Node number duplication error	Make sure that there are no duplicate node numbers.

Main code	Sub-code	Probable cause	Remedy
02: Destination node error	01	Destination node not part of Network	Add to Network.
	02	No node with the specified node number	Check the destination node's node number.
	03	Third node not part of Network	Check the third node's node number.
		Broadcasting was specified.	Check the control data and specify only one node as the third node.
	04	Busy error, destination node busy	Increase the number of transmit retry attempts or re-evaluate the system so that the destination node is not so busy receiving data.
	05	Response time-out, message packet was corrupted by noise	Increase the number of transmit retry attempts.
		Response time-out, response watchdog timer interval too short	Increase the value for the response watchdog timer interval in the control data.
		Frame lost in transmission	Check the error log and correct the process.
03: Communications controller error	01	Error occurred in the communications controller, ERC indicator is lit	Take corrective action, referring to communications controller errors and remedies table at end of this section
	02	CPU error occurred in the PC at the destination node	Clear the error in the CPU (refer to the PC's operation manuals)
	03	A controller error has prevented a normal response from being returned.	Check network communications status and reset the controller board. If the error still exists, replace the controller board.
	04	Node number setting error	Make sure the node number is within specified range and that there are no duplicate node numbers.
04: Not executable	01	An undefined command has been used.	Check the command code and be sure that the Unit supports it.
	02	Cannot process command because the specified unit model or version is wrong.	Check the unit model and version.
05: Routing error	01	Destination node number is not set in the routing table.	Set the destination node number in the routing table.
	02	Routing table isn't registered.	Set the source nodes, destination nodes, and relay nodes in the routing table.
	03	Routing table error	Set the routing table correctly.
	04	The maximum number of relay nodes (2) was exceeded in the command.	Redesign the network or reconsider the routing table to reduce the number of relay nodes in the command.
10: Command format error	01	The command is longer than the max. permissible length.	Check the command format of the command and set it correctly.
	02	The command is shorter than min. permissible length.	Check the command format of the command and set it correctly.
	03	The designated number of data items differs from the actual number.	Check the number of items and the data, and make sure that they agree.
	04	An incorrect command format has been used.	Check the command format of the command and set it correctly.
	05	An incorrect header has been used. (The local node's relay table or relay node's local network table is wrong.)	Set the routing table correctly.

Main code	Sub-code	Probable cause	Remedy
11: Parameter error	01	A correct memory area code has not been used or Expansion Data Memory is not available.	Check the command's memory area code and set the appropriate code.
	02	The access size specified in the command is wrong, or the first address is an odd number.	Set the correct access size for the command.
	03	The first address is in an inaccessible area.	Set a first address that is in an accessible area.
	04	The end of specified word range exceeds the acceptable range.	Check the acceptable limits of the data area and set the word range within the limits.
			Check the data link tables to be sure the limit to link words has not been exceeded.
	06	A non-existent program no. has been specified.	Check the program number and be sure that it is set correctly.
	09	The sizes of data items in the command block are wrong.	Check the command data and be sure that the sixes of the data items are correct.
			Check the data link tables to be sure all nodes in the refresh parameters are in the common link parameters.
	0A	The IOM break function cannot be executed because it is already being executed.	Either abort the current IOM break function processing, or wait until it is completed and execute the command.
			Check the data link tables for duplicate node numbers.
	0B	The response block is longer than the max. permissible length.	Check the command format and set the number of items correctly.
	0C	An incorrect parameter code has been specified.	Check the command data and reenter it correctly.
			Check the data link table file for corruption.
20: Read not possible	02	The data is protected.	Execute the instruction again after issuing the PROGRAM AREA PROTECT CLEAR command.
		An attempt was made to download a file that is being uploaded.	Check the file name and either interrupt servicing or wait for servicing to complete before re-executing the command.
	03	The registered table does not exist or is incorrect.	Set or reset the registered table.
		Too many files open.	Close open files and re-execute the command.
	04	The corresponding search data does not exist.	---
	05	A non-existing program no. has been specified.	Check the program number and be sure that it is set correctly.
	06	A non-existing file has been specified.	Check whether the correct file name was used.
	07	A verification error has occurred.	Check whether the memory contents are correct and replace if incorrect.
			Check the contents of the file. A read error may have occurred.

Main code	Sub-code	Probable cause	Remedy
21: Write not possible	01	The specified area is read-only or is write-protected.	If the specified area is read-only, the write cannot be performed. If it is write-protected, turn off the write-protect switch and execute the instruction again.
	02	The data is protected.	Execute the instruction again after issuing the PROGRAM AREA PROTECT CLEAR command.
		An attempt was made to simultaneously download and upload a file.	Check the file name and either interrupt servicing or wait for servicing to complete before re-executing the command.
		The data link table cannot be written manual because it is set for automatic generation.	Change the system settings to manual data link table generation.
	03	The number of files exceeds the maximum permissible.	Write the file(s) again after erasing unneeded files, or use different disk or Memory Card that has free space.
		Too many files open.	Close open files and re-execute the command.
	05	A non-existing program no. has been specified.	Check the program number and be sure that it is set correctly.
	06	A non-existent file has been specified.	---
	07	The specified file already exists.	Change the name of the file and execute the instruction again.
	08	Data cannot be changed.	Check the contents of the memory area being written to.
22: Not executable in current mode	01	The mode is wrong (executing).	Check the operating mode.
		Data links are active.	Check the data link status before execution.
	02	The mode is wrong (stopped).	Check the operating mode.
		Data links are active.	Check the data link status before execution.
	03	The PC is in the PROGRAM mode.	Check the PC's mode.
	04	The PC is in the DEBUG mode.	Check the PC's mode.
	05	The PC is in the MONITOR mode.	Check the PC's mode.
	06	The PC is in the RUN mode.	Check the PC's mode.
	07	The specified node is not the control node.	Check which node is the control node.
	08	The mode is wrong and the step cannot be executed.	Check whether the step has active status or not.
23: No Unit	01	A file device does not exist where specified.	Mount the Memory Card or disk
	02	The specified memory does not exist.	Check the specifications of the installed file memory.
	03	No clock exists.	Check the model number.
24: Start/stop not possible	01	The data link table either hasn't been created or is incorrect.	Set the data link table correctly.

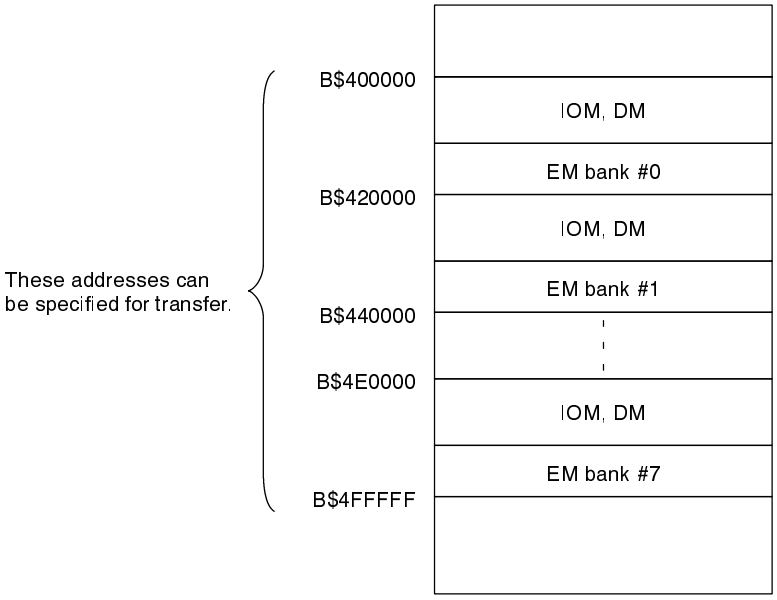
Main code	Sub-code	Probable cause	Remedy
25: Unit error	02	Parity/checksum error occurred because of incorrect data.	Transfer correct data into memory.
	03	I/O setting error (The registered I/O configuration differs from the actual.)	Either change the actual configuration to match the registered one, or generate the I/O table again.
	04	Too many I/O points	Redesign the system to remain within permissible limits.
	05	CPU bus error (An error occurred during data transfer between the CPU and a CPU Bus Unit.)	Check the Unit, Service Boards, and cable connections and issue the ERROR CLEAR command.
	06	I/O duplication error (A rack number, unit number, or I/O word allocation has been duplicated.)	Check the system's settings and eliminate any duplication.
	07	I/O bus error (An error occurred during data transfer between the CPU and an I/O Unit.)	Check the Unit, Service Boards, and cable connections and issue the ERROR CLEAR command.
	09	SYSMAC BUS/2 error (An error occurred during SYSMAC BUS/2 data transfer.)	Check the Unit, Service Boards, and cable connections and issue the ERROR CLEAR command.
	0A	Special I/O Unit error (An error occurred during CPU Bus Unit data transfer.)	Check the Unit, Service Boards, and cable connections and issue the ERROR CLEAR command.
	0D	Duplication in SYSMAC BUS word allocation.	Check and regenerate the I/O table.
	0F	A memory error has occurred in internal memory, in the Memory Card, or in Expansion DM during the error check.	If the error occurred in internal memory or the EM Unit, correct the data in the command an execute it again. If the error occurred in a Memory Card or EM used for file memory, the file data has been corrupted. Execute the MEMORY CARD FORMAT command. If the above remedies do not eliminate the error, replace the faulty memory.
	10	Terminator not connected in SYSMAC BUS System.	Connect the terminator correctly.

Main code	Sub-code	Probable cause	Remedy
26: Command error	01	The specified area is not protected. This response code will be returned if an attempt is made to clear protection on an area that is not protected.	The program area is not protected, so it isn't necessary to clear protection.
	02	An incorrect password has been specified.	Specify a password that is registered.
	04	The specified area is protected.	Execute the command again after the PROGRAM AREA PROTECT CLEAR command.
		To many commands at destination.	The destination has received more than 5 commands. Either interrupt servicing or wait for servicing to complete before re-executing the command.
	05	The service is being executed.	Execute the command again after the service has been completed or aborted.
	06	The service is not being executed.	Execute the service if necessary.
	07	Service cannot be executed from local node because the local node is not part of the data link.	Execute the service from a node that is part of the data link.
		A buffer error has prevented returning a normal response.	Reset the board. If the error persists, replace the board.
	08	Service cannot be executed because necessary settings haven't been made.	Make the necessary settings.
	09	Service cannot be executed because necessary settings haven't been made in the command data.	Check the command format of and make the necessary settings.
	0A	The specified action or transition number has already been registered.	Execute the command again using an action or transition number that hasn't been registered.
	0B	Cannot clear error because the cause of the error still exists.	Eliminate the cause of the error and execute the ERROR CLEAR command.
30: Access right error	01	The access right is held by another device.	Execute the command again after the access right has been released. (The command can be executed after the ACCESS RIGHT FORCED ACQUIRE or ACCESS RIGHT RELEASE command is completed. Releasing the access right might affect processes in progress at the node that held the access right.)
40: Abort	01	Command was aborted with ABORT command.	---

Appendix G
PC Memory Configuration

The PC memory addresses which the user can specify with cyclic service are those in the IOM, DM, and EM Areas. These areas are specified with the absolute PC memory addresses. (The UM Area cannot be specified.)

The following diagram shows the addresses that can be specified.



The PC's memory configuration is shown on the following page. This memory configuration is the same as the one listed in the *CV-series PC Operation Manual: Ladder Diagrams*, but the memory addresses are listed in word units. Memory addresses 0000 through FFFF on the next page correspond to addresses B$400000 through B$41FFFF in the diagram above.

Only even-numbered addresses can be specified.

Addresses are found with the following format: Offset $400000 + memory address × 2

PC Memory Configuration

Memory addresses	PC Memory	Data area addresses
0000 to 0FFF	IOM (I/O Memory) 4K-words	(See expanded view.)
1000 to 13FF	Timer PVs 1K-word	T0000 to T1023
1800 to 1BFF	Counter PVs 1K-word	C0000 to C1023
2000 to 27CF	Data link area 2K-words	D00000 to D01999
27D0 to 2E0F	CPU Bus Unit area 1600 words	D02000 to D03599
2E10 to 7FFF	24K-words of DM (CV500 up to D08191.)	D03600 to D24575
8000 to FFFD	EM banks 0 to 7 32K-words each (CV1000 only)	E00000 to E32765
FFFE to FFFF	Used by the system.	---

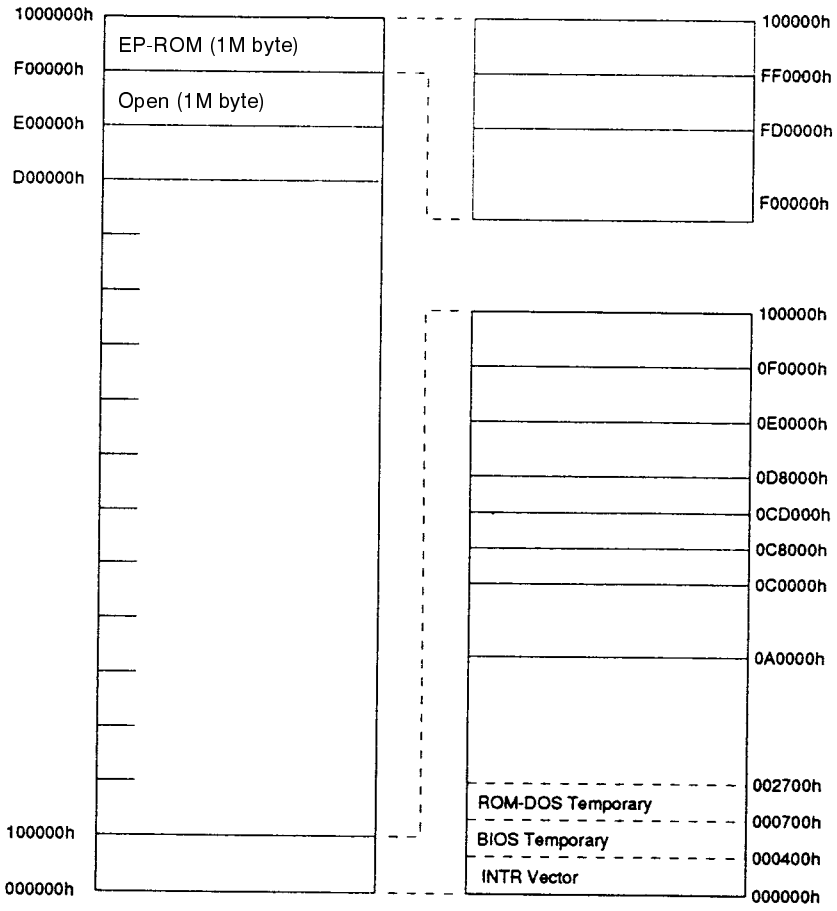
Memory addresses	PC Memory	Data area addresses
0000 to 00C7	I/O Area 200 words	CIO 0000 to CIO 0199
00C8 to 03E7	SYSMAC BUS/2 Area 800 words	CIO 0200 to CIO 0999
03E8 to 04AF	Link Area 200 words	CIO 1000 to CIO 1199
04B0 to 05DB	Holding Area 300 words	CIO 1200 to CIO 1499
05DC to 076B	CPU Bus Unit Area 400 words	CIO 1500 to CIO 1899
076C to 08FB	Work Area 400 words	CIO 1900 to CIO 2299
08FC to 09FB	SYSMAC BUS Area 256 words	CIO 2300 to CIO 2555
09FC to 09FE	Used by the system.	---
09FF	Temporary Relay Area	TR0 to TR7
0A00 to 0AFF	CPU Bus Link Area 256 words	G000 to G255
0B00 to 0CFF	Auxiliary Area 512 words	A000 to A511
0D00 to 0D3F	Transition Area 64 words	TN0000 to TN1023
0E00 to 0E3F	Step Area 64 words	ST0000 to ST1023
0F00 to 0F3F	Timer Area 64 words	T0000 to T1023
0F80 to 0FBF	Counter Area 64 words	C0000 to C1023

CIO Area

Note The CIO acronym is included for clarity; just input the address (0000 to 2555) when specifying words in the CIO Area.

Appendix H
ISA Control Unit Memory Configuration

Memory Configuration



Note 1. This can be set either of two ways, 16K bytes each, within a range of 0D8000H to 0DFFFFh. The setting is made with System Switch 1.

2. Not installed with this Unit.

I/O Configuration

Address (HEX)	Item	Remarks
000 to 01F	DMA controller 1	System resources
020 to 021	Interrupt controller 1	
022 to 03F	Reserved.	
040 to 043	Timer	
044 to 05F	Reserved.	
060 to 06F	Keyboard controller	
070 to 07F	RTC, NMI mask	
080 to 08F	DMA page register	
090 to 09F	Reserved	
0A0 to 0A1	Interrupt controller 2	
0A2 to 0BF	Reserved.	
0C0 to 0DF	DMA controller 2	
0E0 to 0EF	Reserved	
0F0 to 0FF	Coprocessor	
100 to 1EF	For I/O channels	
1F0 to 1FF	IDE hard disk	Used by Hard Disk Unit.
200 to 207	Game I/O adapter	Not used by ISA Control Unit.
208 to 20F	Reserved.	
210 to 217	Not used.	
218 to 21A	Reserved.	
21B to 277	Not used.	
278 to 27F	Parallel port 2	Not used by ISA Control Unit.
280 to 2EF	Not used.	
2F0 to 2F7	Not used.	
2F8 to 2FF	Serial port 2 (COM2)	
300 to 31F	Not used.	
320 to 36F	Not used.	
370 to 37F	Parallel port 1	Not used by ISA Control Unit.
380 to 38F	Not used.	
390	Indicator status register (R)/ Indicator control register A (W)	
391	Reserved	
392	Unit status register (R)/ Indicator control register B (W)	
393	Reserved.	
394	CPU bus interface	
395 to 39F	Not used.	Not used by ISA Control Unit.
3A0 to 3AF	Not used.	
3B0 to 3BF	Not used.	
3C0 to 3DF	Video sub-system	Not used by ISA Control Unit.
3E0 to 3E7	Reserved	
3E8 to 3EF	Not used.	
3F0 to 3F1	Reserved.	
3F2 to 3F7	Floppy disk	Not used by ISA Control Unit.
3F8 to 3FF	Serial port 1 (COM1)	

Interrupt Configuration

The following table indicates the interrupt configuration starting from the highest interrupt level.

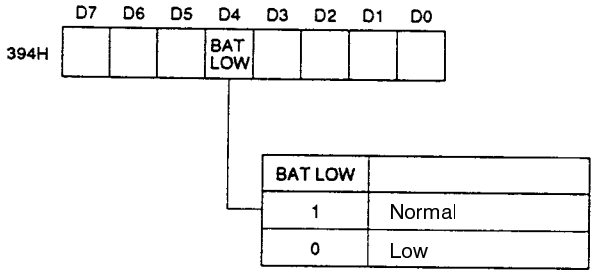
Number	Item	Remarks
NM1	Parity error, I/O point check	
IRQ0	System timer	
IRQ1	Keyboard	
IRQ2	For interrupt controller 2 cascade connection.	
IRQ8	RTC	
IRQ9	VGA controller	Not used by ISA Control Unit.
IRQ10	CPU bus interface/undefined (Note 1)	
IRQ11	CPU bus interface/undefined (Note 1)	
IRQ12	Mouse (keyboard controller)	Not used by ISA Control Unit.
IRQ13	Coprocessor	Not used by ISP01.
IRQ14	HD controller	Used by Hard Disk Unit.
IRQ15	PC card controller	Can be changed by software.
IRQ3	Serial port 2 (COM2)	
IRQ4	Serial port 1 (COM1)	
IRQ5	Parallel port 2	Not used by ISA Control Unit.
IRQ6	FD controller	Not used by ISA Control Unit.
IRQ7	Parallel port 1	Not used by ISA Control Unit.

Note CPU bus interface interrupt signals are set by pin no. 3 of the system switch. Either IRQ10 or IRQ11 can be selected. (The unselected signal will be undefined.)

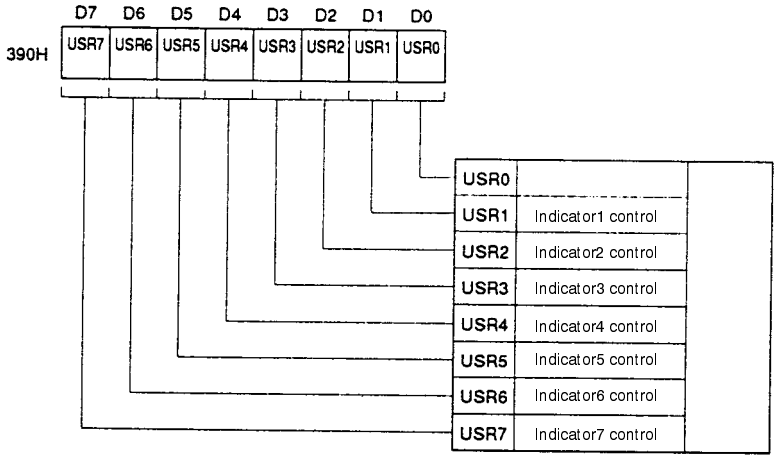
DMA Point Configuration

Number	Item	Remarks
DRQ0	Undefined.	
DRQ1	SDLC	Not used by ISA Control Unit.
DRQ2	FD controller	Not used by ISA Control Unit.
DRQ3	Undefined.	
DRQ4	DMA controller 1 cascade connection	
DRQ5	Undefined.	
DRQ6	Undefined.	
DRQ7	Undefined.	

Battery Low Status Register

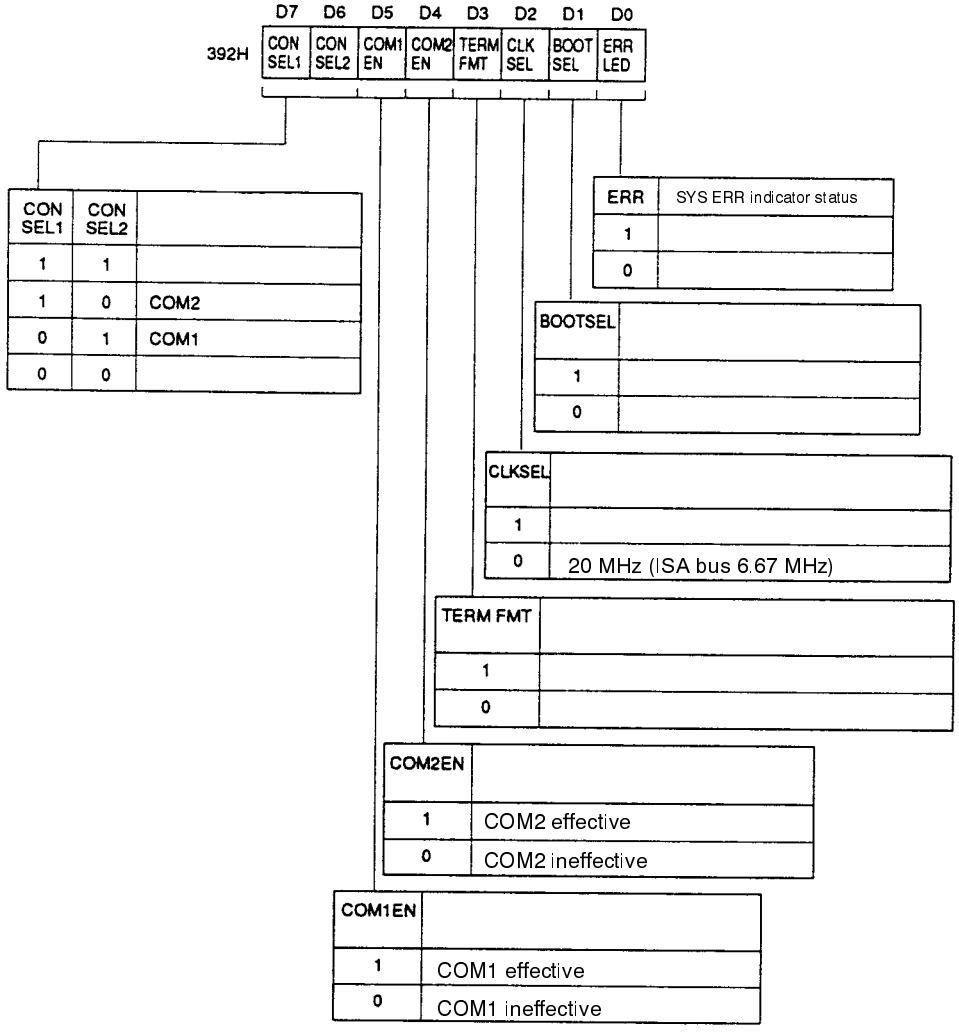


Indicator Register



Unit Status Register

By accessing this register it is possible to monitor the status of the SYS ERR indicator, system switch pin numbers 4 to 8, and the console switch.



Index

Numbers

A–B

C

D

E

F

G–H

I

L

M–O

P

Q–R

S

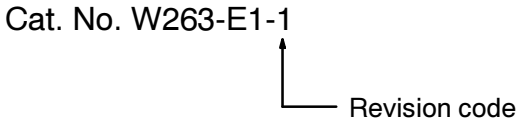
T

U

V–Z

Revision History

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



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	May 1996	Original production