Digital

Human Machine Interface

Pro-face

BOX-Type Industrial Computer
PL-B920 Series
User Manual

When printing out this PDF manual, since the printer setting can effect the quality of the printout, be sure your printer's quality setting is set to "High".

Digital

Digital Electronics Corporation

Introduction

Thank you for purchasing Proface's PL-B920 series Box Type Industrial Computer, hereafter referred to as "the PL". This unit embodies Proface's latest, cost-effective architecture and is designed for Industrial Automation users.

Prior to using your PL, be sure to read this manual thoroughly to familiarize yourself with the unit's operation procedures and functions.

NOTE:

1. It is forbidden to copy the contents of this manual in whole, or in part, without the permission of the Digital Electronics Corporation.
2. The information in this manual is subject to change without notice.
3. This manual was written with care; however, if you should find any errors or omissions, please contact Digital and inform them of your findings.

4) Please be aware that Digital Electronics Corporation shall not be held liable by the user for any damages, losses, or third party claims arising from the uses of this product.

MS-DOS®, Windows® 95, Windows® 98 Second Edition and Windows NT® are registered trademarks of the Microsoft Corporation.

Product names used in this manual are the trademarks of their respective manufacturers.

Essential Safety Precautions

This manual includes the following cautions concerning procedures that must be followed to operate the PL correctly and safely. Prior to operating the PL, be sure to read this manual and any related materials thoroughly to understand the correct operation and functions of this unit.

Safety Icons

To allow you to use the PL correctly, throughout this manual, the following icons are provided next to operations requiring special attention. These icons are used to describe the following situations:

⚠ Warning	Indicates situations where severe bodily injury, death or major equipment damage may occur.
⚠ Caution	Indicates situations where slight bodily injury or machine damage can occur.

⚠ WARNINGS

- **To avoid the possiblity of an electric shock, be sure to connect the power cord to the PL before connecting it to the main power supply.**
- **A fire or electrical shock may occur if voltages used with the PL are beyond the specified range. Be sure to use only the specified voltage.**
- **Before opening the PL's protective cover, be sure to turn the unit's power OFF. This is because the PL's internal parts carry high voltages.**
- **To avoid fires or electrical hazards, do not modify the PL in any way.**
- **If metal particles, water or other types of liquids contact any of the PL's internal parts, immediately turn the unit's power OFF, unplug the power cord, and contact either your PL distributor or the Digital Electronics Corporation.**
- **Read and understand Chapter 4 "Installation and Wiring" thoroughly in order to select an appropriate installation location for the PL.**

⚠ WARNINGS

- Before either plugging in or unplugging a board or interface connector, be sure to turn the PL's power OFF.
- To prevent a possible explosion, do not install the PL in areas containing flammable gases.
- The PL is not appropriate for use with aircraft control devices, aerospace equipment, central trunk data transmission (communication) devices, nuclear power control devices, or medical life support equipment, due to these devices' inherent requirements of extremely high levels of safety and reliability.
- When using the PL with transportation vehicles (trains, cars and ships), disaster and crime prevention devices, various types of safety equipment, non-life support related medical devices, etc. redundant and/or failsafe system designs should be used to ensure the proper degree of reliability and safety.

⚠ CAUTIONS

- Avoid exposing the PL to, or operating the PL in direct sunlight, high temperatures and humidity, and in areas where excessive dust and vibration will occur.
- Avoid using the PL in areas where sudden, extreme changes in temperature can occur. This may cause condensation to form inside the unit, possibly leading to an accident.
- To prevent the PL from overheating, be sure its air circulation vents are clear and clean, and keep the unit's operation area well-ventilated.
- Avoid operating or storing the PL near chemicals, or where chemicals can come into contact with the unit.
- When the Standard display is connected to the PL, after turning the display OFF, be sure to wait at least three (3) seconds before turning it ON again.

⚠CAUTIONS

When PL Hard Disk (HDD) data is lost:

- **The Digital Electronics Corporation cannot be held responsible or provide any compensation for damage(s) caused by the loss of data stored in the PL's hard disk drive (HDD). It is therefore strongly suggested that all important data and software be backed up regularly to an external data backup device.**
- **Please be aware that the Digital Electronics Corporation bears no responsibility for any damages resulting from the customer's application of this unit's hardware or software.**
- **Since the PL unit's hard disk drive (HDD) is a consumable item, i.e. it has a limited lifetime, be sure to back up its data regularly and prepare a spare HDD unit.**
- **To prevent file data damage, be sure to shut down the PL's OS before turning OFF the main power.**
- **After turning OFF the PL's power, wait until the internal HDD stops spinning before turning on the power again (approx. 5 seconds).**
- **When the PL unit's standard display is connected to the PL using the USB connection, wait three (3) seconds before turning the power ON again.**

Table of Contents

CHAPTER 1 PL BASICS

CHAPTER 2 SPECIFICATIONS

CHAPTER 3 INSTALLING OPTIONAL UNITS AND EXPANSION BOARDS

CHAPTER 4 INSTALLATION AND WIRING

CHAPTER 5 SYSTEM SETUP

Documentation Conventions

The list below describes the documentation conventions used in this manual.

Symbol	Meaning
Important	Indicates important information or procedures that must be followed for correct and risk-free software/device operation.
Note:	Provides useful or important supplemental information.
*1	Indicates useful or important supplemental information.
Reference	Refers to useful or important supplemental information
1) , 2)	Indicates steps in a procedure. Be sure to perform these steps in the order given.
PL	Abbreviation for the PL-B920 Series Box Type Industrial Computers.
DU	Standard display unit for the PL-B920 Series Box Type Industrial Computers.

PL Series Model Numbers

```
PL  - B 9 2 * - 4 *
--    -----   -   - -
A       B     C   D E
```

Item	Code	Meaning
A	PL	PL Series Panel Computer
B	B92	“Box” type industrial computer PL-X920 Series unit
C	0	4-Slot type [*1]
	1	2-Slot type [*1]
D	4	CE Marking, UL/c-UL(CSA) Approval
E	*	PL unit's revision number

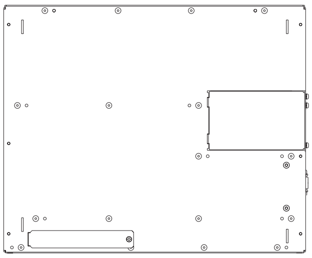
***1** *Refers to the number of available ISA Bus expansion slots.*

Package Contents

The PL's packing box contains the items listed below. Please check to confirm that all items shown below have been included.

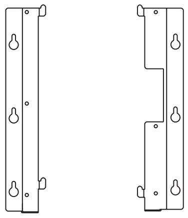
■ **PL Unit**

PL-B920/PL-B921

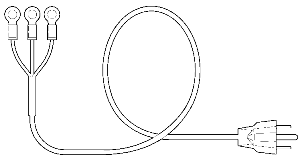


■ **Installation Brackets**

(2 brackets - left and right)
(6 attachment screws)

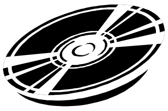


■ **Power Cord**



■ **CD-ROM (1)**

(PL-X920 Series User Manual and Driver CD)



■ **Installation Guide**

(Japanese and English)

Important

- Be careful when handling the PL not to damage the built-in HDD.

- This cord is designed only for AC100/115V use. Any other voltage will require a different cord.

Note:

- The CD-ROM contains User Manual and PL-X920 Series Utility and Driver. For details, Reference Chapter6 - Setting Up Your PL OS.

- When you order a PL unit built to your specifications, that PL package should include the Optional items' installation Guide(s). Be sure to also check the Optional items' Package Contents.

Special Features

The PL series units are equipped with the following unique features:

■ The Latest, High-Performance Architecture

Designed around the Pentium III® (700MHz) CPU, the PL utilizes the type of high performance IBM compatible architecture that offers you superior performance. This allows you to easily use the popular Windows®95, Windows®98 Second Edition, WindowsNT® and Windows®2000 operating systems, as well as others.

■ Highly Expandable

Choose from the PL-B921 series unit - with 2 ISA slots, or the PL-B920 series unit - with 4 ISA slots. In addition, the PL-B921 unit provides a single PCI bus (slot 2), while the PL-B920 unit provides two PCI buses (slots 2 and 3). Also, these slots can accommodate both Digital's own optional boards as well as other commercially available expansion boards.

Digital also offers a wide variety of optional products, such as a -5/-12V DC power unit, DIM memory modules, etc.

■ Separately Designed Main Unit and Display

The PL's separated main unit and display offer you an expanded range of installation options for all your applications.

■ Single Cable for Video and Touch Panel Signals

This all-in-one cable allows you to easily connect the PL's Standard display (DU) and main units together.

■ USB Interface

This standard type USB interface allows you to easily connect any standard type of USB compatible device.

UL/c-UL(CSA) Application Notes

The PL-B92*-T4* series units are (c)UL 1950 recognized products. (UL File No. E171486). Please pay special attention to the following instructions when applying for UL/c-UL approval for machinery which includes any of these PL units.
Equipment with a PL mounted in it requires UL/c-UL evaluation for the combination of the PL and equipment.

The PL conforms as a component to the following standards:
UL 1950, Third Edition, dated March 1,1998 (Standard for Safety of Information Technology Equipment, including Electrical Business Equipment)
CSA-C22.2 No. 950-M95 (Standard for Safety of Information Technology Equipment, including Electrical Business Equipment)
PL-B920-T4* (UL Registration Model: 2880056-02)
PL-B921-T4* (UL Registration Model: 2880056-01)

- The PL should be used as a built-in component of another product.
- Use the PL indoors only.
- When connecting the PL's power cord, be sure to use a cord that is appropriate for the current and voltage used, and that has conductive wires that are 0.75 mm^2 or larger.
- When an end-use product will include the PL, be sure to design the PL's power cut-off switch as a separate disconnect device and locate it where the operator can easily reach it.
- Danger of explosion if backup battery is incorrectly replaced. Replaced only with same or equivalent type recommended by the manufacturer. Dispose of used batteries according to the manufacturer's instructions.
- Be sure the unit the PL is built into is a (c)UL1950 approved structure.

CE Marking Notes

The PL-B92*-T4* series units are CE marked, EMC compliant products.

<These units comply with the following standards>

■ **Safety**

EN60950

■ **EMI**

EN55011 Group1 (Class A), EN61000-3-2, EN61000-3-3

■ **EMS (EN61000-6-2)**

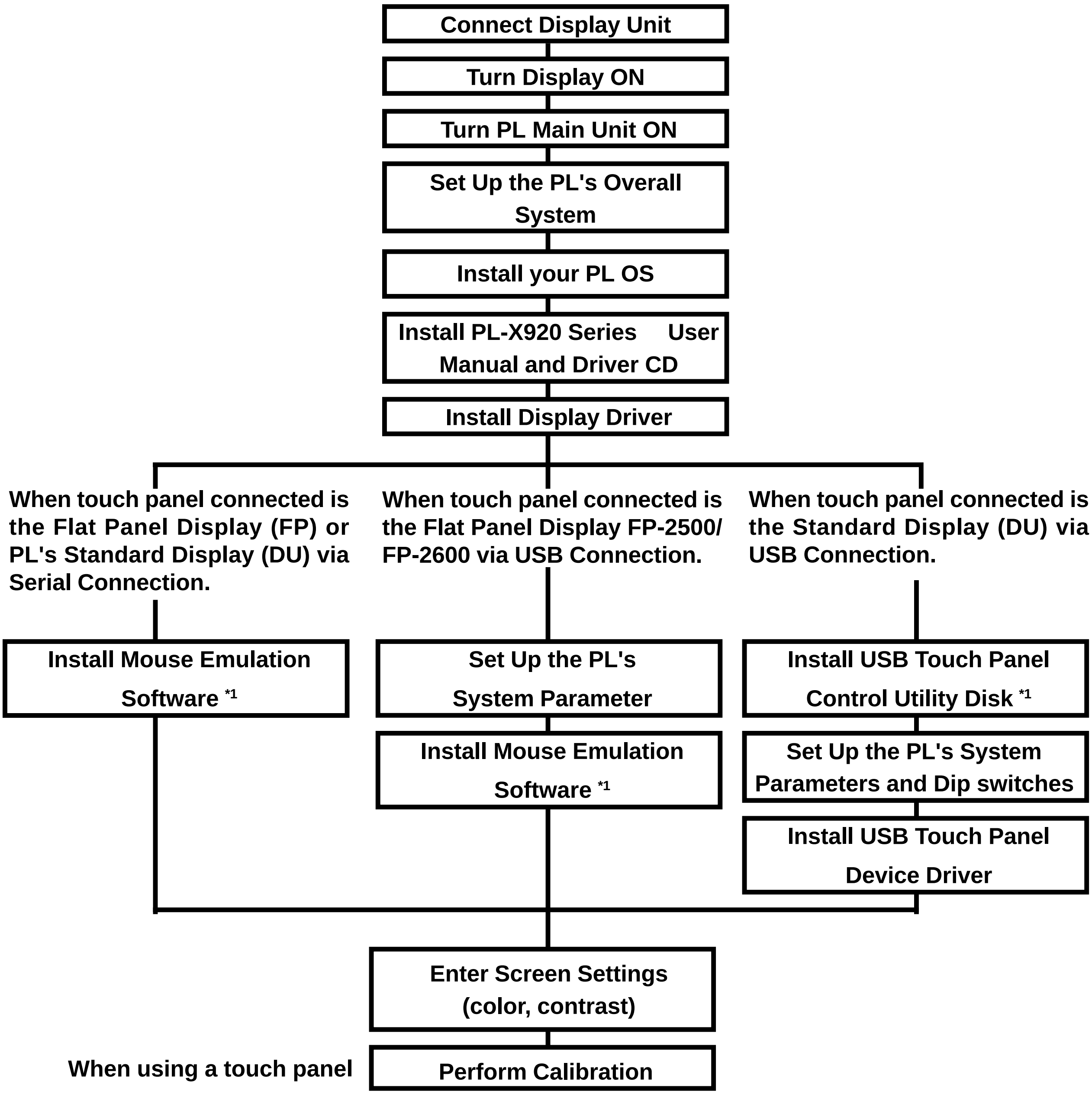
EN61000-4-2, EN61000-4-3, EN61000-4-4, EN61000-4-5, EN61000-4-6, EN61000-4-8, EN61000-4-11

If following requirements are not met, the PL may fail to meet EN60950 standard requirements.

- The PL must be used as a built-in component of an end-use product.
- The PL is intended for indoor use only.
- When connecting the PL's power cord, be sure to use a cord that is appropriate for the current and voltage used and that has conductive wires that are 0.75 mm^2 or larger.
- When installing the PL in a metal panel or cabinet, be sure to place the PL's Power disconnect device (cut-off switch) where the unit's operator can easily reach it.
- There is a danger of explosion if the PL's backup battery is incorrectly replaced. This battery should be replaced only with same or equivalent type recommended by the manufacturer. Dispose of used batteries according to the manufacturer's instructions.
- Be sure the cabinet/enclosure the PL is built into uses an EN60950 approved sheet steel structure.

Chapter 1 PL Basics

1.1 Prior to Operating the PL

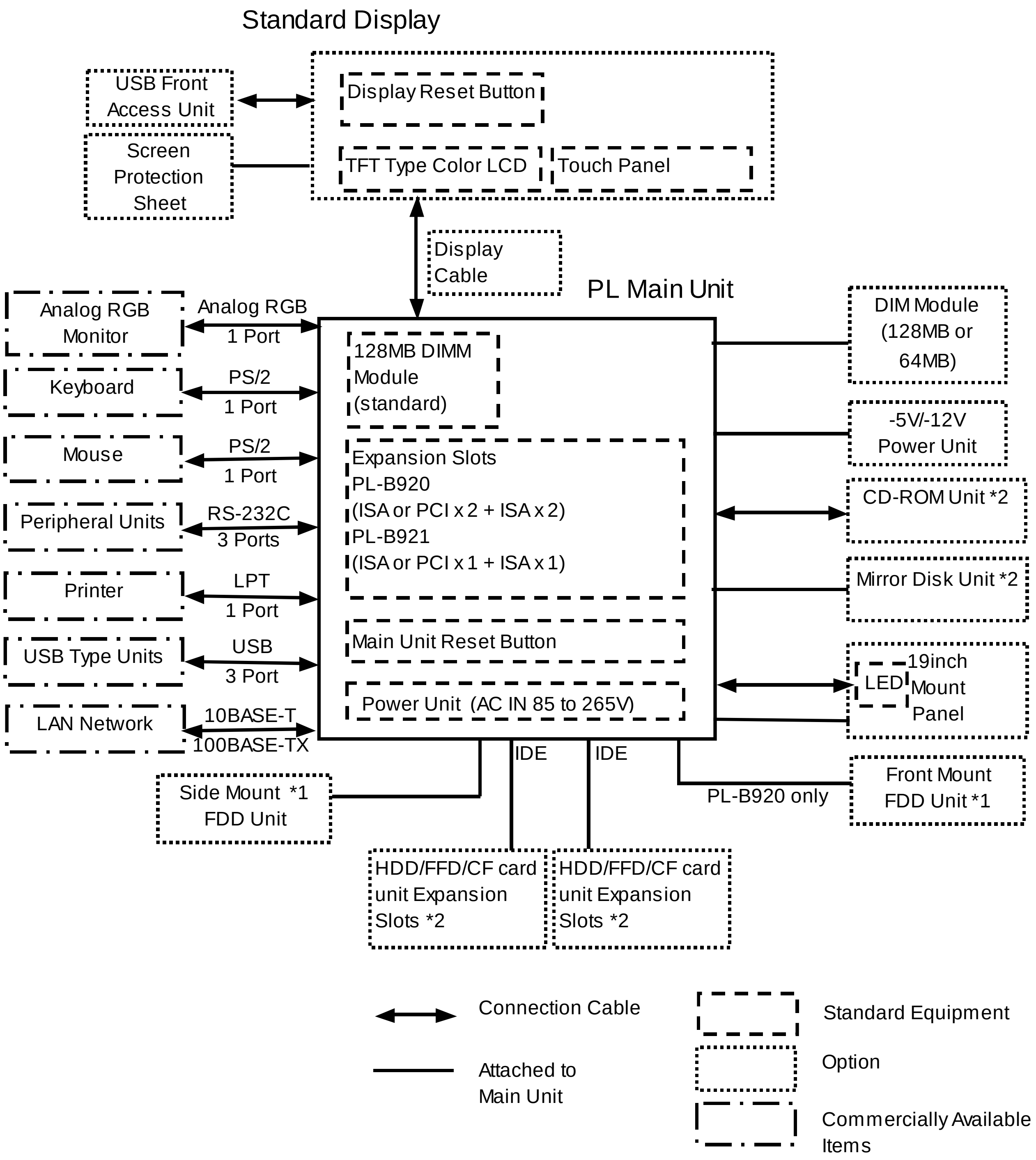

Connect Display Unit
Turn Display ON
Turn PL Main Unit ON
Set Up the PL's Overall System
Install your PL OS
Install PL-X920 Series User Manual and Driver CD
Install Display Driver
When touch panel connected is the Flat Panel Display (FP) or PL's Standard Display (DU) via Serial Connection.
When touch panel connected is the Flat Panel Display FP-2500/ FP-2600 via USB Connection.
When touch panel connected is the Standard Display (DU) via USB Connection.
Install Mouse Emulation Software *1
Set Up the PL's System Parameter
Install USB Touch Panel Control Utility Disk *1
Install Mouse Emulation Software *1
Set Up the PL's System Parameters and Dip switches
Install USB Touch Panel Device Driver
Enter Screen Settings (color, contrast)
When using a touch panel
Perform Calibration

*1 This is required only when using touch panel. For details, Reference 1.2.1 Setting Up the Touch Panel Connection.

Important

- ***After completing the hardware setup, before any data or applications can be placed on the HDD drive, the OS (Windows® or MS DOS®, etc.) must be used to initialize the HDD and create partitions. For details concerning these procedures, refer to the OS maker's installation manual.***
- ***After turning OFF the PL's power, wait until the internal HDD stops spinning before turning on the power again (approx. 5 seconds).***
- ***When the PL unit's standard display is connected to the PL using the USB connection, wait three (3) seconds before turning the power ON again.***

1.2 PL System Configuration



**1 The PL-B920 side and front mount FDD units can be used individually, not simultaneously.*

2 Certain limitations exist for the combinations of the HDD unit, the FFD unit, the CF card unit, the Mirror Disk unit and the CD-ROM drive.* Reference ***1.3 Optional Items*

Important

The above system configuration represents the internal design and peripheral equipment connectable to the PL. The user's actual combination of these items may differ.

1.2.1 Setting Up the Touch Panel Connection

The PL can be connected to a Standard Display (DU), Flat Panel Display (FP) or commercialy available CRT monitor. Depending on the display, the cable used will be different. For detailed connection information,

Reference 3.1.8 Connecting the Standard Display and 3.1.9 Connecting an Analog RGB Monitor.

The connection method used can be via either a serial or USB interface. Also, the type of OS and Touch Panel Device Driver (mouse emulator) used will depend on the type of connection method used for the display and for the touch panel.

Display Type	Touch Panel Connection	Compatible OS Types	Touch Panel Device Driver (Mouse Emulator)
PL-DU6900 **PL-DU7900**	Serial (RS-422)	Windows® 95, Windows® 98 Second Edition, Windows NT ®4.0, Windows® 2000	U-TP (Gunze Inc.)*1
	USB	Windows® 98 Second Edition, Windows® 2000	USB Touch Panel Device Driver (included with DU)
FP-2500 **FP-2600**	Serial (RS-232C)	Windows® 95, Windows® 98 Second Edition, Windows NT ®4.0, Windows® 2000	PL-TD000
	USB	Windows® 98 Second Edition, Windows® 2000	
FP-790	Serial (RS-232C)	Windows® 95, Windows® 98 Second Edition, Windows NT ®4.0, Windows® 2000	U-TP (Gunze Inc.)*1

**1 For product details, please contact your local PL distributor or representative.*

When the connection method is changed to USB, the following BIOS level System settings must also be changed.

Reference ***Chapter 5 - System Setup***

System Setting Menu	Setting Item	USB Setting
Integrated Peripherals	USB Controller	Enabaled
	Onboard Serial Port 4	Disabled
PnP/PCI Configuration	Assign IRQ For USB	Enabled

When the DU display unit is used, the PL's dip switch settings must also be changed. For details,

Reference ***2.4 PL Part Names and Features***

1.2.2 Using the USB Interface

The PL unit's USB interface cannot be used without changing the factory settings. When connecting peripheral devices to the USB port, change the System Settings as shown below.

System Setup Menu	Menu Item	USB Setting
Integrated Peripherals	USB Controller	Enabled
PnP/PCI Configuration	Assign IRQ For USB	Enabled

Important

- ***A USB hub cannot be used with the DU unit's USB connector, or with a PL-B920 unit's single USB connector. When connecting a USB hub, be sure to use the PL-B920 Series unit's two side-by-side USB connectors.***
- ***When using a commercial-type USB hub, additional hubs cannot be attached to the first hub. Only a single "level" USB hub can be used when connecting USB devices.***

Note: **When the DU display unit's USB connector is used, the PL unit's Touch Panel Connection Method must be changed to "USB".**

1.2.3 Using the LAN Interface

The PL unit's LAN interface cannot be used without changing the factory settings. When connecting peripheral devices to the LAN port, change the System Settings as shown below.

System Setup Menu	Menu Item	LAN Setting
Integrated Peripherals	Onboard LAN	Enabled

1.3 Optional Items

Name	Model number	Description
Standard Display (DU)	PL-DU6900-T4*	12.1 inch TFT display panel made by the Digital Electronics Corporation. Connects to PL Main Unit.
	PL-DU7900-T4*	15 inch TFT display panel made by the Digital Electronics Corporation. Connects to PL Main Unit.
Flat Panel Display (FP)	FP790-T**	14.1 inch TFT color touch panel display.
	FP2500-T**	10.4 inch TFT color touch panel display.
	FP2600-T**	12.1 inch TFT color touch panel display.
USB Front Access Unit	PL-US200	USB Front Access Unit for DU.
Display Unit Cable	PL-CB200-5M	Connects Standard Display and Main Unit (5m).
	PL-CB200-10M	Connects Standard Display and Main Unit (10m).
RGB Cable	FP-CV00	Analog RGB interface cable used to output image signals from the PL to the FP-790 and CRT monitor (2.5 m). Compatible with analog RGB interface (D-sub 15 pin male connector).
	FP-CF01	Analog RGB interface cable used to output image signals from the PL to the FP-790 and CRT monitor (5 m). Compatible with analog RGB interface (D-sub 15 pin male connector).
	FP61V-IV000-0	Analog RGB interface cable used to output image signals from the PL to the FP-2500/FP-2600 (5 m). Compatible with analog VGA interface (D-sub 15 pin male connector).
USB Cable	FP-US000	USB interface cable used for touch panel data transmission between the PL and the FP or command transmission to the FP (5m). A-B type cable.
DIM Module	PL-EM500	SDRAM (DIMM) Provides 64MB of memory
	PL-EM128	SDRAM (DIMM) Provides 128MB of memory
-5V/-12V Power Unit	PL-PW100	Provides –5V and –12V power to expansion slots. Can provide a total of 200mA of current. (sum of both slots)

Name	Model number	Description
FDD Unit	PL-FD200	IBM PC Compatible 3.5” FDD unit (Attaches to side slot)
	PL-FD210	IBM PC Compatible 3.5” FDD unit (Attaches to front slot) (for PL-B920)
FFD Unit	PL-FF200	32MB Flash File Disk with IDE interface. Used the same way as an HDD.
CD-ROM Unit	PL-DK200	IDE (AT API) compliant CD-ROM drive unit – for development and maintenance use(special connection cable is included)
CF Card Unit	PL-CF200	Designed exclusively for 5V type cards.
CF Card	GP077-CF20	CF card (16M). CF Card Unit PL-CF200 is required.
	GP077-CF30	CF card (32M). CF Card Unit PL-CF200 is required.
HDD Unit	PL-HD220	20.0GB 2.5" HDD Unit (OS not included)
Mirror Disk Unit	PL-MD200-HU01	IDE type mirror disk unit without OS. Continuously backs up data to 2nd drive, to protect against an HDD malfunction.Contains two 2.5 inch, 2.1GB hard disk drives.
Soft Mirroring Utility	PL-SM900	Provides RAID Level 1 protection, without Mirror Disk unit.
Full-sized Cover	PL-FC200	Attached when an ISA or PCI bus full-sized board is used in the expansion slot. (for PL-B921)
	PL-FC210	Attached when an ISA or PCI bus full-sized board is used in the expansion slot. (for PL-B920)
Mouse Emulator V2	PL-TD000	Mouse Emulator software for FP-2500/FP-2600.
Mouse Emulator	FP-ME000	Mouse Emulator software for FP-790/PL-DU6900/PL-DU7900.
19 Inch Mounting Panel	PL-RM200	Used to install the PL into a 19 inch rack.

Note: ***Both the PL-FD200 and the PL-FD210 cannot be used at the same time.***

The PL is equipped with three IDE interfaces, two (2) of which can be used by the HDD, FFD or CF card units, and one (1) which can be used by either the CD-ROM drive or the Mirror Disk unit. Physically, even though up to three (3) IDE drive units can be connected at the same time, IDE interface specifications require that a controller's simultaneous operation be limited to a single master and slave unit, for a total of two devices.

The following chart shows the combinations available when using two IDE units.

HDD Unit	MS	M	M	M	M	S					S				S			
FFD Unit		S				M	MS	M	M	M		S				S		
Mirror Disk Unit			S					S			M	M	M	M			S	
CD-ROM Drive Unit				S					S				S					S
CF Card Unit					S					S				S	M	M	M	M

MS: Combination of 2 units - Master or Slave, is possible.

M: Used only for Master.

S: Used only for Slave.

■ Maintenance Options

Name	Model Number	Description
Mirror Disk Unit Replacement HDD	PL-MD200-MD01	Mirror Disk Unit’s replacement HDD (1)

- ***Since the PL unit's hard disk drive (HDD) is a consumable item, i.e. it has a limited lifetime, be sure to back up its data regularly and prepare a spare HDD unit.***
- ***The Hard Disk lifetime given here may be reduced due to unforeseen environmental factors, however, generally speaking, at an operating temperature of 20°C the disk should last for 20,000 hours (of operation) or approximately 5 years, whichever comes first.***

■ Commercially Available Items

The PL-B920 Series units can all use commerically available expansion boards (PCI/ISA compatible) as well as a standard keyboard, mouse, printer, etc. The PL-B920 Series units, can also use USB compatible devices. However, among the commercially available USB devices, not all will be compatible with the PL unit.

Note:

- **Since the PL expansion slots do not supply DC-5V and DC-12V current. To use expansion boards requiring DC-5V or DC-12 V, the optional PL-PW100 power supply should be installed.**
- **Be sure to use only DIM modules made by Digital Electronics Corporation. Certain commercial DIM modules may fail to meet the PL unit's specifications and cause the PL to malfunction.**
- **When using USB type devices, be sure they are USB compatible, and be sure to read that device's installation guide prior to connecting it to the PL.**

Memo

Chapter 2 Specifications

2.1 General Specifications

2.1.1 Electrical

	PL-B920	PL-B921
Rated Voltage	AC100V to AC240V	
Allowable Voltage Range	AC85V to AC265V	
Rated Frequency	50/60Hz	
Allowable Voltage Drop	1 cycle or less (however, pause occurrences must be more than 1 second apart)	
Power Consumption	150VA or less	120VA or less
Voltage Endurance	AC1500V 20mA for 1 minute (between charging and FG terminals)	
Insulation Resistance	10MΩ or higher at DC500V (between charging and FG terminals)	

***1** *When PL's internal fan is removed.*

2.1.2 Structural

Ambient Operating Temperature (Cabinet Interior)	With Fan: 5°C to 50°C (With HDD attached)
	Without Fan[*1]: 5°C to 40°C (With HDD attached)
Storage Temperature	-10°C to 60°C
Maximum Wet Bulb Temperature	29°C (Recommended)
Ambient Humidity	10%RH to 85%RH
Air Purity Level	Free of conductive particles and dust
Atmosphere	Free of corrosive gasses
Vibration Resistance	$19.6 m/s^2$ at 10Hz to 25Hz in X, Y, Z directions for 30 minutes With Hard Disk attached: $4.9 m/s^2$ With Floppy Disk attached: $9.8 m/s^2$
Noise Endurance (Impulse Noise)	Noise Voltage: 1500V Pulse Width: 50ns, 500ns, 1μs Rise Time: 1ns (via noise simulator)
Electrostatic Discharge Immunity	4kV IEC 61000-4-2
Noise Immunity (First transient burst noise)	Power Line : 2kV IEC61000-4-4 COM Port : 1kV IEC61000-4-4

Important

- ***When using any of the PL's optional devices, be sure to check that device's specifications for any special conditions or cautions that may apply to its use.***
- ***When using a full sized expansion board, be sure to check its dimensions and shape, since they will affect the PL's environment specifications, such for vibration, etc.***
- ***Since the PL unit's hard disk drive (HDD) is a consumable item, i.e. it has a limited lifetime, be sure to back up its data regularly and prepare a spare HDD unit.***
- ***The Hard Disk lifetime given here may be reduced due to unforeseen environmental factors, however, generally speaking, at an operating temperature of 20°C the disk should last for 20,000 hours (of operation) or approximately 5 years, whichever comes first.***
- ***Using the Hard Disk in an environment that is excessively hot and/or humid will shorten the disk's usage lifetime. A maximum wet bulb temperature of 29°C is recommended. This is equivalent to the following data.***

Temperature	Humidity
at 35°C	no higher than 64%RH
at 40°C	no higher than 44%RH

2.1.3 Environmental

	PL-B920	PL-B921
Grounding	Exclusive grounding: Use your country's applicable standard. [*1]	
Installation Method	Independently mounted (Inside a solid, flat panel)	
Cooling Method	Via cooling pipes and power supply cooling fan	
Weight	8.0kg (17.6lb) or less (with HDD and FDD installed)	7.0kg (15.4lb) or less (with HDD and FDD installed)
External Dimensions	W 330mm[12.99in.] x H 271mm[10.76in.] x D 162mm[6.38in.] (excluding projections)	W 330mm[12.99in.] x H 271mm[10.76in.] x D 116mm[4.75in.] (excluding projections)
Dimensions Including Full-sized Cover	W 385mm[15.16in.] x H 271mm[10.76in.] x D 162mm[6.38in.] (excluding projections)	W 385mm[15.16in.] x H 271mm[10.76in.] x D 116mm[4.75in.] (excluding projections)
Dimensions Including Mirror Disk Unit	W 330mm[12.99in.] x H 271mm[10.76in.] x D 166mm[6.54in.] (excluding projections)	W 330mm[12.99in.] x H 271mm[10.76in.] x D 166mm[6.54in.] (excluding projections)
Dimensions Including RS-232C/RS-485 Conversion Unit	W 330mm[12.99in.] x H 271mm[10.76in.] x D 16mm[6.38in.] (excluding projections and cable)	W 330mm[12.99in.] x H 271mm[10.76in.] x D 138mm[5.43in.] (excluding projections and cable)

*[*1] Reference Refer to **4.3.3 Grounding Cautions**.*

2.2 Functional Specifications

2.2.1 General

CPU		Pentium III 700MHz Processor		
DRAM (SDRAM DIMM)		128MB Standard (2 DIMM sockets: max. of 256MB)		
BIOS		AWARD PC/AT Compatible		
Secondary Cache Memory		256KB (built-in)		
Graphics		VGA (640 x 480 dots) to XGA (1024 x 768 dots) VESA 16 colors/256 colors/64K colors		
Video Memory		UMA type		
Interfaces	**Video**	Analog RGB I/F		
	Touch Panel	COM4 : uses Mouse Emulator USB: uses USB Touch Panel Driver		
	Serial	RS-232C (w/FIFO)	COM1	D-Sub 9 pin (male)
			COM2	D-Sub 9 pin (male) (RI/+5V Changeover)
			COM3	D-Sub 9 pin (male) (RI/+5V Changeover)
	Printer	Centronics Standard (ECP/SPP/EPP equivalent) D-sub 25 pin, female		
	Keyboard	PS/2 Interface (mini DIN 6 pin, female) side 1port		
	Mouse	PS/2 Interface (mini DIN 6 pin, female) side 1port		
	RAS	RAS Interface (D-sub 25 pin, male)		
	Disk I/F	FDD Unit	Side Access/ 2 modes/ 3.5 inch FD	
			Front Access/ 2 modes/ 3.5 inch FD (Available for only PL-B920)	
		IDE I/F	2.5 inch HDD I/F CD-ROM Drive (PL-DK200) Mirror Disk Drive (PL-MD200-HU01)	
	USB[*1]	USB 1.1 compatible, 1channel, side 3 ports		
	LAN[*1]	Satisfies IEEE802.3 standard (10BASE-T/100BASE-TX Autochangeover)		

**1 BIOS setting must be changed.* Reference *5.2.5 Integrated Peripherals*

2.2.2 Expansion Slots

	PL-B920 (4-slot type)	PL-B921 (2-slot type)	Board Size		Slot Pitch	Actual Thickness of Expansion Board
			Without Full-sized cover	With Full-sized cover		
1st slot	ISA	ISA	163 x 122 mm	163 x 122 mm	20 mm	Less than 13 mm
2nd slot	PCI	PCI	250 x 122 mm	250 x 122 mm	25 mm	Less than 18 mm
	ISA	ISA		338 x 122 mm		
3rd slot	PCI/ISA	None	250 x 122 mm	338 x 122 mm	25 mm	Less than 18 mm
4th slot	ISA	None	250 x 122 mm	338 x 122 mm	20 mm	Less than 13 mm
Power Supply	5V: 4A, 12V: 1A (total for 4 slots)	5V: 2A, 12V: 0.5A (total for 2 slots)				

Note: **For the 2nd and 3rd slots on the PL-B920, and the 2nd slot on the PL-B921, either a PCI or an ISA type expansion board can be used.**

Reference For PL dimensions when using the full-sized cover, refer to **1-3 Optional Items**.

2.2.3 Clock (RTC) Accuracy

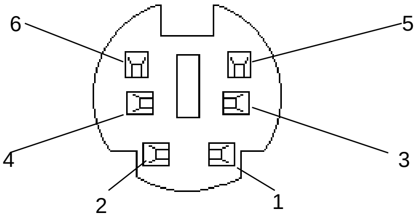
Clock(RTC) accuracy	**±180 seconds per month**

The PL unit's built-in clock (RTC) has a slight error. At the PL's specified ambient temperature and with the power turned OFF the error is +180 seconds per month. However, ambient temperature fluctuations and the age of the unit may increase this error to +300 seconds per month. If the PL unit's RTC clock accuracy is vital to system performance, regular adjustment of this clock is required.

2.3 Interface Specifications

2.3.1 Keyboard Interface (KEYBOARD)

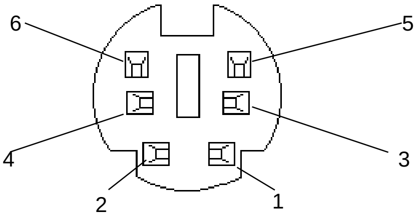
Mini - DIN 6 pin (Female)



Pin No.	Signal Name
1	KEY DATA
2	NC
3	GND
4	+5V
5	KEY CLK
6	NC
SHIELD	GND

2.3.2 Mouse Interface (MOUSE)

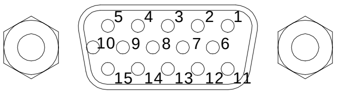
Mini - DIN 6 pin (Female)



Pin No.	Signal Name
1	Mouse DATA
2	NC
3	GND
4	+5V
5	Mouse CLK
6	NC
SHIELD	GND

2.3.3 CRT Interface

D-sub 15 pin (Female)

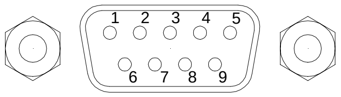


Screw Size: (4-40): Inch Type

Pin No.	Signal Name	Condition
1	Analog R	R signal input
2	Analog G	G signal input
3	Analog B	B signal input
4	Reserved	NC
5	GND	Digital signal ground
6	Return R	R signal GND
7	Return G	G signal GND
8	Return B	B signal GND
9	Reserved	NC
10	GND	Digital signal ground
11	Reserved	NC
12	Reserved	NC
13	H-SYNC	Horizontal signal Input
14	Y-SYNC	Vertical signal input
15	Reserved	NC

2.3.4 RS-232C Interface (COM1/COM2/COM3)

D-sub 9 pin (Male)



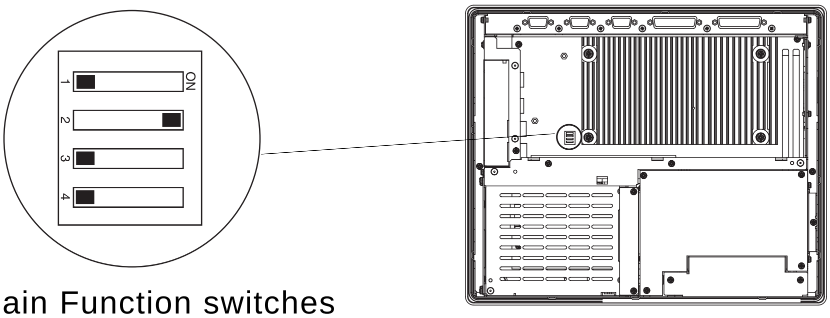
Screw Size: (4-40): Inch Type

Pin No.	Signal Name	Pin No.	Signal Name
1	CD	**6**	DSR
2	RXD	**7**	RTS
3	TXD	**8**	CTS
4	DTR	**9**	RI/+5V
5	GND		

Important

The GND terminal is the signal ground. Be sure to connect it with the cable's opposite side SG terminal.

Number 9 pin's [RI/+5V] changeover is COM2 and COM3. COM1 becomes [RI]. To set the changeover for COM2 and COM3, remove the PL's rear maintenance cover and locate the main function switches, next to the PL's circuit board. To change COM2, set main function switch SW2 to ON, and COM2 will change to +5V. The factory setting is OFF and [RI]. To change COM3, simply turn SW3 to ON. Reference ***3.1.1 Removing the Rear Maintenance Cover***



Main Function switches

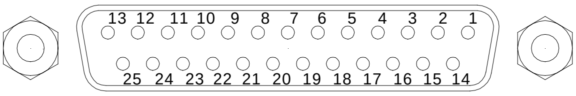
PL-B920(4Slot)

Important

- ***Main Function switches SW1 and SW4 are reserved (OFF). Be sure not to change these settings.***
- ***Be sure to confirm what settings will be used by the other device and set the dip switches accordingly. Failure to do so can result in a unit malfunction or damage.***
- ***Whenever changing the PL Main Function switches, be sure to first turn the PL's power supply OFF. Failure to do so can cause a PL malfunction.***

2.3.5 Printer Interface (LPT1)

D-sub 25 Pin (Female)



Screw Size: (4-40): Inch Type

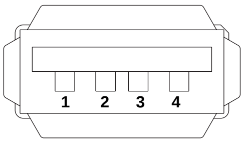
Pin No.	SPP/ECP Mode Signal Name	EPP Mode Signal Name	Direction	Electrical Specif.	Pin No.	SPP/ECP Mode Signal Name	EPP Mode Signal Name	Direction	Electrical Specif.
1	$\overline{\text{STRB}}$	$\overline{\text{WRITE}}$	In/Output	O.D/T.S	14	$\overline{\text{AUTOFD}}$	$\overline{\text{DSTRB}}$	In/Output	O.D/T.S
2	DATA0	DATA0	In/Output	T.S	15	$\overline{\text{ERROR}}$	$\overline{\text{ERROR}}$	Input	TTL
3	DATA1	DATA1	In/Output	T.S	16	$\overline{\text{INIT}}$	$\overline{\text{INIT}}$	In/Output	O.D/T.S
4	DATA2	DATA2	In/Output	T.S	17	$\overline{\text{SLCTIN}}$	$\overline{\text{ADSTRB}}$	In/Output	O.D/T.S
5	DATA3	DATA3	In/Output	T.S	18	GND	GND		
6	DATA4	DATA4	In/Output	T.S	19	GND	GND		
7	DATA5	DATA5	In/Output	T.S	20	GND	GND		
8	DATA6	DATA6	In/Output	T.S	21	GND	GND		
9	DATA7	DATA7	In/Output	T.S	22	GND	GND		
10	$\overline{\text{ACKNLG}}$	$\overline{\text{ACKNLG}}$	Input	TTL	23	GND	GND		
11	BUSY	$\overline{\text{WAIT}}$	Input	TTL	24	GND	GND		
12	PE	PE	Input	TTL	25	GND	GND		
13	SLCT	SLCT	Input	TTL					

O.D.:Open Drain T.S.:3-state Input Output TTL:TTL Input

Note: **Pins 1, 4, 16 and 17 will become O.D when the SPP mode specification is used. If the mode changes to ECP or EPP, these pins will become T.S**

2.3.6 USB Interface (USB)

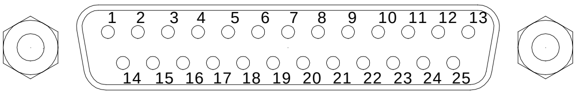
Receptacle



Pin No.	Name
1	Vcc
2	- Data
3	+ Data
4	GND

2.3.7 RAS Interface (RAS)

D-Sub 25 pin (Male)



Screw Size: (4-40): Inch Type

Pin No.	Signal Name	Pin No.	Signal Name
1	GND	14	GND
2	+5V (max. 100mA)	15	+5V
3	+12V (max. 100mA)	16	NC
4	NC	17	NC
5	RESET INPUT (+)	18	NC
6	DIN 0 (+)	19	NC
7	DOUT (-)	20	NC
8	DOUT (+)	21	LAMP OUT (-)
9	ALARM OUT (-)	22	LAMP OUT (+)
10	ALARM OUT (+)	23	NC
11	RESET INPUT (-)	24	DIN1 (-)
12	DIN 0 (-)	25	NC
13	DIN 1 (+)		

Important

Be sure to use only the rated voltage level when using the No. 2 [+5V] and No. 3[12V] for external power output. Failure to do so can lead to a unit malfunction or accident.

Note:

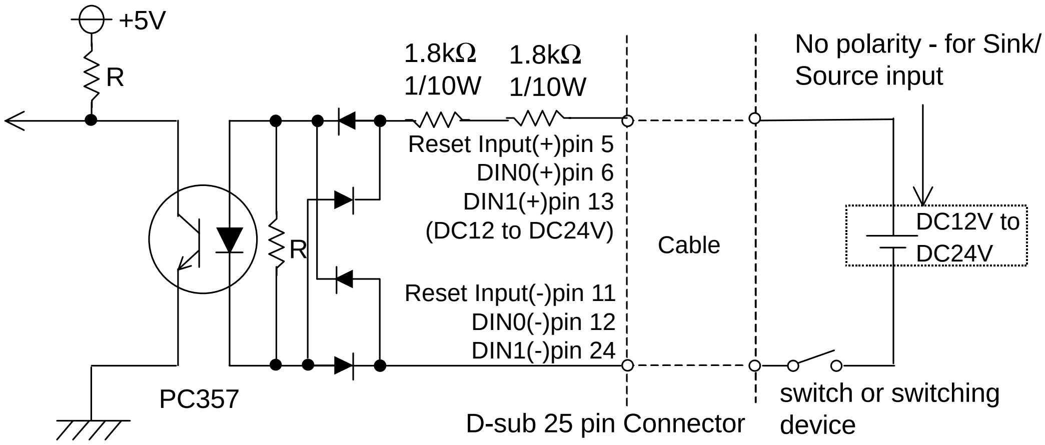
For detailed RAS Feature information,

Reference *Appendix 2 RAS Feature.*

■ External Input Signal (Dual use of DIN, Remote Set Input Port)

Input Voltage	DC12V to DC24V
Input Current	7mA
Operating Voltage	ON voltage: 9V (min), OFF voltage:3V (max)
Isolation Method	Via photocoupler

(Interface Circuit) (Connection Example)

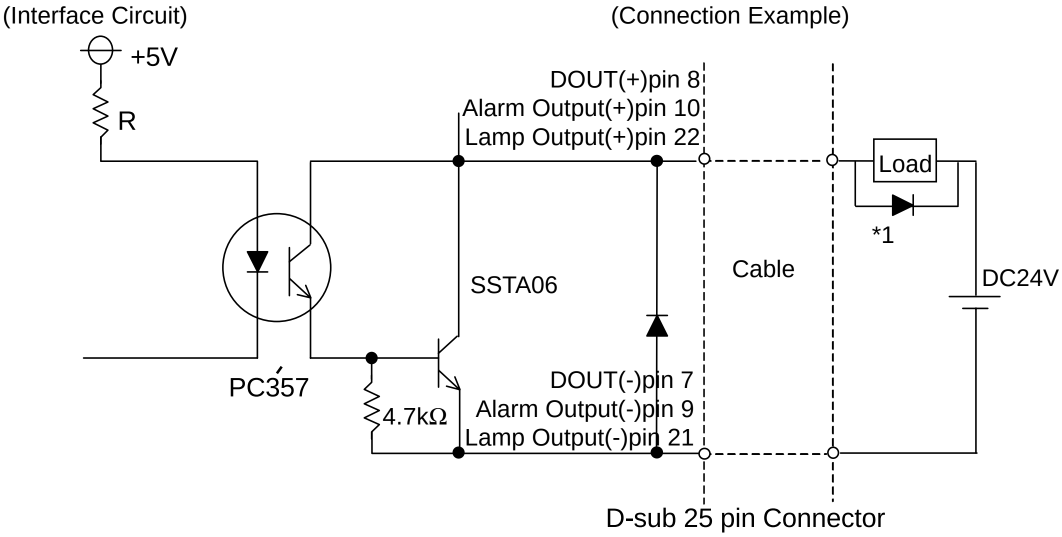


Important

- ***General Purpose Input (DIN) level must be 1.5S or longer to be detected.***
- ***Be sure the voltage value between terminals is controlled via the input voltage, so that the PL is operated within its recommended range. If the input voltage exceeds this range, a malfunction or PL damage may occur.***

■ External Output Signal (DOUT, Alarm Output, Lamp Output Port)

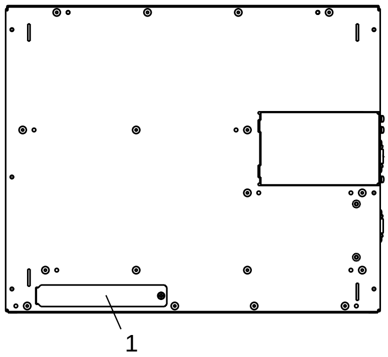
Rated Load Voltage	DC12V to DC24V
Maximum Load Current	100mA/point
Maximum Voltage Drop between Terminals	1.5V (at 100mA load current)
Isolation Method	Via photocoupler



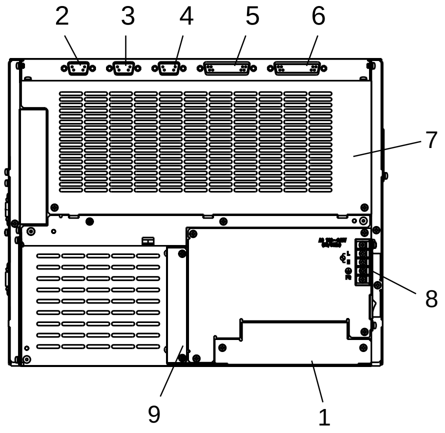
Important

- ***Be sure to operate the unit within its maximum load current. If the maximum load current exceeds this range, a malfunction or PL damage may occur.***
- ***Design your electrical system by adding the load current and voltage values to the terminal voltage. If load current value used is large, a maximum voltage of 1.5V will exist between the terminals.***
- ***When connecting an induction load, be sure to connect the above drawing's protection diode(*1).***

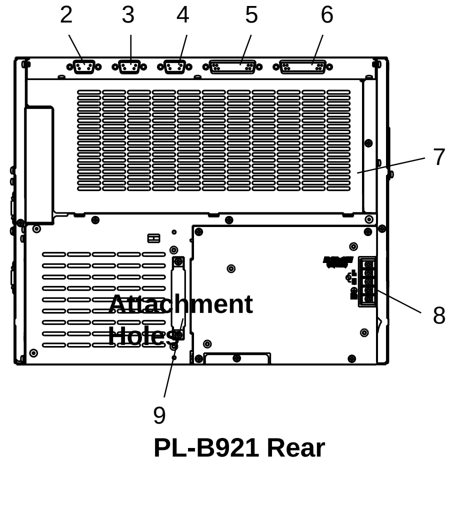
2.4 PL Part Names and Features



Front



PL-B920 Rear



PL-B921 Rear

1: Front Face FDD Unit Attachment Slot

Digital's optional PL-FD210 is attached here.(Only for PL-B920)

2: RS-232C Connector (COM1)

3: RS-232C Connector (COM2) (RI/+5V changeover)

4: RS-232C Connector (COM3) (RI/+5V changeover)

These RS-232C interfaces (D-sub 9 pin male connectors), allow communication with other computers and connection to peripheral devices.

5: Printer Connector (LPT1)

Centronics standard interface (D-sub 25 pin female connector), which connects a parallel device, such as a printer .(supports ECP/SPP/EPP)

6: RAS Connector (RAS)

Interface for DIN, DOUT, Watchdog, and Remote Reset. (D-sub 25 pin male connector)

7: Rear Maintenance Cover

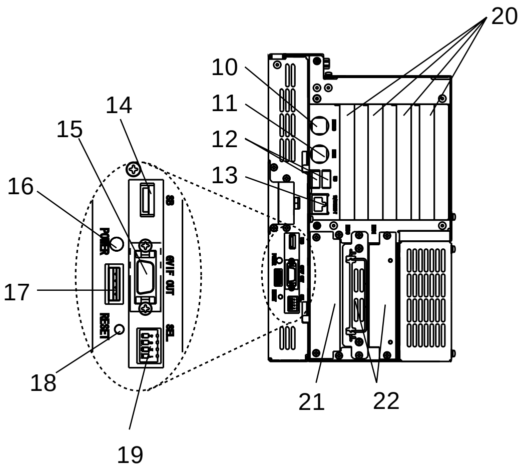
When optional DIM modules or Expansion Boards are installed, this cover must be removed.

8: Power Terminal Block

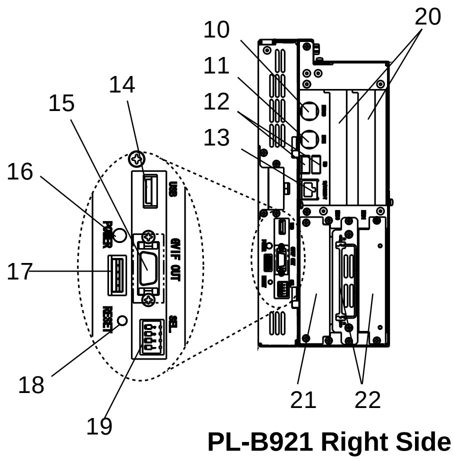
The PL's AC100V/240V power cord terminals are connected here.

9: IDE I/F Cover

To connect the optional CD-ROM drive unit (PL-DK200) and Mirror Disk Unit (PL-MD200-HU01), remove this cover and use this connector.



PL-B920 Right Side



PL-B921 Right Side

10: Keyboard Connector (KEYBOARD)

A PS/2 compatible keyboard can be connected here.

11: Mouse Connector (MOUSE)

A PS/2 compatible mouse can be connected here.

12: USB Connector (USB)

A USB 1.1 compatible device is connected here.

13: LAN Connector (10/100BASE-T) (10BASE-T/100BASE-TX Autochangeover)

Network Interface (meets IEEE802.3 standard).

14: USB Connector (USB)

A USB 1.1 compatible device is connected here.

15: Standard Display Connector (GVIF OUT)

Used to connect Digital’s Standard Display Cable. (PL-CB200-*M)

16: Power LED (POWER)

This LED indicates both the RAS monitor feature’s status and the PL’s power status. An RAS alarm will cause this lamp to blink.

Reference *Appendix 2 RAS Feature*

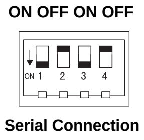
17: Power LED Output Connector

Used when installing the PL in the optional 19” mount to provide power LED output.

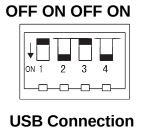
18: Hardware Reset Switch (RESET)

Pushing this switch resets the PL.

19: Dip Switches (SEL.)



Serial Connection



USB Connection

Be sure to set the DU unit’s dip switches to the same setting as the PL.

Reference *PL-X920 Series Standard Display PL-DU6900/PL-DU7900 Users Manual.*

20: Expansion Slots

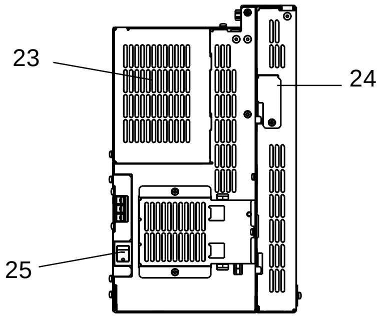
Slots for PCI/ISA Bus compatible expansion boards.

21: Side Mount FDD Slot

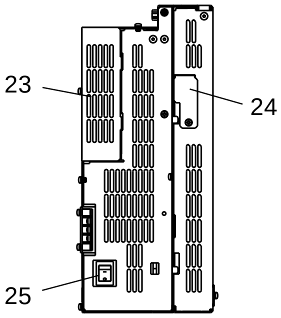
Houses the FDD (PL-FD200) unit.

22: HDD/FFD/CF Card Unit Expansion Slot

Houses either the HDD, FFD or the CF Card unit.



PL-B920 Left Side



PL-B921 Left Side

23: Half Cover

When installing an optional DIM module or expansion board, this cover is removed.

24: Analog RGB Connector Cover

Provides an analog RGB interface. Remove this cover when attaching an RGB (CRT) monitor to the PL.

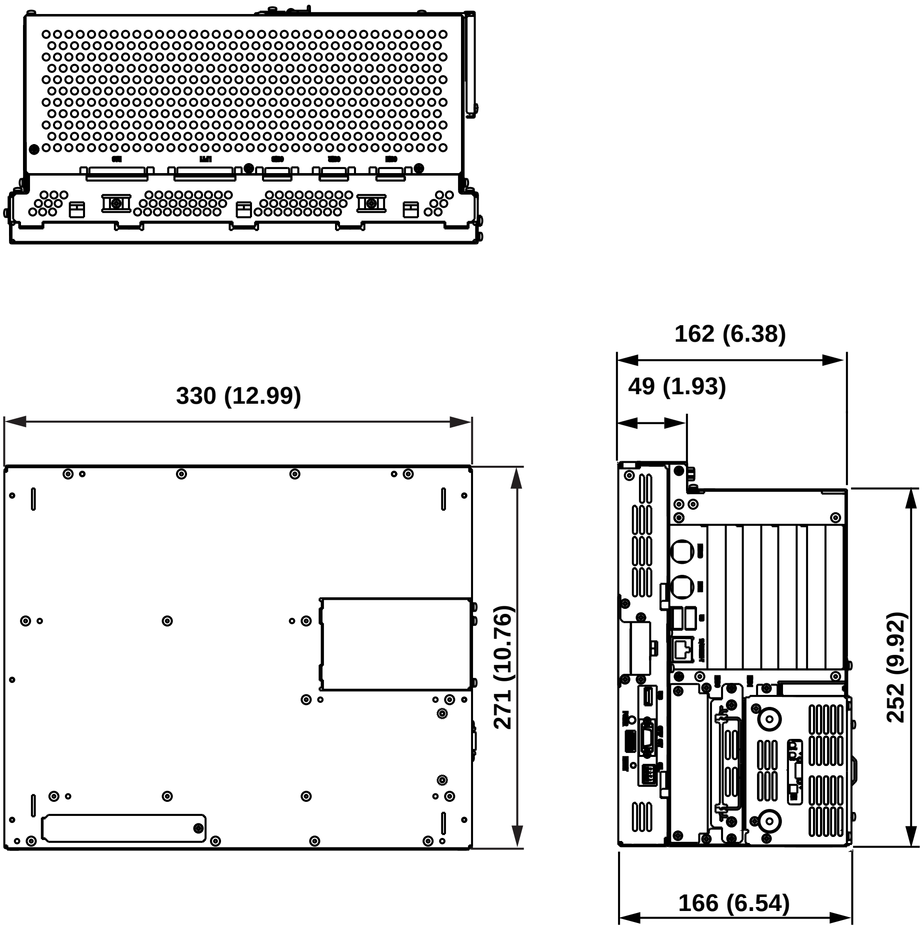
25: Power Switch

Turns the PL main unit ON or OFF.

2.5 PL External Dimensions

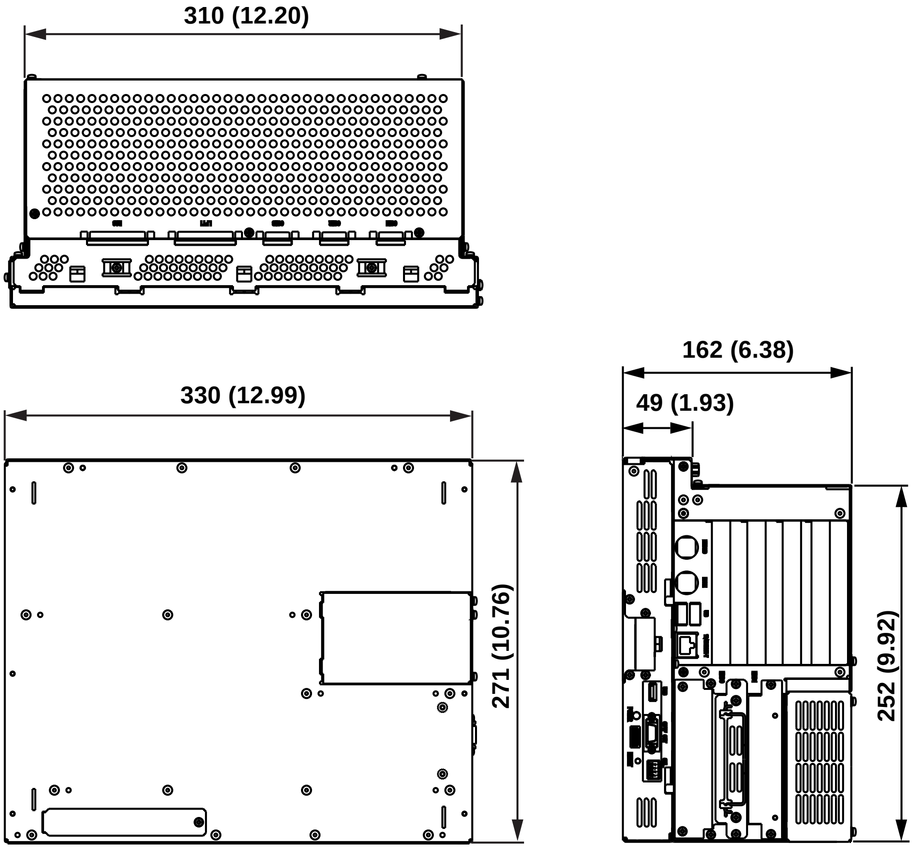
2.5.1 PL-B920 Dimensions

(Unit: mm (in.) - excluding projections)



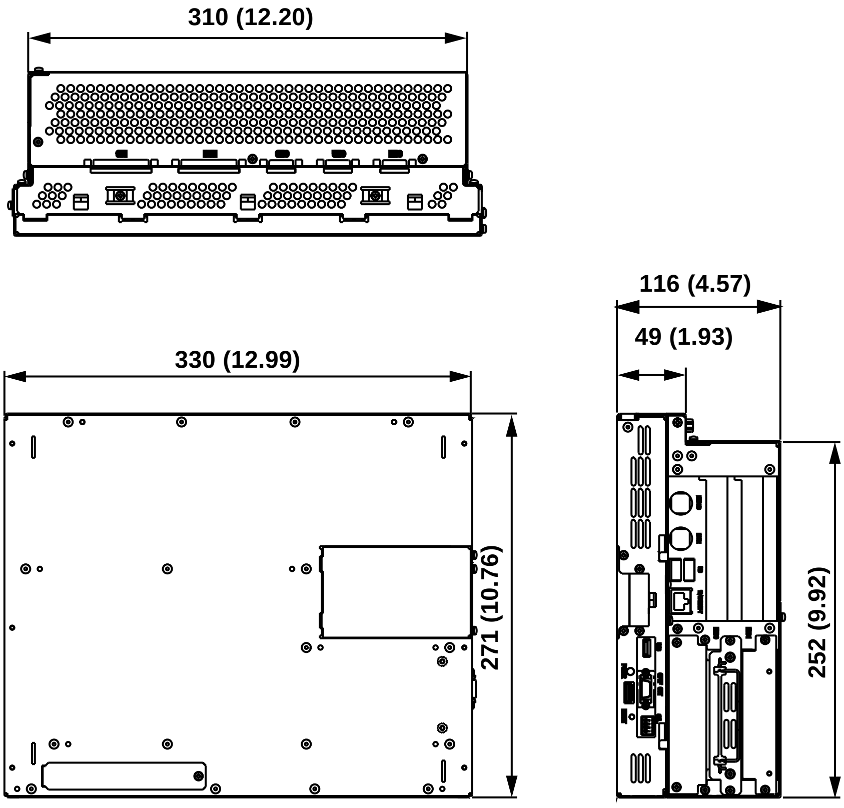
2.5.2 PL-B920 and Mirror Disk Unit Dimensions

(Unit: mm (in.) - excluding projections)



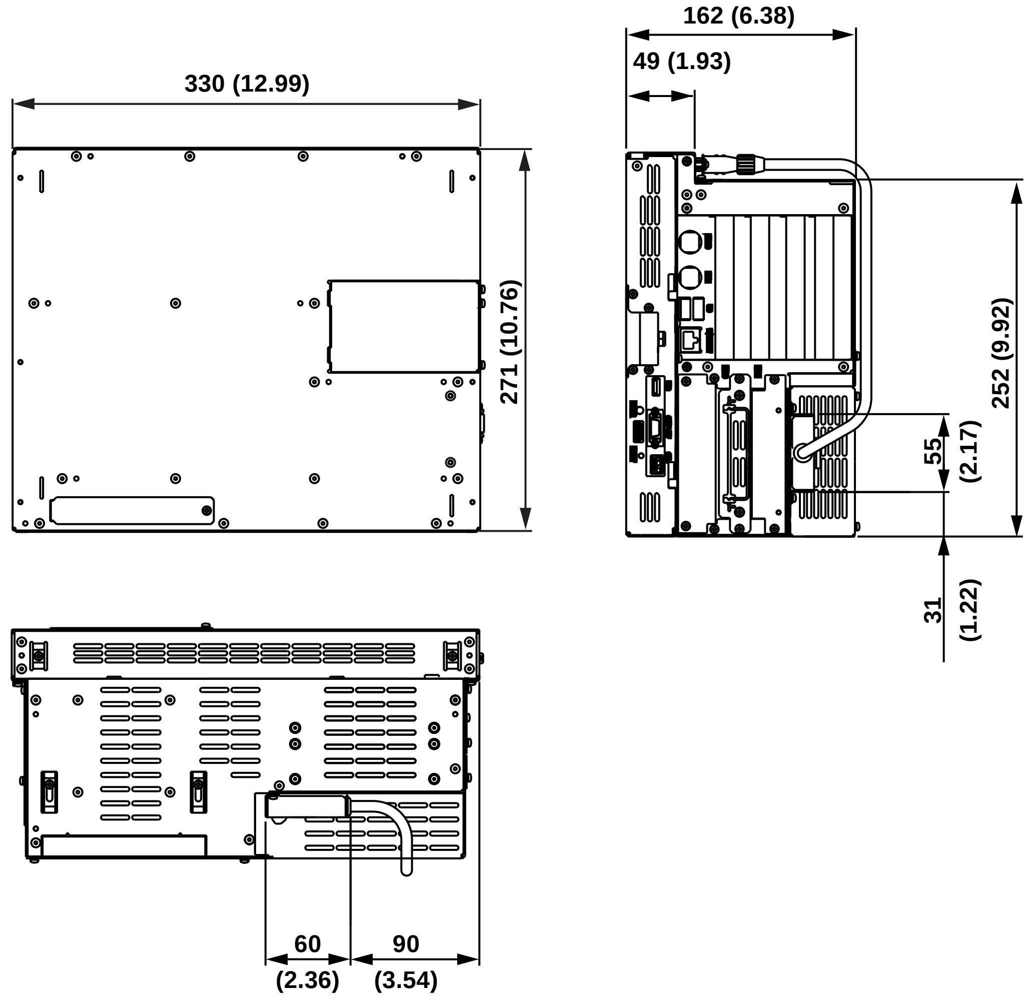
2.5.3 PL-B920 and RS-232C/RS-485 Conversion Unit Dimensions

(Unit: mm (in.) - excluding projections)



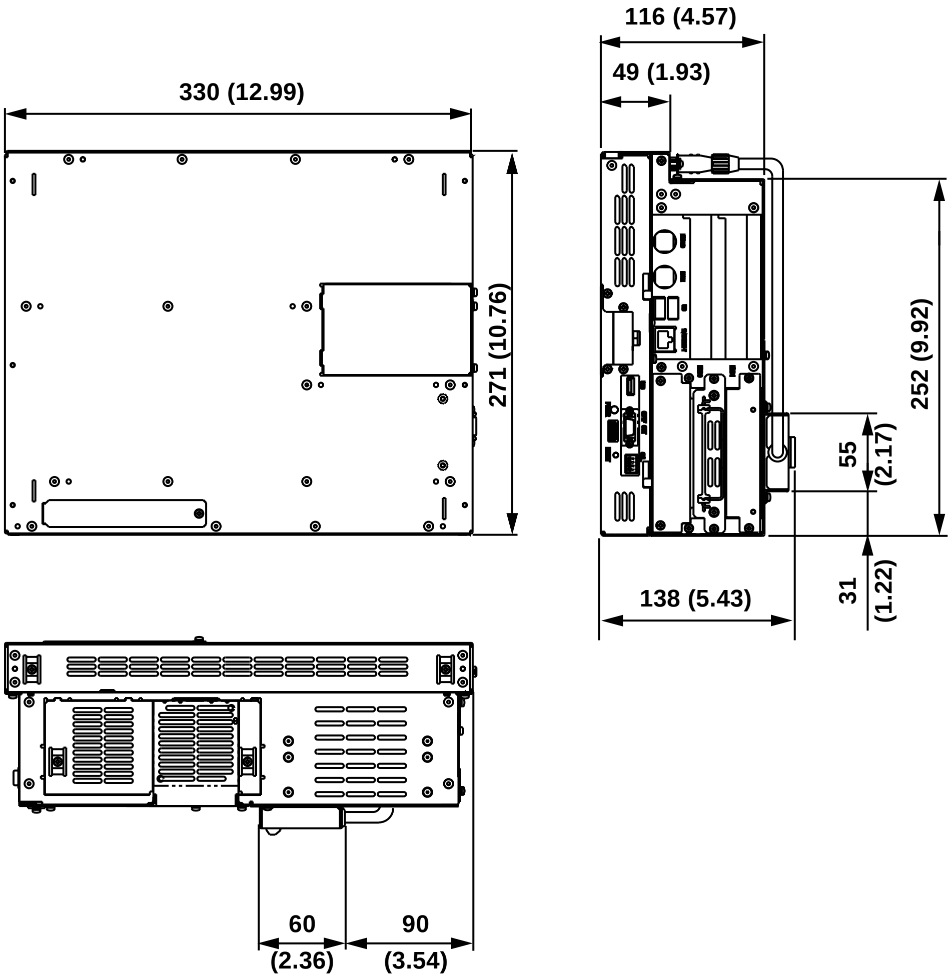
2.5.4 PL-B921 Dimensions

(Unit: mm (in.) - excluding projections)



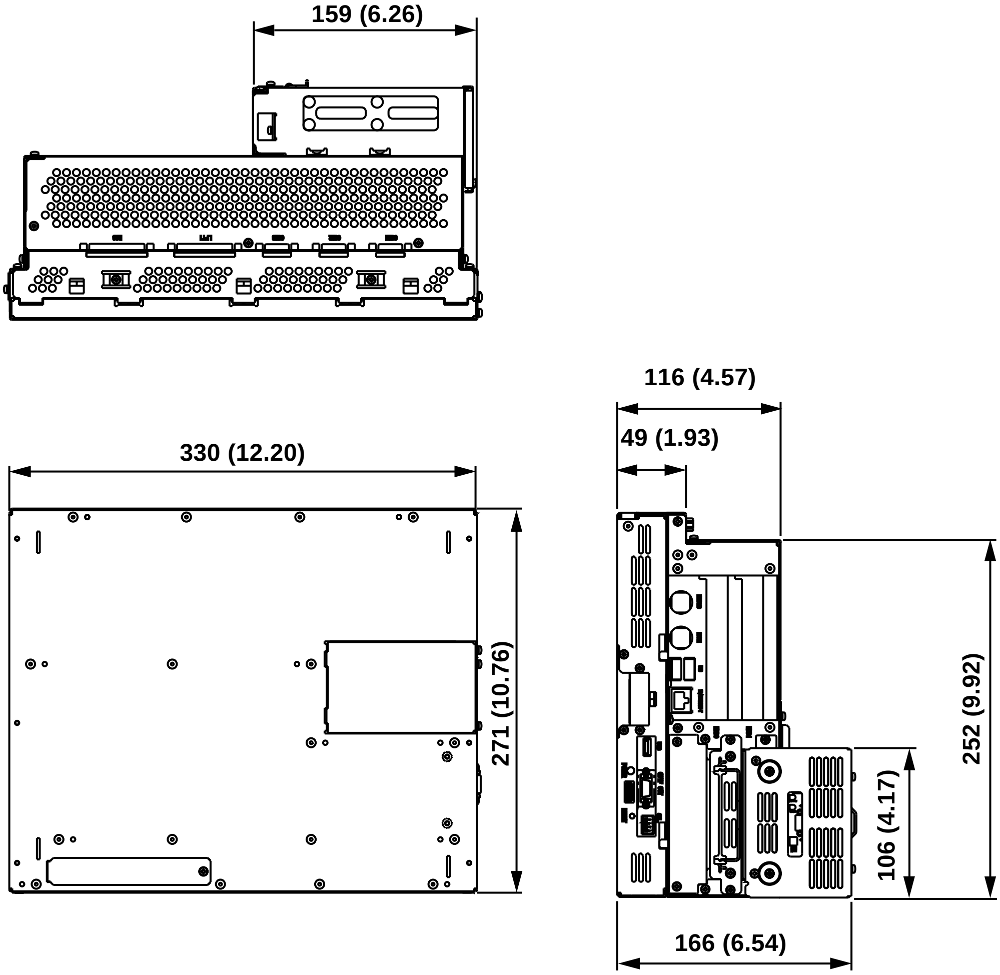
2.5.5 PL-B921 and Mirror Disk Unit Dimensions

(Unit: mm (in.) - excluding projections)



2.5.6 PL-B921 and RS-232C/RS-485 Conversion Unit Dimensions

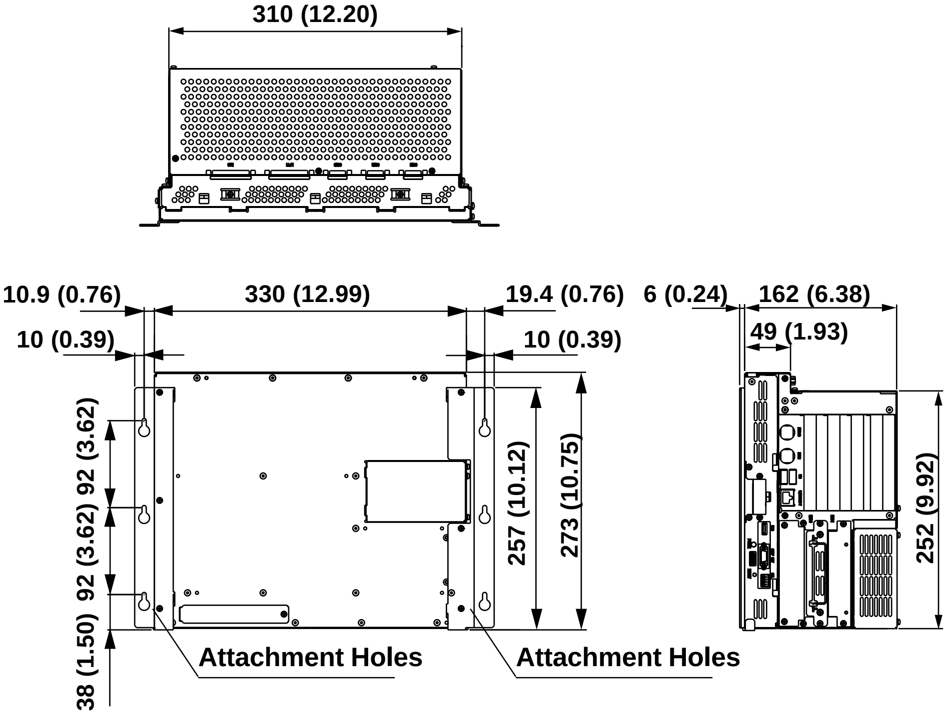
(Unit: mm (in.) - excluding projections)



2.5.7 Installation Fasteners

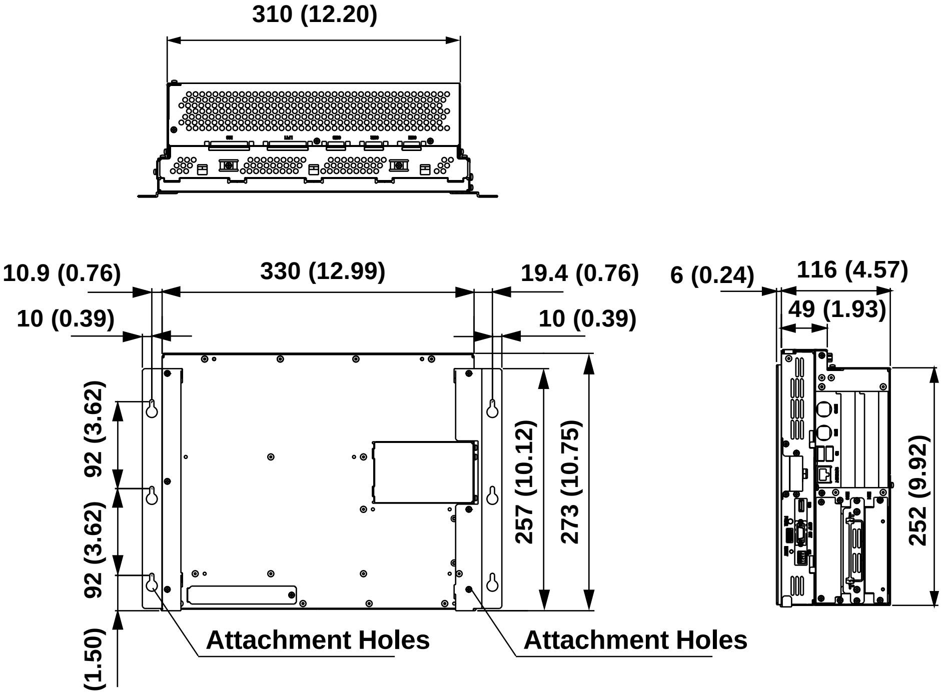
■ PL-B920 (4-slot type) (Unit: mm - excluding projections)

The attachment screw measurements are the same for both units.

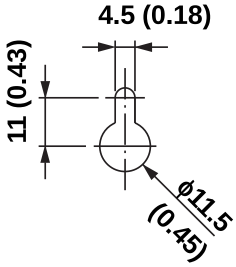


■ PL-B921 (2-slot type) (Unit: mm - excluding projections)

The attachment screw measurements are the same for both units.



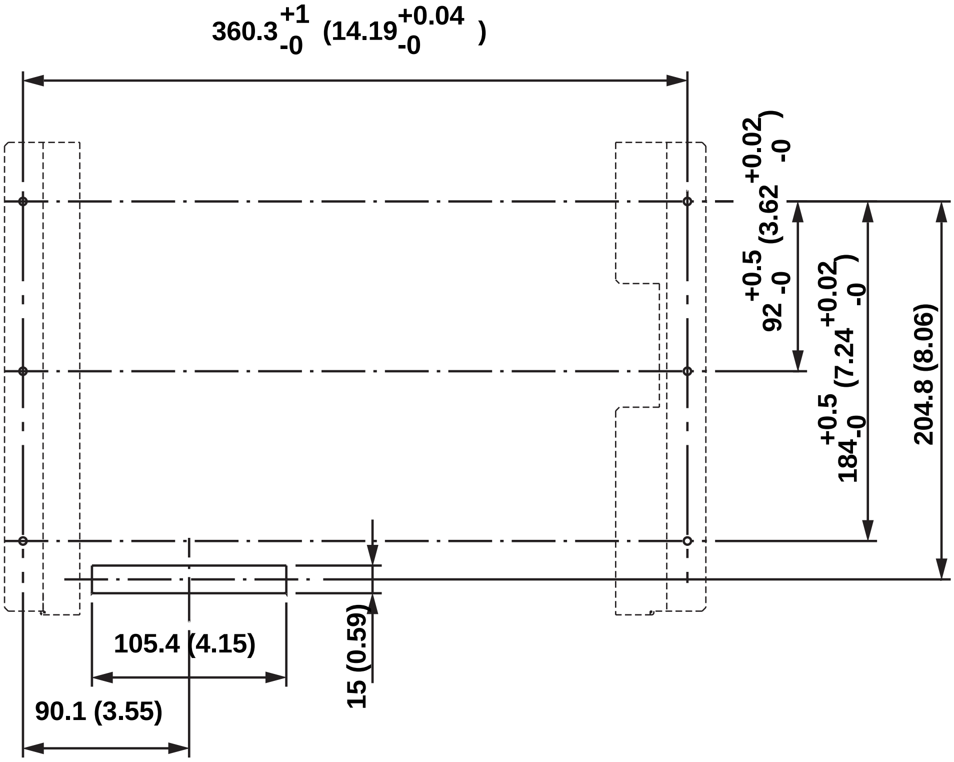
■ Attachment Bracket Attachment Holes (Detail)



■ Drilling Attachment Bracket Attachment Holes

Use the following dimension drawing to drill the attachment bracket attachment holes in the installation panel. Use M4 size screws for the screw holes.

(Unit: mm(in.) - excluding projections)



Front View

- ***Depending on the panel's material and design, the panel's installation surface may need to be strengthened. If high levels of vibration are expected and the PL's installation surface (i.e. an operation panel's door, etc.) can move (i.e.open or close) due consideration should be given to the PL's weight.***

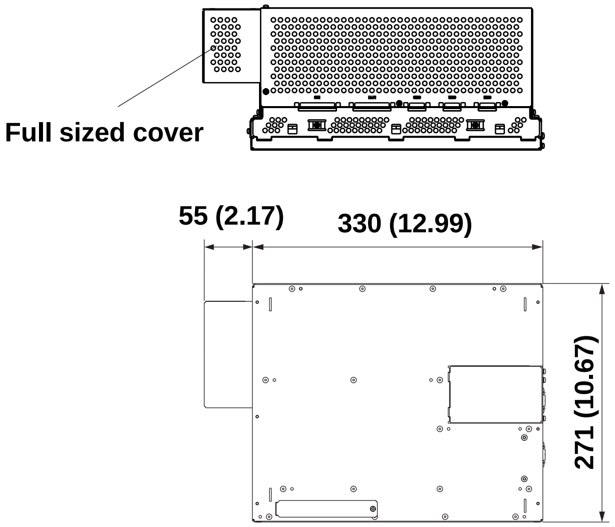
Reference ***2.1.3 Environmental***

- ***Be sure all installation tolerances are maintained to prevent the unit from falling out of its installation panel.***

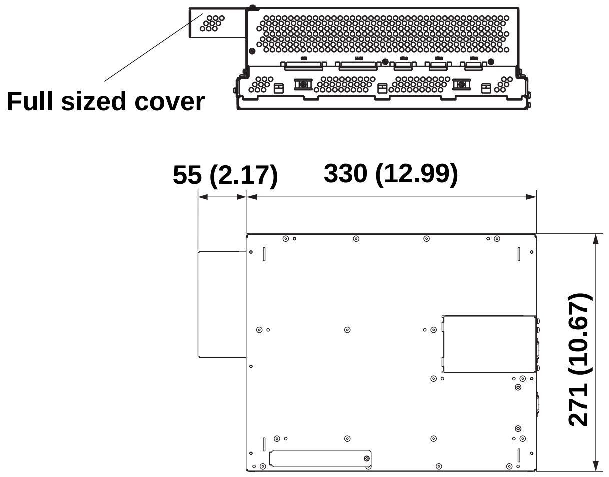
2.5.8 Full-Sized Cover Attachment Dimensions

■ PL-B920 (4-slot type)

(Unit: mm - excluding projections)



■ PL-B921 (2-slot type)



Note: **There are two types of full-sized covers - one for 4 slot units and one for 2 slot units.**

2-Slot Type	PL-FC200
4-Slot Type	PL-FC210

Reference *1.3 Optional Items*

Important

- ***Prior to installing a full-sized board and the PL's full-sized cover (PL-FC200/PL-FC210), be sure that the PL is mounted in its attachment panel/ cabinet. Due to dimension differences, the PL-FC200 or a full sized expansion board can not be attached prior to installing the PL into a panel.***
- ***When using a full sized expansion board, be sure to check its dimensions and shape, since they will affect the board's environment specifications, such as for vibration, etc.***

Chapter 3 Installing Optional Units and Expansion Boards

A wide variety of optional units and expansion boards made by Digital can be installed in the PL, as well as a number of commercially available PCI-bus or ISA-bus compatible boards. This chapter describes how to install these products in the PL.

3.1 Installation

The following explanation pages describe the installation procedures for the PL's DIM module (PL-EM500/PL-EM128), FDD unit (PL-FD200/PL-FD210), HDD unit (PL-HD220), expansion boards, CD-ROM drive unit (PL-DK200), Standard Display (PL-DU6900/PL-DU7900) and the PL's cooling fan.

For information about the installation of other option units, please refer to those unit's individual [Operation Instructions].

WARNING

To avoid electric hazards, be sure to turn the PL's power OFF before installing any optional units or expansion boards.

Important

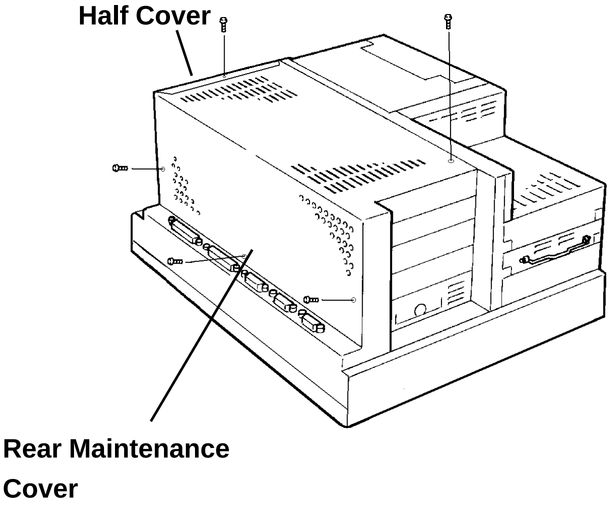
- ***Use a screwdriver to loosen or tighten the screws. Be careful not to tighten screws to tightly, since it may damage the equipment.***
- ***Be careful when removing or inserting any screws that they do not fall inside the PL.***

3.1.1 Removing the Rear Maintenance Cover

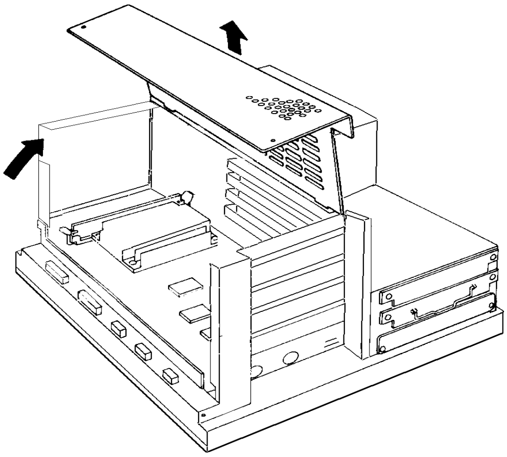
Important

Be sure to handle the rear maintenance cover carefully, since it is made of aluminum and is easily bent.

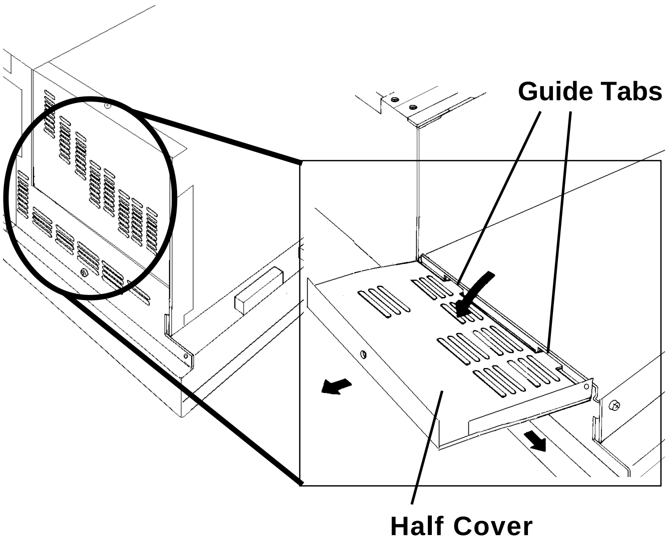
- **PL-B920 Units** **(4 -slot type)**



1) Unscrew the five (5) attachment screws used to hold the rear maintenance cover and half cover.

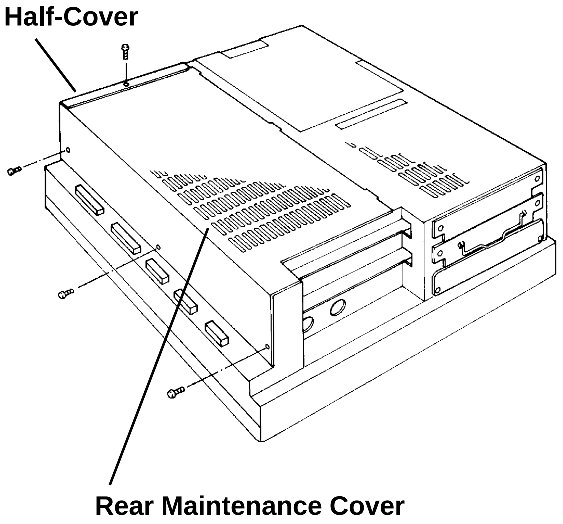


2) Remove the rear maintenance

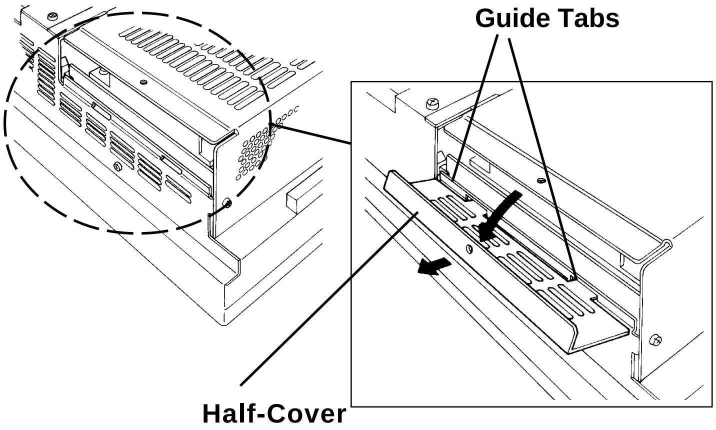


3) Pivot the half cover open and lift up slightly to free the guide tabs. Next, remove the half cover.

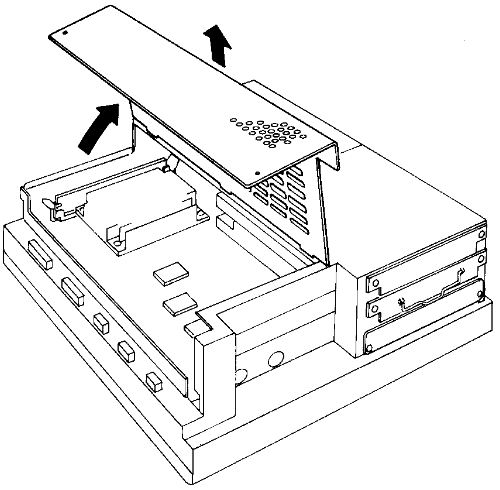
■ PL-B921 Unit (2-slot type)



1) Unscrew the four (4) attachment screws used to hold the rear maintenance cover and half cover.



2) Pivot the half cover open and slide the cover slightly to the side to free the guide tabs. Next, remove the half cover.



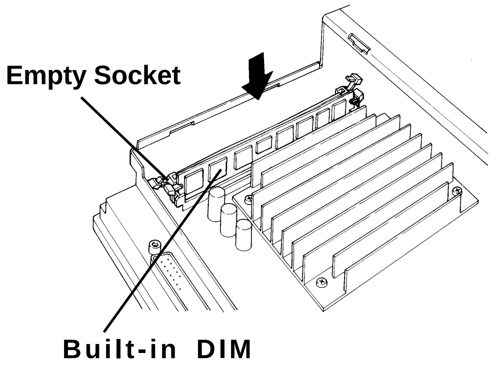
3) Remove the rear maintenance cover.

3.1.2 Installing the DIM Module (PL-EM500 / PL-EM128)

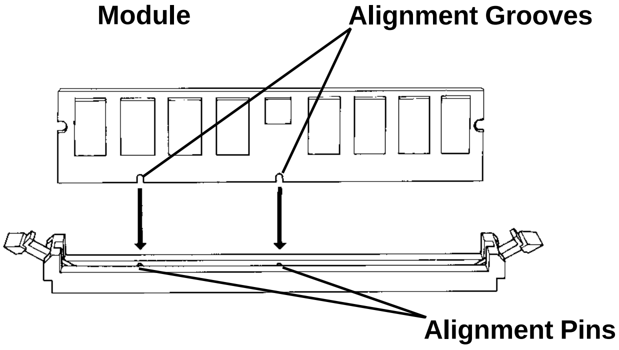
Important

- ***Since DIM module sockets are fragile and break easily, be sure to install the DIM module carefully.***
- ***Do not change the factory installed DIM module's socket position.***

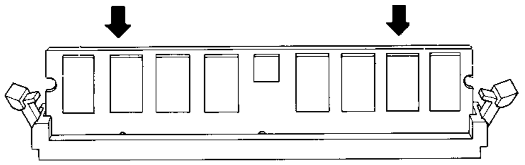
The PL comes with a single, 128MB DIM module pre-installed. There is one more empty socket that can be used to expand you PL unit's memory. Use the following procedure to install a second DIM module in that socket.



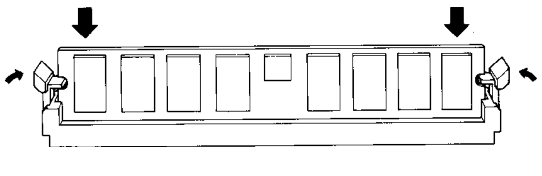
1) Install the DIM expansion module in the empty socket.



2) Position the Alignment Grooves so that they fit the Alignment Pins.



3) Insert the DIM module into the DIM module socket.

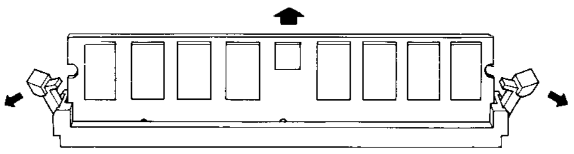


4) Push the DIM module down until the ejector tabs lock.

5) Replace the rear maintenance cover and the half cover and secure them in place with their attachment screws.

◆ To Remove the DIM Module

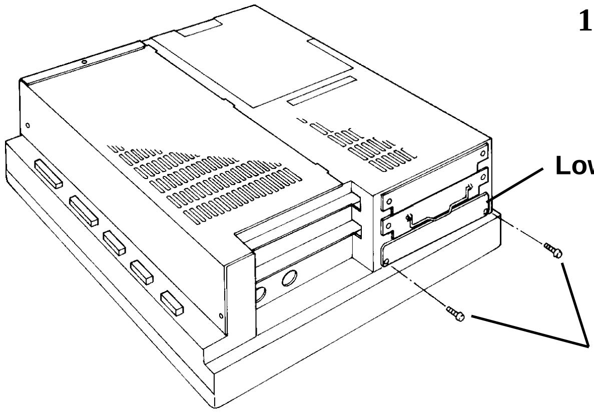
To remove a module, press down on the socket's ejector tabs to release the module.

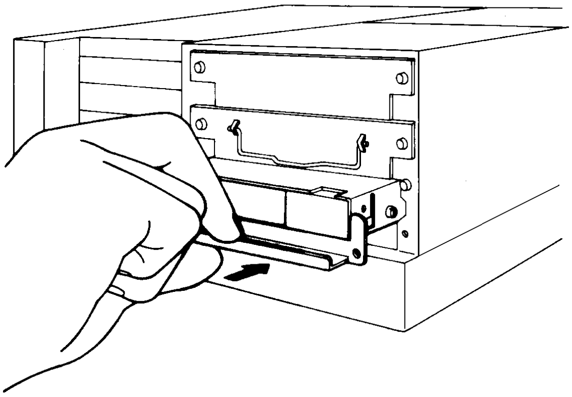


3.1.3 Installing the FDD Unit (PL-FD200)

The PL-FD200 and the PL-FD210 cannot be used at the same time.

1) Remove the two (2) attachment screws from the lowermost Expansion Slot Cover, and remove the cover.





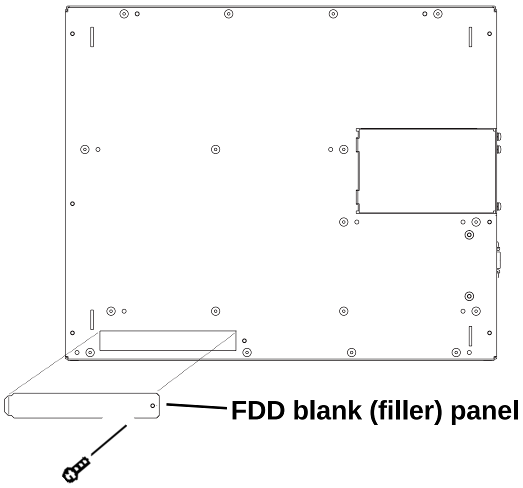
2) Insert the FDD unit so that its guide grooves fit the slot's guide ways. Push the unit in until its rear connector is connected securely.

3) Secure the unit in place with its two(2) attachment screws.

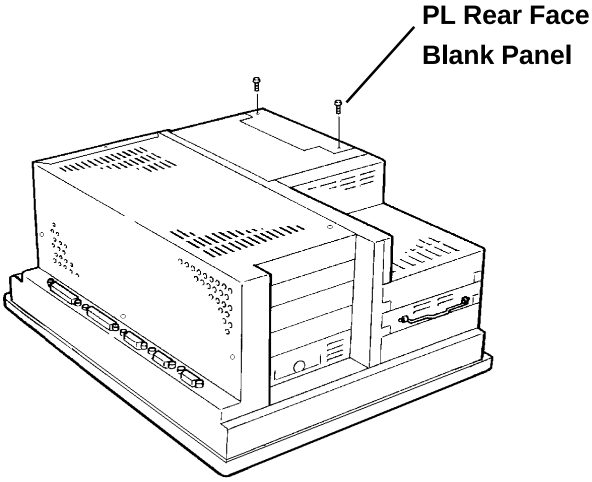
3.1.4 Installing the FDD Unit (PL-FD210)

Note:

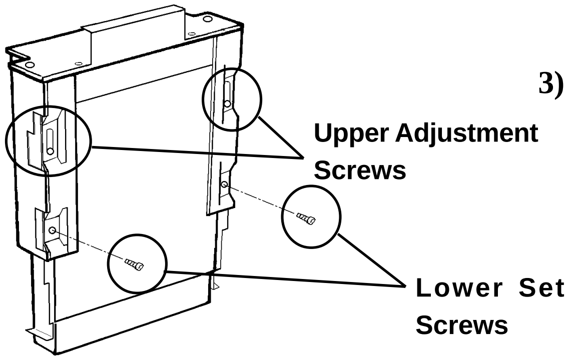
- **The PL-FD210 can only be used with the PL-B920 (4-slot type) units, and cannot be used with the PL-B921 (2-slot type) units.**
- **The PL-FD200 and the PL-FD210 cannot be used at the same time.**



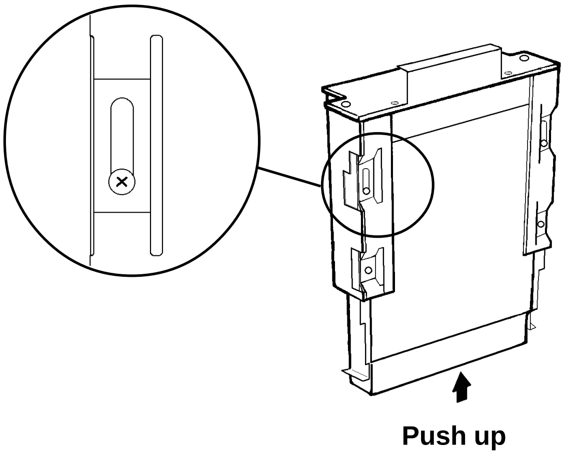
1) Unscrew the PL front face's FDD blank panel's attachment screw (1). Remove the blank panel.



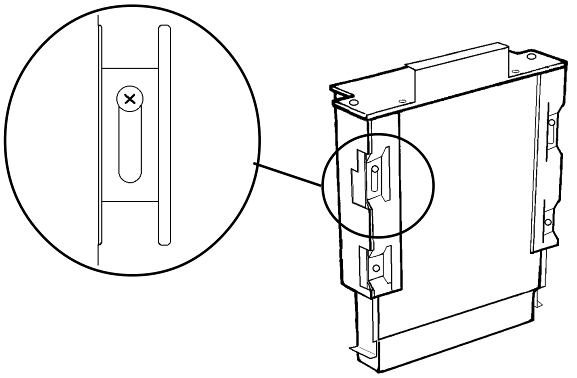
2) Unscrew the two (2) attachment screws from the PL's Rear Face Blank Panel, and remove it.

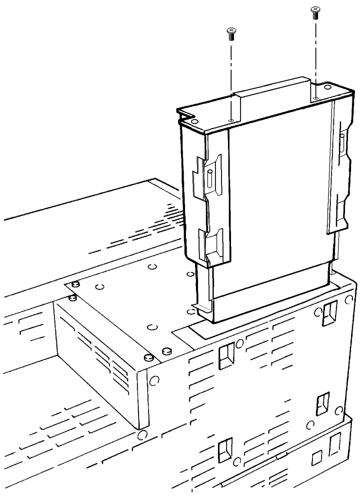


3) Loosen the FDD unit chassis two (2) upper adjustment screws, and remove the unit's two (2) lower set screws to allow the FDD unit to slide forward.



4) Push the unit in the direction of the arrow so that the adjustment screw is in the top-most position. Next, secure the unit in place by tightening all four (4) (upper and lower) screws.





5) Insert the FDD unit so that its guide grooves align with the chassis holder guideways. Push the unit into the PL holder until its connector (middle of unit) is connected securely.

Important

When installing the PL-FD210, insert it slowly into the PL's installation opening and be sure it is securely attached.

6) Fix the unit in place with its two(2) attachment screws.

7) Last, replace the Blank Panel (removed in step 3).

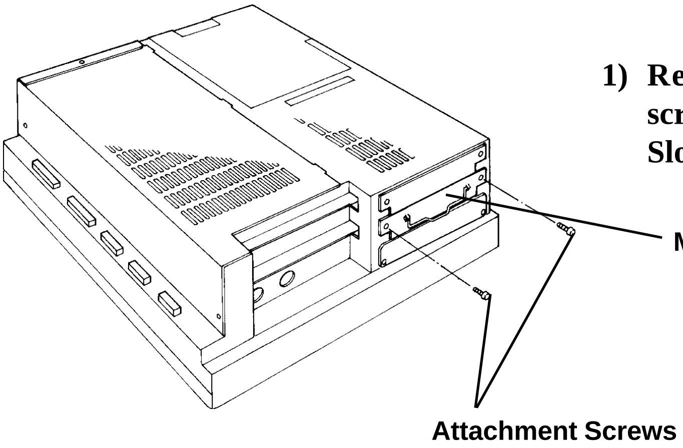
3.1.5 Removing/ Installing the HDD Unit (PL-HD220)

Note:

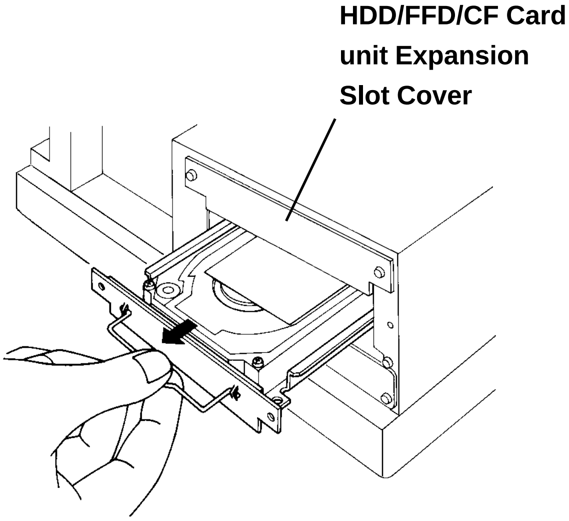
- **The following insertion/removal procedure is applicable also to the FFD(PL-FF210) unit and the CF Card unit (PL-CF200).**
- **Certain usage limitations apply to the HDD, FFD, CF card, Mirror Disk and CD-ROM Drive.** Reference *1.3 Optional Items*

Important

Since the HDD unit is a precision instrument, be sure not to subject it to sudden shaking or shocks.



1) Remove the two(2) attachment screws from the middle Expansion Slot Cover.



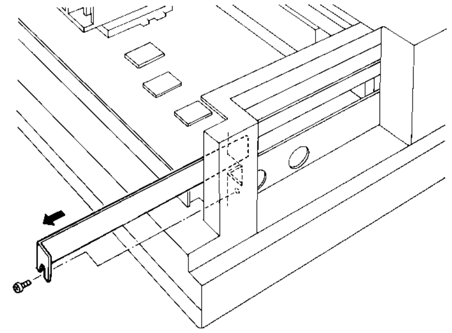
2) Grasp the HDD unit's handle and pull the unit slowly out of the PL. Be sure you do not damage the unit.

3) Insert the new HDD unit into the PL's guideways and push it in until its rear connector is securely connected.

4) Fix the unit in place with its two(2) attachment screws.

Note: **Reattach the Expansion Slot cover using the same steps.**

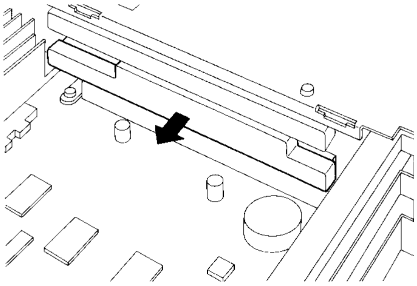
3.1.6 Installing an Expansion Board



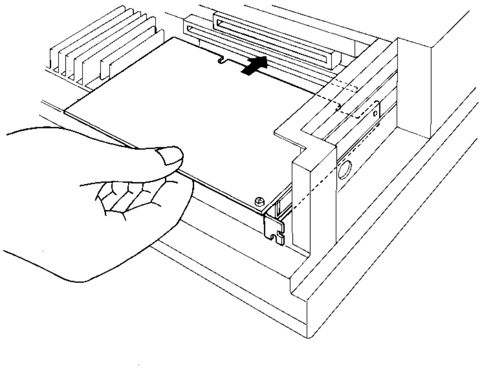
1) **Unscrew the desired expansion slot's cover attachment screw, and remove the cover.**

   Reference ***3.1.1, Removing the Rear Maintenance Cover***

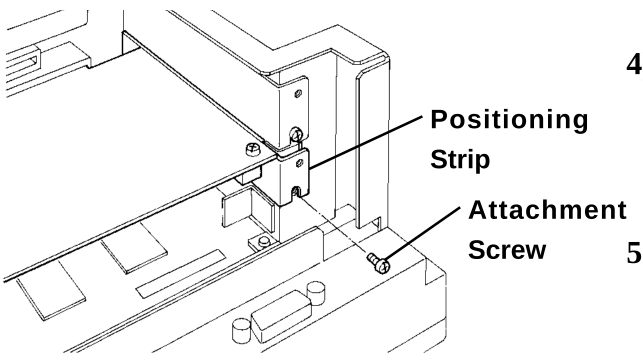
   Unscrew the Blank Panel's attachment screw to remove the Blank Panel.



2) **Remove the expansion slot's duster cover.**



3) **Insert the expansion board into the expansion slot.**



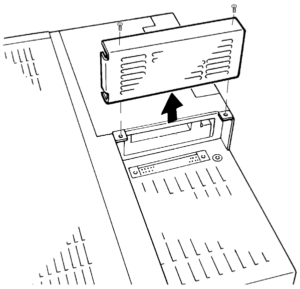
4) **Secure the expansion board's metal positioning strip in place with its attachment screw.**

5) **Last, replace the rear maintenance cover and half cover and secure them in place with their attachment screws.**

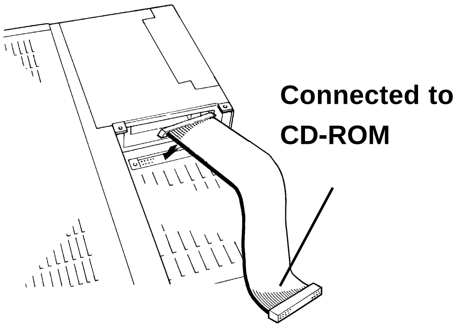
3.1.7 Connecting the CD-ROM Unit (PL-DK200)

When using the CD-ROM unit, be sure to change the System Information area's "Integrated Peripherals" [IDE Primary Master PIO] and [IDE Primary Slave PIO] settings to [Mode 2], or set [IDE Primary Master UDMA] or [IDA Primary Slave UDMA] to [Disabled]. Reference *5.2.9 Integrated Peripherals.*

■ PL-B920 Unit



1) Unscrew the two (2) IDE I/F cover attachment screws, and remove the cover.

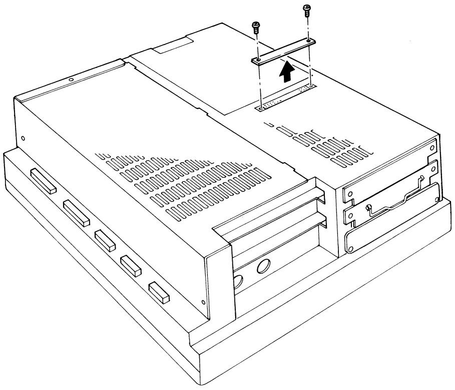


2) Connect the CD-ROM unit cable to the PL's IDF I/F connector.

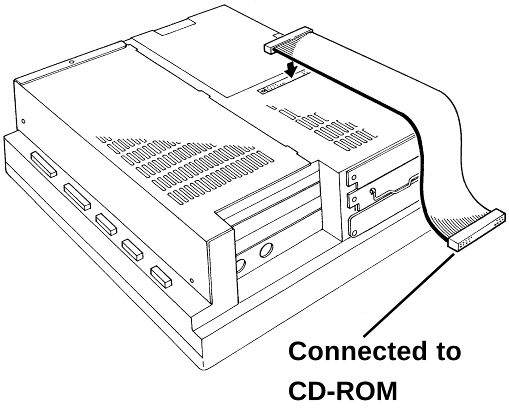
Important

Be sure that the cable is securely connected before turning ON the PL's power switch.

■ PL-B921 Unit



1) Unscrew the two(2) IDF I/F cover attachment screws, and remove the cover.



2) Connect the CD-ROM unit's cable to the IDF I/Fconnector.

Important

Be sure that the cable is securely connected before turning ON the PL's power switch.

3.1.8 Connecting the Standard Display (DU-6900/DU-7900)

*Use the PL-CB200-*M Display Cable to connect the PL's Standard Display Connector to the DU unit's connector. The following table shows the communication format and the display cables that can be used.*

Communication Format	Display Cable
Serial Connection	PL-CB200-5M (5m) PL-CB200-10M (10m)
USB Connection	PL-CB200-5M (5m)

Important

- ***When the DU is connected to the PL using the USB connection, wait three (3) seconds before turning the power ON again.***
- ***When connecting to a DU using USB Connection, the PL unit's System settings must be changed. For details about the System Setup, Reference Chapter5 - System Setup.***

Note:

- **Be sure not to bend, crimp or pull the PL's Standard Display Cable.**
- **Be sure to insert the cable connectors together until they lock securely.**

■Dip Switch Settings

Depending on the communication format used, the DU unit's rear face dip switch settings will need to be changed. These settings must also be set on the PL unit's side face dip switchs.

◆Serial Connection

PL Settings

SW No.	1	2	3	4
Position	ON	OFF	ON	OFF

DU Settings

SW No.	1	2	3	4	5	6	7	8
Position	ON	OFF	ON	OFF	Reserved			ON

◆USB Connection

PL Settings

SW No.	1	2	3	4
Position	OFF	ON	OFF	ON

DU Settings

SW No.	1	2	3	4	5	6	7	8
Position	OFF	ON	OFF	ON	Reserved			OFF

3.1.9 Connecting an Analog RGB Monitor

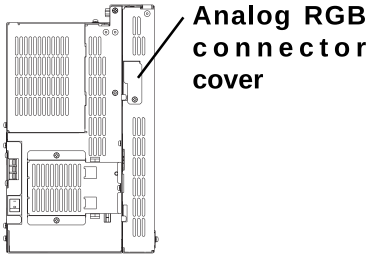
The following explantion connects the PL to a commercially available CRT monitor, or to a Flat Panel display (FP).

The following table shows the type of displays, the communication format and the cable that can be used.

Display Type	Touch Panel Communication Format	RGB Cable
FP-2500 FP-2600	Serial Connection	FP61V-IV00-O (5m)
	USB Connection	FP-US00 (5m)
FP-790 CRT Monitor	Serial Connection	FP-CV00 (2.5m) FP-CV0 (5m)

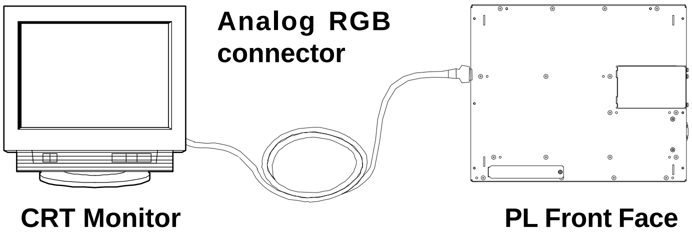
■ Connecting a CRT Monitor

When connecting the CRT monitor, be sure that it is a standard analog multiscan VGA compliant type.



1) Unscrew the Analog connector cover's attachment screw, and remove the cover.

2) Connect the CRT monitor's cable to the PL's analog RGB connector.



■ Connecting a Flat Panel Display (FP)

Digital's FP-790/FP-2500/FP-2600 monitor can also be connected to the PL unit.

When using the Flat Panel touch panel, a mouse emulation software is required. For details about the mouse emulation software, Reference ***1.2 .1 Setting Up the Touch Panel Connection.***

The connection method can be the same as the CRT Monitor shown above.

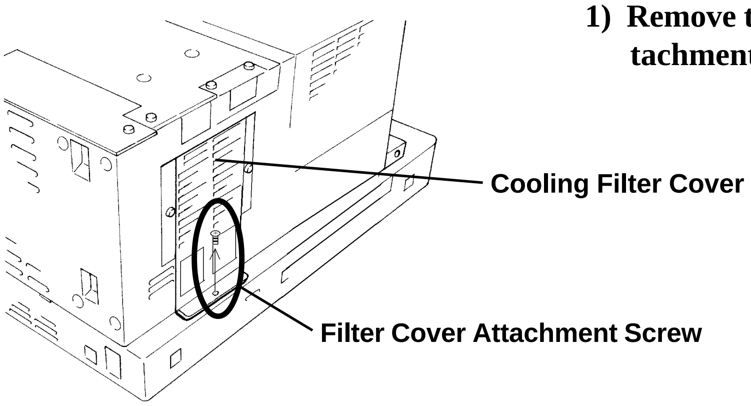
Important

- ***When connecting the FP-2500/FP-2600 using a USB connection, the PL unit's System settings must be changed. For details about the System Setup,*** Reference ***Chapter 5 - System Setup.***

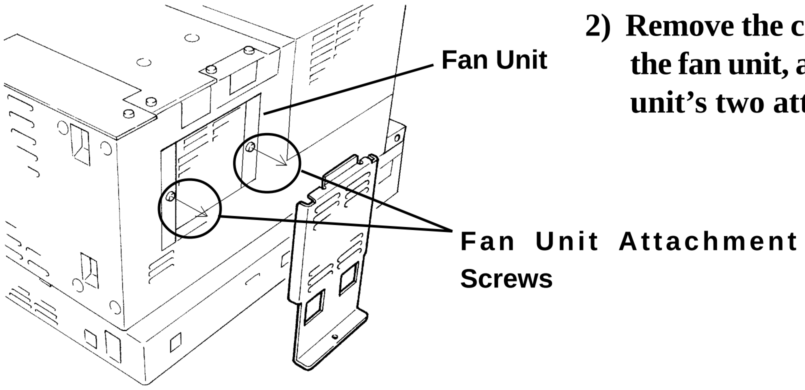
3.1.10 Removing the Cooling Fan Unit

The PL units can be operated without their bottom face cooling fan unit. The user should, however, be aware that doing so (i.e. removing the fan unit) will cause the PL's ambient temperature to rise. Reference *2-1-2 Environment Specifications*

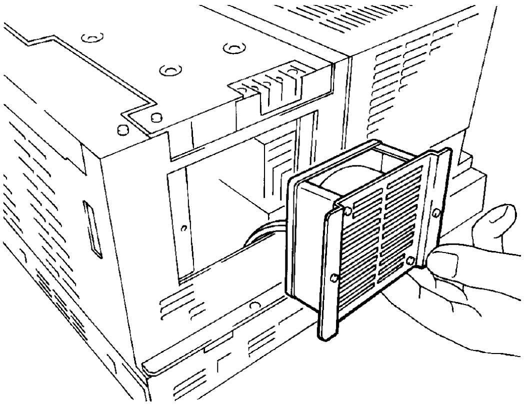
■ PL-B920 Unit



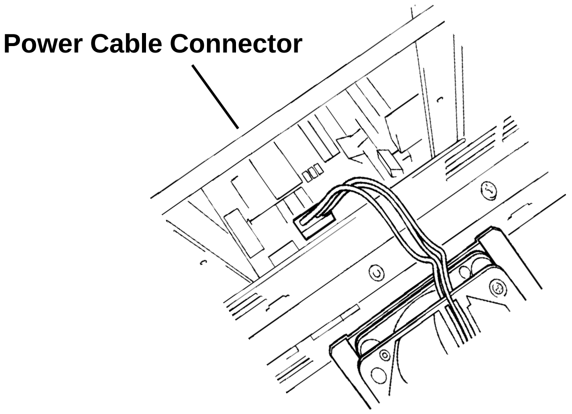
1) Remove the cooling filter cover's attachment screw.



2) Remove the cooling filter cover from the fan unit, and then remove the fan unit's two attachment screws.

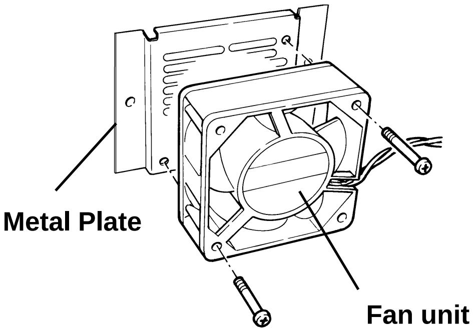


3) Remove the fan unit from the PL.



4) Disconnect the fan unit's power cable

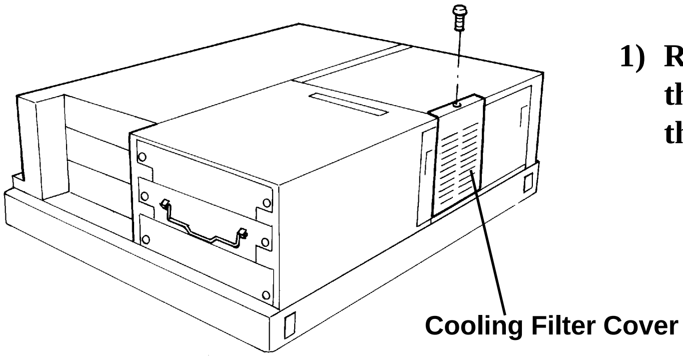
5) Replace the cooling filter cover and reattach the cooling filter cover's attachment screw.



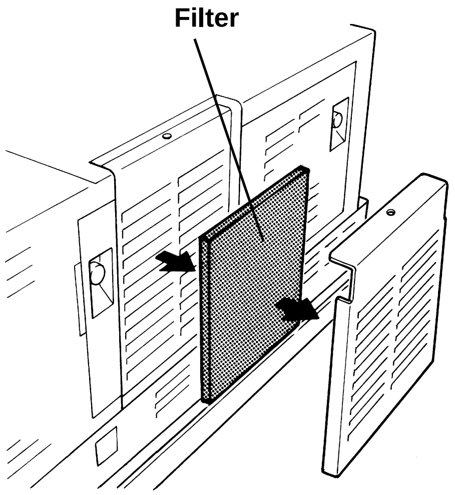
6) Unscrew the two(2) fan unit's attachment screws and remove the fan unit from the metal plate holding the fan unit.

7) Attach the metal plate and the cooling filter cover to the PL.

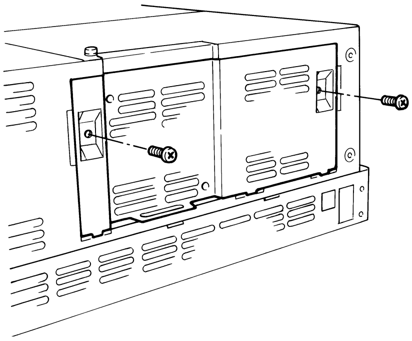
■ PL-B921 Unit



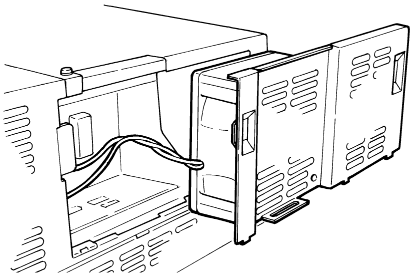
1) Remove the attachment screw on the cooling filter cover, and remove the cover.

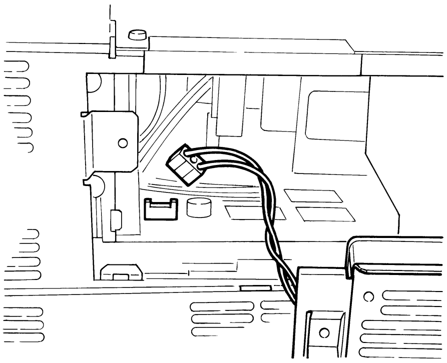


2) Remove the filter.

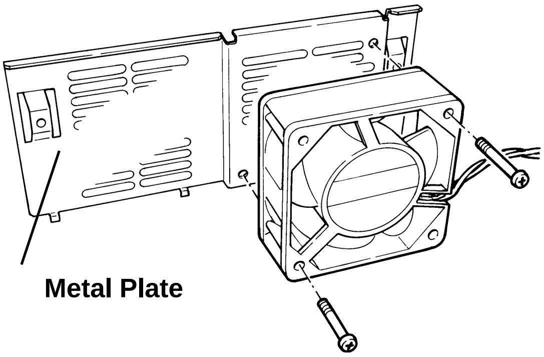


3) Remove the two(2) fan unit attachment screws, and take out the fan unit.





4) Unplug the fan unit's power cable connector from the PL.



5) Unscrew the two (2) attachment screws and remove the fan unit, and then remove the fan unit from the metal plate.

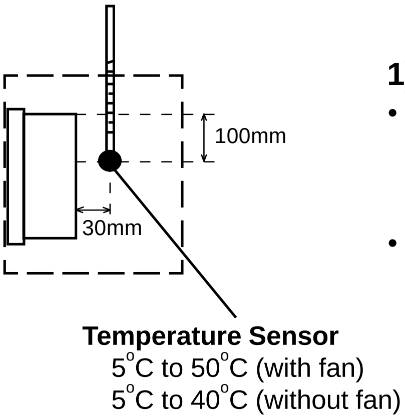
6) Reattach the metal attachment plate and the cooling filter cover to the PL.

Memo

Chapter 4 Installation and Wiring

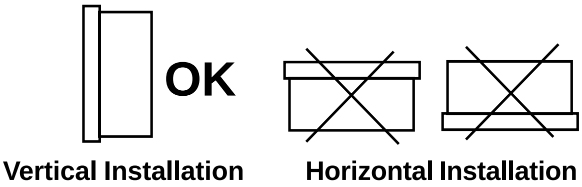
This chapter explains how to install and wire the PL-B920 series units, as well as the cautions required both before and during installation.

4.1 Installation Cautions



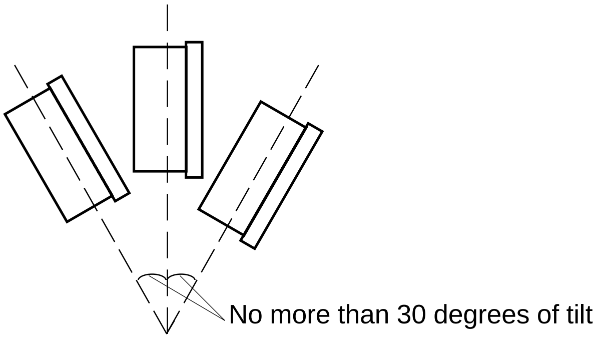
1) Temperature Related Cautions

- The PL should be installed in a vertical position, and forced air cooling should be used, instead of natural air circulation.
- Be sure to confirm that the area near the PL will be within the allowable temperature range by placing a temperature sensor in the location shown in the left-side drawing. If this area's temperature exceeds the allowed limit, a machine breakdown can occur.



2) Installation Positioning Cautions

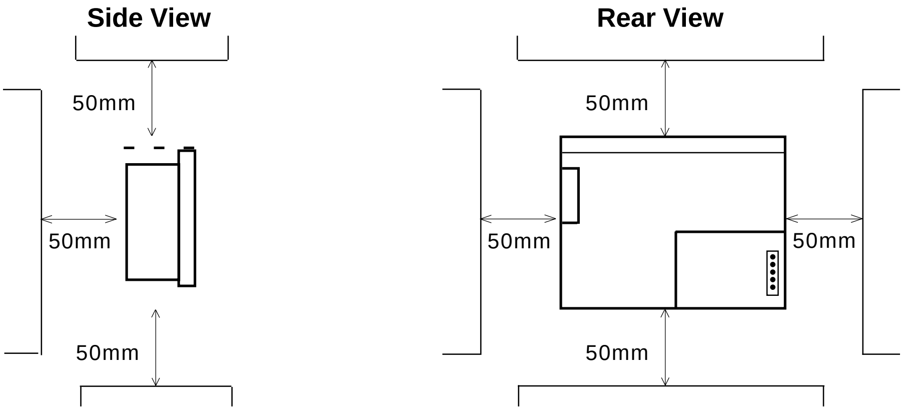
Be sure to install the panel in an upright (vertical) position.



Also, be sure that the panel's viewing angle is tilted no more than 30 degrees from parallel to the operator (i.e. directly in front).

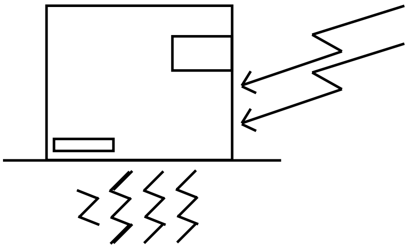
Installation Location

- Avoid placing the PL next to other devices that might cause overheating.
- Keep the PL away from arc-generating devices such as magnetic switches and non-fuse breakers.
- Avoid using the PL in environments where corrosive gases are present.
- To ensure the reliability, operability and ventilation of the PL, be sure to install it in locations that are more than 50mm away from adjacent structures or equipment. Also, consider the need for installing or removing expansion boards, or connectors when designing and installing your PL.



Vibration and Shocks

If the PL is moved when its enclosure doors are open, or while it is installed in a rack equipped with caster wheels, the hard disk can receive excessive vibration or jolting. Be especially careful at this time.



PL Configuration	Can Withstand
HDD	Up to 4.9m/s^2
FDD	Up to 9.8m/s^2
No drives	Up to 19.6m/s^2

Important

- ***The Hard Disk Drive is precision equipment and should not be moved or jolted . Especially when the PL is turned ON, even changing the PL's direction while it is on a table, or repositioning the unit should not be performed, since it can lead to a hard disk crash or malfunction.***
- ***When using a fan to cool the PL unit, be sure that the fan does not point directly at any of the PL's disk drive units, since it can lead to a hard disk crash or malfunction.***

4.2 Installing the PL

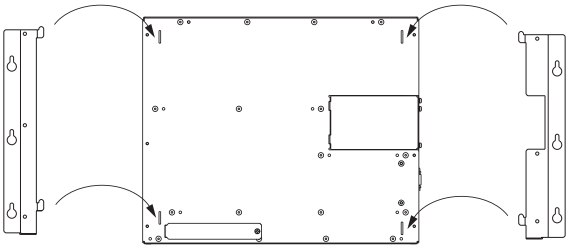
4.2.1 Installation Procedures

Use the following procedures to install the PL into a solid panel.

Important ***Be sure to read the previous section's Installation Cautions in order to install the PL safely.***

Note: **When using the optional 19 inch Mount Panel, refer to the PL-RM200 unit's Installation guide for details.**

1) Use the six (6) attachment screws included in the PL's packing box to attach the mounting brackets tothe PL main unit. Be careful that each bracket is attached to the correct side.



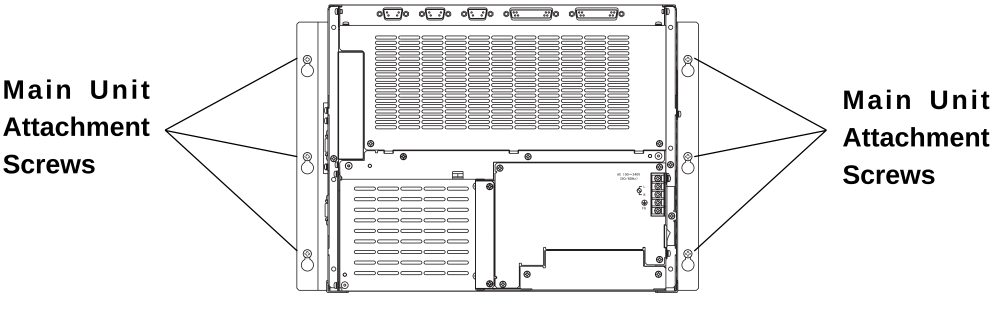
2) Drill the attachment holes in the Installation Panel. Be sure to follow the dimensions given for the attachment holes.

Reference *2.5.3 Full Sized Cover Attachment Dimensions*

3) Attach the PL to the Installation Panel with M4 screws.

First, insert the main unit attachment screws into the Installation Panel's holes, but do not tighten them. Then, place the PL main unit on the panel so that the bracket holes and the screw heads align. Next, slide the PL down so that the main unit is supported by these attachment screws. Last, tighten the screws until the PL is secured in place.

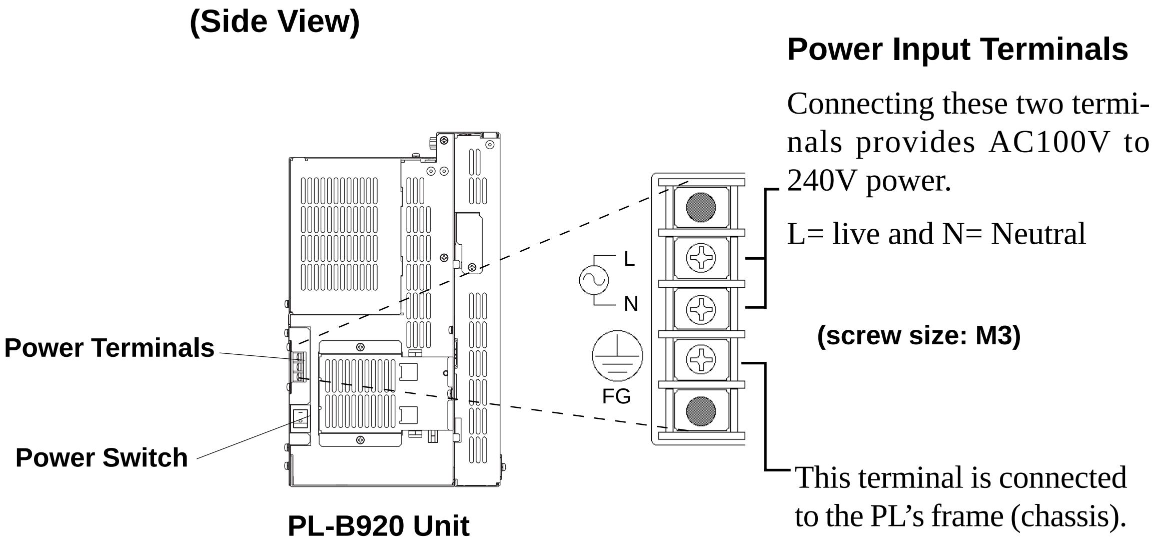
Important ***Do not use excessive force when tightening the main unit attachment screws. The torque required is from 0.5 to 0.6N•m.***



4.3 Wiring the PL

4.3.1 Connecting the Power Cord

Connect the PL's power cord to the PL's rear face power terminals.



Connecting these two terminals provides AC100V to 240V power.

L= live and N= Neutral

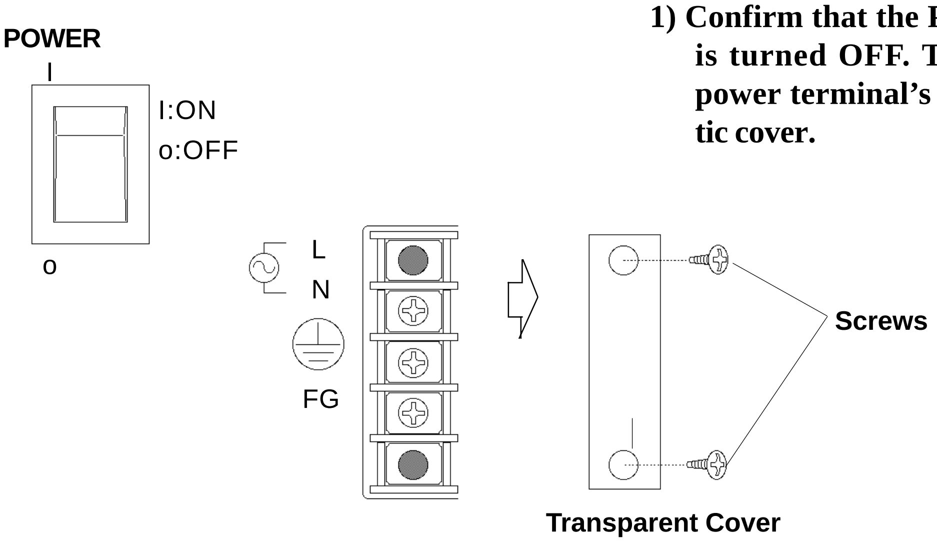
This terminal is connected to the PL's frame (chassis).

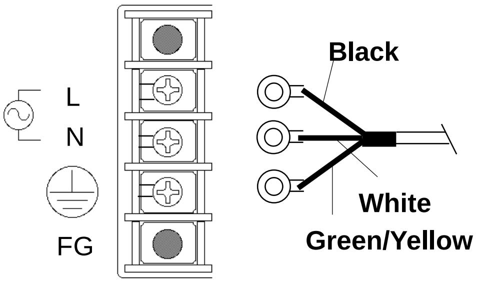
Use the following steps when connecting the power cord to the PL's power terminals.

WARNING

- **To prevent electric shocks, be sure to turn the PL OFF before connecting the power cord terminals to the PL.**
- **To avoid the dangers of fire, electrical hazards and equipment damage, be sure to use only the specified power supply voltage when operating the PL.**

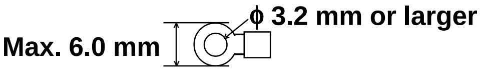
1) Confirm that the PL's power switch is turned OFF. Then, remove the power terminal's transparent plastic cover.



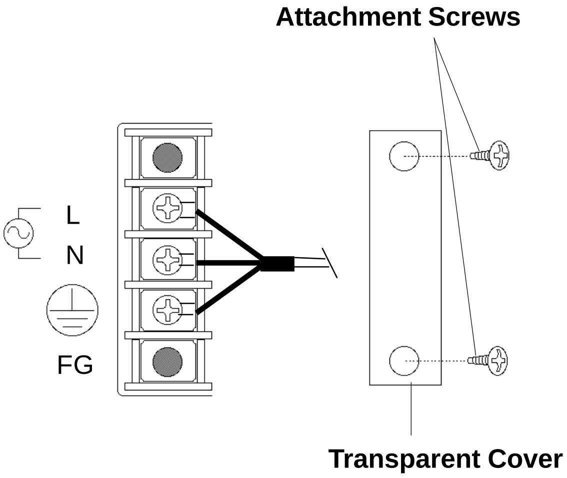


2) Loosen and remove the middle three screws from the terminal strip. Align the crimp terminals with each screw hole, and tighten the screws.

- **Crimp Terminal Types : V1.25-3, by J.S.T. or equivalent (JIS standard part number : RAV1.25-3)**
- **Crimp terminals must be the same as shown below.**



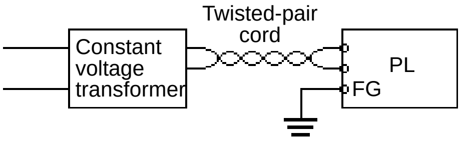
- ***The colors used in these figures are for the cable which came with the PL.***
- ***This power cable is designed only for AC100V/115V use. When using other than AC100V/115V power, use a cable that is designed specifically for that level of power.***



3) Reattach the terminal strip’s transparent cover and secure it in place with its attatchment screws.

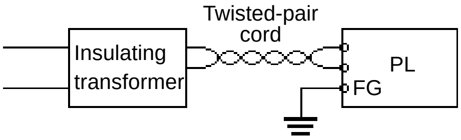
4.3.2 Power Supply Cautions

When connecting the PL unit's AC power terminals, please be aware of the following:



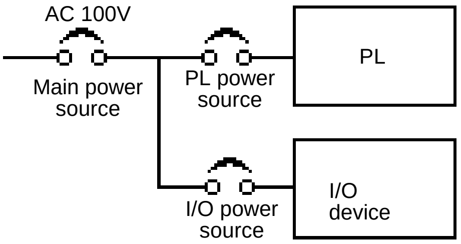
- If voltage fluctuations are expected to vary beyond the specified range, connect a constant voltage transformer.

Reference *For information about the specified voltage, refer to **2-1 General Specifications***

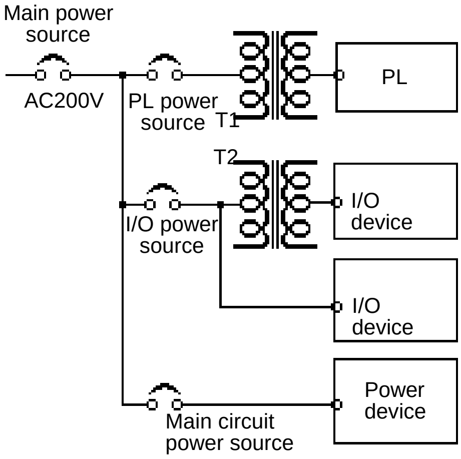


- Use a low-noise power supply both between the lines and between the PL and its ground. If there is still excess noise, connect an insulating transformer (noise-prevention type).

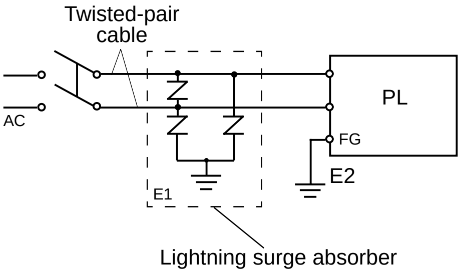
Note: **Be sure any constant or insulating transformer used has a capacity of 200VA or more.**



- Wire the power cords of the PL, I/O devices, and power supply devices separately.



- To improve noise immunity, it is recommended to attach a ferrite core to the power cord.
- Isolate the main circuit (high voltage, large current) line, I/O signal lines, and power cord, and do not bind or group them together.



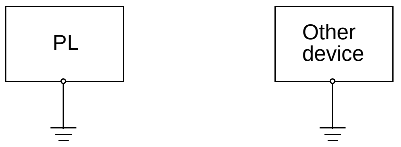
- To prevent damage from lightning, connect a lightning surge absorber.

Important

- ***Ground the lightning surge absorber (E1) and the PL (E2) separately.***
- ***Select a lightning surge absorber which will not exceed the allowable circuit voltage, even when the voltage rises to the maximum.***

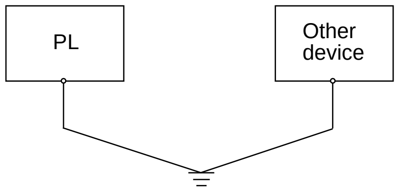
4.3.3 Grounding Cautions

(a) Exclusive Ground - best [*1]



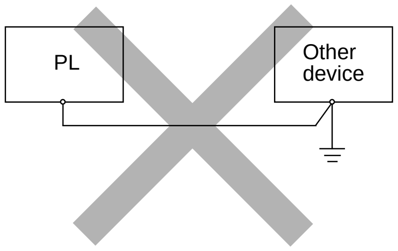
- Set up a dedicated ground when using the rear panel's FG terminal.

(b) Shared Ground - allowed [*1]



- If a dedicated ground is not possible, use a shared ground, as shown in figure (b).
- The grounding point must be as close to the PL as possible, and the grounding wires must be as short as possible. If the wires must be long, use thick, insulated wires and run them through conduits.

(c) Shared ground - not allowed



4.3.4 Cautions When Connecting I/O Signal Lines

- I/O signal lines must be wired separately from charged lines. If the power cord needs to be wired together with the (I/O) signal lines for any reason, use shielded lines and ground one end of the shield to the PL's FG terminal.
- To improve noise immunity, attaching a ferrite core to the power cord is recommended.

[*1] *Use a grounding resistance of less than 100 Ω and a 2mm^2 or thicker wire, or your country's applicable standard. For details, contact your local PL distributor.*

.

Memo

Chapter 5 System Setup

This chapter explains how to enter a PL-6920/PL-7920 Series unit's system settings, as well as the cautions required both before and during set up.

5.1 Setup Procedures

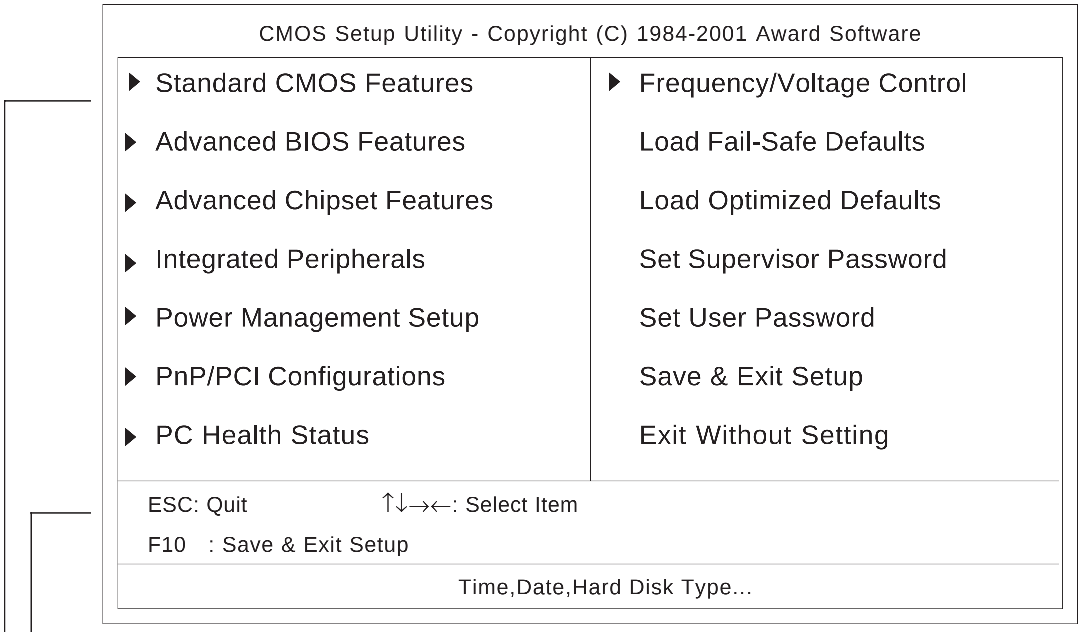
Important **_Normally, use only the factory (default) settings._**

Note: **The following settings are those pre-set at the factory.**

1) Connect a keyboard to the PL.

2) Turn the PL's power ON.

3) After the message "Press <DEL> to Enter SETUP" appears, press the [DEL] key until the following screen appears.



```
            CMOS Setup Utility - Copyright (C) 1984-2001 Award Software
▶ Standard CMOS Features          ▶ Frequency/Voltage Control
▶ Advanced BIOS Features            Load Fail-Safe Defaults
▶ Advanced Chipset Features         Load Optimized Defaults
▶ Integrated Peripherals            Set Supervisor Password
▶ Power Management Setup            Set User Password
▶ PnP/PCI Configurations            Save & Exit Setup
▶ PC Health Status                  Exit Without Setting

  ESC: Quit              ↑↓→←: Select Item
  F10  : Save & Exit Setup

                         Time,Date,Hard Disk Type...
```

KEYBOARD ACTION KEYS
Provides a summary of the keyboard keys used to carry out the set up.

SYSTEM SETTING SELECTION AREA

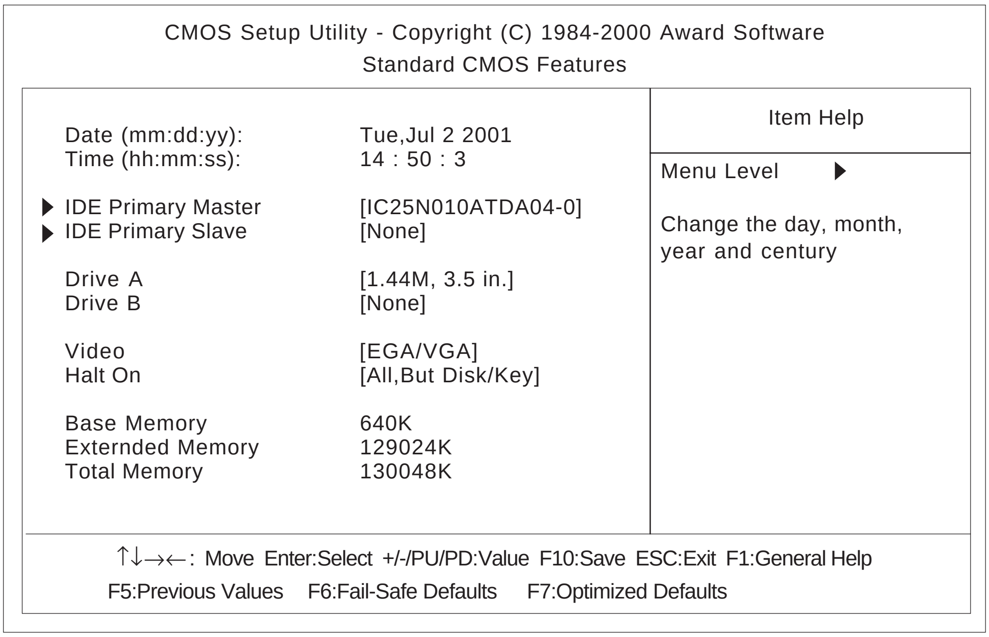
Each of the titles (areas) listed here contains system setting items.

4) Use the arrow keys to move the cursor to the desired selection.

5.2 System Parameters

5.2.1 Standard CMOS Features

Selecting the STANDARD CMOS FEATURES menu item produces the following screen.



```
          CMOS Setup Utility - Copyright (C) 1984-2000 Award Software
                            Standard CMOS Features

   Date (mm:dd:yy):          Tue,Jul 2 2001            Item Help
   Time (hh:mm:ss):          14 : 50 : 3
                                                  Menu Level     ▶
 ▶ IDE Primary Master        [IC25N010ATDA04-0]
 ▶ IDE Primary Slave         [None]               Change the day, month,
                                                  year and century
   Drive A                   [1.44M, 3.5 in.]
   Drive B                   [None]

   Video                     [EGA/VGA]
   Halt On                   [All,But Disk/Key]

   Base Memory               640K
   Externded Memory          129024K
   Total Memory              130048K

   ↑↓→←: Move  Enter:Select  +/-/PU/PD:Value  F10:Save  ESC:Exit  F1:General Help
        F5:Previous Values   F6:Fail-Safe Defaults   F7:Optimized Defaults
```

- **Date/Time**

  This data sets the PL's internal time and date.

Hours	:00 - 23
Minutes	:00 - 59
Seconds	:00 - 59

- **IDE Primary Master (Slave)**

  Displays the size of the Hard Disk installed in the PL. Pressing the [Enter] key will call up the Parameter settings menu.
  For details, refer to ***5.2.2 IDE HDD Auto Detection***

- **Drive A (B)**

  This setting determines the format used by the PL's internal floppy disk drive. The available settings are [None], [720K - 3.5in], [1.44M - 3.5in], or [2.88M, 3.5in]. The factory settings are Drive A [1.44M - 3.5in] and Drive B [None] and recommended for most users.

- **Video**

  The selections for the screen (video) mode. The available settings are [EGA/VGA], [CGA40], [CGA80] and [MONO]. The [EGA/VGA] selection is factory set and recommended for most users.

■ Halt On

Designates the type of processing that will be performed when an error occurs during the Initial Start-Up's Self Test. The [All But Disk /Key] selection is factory set and recommended for most users.

[All Errors] : Displays all errors and stops the unit.

[No Errors] : Displays all errors and does not stop the unit.

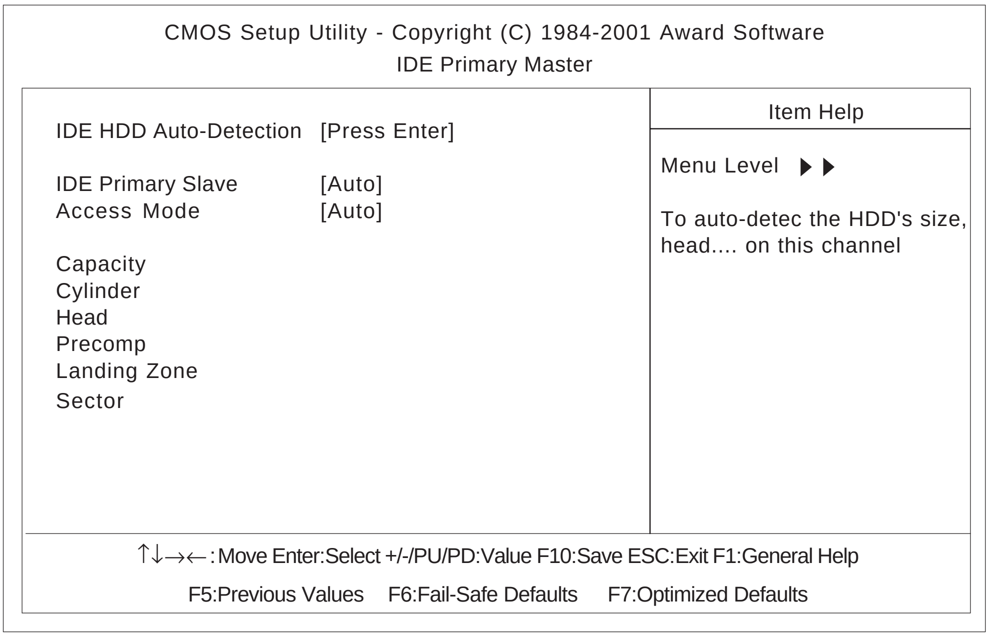
[All,But Keyboard] : Displays all errors, except for those related to the keyboard, and stops the unit. If the User has no keyboard connected, please use this setting.

[All,But Diskette] : Displays all errors, except for those related to the disk drive (FDD), and stops the unit.

[All,But Disk/Key] : Displays all errors, except for those related to the disk drive (FDD) and keyboard, and then stops the unit.

5.2.2 IDE Auto-Detection

The Standard CMOS Features menu is used to select the IDE Primary Master and the IDE Primary Slave.



```
CMOS Setup Utility - Copyright (C) 1984-2001 Award Software
                    IDE Primary Master

IDE HDD Auto-Detection   [Press Enter]      |           Item Help
                                            |  Menu Level  ▶▶
IDE Primary Slave        [Auto]             |
Access Mode              [Auto]             |  To auto-detec the HDD's size,
                                            |  head.... on this channel
Capacity
Cylinder
Head
Precomp
Landing Zone
Sector

↑↓→←: Move Enter:Select +/-/PU/PD:Value F10:Save ESC:Exit F1:General Help
     F5:Previous Values   F6:Fail-Safe Defaults   F7:Optimized Defaults
```

- ■ **IDE HDD Auto-Detection**

  This setting detects the hard disk connected to the IDE interface.

- ■ **IDE Primary Master (Slave)**

  This setting designates the IDE type Hard Disk's parameter setting method. The available settings are [None], [Auto], or [Manual]. The factory default setting is [Auto] and is recommended for most users.

- ■ **Access Mode**

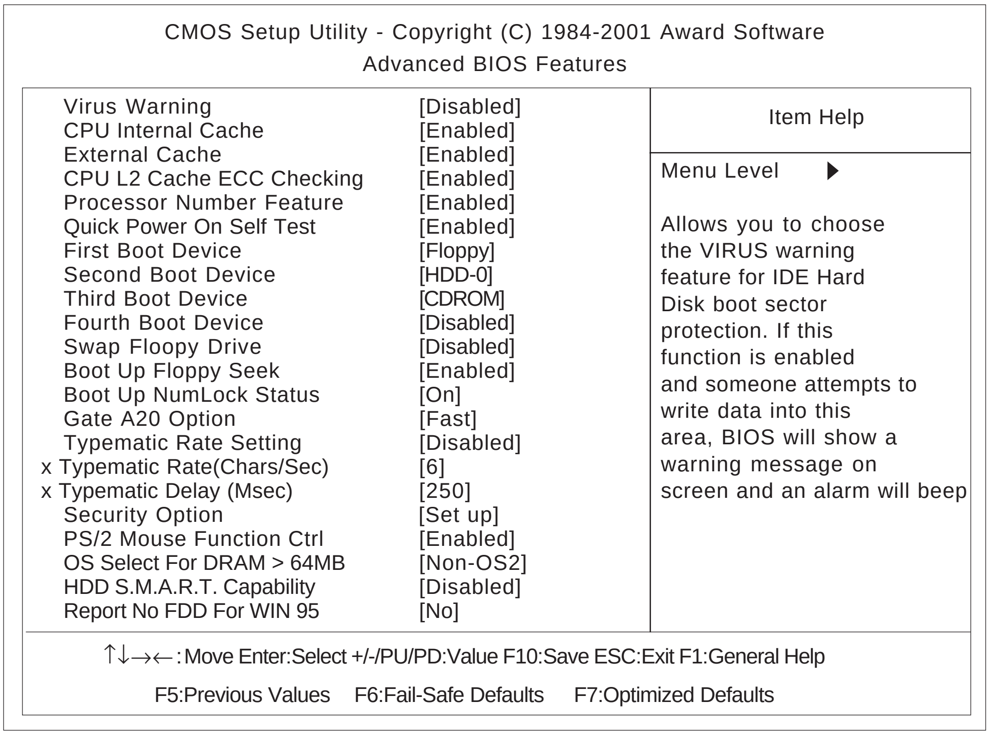
  This setting designates the IDE type Hard Disk's access mode. The available settings are [CHS], [LBA], [Large], or [Auto]. The factory default setting is [Auto] and is recommended for most users.

- ■ **Capacity/Cylinder/Head/Precomp/Landing Zone/ Sector**

  These settings designate individual IDE type Hard Disk parameter settings. These can only be set when the [IDE Primary Master (Slave)] setting is set to [Manual]. When the [IDE Primary Master (Slave)] setting is set to [Auto], these values are automaticaly detected. Capacity is set automaticaly.

5.2.3 Advanced BIOS Features

Selecting the ADVANCED BIOS FEATURES menu item calls up the following screen.



```
      CMOS Setup Utility - Copyright (C) 1984-2001 Award Software
                        Advanced BIOS Features

   Virus Warning                  [Disabled]  |        Item Help
   CPU Internal Cache             [Enabled]   |
   External Cache                 [Enabled]   | Menu Level    ▶
   CPU L2 Cache ECC Checking      [Enabled]   |
   Processor Number Feature       [Enabled]   | Allows you to choose
   Quick Power On Self Test       [Enabled]   | the VIRUS warning
   First Boot Device              [Floppy]    | feature for IDE Hard
   Second Boot Device             [HDD-0]     | Disk boot sector
   Third Boot Device              [CDROM]     | protection. If this
   Fourth Boot Device             [Disabled]  | function is enabled
   Swap Floopy Drive              [Disabled]  | and someone attempts to
   Boot Up Floppy Seek            [Enabled]   | write data into this
   Boot Up NumLock Status         [On]        | area, BIOS will show a
   Gate A20 Option                [Fast]      | warning message on
   Typematic Rate Setting         [Disabled]  | screen and an alarm will beep
 x Typematic Rate(Chars/Sec)      [6]         |
 x Typematic Delay (Msec)         [250]       |
   Security Option                [Set up]    |
   PS/2 Mouse Function Ctrl       [Enabled]   |
   OS Select For DRAM > 64MB      [Non-OS2]   |
   HDD S.M.A.R.T. Capability      [Disabled]  |
   Report No FDD For WIN 95       [No]        |

    ↑↓→←:Move Enter:Select +/-/PU/PD:Value F10:Save ESC:Exit F1:General Help
         F5:Previous Values    F6:Fail-Safe Defaults    F7:Optimized Defaults
```

- **Virus Warning**

  This setting determines whether to display a warning when a write to the boot sector is attempted. The available settings are [Enabled] or [Disabled]. The factory default setting is [Disabled] and is recommended for most users.

- **CPU Internal Cache**

  This setting determines the usage of the CPU's internal cache memory. The available settings are [Disabled] or [Enabled]. The factory default setting is [Enabled] and is recommended for most users.

- **External Cache**

  This setting determines the usage of the external cache memory (L2). The available settings are [Disabled] or [Enabled]. The factory default setting is [Enabled] and is recommended for most users.

- **CPU L2 Cache ECC Checking**

  This setting turns ON or OFF external(L2) Cache Memory's ECC (Error Check Correction). The available settings are [Disabled] and [Enabled]. The factory default setting is [Enabled] and is recommended for most users.

- **Processor Number Feature**

  This option is for the Pentium® III processor. If this setting is set to Enabled, it will check the CPU's serial number. If you do not need to know the serial number, set this option to [Disabled]. If this is turned ON, use the "Serial Number Control Program" found on Intel's web site to perform the check.

- **Quick Power On Self Test**

  This setting determines whether the quick self test is performed when the power is turned on. The available settings are [Disabled] or [Enabled]. The factory default setting is [Enabled] and is recommended for most users.

- **First/ Second/ Third/ Fourth Boot Device**

  The selections for the search drive sequence of the operating system. The available settings are [Floppy], [HDD-0][*1], [LS120], [HD-0], [SCSI], [CDROM], [HDD-1][*1], [ZIP100], [LAN], and [Disabled].

- **Swap Floppy Drive**

  This setting swaps Drives A and B. The available settings are [Disabled] or [Enabled]. The factory default setting is [Disabled] and is recommended for most users.

- **Boot Up Floppy Seek**

  The setting checks whether the floppy disk drive is installed during the system boot-up process. The available settings are [Disabled] or [Enabled]. The factory default setting is [Enabled] and is recommended for most users.

- **Boot Up Numlock Status**

  This setting specifies the Numlock key status upon the startup. The available settings are [On] and [Off]. The factory default setting is [On] and is recommended for most users.

- **Gate A20 Option**

  The available settings are [Normal] and [Fast]. When [Normal] is selected, the Keyboard control will be employed to control Gate A20. When [Fast] is selected, the Chipset will be employed. The factory default setting is [Fast] and is recommended for most users.

*1 *This has a different meaning from the expansion slot position codes "HDD0" and "HDD1" stamped into the side of the PL unit's metal chassis.*

HDD-0:

-When only one HDD unit is installed.

The HDD setting is recognized as "HDD-0" and is not related to the Master/Slave items.

-When 2 HDD units are installed.

An HDD unit designated as the Master HDD is recognized as "HDD-0".

HDD-1:

-When only one HDD unit is installed.

OS cannot be started. Be sure the BIOS setting is "HDD-0".

-When 2 HDD units are installed.

An HDD unit designated as the Slave HDD is recognized as "HDD-1".

■ Typematic Rate Setting

The setting specifies the keyboard speed used when repeating characters.
The available settings are [Enabled] and [Disabled]. The factory default setting is [Disabled] and is recommended for most users.

■ Typematic Rate (Chars/ Sec)

This setting specifies the actual typematic rate (repeated character input per second) when the [Typematic Rate Setting] option is set to [Enabled]. The settings are used to determine how many repeats are performed in one second. The factory default setting is [6] and is recommended for most users.

■ Typematic Delay (Msec)

When [Typematic Rate Setting] is set to [Enabled], this setting determines the delay period until the initial repetition is started. The [250] selection is factory set and is recommended for most users.

■ Security Option

This setting designates the area to request a password. If during BIOS setup you are requested to enter a password, select [Setup]. If during System Startup you are requested to enter a password, select [System]. This setting is NOT available if the password is not set in the [SET SUPERVISOR PASSWORD] or [SET USER PASSWORD] areas. The factory default setting is [Setup] and is recommended for most users.
[SET SUPERVISOR PASSWORD] Reference **5.2.14 Set Supervisor Password**
[SET USER PASSWORD] Reference **5.2.15 Set User Password**

■ PS/2 Mouse Function Ctrl

The available selections are [Enabled] and [Disabled], [Enabled] is factory set and recommended for most users.

■ OS Select For DRAM > 64MB

The available settings are [Non-OS2] and [OS2]. The factory default settting is [Non-OS2] and is recommended for most users.

■ HDD S.M.A.R.T Capability

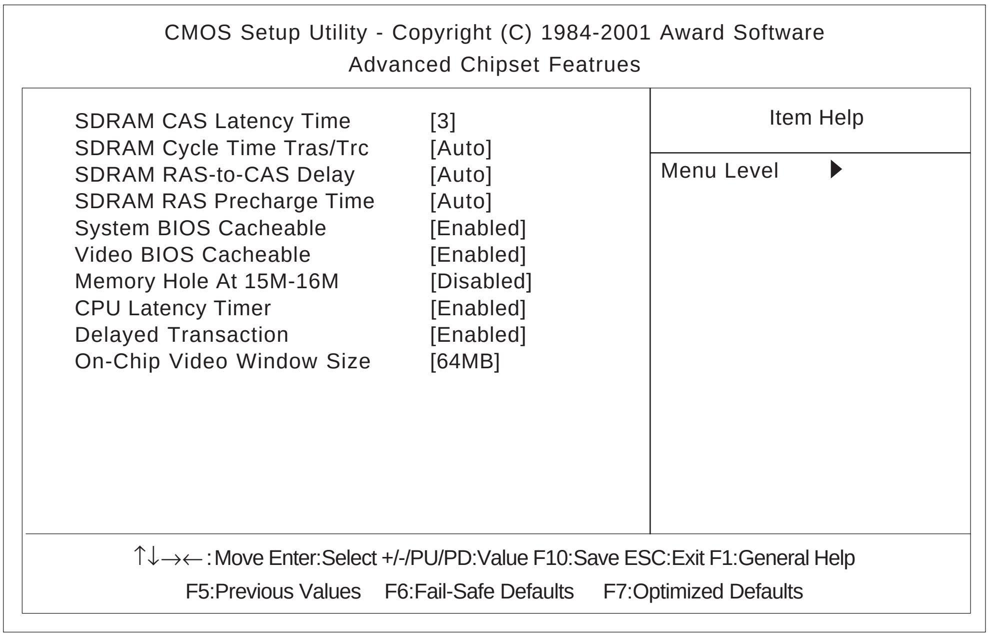
This feature sets the HDD unit's SMART (Self-Monitoring Analysis and Reporting Technology). The available selections are [Enabled] and [Disabled], [Disabled] is factory set and recommended for most users.

■ Report No FDD For WIN 95

This setting determines if the FDD node is reported by BIOS to WIN95. The available settings are [No] and [Yes]. The factory default setting is [No] and is recommended for most users.

5.2.4 Advanced Chipset Features

Selecting the ADVANCED CHIPSET FEATURES menu item calls up the following screen.



```
        CMOS Setup Utility - Copyright (C) 1984-2001 Award Software
                        Advanced Chipset Featrues

   SDRAM CAS Latency Time        [3]                    Item Help
   SDRAM Cycle Time Tras/Trc     [Auto]
   SDRAM RAS-to-CAS Delay        [Auto]          Menu Level   ▶
   SDRAM RAS Precharge Time      [Auto]
   System BIOS Cacheable         [Enabled]
   Video BIOS Cacheable          [Enabled]
   Memory Hole At 15M-16M        [Disabled]
   CPU Latency Timer             [Enabled]
   Delayed Transaction           [Enabled]
   On-Chip Video Window Size     [64MB]

 ↑↓→←: Move Enter:Select +/-/PU/PD:Value F10:Save ESC:Exit F1:General Help
     F5:Previous Values    F6:Fail-Safe Defaults    F7:Optimized Defaults
```

- **SDRAM CAS Latency Time**

  Designates the clock counts used, from the enabling of CAS to the start of the burst transmission. Can be set to either [3] or [2]. Factory default setting is [3] and strongly recommended for most users.

- **SDRAM Cycle Time Trans/Trc**

  Designates the number of SLCK's for an access cycle, i.e. the minumum required time from when a bank is activated to the activation of an identical bank. Settings are [7/9], [5/7] or [Auto]. Factory default setting is [Auto] and strongly recommended for most users.

- **SDRAM RAS-to-CAS Delay**

  Designates the timing delay used between RAS and CAS strobe signals. Settings are [2], [3], or [Auto]. Factory default setting is [Auto] and strongly recommended for most users.

- **SDRAM RAS Precharge Time**

  Designates the pre-charge time value used, to allow RAS to accumulte its charge before DRAM refresh. Settings are [2], [3], or [Auto]. Factory default setting is [Auto] and strongly recommended for most users.

- **System BIOS Cacheable**

  Sets whether the System BIOS' Cache is used or not. When the OS is set to use this cache, the PL unit's processing speed will increase. Settings available are [Enabled] and [Disabled]. The factory setting is [Enabled] and is recommended for most users.

■ Video BIOS Cacheable

Sets whether the Video BIOS' Cache is used or not. Settings available are [Enabled] and [Disabled]. The factory setting is [Enabled]. When this feature is [Enabled], the OS' BIOS ROM range available for caching is from C0000h - F7FFFh, which will improve the video performance. However, if another program tries to write to this area of memory, a system error may occur.

■ Memory Hole At 15M-16M

This setting determines whether to designate the memory space from 15MB to 16MB as the buffer area for the ISA bus card. The available settings are [Disabled] and [Enabled]. The factory default setting is [Disabled] and is recommended for most users.

■ CPU Latency Timer

If Enabled, a deferrable CPU cycle will only be Deferred after it has been in a Snoop Stall for 31 scans and another ADS# has arrived. If Disabled, changeable CPU cycles will be changed when an ADS number is received.

■ Delayed Transaction

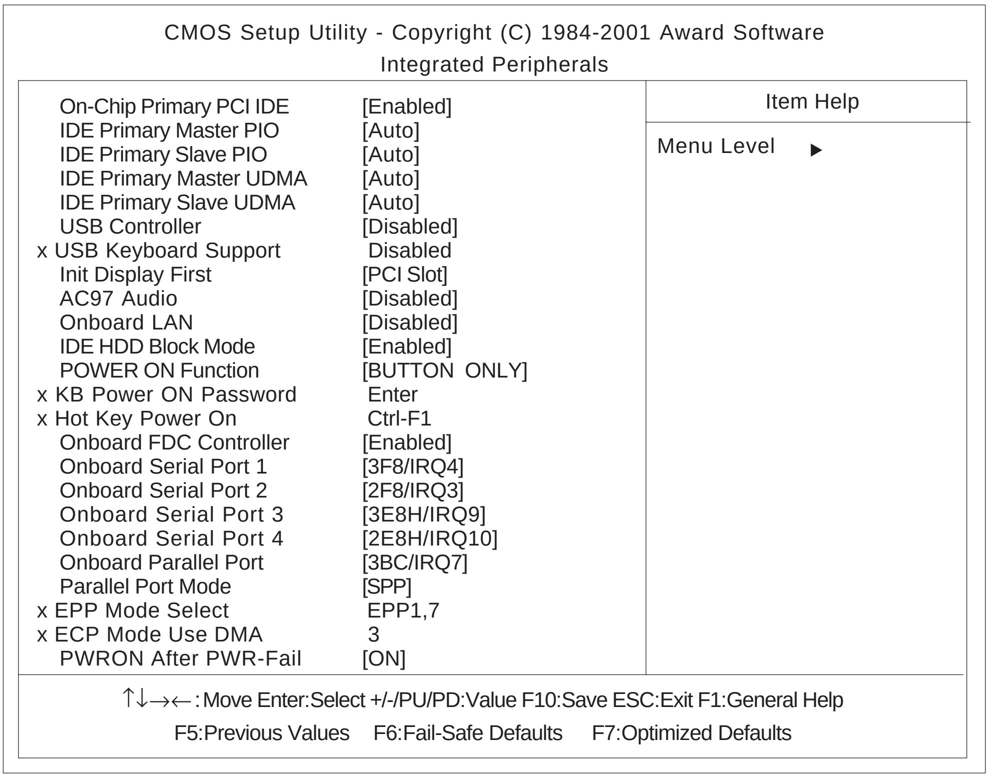
Designates the length of the pre-charge time. The available settings are [Enabled] and [Disabled]. The factory default setting is [Enabled] and is recommended.

■ On-Chip Video Window Size

Designates the on-chip video window size used by the VGA driver. The available settings are [Disabled] and [64MB]. The factory default setting is [64MB] and is recommended for most users.

5.2.5 Integrated Peripherals

Selecting INTEGRATED PERIPHERALS SETUP menu item displays the following screen.



```
        CMOS Setup Utility - Copyright (C) 1984-2001 Award Software
                          Integrated Peripherals

   On-Chip Primary PCI IDE       [Enabled]            Item Help
   IDE Primary Master PIO        [Auto]
   IDE Primary Slave PIO         [Auto]           Menu Level   ▶
   IDE Primary Master UDMA       [Auto]
   IDE Primary Slave UDMA        [Auto]
   USB Controller                [Disabled]
 x USB Keyboard Support           Disabled
   Init Display First            [PCI Slot]
   AC97 Audio                    [Disabled]
   Onboard LAN                   [Disabled]
   IDE HDD Block Mode            [Enabled]
   POWER ON Function             [BUTTON  ONLY]
 x KB Power ON Password           Enter
 x Hot Key Power On               Ctrl-F1
   Onboard FDC Controller        [Enabled]
   Onboard Serial Port 1         [3F8/IRQ4]
   Onboard Serial Port 2         [2F8/IRQ3]
   Onboard Serial Port 3         [3E8H/IRQ9]
   Onboard Serial Port 4         [2E8H/IRQ10]
   Onboard Parallel Port         [3BC/IRQ7]
   Parallel Port Mode            [SPP]
 x EPP Mode Select                EPP1,7
 x ECP Mode Use DMA               3
   PWRON After PWR-Fail          [ON]

   ↑↓→←: Move Enter:Select +/-/PU/PD:Value F10:Save ESC:Exit F1:General Help
       F5:Previous Values   F6:Fail-Safe Defaults    F7:Optimized Defaults
```

- **On-Chip Primary PCI IDE**
  Designates the internal IDE port's setting. The two selections available are [Disabled] and [Enabled]. The [Enabled] selection is factory set and recommended for most users.

- **IDE Primary Master (Slave) PIO**
  Designates the Master Drive's Operation Mode. Only after the "Internal PCI/ IDE" has been set to [Primary] can these IDE Primary Master PIO settings be entered. The two selections available are [Auto], [Mode0], [Mode1], [Mode2], [Mode3], and [Mode4]. The [Auto] selection is factory set and recommended for most users. When using the CD-ROM drive, select [Mode 2].

- **IDE Primary Master (Slave) UDMA**
  Designates the Master Drive's UDMA Operation Mode. The two selections available are [Auto] or [Disabled]. The [Auto] selection is factory set and recommended for most users.When using the CD-ROM drive, select [Mode 2].

■ **USB Controller**

Select [Enabled] or [Disable] when attaching a USB device. The [Disabled] selection is factory set and recommended for most users. When using the USB I/F or the USB connector for connecting the touch panel, set this item to "Enabled".

Important ***If a USB device is connected, change the [USB Controller] and [PnP/PCI Configration] menu's [Assign IRQ For USB] settings to [Enabled].***

■ **USB Keyboard Support**

Select [Enabled] or [Disable] if your system contains a USB controller and you wish to use a USB keyboard. The available settings are [Disabled] and [Enabled]. The factory default setting is [Disabled]. Either [Disabled] or [Enabled] can be selected. If, however, the USB Controller is set to [Disabled], this setting cannot be changed.

Important ***If a USB device is connected, change the [USB Controller] and [PnP/PCI Configration] menu's [Assign IRQ For USB] settings to [Enabled].***

■ **Init Display First**

When both a PCI and an AGP device are installed, this setting designates which will output first. Settings are [PCI slot] or [Onboard/AGP]. The factory default setting is [PCI slot].

■ **AC97 Audio**

Enables or disables the 815 chipset family to support AC97 Audio. The available settings are [Disabled] and [Auto]. The factory default setting is [Disabled].

■ **Onboard LAN**

The available settings are [Disabled] and [Enabled]. The factory default setting is [Disabled].

■ **IDE HDD Block Mode**

This setting determines whether to enable the Block Mode on the HDD supporting the Block Mode. The available settings are [Disabled] and [Enabled]. The factory default setting is [Enabled] and is recommended for most users.

■ **Power ON Function**

This feature is not supported.

■ **KB Power ON Password**

This feature is not supported.

■ **Hot Key Power ON**

This feature is not supported.

■ Onboard FDC Controller

Designates whether the PL's FDD(Floppy Disk Drive) controller is used or not. Settings available are [Disabled] or [Enabled]. The [Enabled] selection is factory set.

■ Onboard Serial Port 1

Designates the PL's Serial Port 1 I/O address. The selections include [Disabled], [Auto], [3F8/IRQ4], [2F8/IRQ3], [3E8/IRQ4] and [2E8/IRQ3]. The [3F8/IRQ4] selection is factory set and recommended for most users.

■ Onboard Serial Port 2

Designates the PL's Serial Port2 I/O address. The selections include [Disabled], [Auto], [3F8/IRQ4], [2F8/IRQ3], [3E8/IRQ4] and [2E8/IRQ3]. The [2F8/IRQ3] selection is factory set and recommended for most users.

■ Onboard Serial Port 3

Designates the PL's Serial Port3 address setting. The selections include [Disabled], [3F8H/IRQ9], [2F8H/IRQ10], [3E8H/IRQ9] or [2E8H/IRQ10]. The [3E8H/IRQ9] selection is factory set and recommended for most users.

■ Onboard Serial Port 4

Designates the PL's Serial Port3 address setting. The selections include [Disabled], [3F8H/IRQ9], [2F8H/IRQ10], [3E8H/IRQ9] or [2E8H/IRQ10]. The [2E8H/IRQ10] selection is factory set and recommended for most users.When using the USB connector for connecting the touch panel, set this item to “Disabled”.

■ Onboard Parallel Port

Selects the PL Parallel Port 1's I/O Address. The selections include [Disabled], [3BC/IRQ7], [378/IRQ7] or [278/IRQ5]. The [3BC/IRQ7] selection is factory set and recommended for most users.

■ Parallel Port Mode

Here, the parallel port's operation mode can be selected ([SPP], [EPP1.7], [EPP1.9], [ECP], or [ECP + EPP]). Normally, [SPP] is used and recommended for most users. When either [ECP] or [ECP+EPP] is selected, the [ECP Mode Use DMA] setting can be used.

■ EPP Mode Select

When [EPP] or [ECP + EPP] mode is selected, this feature allows you to select the EPP mode protocol. [EPP 1.7] or [EPP 1.9] can be selected.

■ ECP Mode Use DMA

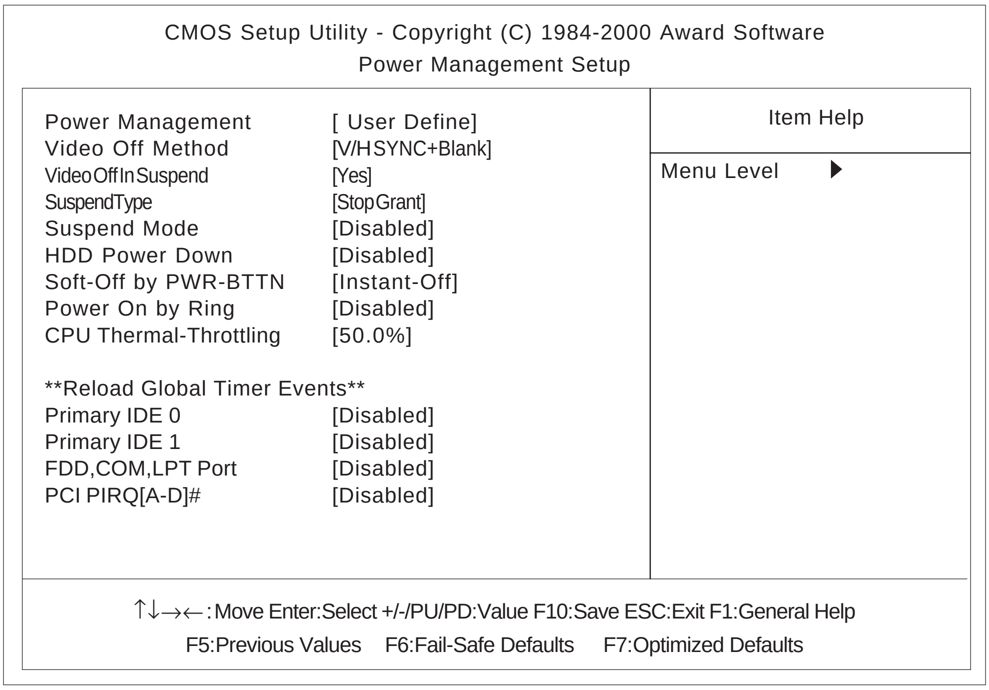
Designates the ECP mode's DMA channel. [1] and [3] are available selections, however, this item can be set only when the Parallel Port Mode is set to either [ECP] or [ECP+EPP].

■ PWRON After PWR-Fail

This feature is not supported.

5.2.6 Power Management Setup

Selecting the POWER MANAGEMENT SETUP menu item calls up the following screen.



```
CMOS Setup Utility - Copyright (C) 1984-2000 Award Software
                  Power Management Setup

Power Management            [ User Define]          Item Help
Video Off Method            [V/HSYNC+Blank]
Video Off In Suspend        [Yes]                   Menu Level   ▶
Suspend Type                [Stop Grant]
Suspend Mode                [Disabled]
HDD Power Down              [Disabled]
Soft-Off by PWR-BTTN        [Instant-Off]
Power On by Ring            [Disabled]
CPU Thermal-Throttling      [50.0%]

**Reload Global Timer Events**
Primary IDE 0               [Disabled]
Primary IDE 1               [Disabled]
FDD,COM,LPT Port            [Disabled]
PCI PIRQ[A-D]#              [Disabled]

↑↓→←:Move Enter:Select +/-/PU/PD:Value F10:Save ESC:Exit F1:General Help
   F5:Previous Values    F6:Fail-Safe Defaults    F7:Optimized Defaults
```

- **Power Management**

You can choose from three power management options. These are [User Define], [Min Saving] or [Max Saving]. The [User Define] selection is factory set and recommended for most users.

- **Video Off Method**

This setting determines the method to blank the display screen. The available settings are [Blank Screen], [V/H SYNC+Blank], and [DPMS Supported]. The [Blank Screen] selection blanks the display. The [V/H SYNC+Blank] blanks the display and also suspends the Vertical/Horizontal synchronization signal of the display. The [DPMS Supported] selection controls the operation when a CRT that supports DPMS is used. The factory default setting is [V/H SYNC+Blank] and is recommended for most users.

- **Video Off In Suspend**

Designates how the monitor is blanked. The available settings are [Yes] and [No]. The factory default setting is [Yes].

- **Suspend Type**

Designates the type of suspend method used. The available settings are [Stop Grant] and [PWRON Suspend]. The factory default setting is [Stop Grant].

- **Suspend Mode**

When enabled, designates the period of time before all devices except the CPU are shut down. The available settings are [1Min], [2Min], [4Min], [8Min], [12Min], [20Min], [30Min], [40Min], [1Hour] and [Disabled]. The factory default setting is [Disabled].

- **HDD Power Down**

  Designates the length of time until the motor of the hard disk stops. The available selections are[1Min]→[15Min] and [Disabled]. [Disabled] is factory set and recommended for most users.

- **Soft-Off by PWR-BTTN**

  Set up the power buttons used for software control. Available selections are [Delay 4 sec] and [Instant-off], however, since the PL unit has no power button for software control, be sure to select [Instant-off]. Factory setting is [Instant-off].

- **Power On by Ring**

  When enabled, the system will boot up when the modem receives a call. When disabled, the sytstem will ignore all calls received by the modem. The available settings are [Enabled] and [Disabled]. The factory default setting is [Disabled].

- **CPU Thermal-Throttling**

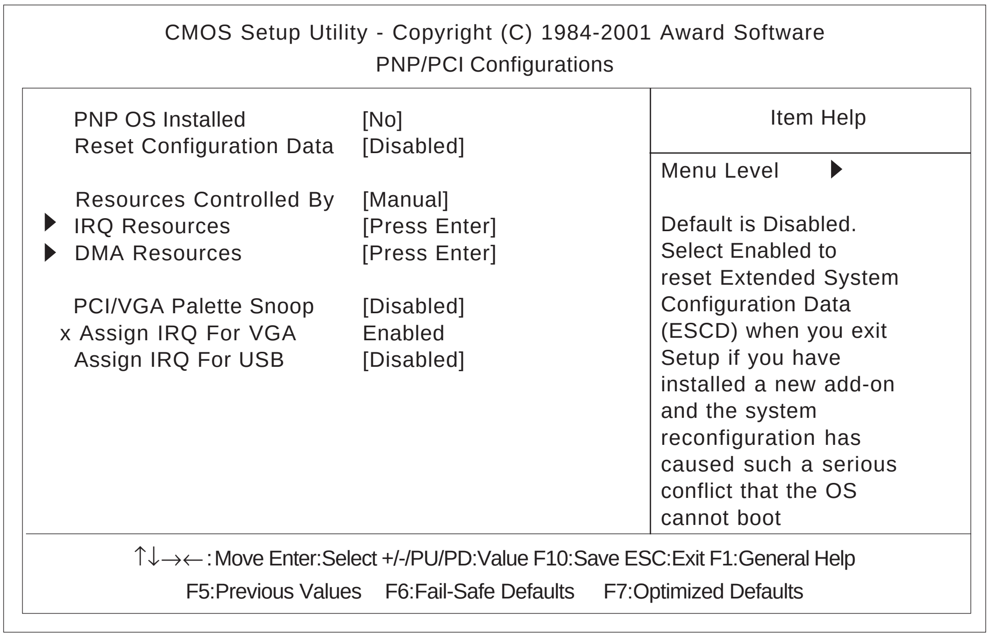
  Designates the CPU THRM-Throttling rate. The available settings are [12.5%], [25.0%], [37.5%], [50.0%], [62.5%], [75%] and [87.5%]. The factory default setting is [50.0%].

- ****Reload Global Timer Events****

  This feature sets the event that reloads the timer used to designate the amount (period) of PL unit idle time prior to changing to energy-saving mode. Thus, when an event occurs for any of the items in this area that are set to [Enabled], the system will reload the Timer.

5.2.7 PnP/PCI Configurations

Selecting the PnP/PCI CONFIGURATION menu item displays the following screen.



```
          CMOS Setup Utility - Copyright (C) 1984-2001 Award Software
                            PNP/PCI Configurations

     PNP OS Installed            [No]                       Item Help
     Reset Configuration Data    [Disabled]
                                                   Menu Level    ▶
     Resources Controlled By     [Manual]
   ▶ IRQ Resources               [Press Enter]     Default is Disabled.
   ▶ DMA Resources               [Press Enter]     Select Enabled to
                                                   reset Extended System
     PCI/VGA Palette Snoop       [Disabled]        Configuration Data
   x Assign IRQ For VGA          Enabled           (ESCD) when you exit
     Assign IRQ For USB          [Disabled]        Setup if you have
                                                   installed a new add-on
                                                   and the system
                                                   reconfiguration has
                                                   caused such a serious
                                                   conflict that the OS
                                                   cannot boot

  ↑↓→←: Move Enter:Select +/-/PU/PD:Value F10:Save ESC:Exit F1:General Help
          F5:Previous Values    F6:Fail-Safe Defaults    F7:Optimized Defaults
```

■ **PNP OS Installed**

Setting used when the user's OS complies with Plug-And-Play standards. The selection options are either [Yes] or [No]. The [No] setting is factory set and recommended for most users.

■ **Reset Configuration Data**

Designates whether ESCD (Extended System Configuration Data) data should be erased or not. The two selections available are [Disabled] and [Enabled]. The [Disabled] selection is factory set and recommended for most users.

■ **Resources Controlled By**

The Plug and Play feature allows you to designate whether the allocation of I/O Port, IRQ and DMA resources is performed automatically or manually. The two selections available are [Manual] or [Auto(ESCD)]. If [Auto(ESCD)] is selected, the IRQ Resources and DMA Resources selection will be disabled. The [Manual] selection is factory set and recommended for most users.

■ **IRQ Resources**

When resources are controlled manually, assign each system interrupt used a type, that reflects the type of device using the interrupt.

Reference **5.2.8 IRQ Resources**

■ **DMA Resources**

This menu provides control over the DMA resources used.

Reference **5.2.9 DMA Resources**

■ **PCI/VGA Pallet Snoop**

However, depending on the VGA or MPEG board used, select [Enabled]. For setup details, refer to that board's installation guide.

- **Assign IRQ for VGA**

Designates whether the VGA interrupt is allocated or not. Set to [Enabled].

- **Assign IRQ for USB**

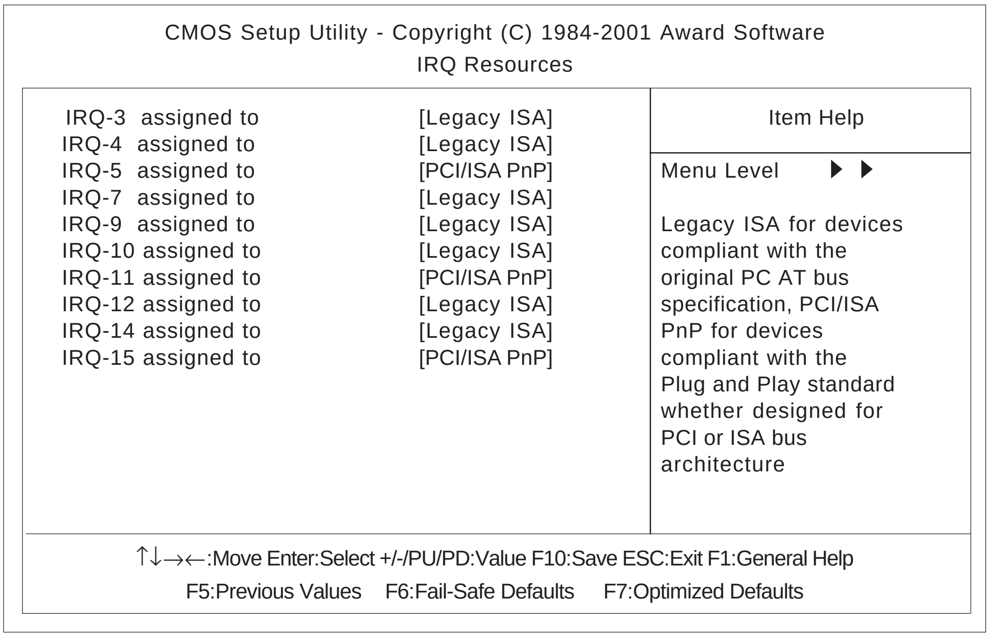
The two selections available are [Disabled] and [Enabled]. The [Disabled] selection is factory set and recommended for most users.When using the USB I/F or the USB connector for connecting the touch panel, set this item to “Enabled”.

Important

If a USB device is connected, change the [Integrated Peripherals] menu's [USB Controller] and [Assign IRQ For USB] settings to [Enabled].

5.2.8 IRQ Resources

Select IRQ Resources from the PnP/ PCI Configurations menu and the following screen will appear.



```
        CMOS Setup Utility - Copyright (C) 1984-2001 Award Software
                              IRQ Resources

  IRQ-3  assigned to          [Legacy ISA]   |        Item Help
  IRQ-4  assigned to          [Legacy ISA]   |
  IRQ-5  assigned to          [PCI/ISA PnP]  | Menu Level     ▶ ▶
  IRQ-7  assigned to          [Legacy ISA]   |
  IRQ-9  assigned to          [Legacy ISA]   | Legacy ISA for devices
  IRQ-10 assigned to          [Legacy ISA]   | compliant with the
  IRQ-11 assigned to          [PCI/ISA PnP]  | original PC AT bus
  IRQ-12 assigned to          [Legacy ISA]   | specification, PCI/ISA
  IRQ-14 assigned to          [Legacy ISA]   | PnP for devices
  IRQ-15 assigned to          [PCI/ISA PnP]  | compliant with the
                                             | Plug and Play standard
                                             | whether designed for
                                             | PCI or ISA bus
                                             | architecture

   ↑↓→←:Move Enter:Select +/-/PU/PD:Value F10:Save ESC:Exit F1:General Help
       F5:Previous Values    F6:Fail-Safe Defaults    F7:Optimized Defaults
```

- **IRQ-3 assigned to ~ IRQ-15 assigned to**

This setting determines the type of device assigned to the IRQ. This function is available when the [Resource Control By] option under the [PnP/ PCI Configurations] menu is set to [Manual].

[PCI/ISA PnP] .. Select to use a PnP-ready PCI or ISA card.

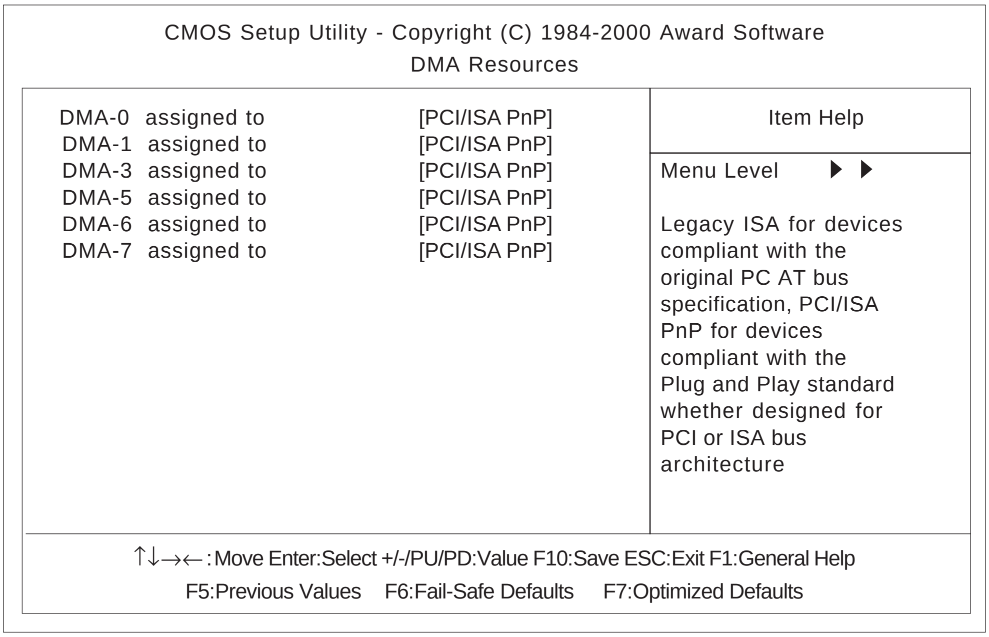
[Legacy ISA] ..... Select to use a non-PnP ISA card.

The initial settings are as shown below.

	Initial Value		Initial Value
IRQ-3 assigned to	Legacy ISA	IRQ-10 assigned to	Legacy ISA
IRQ-4 assigned to	Legacy ISA	IRQ-11 assigned to	PCI/ISA PnP
IRQ-5 assigned to	PCI/ISA PnP	IRQ-12 assigned to	Legacy ISA
IRQ-7 assigned to	Legacy ISA	IRQ-14 assigned to	Legacy ISA
IRQ-9 assigned to	Legacy ISA	IRQ-15 assigned to	PCI/ISA PnP

5.2.9 DMA Resources

Selecting DMA Resources from the PnP/PCI Configuration menu and the following screen will appear.



```
           CMOS Setup Utility - Copyright (C) 1984-2000 Award Software
                              DMA Resources

  DMA-0  assigned to          [PCI/ISA PnP]   |          Item Help
  DMA-1  assigned to          [PCI/ISA PnP]   |
  DMA-3  assigned to          [PCI/ISA PnP]   | Menu Level     ▶ ▶
  DMA-5  assigned to          [PCI/ISA PnP]   |
  DMA-6  assigned to          [PCI/ISA PnP]   | Legacy ISA for devices
  DMA-7  assigned to          [PCI/ISA PnP]   | compliant with the
                                              | original PC AT bus
                                              | specification, PCI/ISA
                                              | PnP for devices
                                              | compliant with the
                                              | Plug and Play standard
                                              | whether designed for
                                              | PCI or ISA bus
                                              | architecture

     ↑↓→← : Move Enter:Select +/-/PU/PD:Value F10:Save ESC:Exit F1:General Help
          F5:Previous Values    F6:Fail-Safe Defaults    F7:Optimized Defaults
```

■ **DMA-0 assigned to ~ DMA-7 assigned to**

This setting determines the type of device assigned to the port address. This function is available when the [Resource Control By] option under the [PnP/PCI Configurations] menu is set to [Manual].

[PCI/ISA PnP]...Select to use a PnP PCI or ISA card.

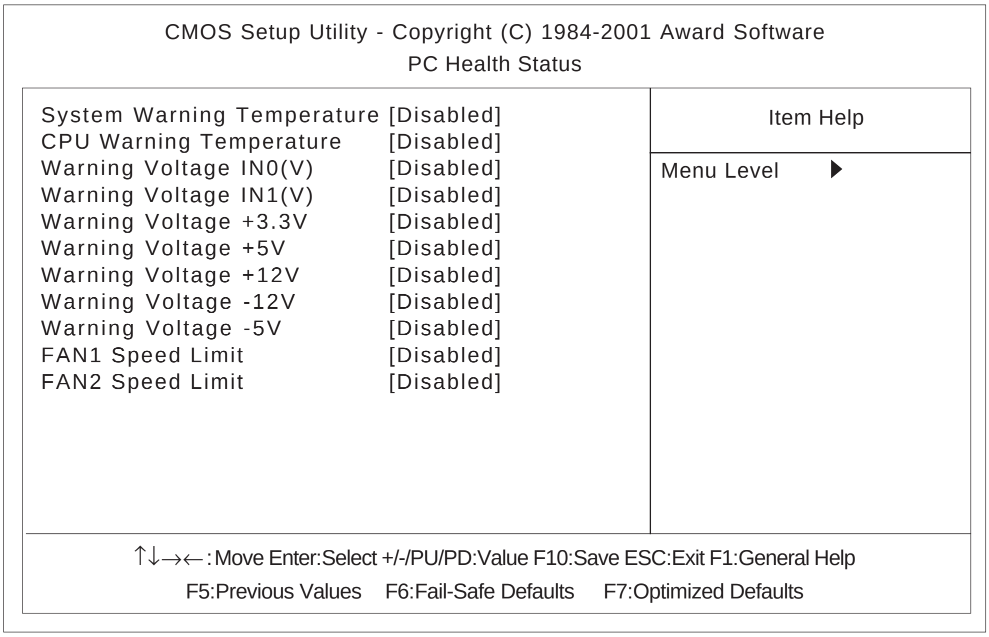
[Legacy ISA]...Select to use a non-PnP ISA card.

The initial settings are as shown below.

	Initial Value		Initial Value
DMA-0 assigned to	PCI/ISA PnP	DMA-5 assigned to	PCI/ISA PnP
DMA-1 assigned to	PCI/ISA PnP	DMA-6 assigned to	PCI/ISA PnP
DMA-3 assigned to	PCI/ISA PnP	DMA-7 assigned to	PCI/ISA PnP

5.2.10 PC Health Status

Select PC Health Status from the Main Menu and the following screen will appear.



```
CMOS Setup Utility - Copyright (C) 1984-2001 Award Software
                    PC Health Status

System Warning Temperature [Disabled]        Item Help
CPU Warning Temperature    [Disabled]
Warning Voltage IN0(V)     [Disabled]     Menu Level   ▶
Warning Voltage IN1(V)     [Disabled]
Warning Voltage +3.3V      [Disabled]
Warning Voltage +5V        [Disabled]
Warning Voltage +12V       [Disabled]
Warning Voltage -12V       [Disabled]
Warning Voltage -5V        [Disabled]
FAN1 Speed Limit           [Disabled]
FAN2 Speed Limit           [Disabled]

↑↓→←:Move Enter:Select +/-/PU/PD:Value F10:Save ESC:Exit F1:General Help
   F5:Previous Values   F6:Fail-Safe Defaults   F7:Optimized Defaults
```

- **System Warning Temp**

  This setting designates the system temperature at which a warning will be issued. The available settings are [40°C/104°F], [45°C/113°F], [50°C/122°F], [55°C/131°F], [60°C/140°F], [65°C/149°F], [70°C/158°F], [75°C/167°F], [80°C/176°F], [85°C/185°F] and [Disabled]. The factory default setting is [Disabled].

- **CPU Warning Temperature**

  This setting designates the CPU temperature at which a warning will be issued. The available settings are [40°C/104°F], [45°C/113°F], [50°C/122°F], [55°C/131°F], [60°C/140°F], [65°C/149°F], [70°C/158°F], [75°C/167°F], [80°C/176°F], [85°C/185°F] and [Disabled]. The factory default setting is [Disabled].

- **Warning Voltage IN0 (V)**

  When enabled, this setting determines the allowable range of the voltage of IN0 (Vcore) and designates that a warning will be issued if the temeperature exceeds that level. The available settings are [+/-4%], [+/-6%] and [Disabled]. The factory default setting is [Disabled].

- **Warning Voltage IN1 (V)**

  When enabled, this setting determines IN1's allowable voltage range and designates that a warning will be issued if the temeperature exceeds that level. The available settings are [+/-4%], [+/-6%] and [Disabled]. The factory default setting is [Disabled].

- **Warning Voltage +3.3V**

  When enabled, this setting designates that a warning will be issued if the temeperature exceeds +3.3V's designated range. The available settings are [+/-4%], [+/-6%] and [Disabled]. The factory default setting is [Disabled].

- **Warning Voltage +5V**

  When enabled, this setting designates that a warning will be issued if the temeperature exceeds +5V's designated range. The available settings are [+/-4%], [+/-6%] and [Disabled]. The factory default setting is [Disabled].

- **Warning Voltage +12V**

  When enabled, this setting designates that a warning will be issued if the temeperature exceeds +12V's designated range. The available settings are [+/-4%], [+/-6%] and [Disabled]. The factory default setting is [Disabled].

- **Warning Voltage -12V**

  When enabled, this setting designates that a warning will be issued if the temeperature exceeds -12V's designated range. The available settings are [+/-4%], [+/-6%] and [Disabled]. The factory default setting is [Disabled].

- **Warning Voltage -5V**

  When enabled, this setting designates that a warning will be issued if the temeperature exceeds -5V's designated range. The available settings are [+/-4%], [+/-6%] and [Disabled]. The factory default setting is [Disabled].

- **FAN1 Speed Limit**

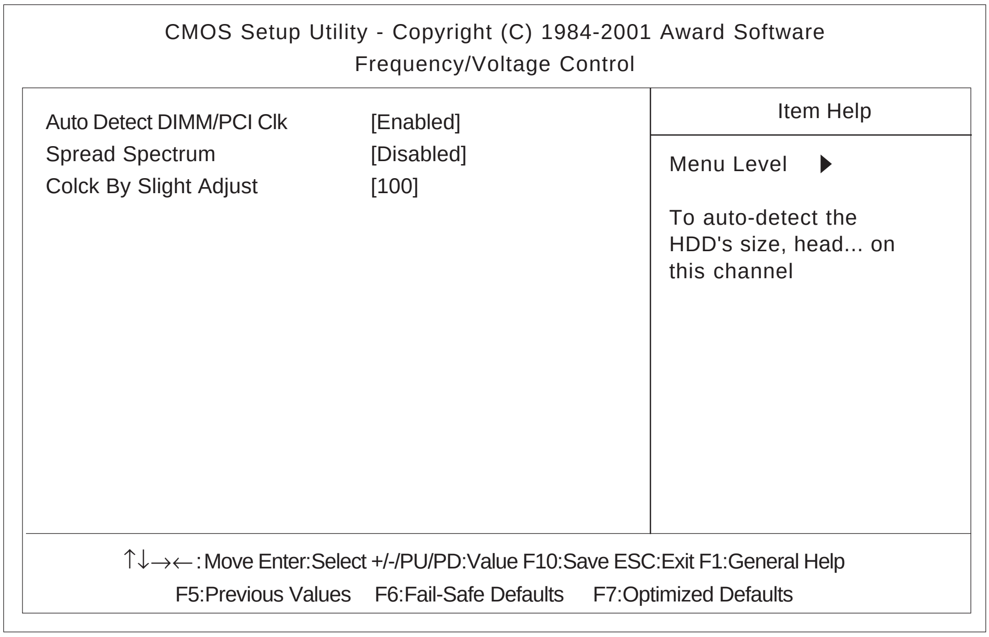
  When enabled, this setting designates the allowed speed range for FAN1. The available settings are [-30%], [-50%] and [Disabled]. The factory default setting is [Disabled]. FAN1 is used for the CPU fan.

- **FAN2 Speed Limit**

  When enabled, this setting designates the allowed speed range for FAN2. The available settings are [-30%], [-50%] and [Disabled]. The factory default setting is [Disabled]. FAN2 is used for the power supply fan.

5.2.11 Frequency/Valtage Control

Selecting the Frequency/Voltage Control menu item produces the following screen.



```
CMOS Setup Utility - Copyright (C) 1984-2001 Award Software
Frequency/Voltage Control

Auto Detect DIMM/PCI Clk        [Enabled]          Item Help
Spread Spectrum                 [Disabled]
Colck By Slight Adjust          [100]              Menu Level   ►

                                                   To auto-detect the
                                                   HDD's size, head... on
                                                   this channel

↑↓→←:Move Enter:Select +/-/PU/PD:Value F10:Save ESC:Exit F1:General Help
F5:Previous Values    F6:Fail-Safe Defaults      F7:Optimized Defaults
```

- **Auto Detect DIMM/PCI CLK**

  This setting designates the auto detect of the DIMM/PCI clock. The available settings are [Enabled] and [Disabled]. The factory default setting is [Enabled] and strongly recommended for users.

- **Spread Spectrum**

  This setting allows you to set the CPU Clock/Spread Spectrum. The available settings are [Enabled] and [Disabled]. The factory default setting is [Disabled] and strongly recommended for users.

- **Clock By Slight Adjust**

  This setting sets the CPU's clock speed in either of three ranges - 133MHz to 166MHz, 100MHz to 133MHz, or 66MHz to 100MHz, depending on the CPU's host clock. The available settings are [100] to [132]. The factory default setting is [100] and strongly recommended for users.

5.2.12 Load Fail-Safe Defaults

When the Menu screen's [Load Fail-Safe Defaults] is selected, you are able to designated if the minimum number of System Settings is used or not. The selections are [Y] and [N].

5.2.13 Load Optimized Defaults

Selecting [Load Optimized Defaults" designates whether or not you will set up the PL unit's revert to the PL unit's factory settings. The selections are [Y] and [N].

Note: **When the PL unit uses its factory settings, the USB interface cannot be used. For information about connecting the Touch Panel via the USB connector, see 1.2.1 Setting Up the Touch Panel Connection. For information about connecting the USB I/F, see 1.2.2 Using the USB Interface.**

■Dip Switch Settings

The USB/Serial connection setting is controlled via the PL unit's side face dip switches.

Depending on the communication format used, the DU unit's rear face dip switch settings will need to be changed. These settings must also be set on the PL unit's side face dip switchs. The factory setting is the Serial Connection. For details about Dip Switch settings, refer to Reference **3.1.8 Connecting the Standard Display (PL-DU6900/PL-DU7900).**

5.2.14 Set Supervisor Password

This password is used to change system information settings. It is designed to prevent unapproved users from changing the system information settings. Entering up to 8 characters here will overwrite the current password.

When you wish to have no password, click on the [Enter] key. Next, the words "PASS-WORD DISABLE" will appear, providing confirmation that the Password is no longer set.

When password input is required, use the [Advanced BIOS Features] area's [Security Option] feature to enter the password. Reference **5.2.3 ADVANCED BIOS FEATURES**

5.2.15 Set User Password

This password is used to view system information settings. It is designed to prevent unapproved users from viewing the system information settings. Entering up to 8 characters here will overwrite the current password.

When you wish to have no password, click on the [Enter] key. Next, the words "PASSWORD DISABLE" will appear, providing confirmation that the Password is no longer set.

When password input is required, use the [Advanced BIOS Features] area's [Security Option] feature to enter the password. See 5.2.3 ADVANCED BIOS FEATURES

Note:

- **Selecting either "Set Supervisor Password" or "Set User Password" allows you to view and change System Settings.**
- **When you enter a setting screen, "Set Supervisor Password" allows you to view or modify System Settings, however, "Set User Password" allows you to only view System Settings.**

5.2.16 Save & Exit Setup

This feature saves the settings entered in the Setup Utility and restarts the PL unit.

5.2.17 Exit Without Setting

This feature quits the Setup Utility program without saving any settings entered.

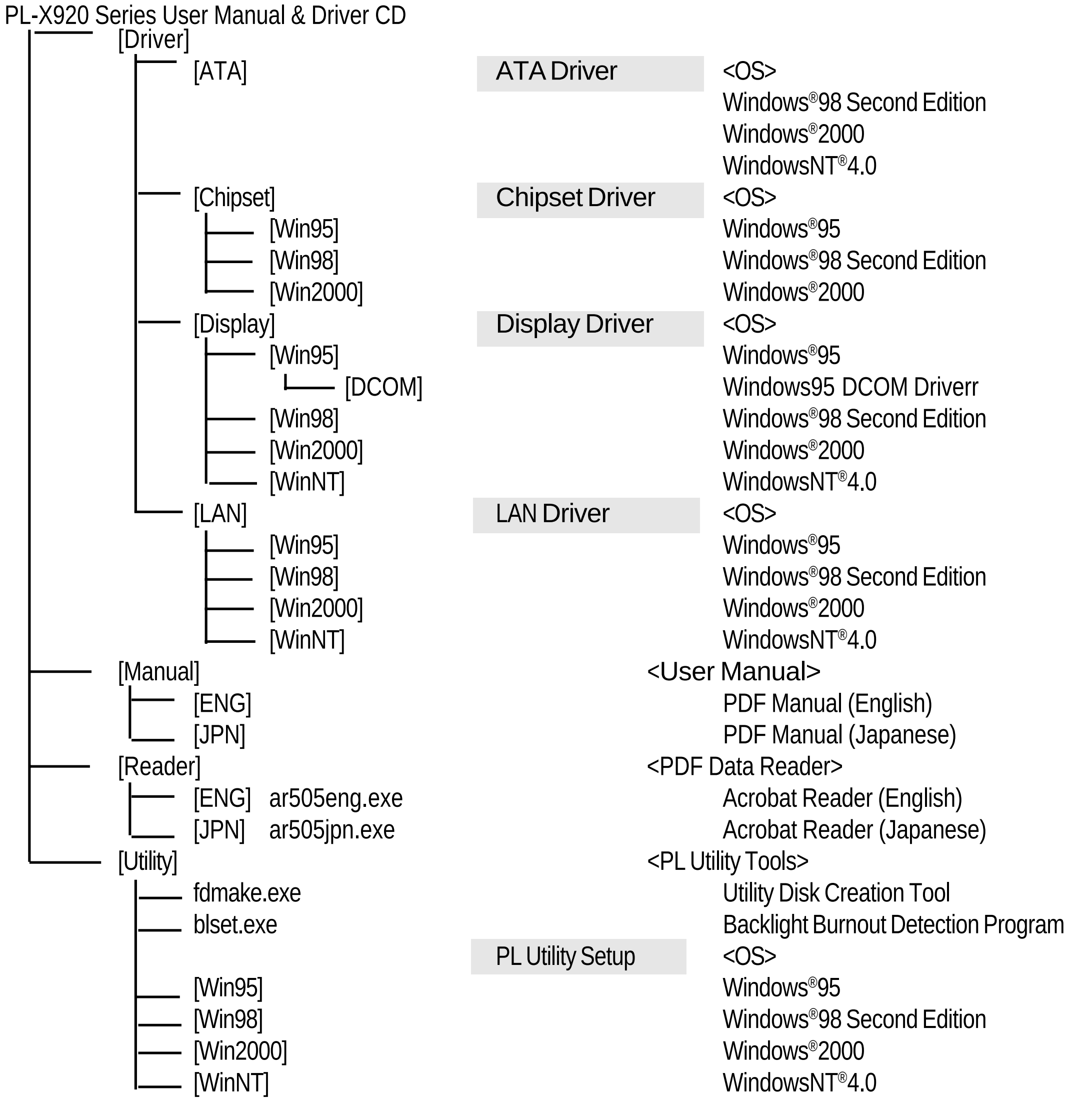
Chapter 6 Setting Up Your PL OS

Pro-face has prepared the following additional program files which are not supported by the standard versions of the Windows® 95 OSR2 or higher, Windows NT® 4.0, Windows® 98 Second Edition and Windows® 2000 operating systems. These files are located on the PL unit's additional CD-ROM.

6.1 CD-ROM Contents

6.1.1 Diagram

The following tree-structure diagram shows the contents of the CD-ROM disk.



```
PL-X920 Series User Manual & Driver CD
├── [Driver]
│    ├── [ATA]                       ATA Driver        <OS>
│    │                                                   Windows®98 Second Edition
│    │                                                   Windows®2000
│    │                                                   WindowsNT®4.0
│    ├── [Chipset]                   Chipset Driver    <OS>
│    │    ├── [Win95]                                    Windows®95
│    │    ├── [Win98]                                    Windows®98 Second Edition
│    │    └── [Win2000]                                  Windows®2000
│    ├── [Display]                   Display Driver    <OS>
│    │    ├── [Win95]                                    Windows®95
│    │    │    └── [DCOM]                                Windows95 DCOM Driverr
│    │    ├── [Win98]                                    Windows®98 Second Edition
│    │    ├── [Win2000]                                  Windows®2000
│    │    └── [WinNT]                                    WindowsNT®4.0
│    └── [LAN]                       LAN Driver        <OS>
│         ├── [Win95]                                    Windows®95
│         ├── [Win98]                                    Windows®98 Second Edition
│         ├── [Win2000]                                  Windows®2000
│         └── [WinNT]                                    WindowsNT®4.0
├── [Manual]                                         <User Manual>
│    ├── [ENG]                                           PDF Manual (English)
│    └── [JPN]                                           PDF Manual (Japanese)
├── [Reader]                                         <PDF Data Reader>
│    ├── [ENG]  ar505eng.exe                             Acrobat Reader (English)
│    └── [JPN]  ar505jpn.exe                             Acrobat Reader (Japanese)
└── [Utility]                                        <PL Utility Tools>
     ├── fdmake.exe                                      Utility Disk Creation Tool
     ├── blset.exe                                       Backlight Burnout Detection Program
     │                               PL Utility Setup  <OS>
     ├── [Win95]                                         Windows®95
     ├── [Win98]                                         Windows®98 Second Edition
     ├── [Win2000]                                       Windows®2000
     └── [WinNT]                                         WindowsNT®4.0
```

6.2 Setting Up Your PL OS

Prior to using the PL unit with the Windows® 95/Windows® 98 Second Edition/ WindowsNT® 4.0/Windows® 2000 operating system, certain utility software must be installed.

■ Installing the HDD Unit

A hard disk unit must be installed in the PL unit. Reference **PL-HD220 Installation Guide.**

■ System Parameters Setup

System Parameters Setup must also be entered into the PL unit. After these setting are entered, check that the HD drive is correctly recognized by the PL.

Reference **PL-HD220 Installation Guide.**

■ OS Setup

The PL unit is designed to operate using the following OS types.

Important

The PL unit is designed to operate under the following standard Windows OS versions. PL operation with any other maker's OS is not guranteed.

Windows® 95 OSR2 or higher

Windows® 98 Second Edition

Windows NT® 4.0 (Windows Service Pack 3 or higher)

Windows® 2000

■ PL Utility Setup

Use the "PL-X920 Series User Manual & Driver CD" to install the necessary drivers and utility software.

Note:

- **To set up the PL, a PS/2 type (Mini DIN) keyboard is required.**
- **To use the PL unit's touch panel, the touch panel device driver (PL-TD000) is required. When installing this driver, be sure to designate the COM port as COM4.**

For installation details, refer to the Installation Guide included with the PL-TD000.

◆ Installing Software from the CD-ROM

Note: **To install the software on the PL, Digital’s CD-ROM drive (PL-DK200) is required.**

To set up the correct software for your PL unit’s OS, be sure to use the “Disk1” folder’s “Setup.exe” file.

Ex. When running Windows® 98 Second Edition, and the CD-ROM drive is “D”:

D:Utility\Win98[*1]\Disk1\Setup.exe

◆ Installing Software from a FD

Note: **To install the software on the PL, Digital’s FDD drive (PL-FD200 or PL-FD210) is required.**

Use the PL’s floppy disk drive to create a floppy disk that can be used to install the Driver & Utility programs designed specifially for your PL unit’s OS. This floppy disk creation program is designed to run on Windows®.

Step 1 Insert the PL unit's additional CD-ROM disk "PL-X920 Series User's Manual & Driver CD" in the PL's CD-ROM drive.

Step 2 Double click on the “Fdmake.exe” program, that is located in the CD-ROM disk’s [OS] -> [Disk 1] folder. This will create the FD (Utility Disk).

D:Utility\Fdmake.exe (CD-ROM drive is “D”)

Step 3 Insert the FD(Utility Disk) in the PL's FDD unit.

Step 4 Insert the FD into drive A: and double click on the “Setup.exe” file. Follow the instructions given by the Installer program to complete the installation.

***1** *Windows® 95 OSR2 or higher:*	*“Win95”*
Windows® 98 Second Edition:	*“Win98”*
Windows NT® 4.0 (Windows Service Pack 3 or higher):	*“WinNT”*
Windows® 2000 :	*“Win2000”*

◆ Proface Folder Contents (on PL hard disk)

When you set up the PL unit's utility software, the folder [Proface] will automatically be created on the C: drive. Inside that folder are the following programs.

(Same for all Windows OS types.)

[Proface]

[69api]	API-DLL
[Ata]	ATA Driver (not included in Windows® 95)
[Blsaver]	Backlight control screen saver
[Chipset]	Chipset driver (not included in Windows NT® 4.0)
[Disp]	Display ON/OFF utility
[Display]	Graphic Accelerator Driver
[Keyclick]	On-screen Keyboard Emulator
[Lan]	LAN driver
[Sysmon]	System monitor/RAS application

Note:

- **The contents of the hard disk (shown above) may change, depending on the type of OS installed.**
- **In your PL hard disk's system folder you will find the following control-related drivers.**

PLSYSMON.VXD Hardware control driver (Windwos® 95)
PLSYSMON.SYS Hardware control driver (WindowsNT® 4.0, Windows® 2000)
BLCTRL.VXD Backlight control driver (Windows® 95)
BLCTRL.SYS Backlight control driver (WindowsNT® 4.0, Windows® 2000)

6.3 Installing Drivers

In order to use the PL unit's special features, 4 types of drivers have been created. ([ATA], [Chipset], [Graphic Accelerator], [LAN].

If your PL has no pre-installed OS, or has had its OS recovered, please install the following drivers as required.

The following explanation assumes the utility progams have been previously installed on your PL unit's hard disk in the [Proface] folder.

Drivers set up manually cannot be uninstalled.

■ Installing the ATA Driver

Use the following explanation to install the PL-X920 series ATA driver in your PL unit. Installing this driver will speed up your hard disk access. This driver, however, cannot be used with Windows® 95.

1) Click on the "iaa23_multi.exe" file in your Proface folder's "Ata" folder. Follow the instructions given to complete the installation.
C:\Proface\Ata\iaa23_multi.exe

Note: **When your PL unit's OS is Windows®2000, after installation, select the [Device Manager] area's [Secondary IDE Channel]. Click on the [Device Usage] item's [Disable in this hardware profile] check box.**

■ Installing the Chipset Driver

Use the following explanation to install the PL-X920 series Chipset driver in your PL unit. Installing this driver will cause your OS to recognize your hard disk. This driver, however, cannot be used with Windows NT® 4.0.

1) Depending on you PL OS, click on the following file in your Proface folder's "Chipset" folder. Follow the instructions given to complete the installation.
C:\Proface\Chipset\infinst_enu.exe (Windows® 95)
C:\Proface\Chipset\infinst_autol.exe (Windows® 98, Windows® 2000)

■ Installing the Graphic Accellerator Driver

Use the following explanation to install the PL-X920 series Graphic Accellerator driver in your PL unit. Installing this driver will speed up your PL unit's display, using special hardware features.

◆ With Windows® 95

1) Prior to installing this driver, start up the following file.
C:\Proface\Display\Dcom\Dcom95.exe

2) Start up C:\Proface\Display\win9xm66.exe. Follow the instructions given to complete the installation.

◆ With Windows® 98 Second Edition, Windows® 2000, Windows NT® 4.0

1) Depending on your PL OS Start up the following file and follow the instructions given.

C:\Proface\Display\win9xm67.exe (Windows® 98)

C:\Proface\Display\winnt4m67.exe (WindwosNT® 4.0)

C:\Proface\Display\win9xm67.exe (Windwos® 2000)

■ Installing the LAN Driver

Use the following explanation to install the PL-X920 series LAN driver in your PL unit. Installing this driver allows you to access a LAN.

◆ Installing the LAN Driver

1) Change the BIOS [Integrated Peripherals] menu's [Onboard LAN] setting to [Enabled]. Reference **5-2-5 Integrated Peripherals**

2) Start up the PL unit's OS. With Windows® 95 and Windows® 98, the Installation Wizard will appear.

◆ With Windows® 95

Note: **Be sure the PL unit's optional CD-ROM drive (PL-DK200) is connected and operating correctly prior to inserting the Windows® 95 OS CD-ROM into the CD-ROM drive.**

1) Click on [Next].

2) Click on [Other Locations].

The location designation wizard will appear.

3) Enter "C:\Proface\lan " and click [OK].

4) Enter "C:\Proface\lan " and click [OK] again.

The device driver wizard will appear.

5) Click on [Finish].

6) Enter "D:\Win95" and click on [OK].

Files will be copied from the Windows95 CD-ROM to the PL unit.

7) Click on [Yes] and restart the PL unit to complete the installation.

◆ With Windows® 98 Second Edition

Note: **Be sure the PL unit's optional CD-ROM drive (PL-DK200) is connected and operating correctly prior to inserting the Windows® 98 OS CD-ROM into the CD-ROM drive.**

1) Click on [Next].

2) Select [Search for a better driver than the one your device is using now. (Recommended)] and click on [Next].

3) Click on the [Specify a location] check box, enter "C:\Proface\Lan" in the location window, and click on [Next].

4) Click on [Next].

The file copy dialog box will appear and files will be copied from the Windows98 CD-ROM.

5) Click on [Finish]

The system settings dialog box will appear.

6) Click on [Yes] to restart your PL.

◆ With Windows NT® 4.0

Click the [Start] button, point to [Settings] and click on [Control Panel (C)].

1) Double click on the [Control Panel]'s [Network] icon.

The [Network Configuration] dialog box will appear.

2) Click on [Yes].

The Network Setup Wizard will appear.

3)Select [Wired to the network:], and click on [Next].

4) Click on [Select from list].

The Network Adaptor selection dialog box will appear.

5) Click on [Have disk].

The "Insert floppy disk" dialog box will appear.

6) Enter "C:\Proface\lan" and click [OK].

The "Select OEM Option" dialog box will appear.

7) Click on [OK].

The Network Setup wizard will appear.

8) Click on [Next].

9) Select the desired network protocol and click on [Next].

10) Select the desired service to install and click on [Next].

11) Click on [Next].

The WindowsNT setup dialog box will appear.

12) Enter "D:\I386" and click [Continue].

13) Enter "C:\Proface\lan" and click [Continue].

The "Speed /Duplex mode" dialog box will appear.

14) Click on [Continue].

The "Input Network Address" dialog box will appear.

15) Click on [OK].

The "Input Tx Early Threshold" dialog box will appear.

16) Click on [OK].

The TCP/IP dialog box will appear.

17) Enter the settings to connect with your network.

The Network Setup Wizard will appear.

18) Click on [Next].

19) Click on [Next].

20) Enter the Computer and Workgroup names, and click on [Next].

21) Click on [Finish].

The Network Settings Change dialog box will appear.

22) Click on [Yes(Y)] to restart your PL.

The Service Control Manager dialog box will appear.

23) Restarting the PL will cause an error message to appear, which requires the PL's Service Pack to be reinstalled.

24) After the Service Pack is reinstalled, restart the PL.

◆ With Windows® 2000

Click the [Start] button, point to [Settings] and click on [Control Panel (C)].

1) Double click on the [Control Panel]'s [Network and Dial-Up Connections] icon.

The Network and Dialup Connection window will appear.

2) Right click on the [Local Area Connection] icon, and select [Properties].

The Local Area Connection properties will appear.

3) Click on [Configure].

The Realtek RTL8139(A) PCI Fast Ethernet Adapter properties will appear.

4) Click on [Update Driver].

The Upgrade Device Driver Wizard will appear.

5) Click on [Next].

6) Select [Search for a suitable driver for my device (recommended)] and click [Next].

7) Select [Specify a location] and click [Next].

8) Enter “C:\Proface\lan “ and click [OK].

The search for the driver will start.

9) Click on [Next].

10) Click on [Finish] to restart your PL.

6.4 Application Features

The PL unit is equipped with the following special features. The following files have been copied to the PL unit's hard disk and are contained in the [Proface] folder.

File Name	Windows®95/Windows®98 Second Edition	Windows NT®4.0/Windows®2000
PL_BLIOC.DLL	C:\Windows\System	C:\Winnt\System32
PL_DLL.DLL		
PL_IOC.DLL		
Backlight Control.scr		
Disp.exe	C:\Proface\Disp	
Keyclick.exe	C:\Proface\Keyclick	
PL_Smon.exe	C:\Proface\Sysmon	
PL_Wps.exe	C:\Proface\Sysmon	

■ **API-DLL**

This is a dynamic library designed to provide access to the System BIOS' RAS feature for User applications. API-DLL consists of three types, which are explained below.

◆ **Backlight Control API-DLL file (PL_BLIOC.dll)**

This API-DLL file provides a dynamic library that allows User-created applications to utilize the PL-X920 series' backlight control feature. This file must be installed into the same directory as the User's application.

For details, Reference ***Appendix 4 - Backlight Control Feature API-DLL***

◆ **System Monitor API-DLL (PL_DLL.DLL)**

This API-DLL file provides a dynamic library that allows User-created applications to utilize the PL-X920 series' System Monitor feature. This file must be installed into the same directory as the User's application.

◆ **RAS Feature API-DLL (PL_IOC.DLL)**

This API-DLL file provides a dynamic library that allows User-created applications to utilize the PL-X920 series' System BIOS' RAS feature.

For details, Reference ***Appendix 3 - System Monitor/RAS Feature API-DLL***

■ **Backlight OFF Screen Saver (Backlight control.scr)**

This software is used to turn OFF the PL's backlight after a specified period of inactivity. The use of this feature will help to extend the life of the PL's backlight.

Note: **Certain application programs may not allow the PL's backlight to turn OFF. Please test each program individually to check if the screen saver will operate correctly.**

■ Screen Display ON/OFF Utility (DISP.EXE)

This command line utility is used to turn OFF both the PL's backlight and display.

Settings Used	DISP [ON/OFF]
Option Switch	ON: Displayed / OFF: Not Displayed
Return Value	0: Completed Normally / -1: Option Switch Error

■ Keyboard Emulator (Keyclick32)

This program allows the User's mouse operation to perform keyboard-like data input.

Note:

- **Certain application programs do not support this keyboard emulator. Please test each application individually to check if the keyboard emulator will operate correctly.**
- **This application cannot be used to enter Windows® startup screen User Name and Password information.**
- **To change the Keyclick program's font size a keyboard is required.**

- **For details concerning the Keyclick program's operation, simply click on the HELP button to call up the program's online help data.**

■ System Monitor/RAS Application PL_SMON.EXE / PL_WPS.EXE

This utility provides monitoring of the PL's temperature, voltage level, and fan's operation, via the system BIOS' RAS and system monitoring functions.

◆ System Monitor Program Pl_Smon.exe

For details, Reference ***Appendix 3.3 System Monitor Operation***

◆ Monitor Parameter Setting Program Pl_Wps.exe

For details, Reference ***Appendix 3.2 System Monitor Property Settings***

6.4.1 Uninstalling PL-X920 Driver and Utility

1) Click the Windows main screen's bottom left corner [Start] button.

2) Click on [Settings] and then on [Control Panel].

3) Click the Control Panel's [Add/Remove Programs] icon.

4) Select the "PL-X920 Driver and Utility" and click on the Add/Remove button to remove the software.

Important

All drivers installed with this program cannot be uninstalled.

6.5 WindowsNT®4.0/Windows®2000 Cautions

Perform the following settings as required by your OS.

6.5.1 Automatic System Log-On Setup

■ **When using Windows NT® 4.0**

1) Click on the main screen's "Start" button, and select the "Enter Filename" item. Enter the text "C:\WINNT\REGEDIT.EXE" and press [Enter] to start the program.

2) When the REGEDIT Registry Tree appears, select the "Winlogon" subkey via the followingtext:

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\WindowsNT\ Current Version\Winlogon.

3) In the "DefaultUserName" field, enter the User name to be used for the Automatic Log-on.

4) Select the Edit menu's [New/String Value] feature.

5) To the Data Items present, add "AutoAdminLogon" to the Name column's data, and then enter "1" in that entry's Data field.

6) Add "Default Password" to the Name column's data, and enter the password used previously for the DefaultUserName in the Data field.

Important

A user with no password cannot automatically log on.

7) REGEDIT data entry is now finished.

Note:

- **If a User attempting to automatically log on is not attached to an "Administrators" group, i.e. no Default Password string is specified, Windows NT automatically changes the value of the AutoAdminLogon key from 1(true) to 0(false), thereby disabling the AutoAdminLogon feature. In that case, if the Shift key is held down during Logoff, the "Login Data" dialog box will appear and the User can use the name of another, registered User to log-on successfully.**
- **If Auto LogOn Setting are not entered, when logging on, a PS/2 type keyboard is required.**

■ **When using Windows® 2000**

1) Start the Control Panel's [Users and Passwords].
2) Select an automatic log-on user and deselect the [Users must enter a user name and password to use this computer] checkbox.
3) Click on the [Advancedl] tab and deselect the [Require users to press Ctrl-Alt-Delete before logging on] checkbox.
4) Click the [Apply] button and when the automatic login dialog box appears, enter your password.

6.5.2 Using an Uninterrupted Power Supply

Prior to turning OFF the PL's power, be sure to shut down the NT correctly via the NT OS' "Shutdown" feature. It is recommended that an Uninterrupted Power Supply Device is used to prevent the accidental loss of User data, due to an unexpected power outage.

When using an Uninterrupted Power Supply (UPS), the unit can be set to switch to backup power, which will provide enough time to safely shut down your PL, or it can even shut down your PL for you.

For details, please consult your local dealer of UPS units.

6.5.3 When Changing the System Design

When the PL unit is connected to a printer or to a LAN network, the Windows system settings must be changed.

■ **When using Windows NT® 4.0**

- **Changing the System Design**

  When the Windows NT® 4.0 system design is changed, the following messages will appear.

Windows NT Setup

Setup needs to copy some Windows NT files

Setup will look for the files in the location specified below. If you want Setup to look in a different place, type the new location. When the location is corrent, click Continue.

Files Needed

Some files on WindowsNT Workstation CD-ROM are needed. Insert WindowsNT Workstation CD-ROM into the drive selected below, and then click OK.

Even when one of these messages appears, designate a new location for the system design change folder (Windows NT® 4.0 CD-ROM's [I386]) and click on [Next].

D:\I386 (CD-ROM drive is "D")

- **Reinstalling Service Pack Data**

  When changing the Windows NT system settings, the system files are written over your existing Service Pack 1 files. Be sure to set up your Service Pack data again.

■ **When using Windows® 2000 Setup**

- **Changing the System Design**

  When the Windows® 2000 system design is changed, the following messages will appear.

> **Please insert the floppy disk labeled'Windows2000 Professional CD-ROM' into drive D and then click OK.**
>
> **You can also click OK if you want files to be copied from an altemate location. such as a network sever or a compact disc.**

Designate the new folder location for the system settings (Windows NT® 2000 CD-ROM's [I386]) and click on [Next].

D:\I386 (CD-ROM drive is "D")

6.5.4 Changing to the NTFS File System

■ **With a hard disk using WindowsNT® 4.0 and Windows® 2000**

If your hard disk was formatted using the Windows DOS compatible FAT32 system, you can use Command Prompt to convert the hard disk to an NTFS system.

User the following comand.

convert x:/fs:ntfs, where "x" is the drive name of your hard disk.

Note:
- **After converting data to the NTFS file system, it cannot be converted back to the FAT32 (DOS compatible) file system.**

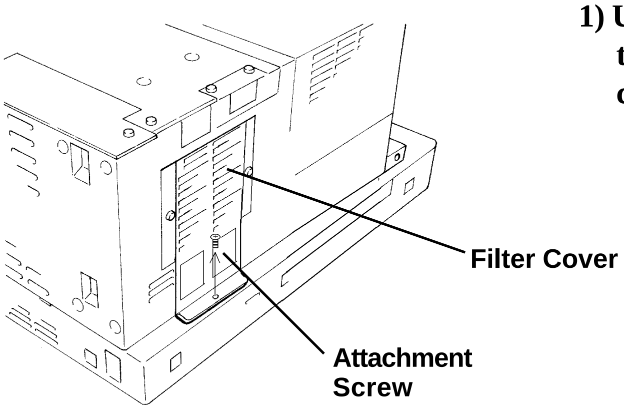
Memo

Chapter 7 Maintenance and Inspection

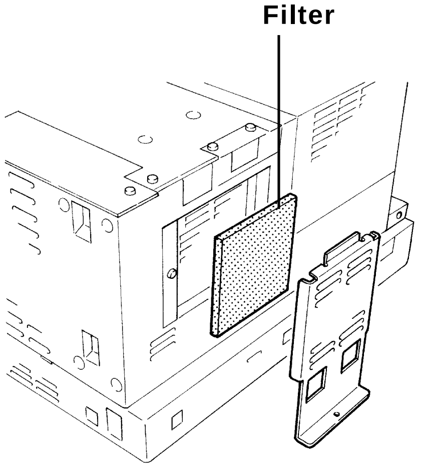
7.1 Cleaning the Filter

Since excessive dirt and dust in the filter of the PL's cooling fan can potentially affect the performance of the unit, regular inspection and cleaning of the filter is strongly recommended.

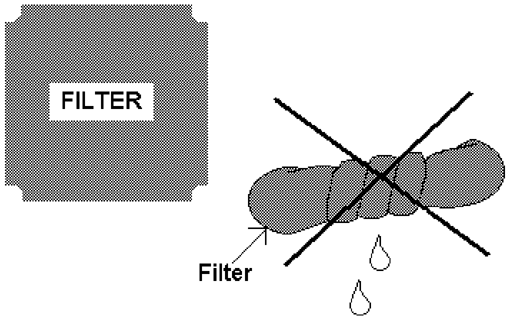
■ PL-B920 Unit



1) Unscrew the cooling fan filter cover's attachment screw and remove the filter cover.



2) Remove the filter from the fan cover.



3) Clean the filter completely.

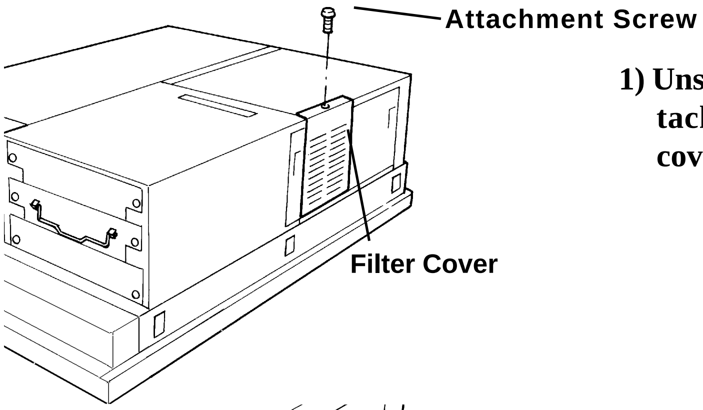
If stubborn dirt or stains are present, wash the filter with a neutral detergent.

Important

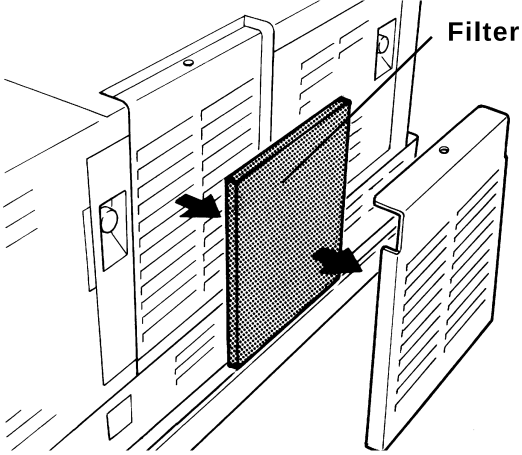
- ***Do not wring the filter to dry it.***
- ***Be sure the filter is completely dry before reattaching the filter to the fan cover.***
- ***Allow it to air dry. Do not leave the filter in direct sunlight.***

4) After reattaching the filter to the filter cover, reattach the cover to the cooling unit with the filter cover attachment screw.

■ PL-B921 Unit



1) Unscrew the cooling fan filter cover's attachment screw and remove the filter cover.



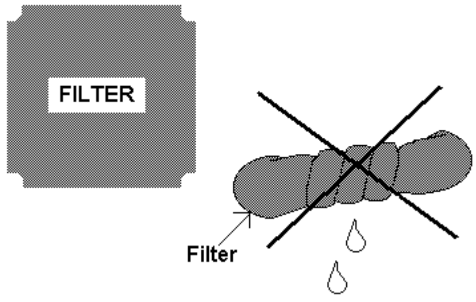
2) Remove the filter from the fan cover.

3) Clean the filter completely.

If stubborn dirt or stains are present, wash the filter with a neutral detergent.

Important

- ***Do not wring the filter to dry it.***
- ***Be sure the filter is completely dry before reattaching the filter to the fan cover.***
- ***Allow it to air dry. Do not leave the filter in direct sunlight.***



4) After reattaching the filter to the filter cover, reattach the cover to the cooling unit with the cover attachment screw.

7.2 Periodic Maintenance Points

Check the PL periodically to ensure it is in good working condition.

Ambient environment check

- Is the ambient temperature within the specified range?

	With HDD, FDD	Without HDD, FDD
With Fan	5 to 50 °C	0 to 50 °C
Without Fan	5 to 40 °C	0 to 40 °C

- Is the ambient humidity within the specified range (10%RH to 85 %RH) ?
- Is the atmosphere free of corrosive gas ?

Electrical specifications check

- Is the voltage adequate (AC85V to AC265V, 50/60 Hz) ?

Installation check points

- Is the connection cable firmly connected (not loose) ?
- Are any bolts or screws loose ?
- Are there any flaws or tears in the moisture resistant gesket ?

Display check

- Is the display bright enough ?

Note: **When the PL’s Standard Display (DU)’s backlight needs to be replaced, please contact your local PL distributor.**

Memo

Appendices

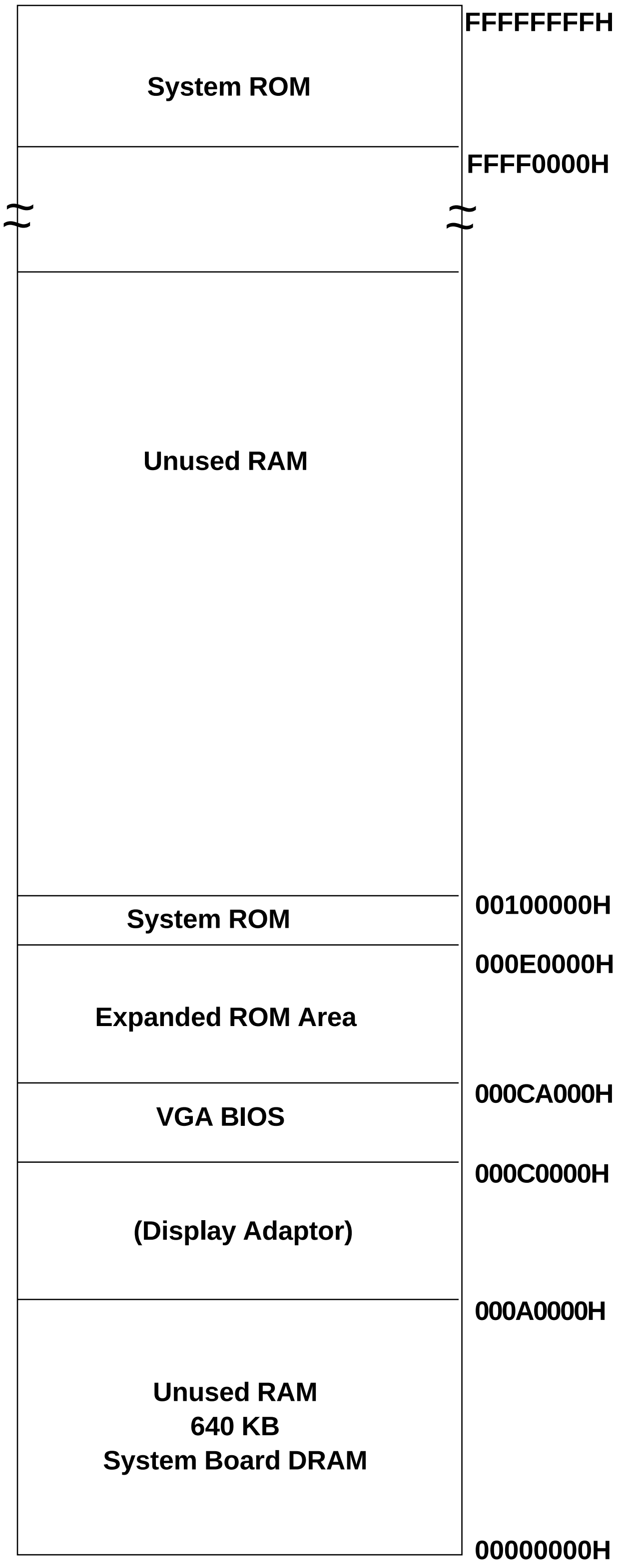
A.1 Hardware Configuration

The following data explains the design of the I/O Map, Memory Map and Interrupt Map, as well as additional hardware design items including the RAS feature.

A.1.1 I/O Map

Address	AT System Device	System Device
0000H - 001FH	DMA controller (8237)	
0020H - 003FH	Interrupt controller (8259A)	
0040H - 005FH	System timer (8254)	
0060H - 006FH	Keyboard Controller	
0070H - 007FH	Real-time clock, NMI mask	
0080H - 009FH	DMA page register	
00A0H - 00BFH	Interrupt controller 2 (8259A)	
00C0H - 00DFH	DMA controller 2 (8237)	
00F0H - 00FFH	Numeric data processor	
01F0H - 01FFH	Hard disk (IDE)	
0200H - 0207H	Game I/O	
0290H - 029FH	Reserved	
02E8H - 02EFH	Reserved	Touch Panel Serial Port 4 (COM4) (for Standard Display)
02F8H - 02FFH	Serial port 2 (COM2) : General Use	
03B0H - 03BBH	Video controller (VGA)	
03BCH - 03BFH	Parallel port 1 (LPT1)	
03C0H - 03DFH	Video controller (VGA)	
03E8H - 03EFH	Reserved	Serial port 3 (COM3)
03F0H - 03F7H	Floppy disk controller	
03F8H - 03FFH	Serial port 1 (COM1) :General Use	

A.1.2 Memory Map



A.1.3 Interrupt Map

■ Hardware Interrupt List

	Description
NMI 0	Parity Error or I/O Channel Check
IRQ 0	Timer (in the Chipset)
IRQ 1	Keyboard
IRQ 2	Cascade from Controller 2
IRQ 3	Serial Port 2 (COM2): General Use Port
IRQ 4	Serial Port 1 (COM2): General Use Port
IRQ 5	Available for users *
IRQ 6	Floppy Disk Controller
IRQ 7	Parallel Port 1 (LPT1) : Printer Port
IRQ 8	Real Time Clock
IRQ 9	Serial Port 3 (COM3): General Use Port
IRQ 10	Serial Port 4 (COM4): Touch Panel (for Standard monitor)
IRQ 11	Available for users *
IRQ 12	PS/2 Mouse
IRQ 13	Numeric Data Processor
IRQ 14	Hard Disk (IDE)
IRQ 15	Available for users *

Important

The "*" mark indicates that, depending on the customer's settings, the following devices are automatically allocated as Plug and Play devices.

Display Controller

SMBus Controller

Multimedia Device

Network Controller

USB Controller

■ DMA Channel List

	Description	
DMA 0		For 8-bit transmission
DMA 1		
DMA 2	Floppy disk controller	
DMA 3		
DMA 4	Cascade to controller 1	For 16-bit transmission
DMA 5		
DMA 6		
DMA 7		

A.2 RAS Feature

A.2.1 PL's RAS Features

RAS, which stands for Reliability, Availability and Serviceability, is a device-level monitoring function that provides a variety of features to improve the reliability of your PL system.

Though the standard set of RAS features used will vary depending on the devices used, the following features are used to provide Alarm Monitoring and External Input Signal support.

Alarm Monitoring	Power Voltage Alarm Cooling Fan Alarm Internal Temperature Alarm Watchdog Timer Time Up Soft Mirror Disk Alarm [*3] Hard Mirror Disk Alarm [*1 *3]
External Input Signal	General Purpose Input (DIN 2 bit) Remote Reset Input [*2]

Important

General Purpose Input (DIN) Level must be 1.5S or longer to be detected.

Also, when either the one of the above mentioned alarms occurs, or an external signal input is received, the following types of alarm processing output signals and features are supported.

External Output Signal	General Purpose Output (DOUT 1 bit) Alarm Output (1 point) Lamp Output (1 point)
Types of Processing (all units)	LED Indicator (2-state display – 1 point) Pop-up Message Output Buzzer Output System Shutdown

Furthermore, using the PL's System Monitor feature (included in the PL's software utility disk), allows the easy setting and control (Enable/Disable) of the aforementioned Alarm Monitor and External Input Signals.

Reference *A.3 - System Monitor*

Last, the system monitor feature's use of an Application Link Library (API-DLL) allows it to also be used with other applications.

*[*1] When a Hardware Mirror Disk Alarm occurs and the standard RAS feature settings are used, the alarm output is limited to the Mirror Disk unit's LED indicator. (Alternately flashing orange and green)*

*[*2] The remote reset feature's input can be either enabled or disabled, the alarm output setting cannot be set to trigger a forced system reset.*

*[*3] The Hardware Mirroring Disk and Software Mirroring Disk must be purchased separately.*

A.2.2 RAS Feature Details

■ Alarm Monitoring

◆ Power Voltage Alarm

Monitors the condition of the PL’s internal and CPU power.

◆ Cooling Fan Alarm

Monitors the condition of the PL’s internal power and CPU cooling fans.

◆ Internal Temperature Alarm

Monitors the PL’s internal and CPU vicinity temperatures.

The degree of monitoring (3 levels) and the enabling or disabling of the above three items is performed via the System Setup Area’s settings.

For detailed information about the monitoring level settings,

Reference ***5.2.10 PC Health Status***

This utility can also be used to enable or disable the above mentioned features, as well as designate what type of processing is to be performed.

◆ Watchdog Timer Time Up

This feature alternately writes Time Up Count values from the CPU to the RAS feature’s special programmable timer and then periodically clears them, which provides a means of monitoring the CPU’s performance. If the clearing of this count value is stopped, the timer will overflow and an alarm will be detected. The System Monitor utility can be used to enable or disable this feature, as well as designate what type of processing is to be performed.

◆ Software Mirroring Disk Alarm

This optional product monitors the performance of the PL unit's hard disk, via the optional Software Mirroring disk.

◆ Hardware Mirroring Disk Alarm

Whenever a disk crash, or other alarm event occurs to the optional Hardware Mirroring Disk unit, the PL unit’s LED indicator will flash (either orange or green) to indicate there is a problem.

■ External Input Signal

The PL’s RAS interface connector uses the following input signals.

General Purpose Input (DIN)

This standard digital input is used for alarm detection in external devices. The input signal uses two bits.

The System Monitor utility can be used to enable or disable this feature, as well as designate what type of processing is to be performed once a signal is received.

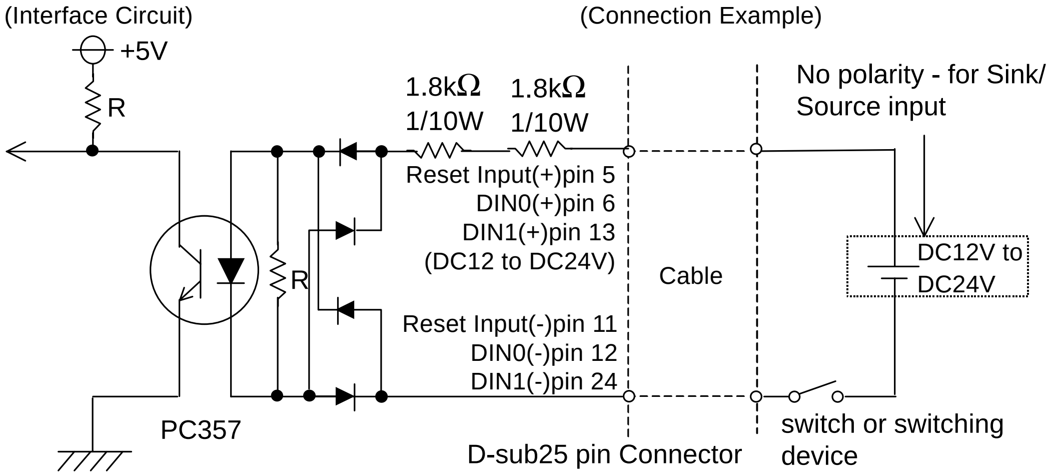
Remote Reset Input

This is the reset signal sent from an external device to the PL. When this signal is enabled, a forced reset of the PL is performed.

The System Monitor utility can be used to enable or disable this feature

External Input Signal (for both DIN and Remote Reset Input)

Input Voltage	DC12V to DC24V
Input Current	7mA
Operating Voltage	ON voltage: 9V (min), OFF voltage:3V (max)
Isolation Method	Via photocoupler



Important

- ***General Purpose Input (DIN) level must be 1.5S or longer to be detected.***
- ***Be sure the voltage value between terminals is controlled via the input voltage, so that the PL is operated within its recommended range. If the input voltage exceeds this range, a malfunction or PL damage may occur.***
- ***With Sink/Source input, even if the D(-), and RESET(-) are positive, and D(+), RESET(+) are negative, no problems are created. Be sure to operate the unit within the recommended voltage range.***

Note:

For connection pin location details,

Reference ***2.3.7 RAS Interface***

■ External Output Signal

The PL's RAS interface connector uses the following output signals.

General Purpose Output (DOUT)

This general purpose digital output signal provides system condition information to external devices.

The System Monitor's API-DLL are used by applications to control this signal.

Alarm Output (1 point)

Lamp Output (1 point)

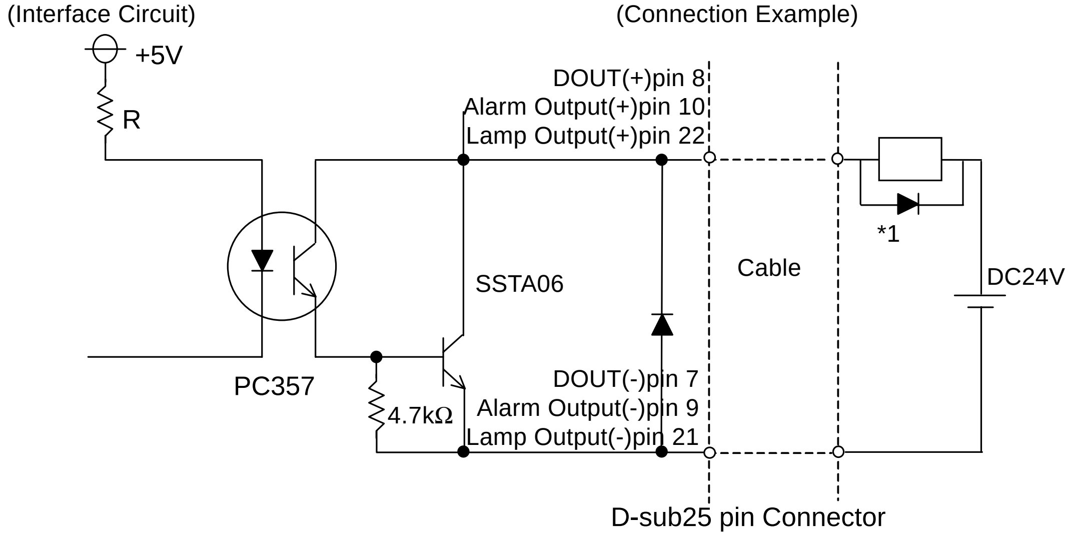
The above mentioned general purpose digital output signals provide system condition information to external devices.

The System Monitor utility can be used to enable or disable any of these output signals.

Also, when alarm output is enabled, the orange colored LED indicator will also blink.

External Output Signal (used for DOUT, Alarm Output, Lamp Output)

Rated Load Voltage	DC12V to DC24V
Maximum Load Current	100mA/point
Maximum Voltage Drop between Terminals	1.5V (at 100mA load current)
Isolation Method	Via photocoupler



- **Be sure to operate the unit within its maximum load current. If the maximum load current exceeds this range, a malfunction or PL damage may occur.**
- **Design your electrical system by adding the load current and voltage values to the terminal voltage. If load current value used is large, a maximum voltage of 1.5V will exist between the terminals.**
- **When connecting an induction load, be sure to connect the above drawing's protection diode (*1) .**

For connection pin location details,

Reference ***2.3.7 RAS Interface***

■ Types of Processing (all units)

The PL provides system condition information via the following methods.

LED Indicator (2-state display – 1 point)

In addition to indicating if the unit's power is ON or OFF, the 2-state LED indicator (power lamp) provides the following system condition information. PL-B920 LED indicator is on the left side of PL unit's. When connecting to a DU, the DU's LED indiator is on the left bottom of the DU screen.

Color	System Condition	Output Created
Green	Normal Operation (Power ON)	None
Orange	RAS has detected a system alarm.	The alarm output set in the system monitor is enabled.
Flashing Orange/ Green	Soft Mirror Disk Alarm and Mirror Disk Alarm has occurred	None

Pop-up Message Output

This feature uses the Windows® system's pop-up message feature to indicate that an alarm has occurred.

Buzzer Output

This feature uses the PL's internal speaker to indicate the system's condition.

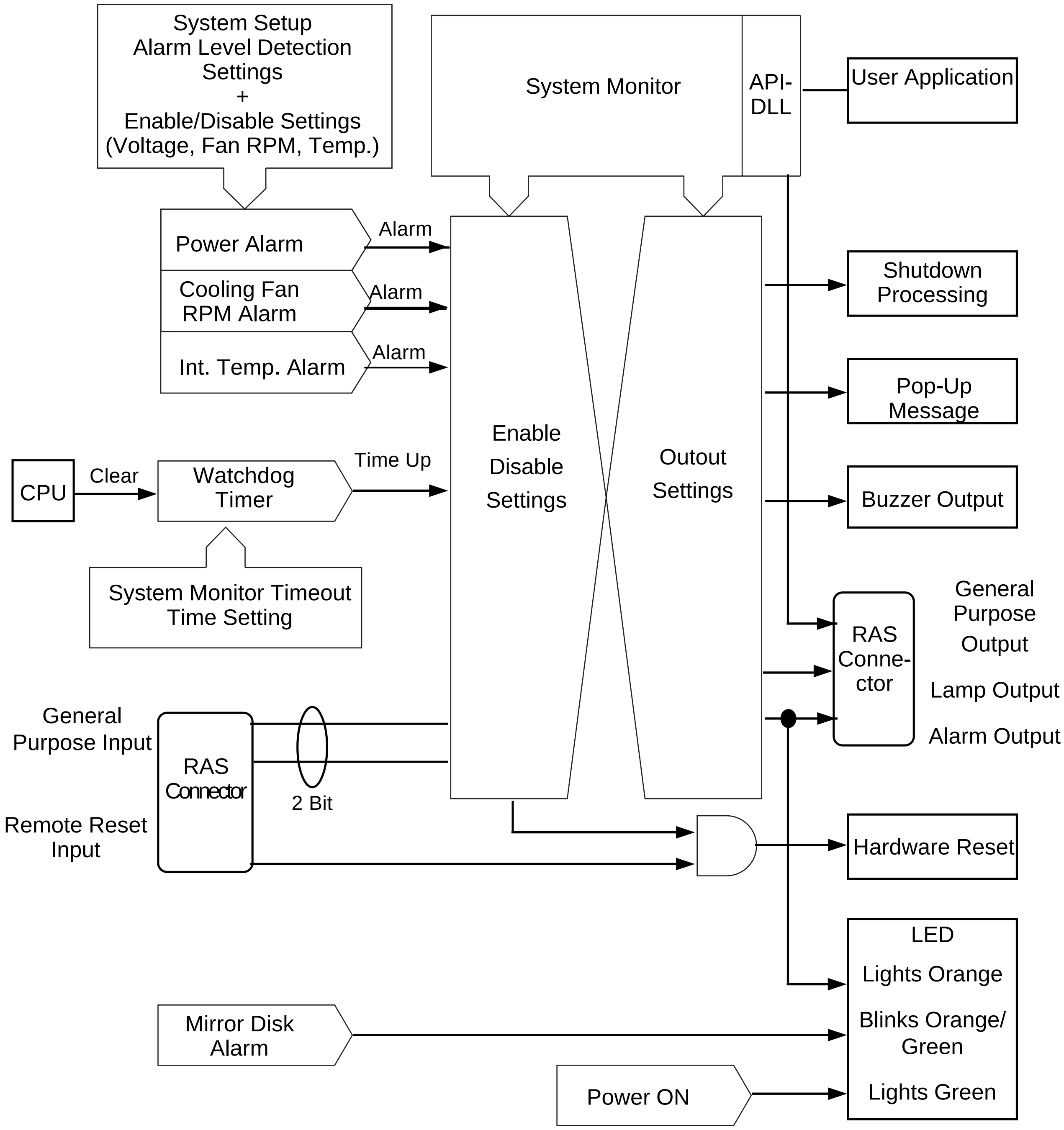
System Shutdown

This feature shuts down the PL's OS (Windows®95/ Windows®98 Second Edition/ Windows®NT 4.0). The System Monitor utility can be used to enable or disable this feature.

System Reset

Resets the PL unit's system after the Watchdog Timer reaches "Time Up".

A.2.3 RAS Feature Overview

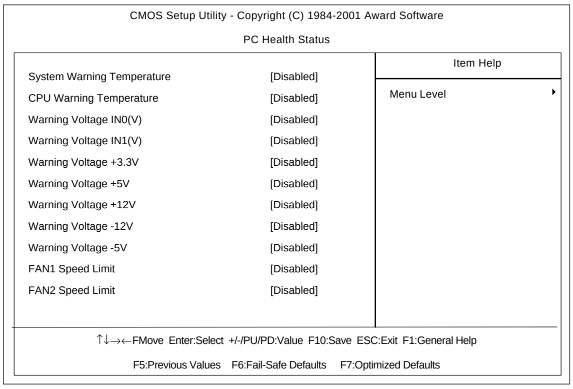


A.3 System Monitor

A.3.1 Setup Procedure

Follow the steps shown here to complete the System Monitor/RAS setup.

■ System Setup Screen Settings



CMOS Setup Utility - Copyright (C) 1984-2001 Award Software

PC Health Status

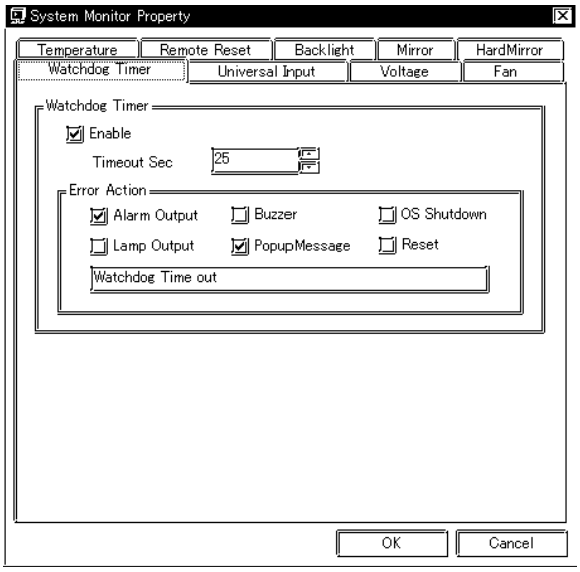
		Item Help
System Warning Temperature	[Disabled]	Menu Level ▸
CPU Warning Temperature	[Disabled]	
Warning Voltage IN0(V)	[Disabled]	
Warning Voltage IN1(V)	[Disabled]	
Warning Voltage +3.3V	[Disabled]	
Warning Voltage +5V	[Disabled]	
Warning Voltage +12V	[Disabled]	
Warning Voltage -12V	[Disabled]	
Warning Voltage -5V	[Disabled]	
FAN1 Speed Limit	[Disabled]	
FAN2 Speed Limit	[Disabled]	

↑↓→←FMove Enter:Select +/-/PU/PD:Value F10:Save ESC:Exit F1:General Help

F5:Previous Values F6:Fail-Safe Defaults F7:Optimized Defaults

In the System Setup area's PC Health Status menu, enable or disable each feature according to your system needs.

■ System Monitor Property Settings



Watchdog Timer

☑ Enable

Timeout Sec: 25

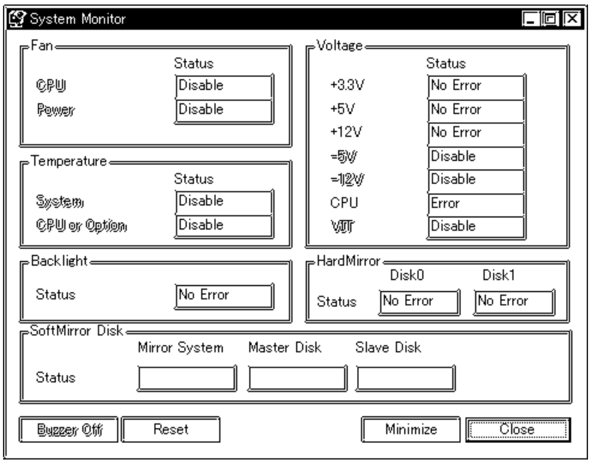
Error Action

☑ Alarm Output □ Buzzer □ OS Shutdown

□ Lamp Output ☑ PopupMessage □ Reset

Watchdog Time out

1) Start the PL unit's OS and click on the [Start] -> [Program] ->[System Monitor]->[System Monitor Property] screen. In this screen enter the System Monitor/RAS Event settings for each feature/tab. Click on the [OK] button and the program will automatically close.



Fan

	Status
CPU	Disable
Power	Disable

Temperature

	Status
System	Disable
CPU or Option	Disable

Voltage

	Status
+3.3V	No Error
+5V	No Error
+12V	No Error
-5V	Disable
-12V	Disable
CPU	Error
VTT	Disable

Backlight

Status: No Error

HardMirror

	Disk0	Disk1
Status	No Error	No Error

SoftMirror Disk

	Mirror System	Master Disk	Slave Disk
Status			

2) Restart the [System Monitor] utility, click on the [System Monitor] button and you can begin to monitor PL activity.

A.3.2 System Monitor Property Settings (PL_Wps.exe)

The following chart shows the features available when any of the monitoring value ranges set in the [PC Health Status] menu is exceeded.

O: Can be set   X: Cannot be set

	Alarm Output	Lamp Output	Buzzer	Popup Message	OS Shutdown	Reset
Watchdog Timer	O	O	O	O	O	O
Universal Input	O	O	O	O	O	X
Voltage	O	O	O	O	O	X
Fun	O	O	O	O	O	X
Temperature	O	O	O	O	O	X
Remote Reset [*1]	X	X	X	X	X	O
Backlight [*2]	O	O	O	O	X	X
Mirror	O	O	O	O	X	X
HardMirror	O	O	O	O	O	X

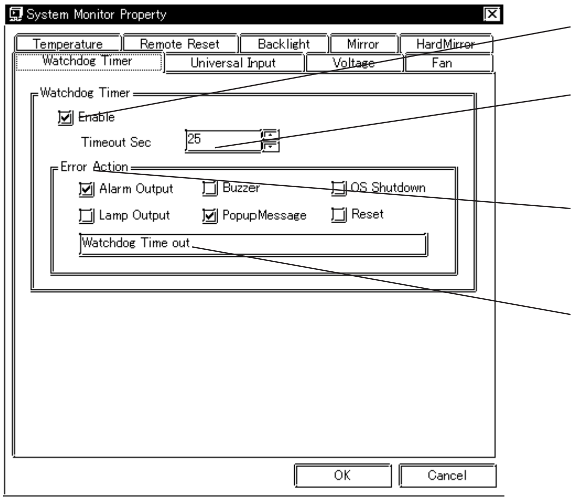
**1 When setting Enable on Remote Reset, the same action as the Reset occurs.*

**2 Cannot be used with a PL-B920 Series unit.*

Each of the above items performs the following operation.

Item	Operation
Alarm Output	RAS Interface Alarm Output (#9 to #10) signal is output.
Lamp Output	RAS Interface Alarm Output (#21 to #22) signal is output.
Buzzer	Buzzer sound is output as an alarm notification. (except for when the OS Shutdown feature is checked)
Popup Message	Error message appears as a Pop-Up Message Screen (on the PL unit's screen)
OS Shutdown	Shuts down the PL unit's OS. This can be set to either display a shutdown confirmation message, or perform a forced OS shutdown.Default is display a shutdown confirmation message.
Reset	Resets the PL unit by performing a forced shutdown.
Enable	Allows setting of monitoring items.

The System Monitor Property Screen details is as shown below.



Watchdog Timer

☑ Enable

Timeout Sec 25

Error Action

☑ Alarm Output ☐ Buzzer ☐ OS Shutdown

☐ Lamp Output ☑ PopupMessage ☐ Reset

Watchdog Time out

Set Enable/Disable of the function

Set the timer

Set the Error action when over the Enabled range

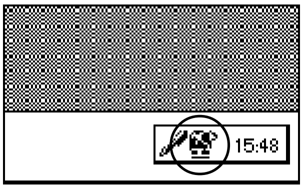
Input the message for the Pop-up Message Screen

A.3.3 System Monitor Operation (PL_Smon.exe)

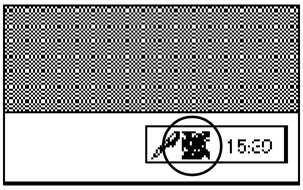
As soon as the PL unit's OS starts up, instead of showing the System Monitor Dialog box, the Alarm Monitoring icon appears in the System Tray. (see below)

Usually, when a System Monitor dialog box appears, the user checks the current condition and then left-clicks the mouse on the system tray icon to call up the System Monitor screen.

When an alarm is detected, the actions set in the System Monitor Property screen are performed, and an "X" appears over the System Tray icon. When this occurs, double-click on the icon to view the alarm contents.

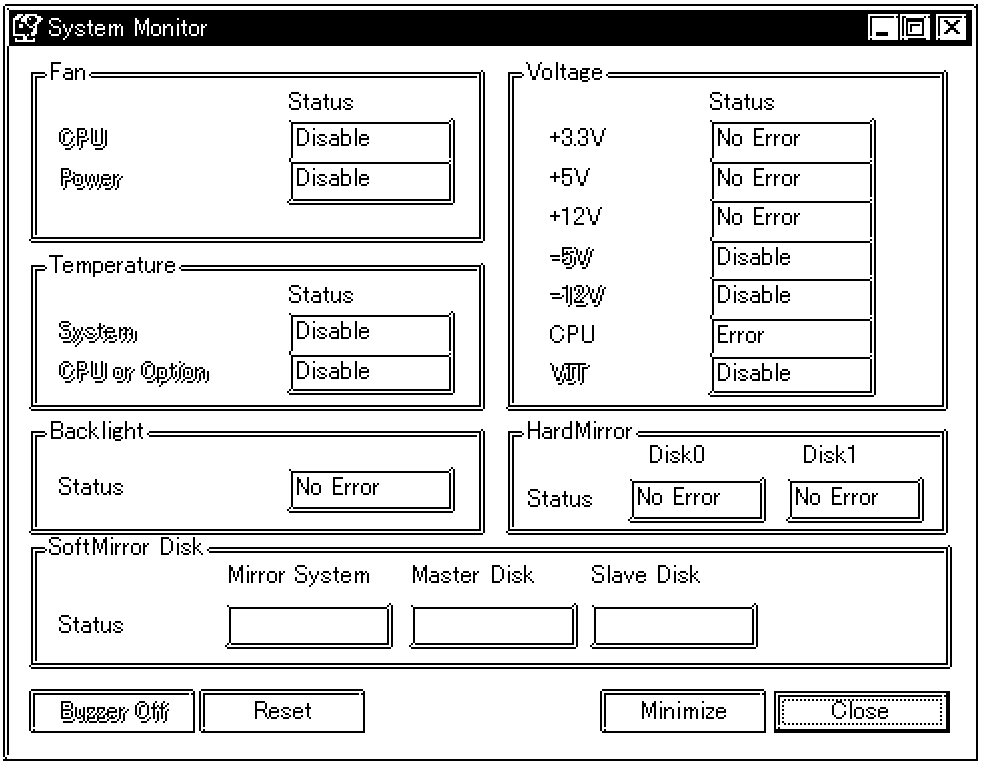


Start-up Icon



Alarm condition Icon

The System Monitor screen is as shown below.



System Monitor

Fan

	Status
CPU	Disable
Power	Disable

Temperature

	Status
System	Disable
CPU or Option	Disable

Voltage

	Status
+3.3V	No Error
+5V	No Error
+12V	No Error
-5V	Disable
-12V	Disable
CPU	Error
VTT	Disable

Backlight

Status: No Error

HardMirror

	Disk0	Disk1
Status	No Error	No Error

SoftMirror Disk

	Mirror System	Master Disk	Slave Disk
Status			

System Monitor Screen

In addition to the "Buzzer Off", "Reset", "Minimize", and "Close" buttons, the System Monitor screen contains the following features.

Button Name	Meaning
Buzzer Off	Stops the Alarm buzzer.
Reset	Resets alarm operation or System Monitor's internal alarm hold condition.
Minimize	Minimizes the System Monitor icon.
Close	Quits the System Monitor software.

Within the System Monitor screen is are the three words "No Error", "Error", and "Disable". They show the current status of each of the monitoring items, such as the Fan, Voltage, Temperature, Backlight, Software Mirroring Disk, and Hardware Mirroring Disk.

Status Name	Meaning
No Error	Normal operation
Error	Alarm
Disable	Not monitored

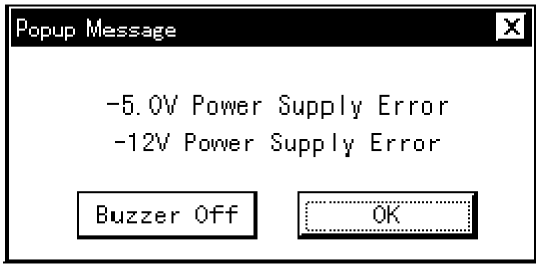
Whenever an alarm occurs for one of the monitored items, or when input (Universal Input) is detected from an external source, The operation designated in the System Monitor Property screen (Error Action) is performed.

Reference *3.2 System Monitor Property Settings*

Each error action, once an error or input is detected, is performed only once.

If +3.3V and +5.0V are monitored and a pop-up message is designated for the error action, when the +3.3V alarm is detected, a pop-up message will appear. Click on [OK] and the box is closed. Then, when a +5.0V alarm occurs, the +5.0V pop-up message will appear.

The pop-up message provides information about the type of error and the error contents. When this message box's [Buzzer Off] button is clicked, the buzzer sound will stop. Clicking on [OK] will close the box.



-5.0V Power Supply Error
-12V Power Supply Error

Pop-up Message Box

Once an alarm has occurred, the System Monitor will continue to remain in the "Alarm" state. (i.e. the alarm detected "X" will continue to be displayed on the System Tray icon. To release this condition, click on the System Monitor dialog box's [Rset] button. Or, turn the PL unit's power OFF, find and solve the problem, and turn the PL unit's power ON again.

A.3.4 Error Messages

The following error messages occur via the settings in the System Monitor, and the System Monitor Property dialog boxes.

■ System Monitor

◆ Alarm Pop-up Messages

When the Pop-up messages related to the Error Action are enabled, the following error messages will appear.

Error Type	Message
CPU Voltage	"CPU Power Supply Error"
+3.3V	"+3.3V Power Supply Error"
+5.0V	"+5V Power Supply Error"
+12V	"+12V Power Supply Error"
-12V	"-12V Power Supply Error"
-5V	"-5V Power Supply Error"
CPU Voltage 2	"VIT Power Supply Error"
Power FAN	"Power FAN Error"
CPU FAN	"CPU or OPTION FAN Error"
System Temperature	"System Temperature Error"
CPU or Option Temperature	"CPU Temperature Error"
Universal Input 0	"Universal Input 0"
Universal Input 1	"Universal Input 1"
Watchdog	"Watch Dog Timer Error"
Software Mirroring	"A Mirror disk error occurred"
Hardware Mirroring	"A Mirror disk error occurred"
Backlight	"Back Light Blowout Error"

Driver Error

"The system monitor driver not found."

"Install the latest driver."

Driver Version Error

"The old system monitor driver version."

"Update the driver."

Overlapping Startup message

"System monitor has started. "

"Terminate the system monitor in starting."

Shutdown Confirmation

"The system monitor is terminated."

"Are you sure?"

■ System Monitor Property Screen

◆ Overlapping startup message

"System Monitor Property has started."

"Terminate the system monitor property in starting."

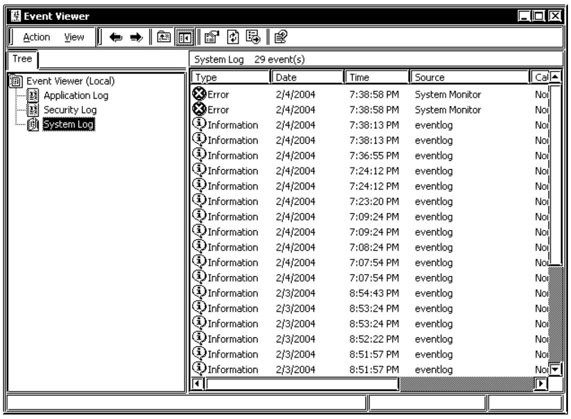
◆ Shutdown Confirmation message

"Save Changes to the registry?"

A.3.5 Error Displays When Using Event Viewer

Error type/location and error actions are recorded as error events in the System Log. Error event information can be checked using Event Viewer.

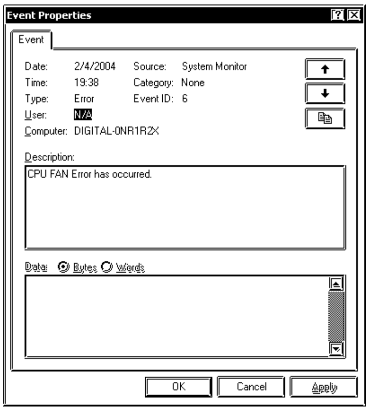
■ Error Message Display



System Log 29 event(s)

Type	Date	Time	Source	Cal
Error	2/4/2004	7:38:58 PM	System Monitor	Noi
Error	2/4/2004	7:38:58 PM	System Monitor	Noi
Information	2/4/2004	7:38:13 PM	eventlog	Noi
Information	2/4/2004	7:38:13 PM	eventlog	Noi
Information	2/4/2004	7:36:55 PM	eventlog	Noi
Information	2/4/2004	7:24:12 PM	eventlog	Noi
Information	2/4/2004	7:24:12 PM	eventlog	Noi
Information	2/4/2004	7:23:20 PM	eventlog	Noi
Information	2/4/2004	7:09:24 PM	eventlog	Noi
Information	2/4/2004	7:09:24 PM	eventlog	Noi
Information	2/4/2004	7:08:24 PM	eventlog	Noi
Information	2/4/2004	7:07:54 PM	eventlog	Noi
Information	2/4/2004	7:07:54 PM	eventlog	Noi
Information	2/3/2004	8:54:43 PM	eventlog	Noi
Information	2/3/2004	8:53:24 PM	eventlog	Noi
Information	2/3/2004	8:53:24 PM	eventlog	Noi
Information	2/3/2004	8:52:22 PM	eventlog	Noi
Information	2/3/2004	8:51:57 PM	eventlog	Noi
Information	2/3/2004	8:51:57 PM	eventlog	Noi

1) Start the [Control Panel] -> [Administrative Tools] ->[Event Viewer], and select [System Log].



Date: 2/4/2004 Source: System Monitor
Time: 19:38 Category: None
Type: Error Event ID: 6
User: N/A
Computer: DIGITAL-0NR1R2X

Description:
CPU FAN Error has occurred.

Data: ● Bytes ○ Words

2) Select the System Monitor's error and click the [Properties] icon. When the [Event Properties] dialog box appears, the error message will appear in the [Description] area.

◆ Error Type/Location

The error type/locations shown by the Event Viewer are as follows.

Error Type/Location	Error Message
+3.3V	+3.3V Error has occurred.
+5.0V	+5.0V Error has occurred.
+12V	+12V Error has occurred.
-12V	-12V Error has occurred.
-5.0V	-5.0V Error has occurred.
CPU valtage	CPU voltage Error has occurred.
Vit valtage	Vit valtage Error has occurred.
CPU FAN	CPU FAN Error has occurred.
Power FAN	Power FAN Error has occurred.
CPU Temp	CPU Temperature Error has occurred.
System Temp	System Temperature Error has occurred.
Universal Input 0	Universal Input 0 Error has occurred.
Universal Input 1	Universal Input 1 Error has occurred.
Watch Dog Timer	Watch Dog Timer Error has occurred.
Backlight	Backlight Error has occurred.
Soft Mirror	Mirror Disk Error has occurred.
Hard Mirror	Mirror Disk Error has occurred.

◆ Error Action

Using the Event Viewer the following actions can be taken when an error occurs.

Note:

- **The data shown in the table's "□" indicate the error type/location.**
- **The actions to take after an error occurs are set via the System Monitor Property screen.**

Error Action	Error Message
Buzzer	Buzzer has sounded because of □□ error.
Popup Message	Popup message has been shown because of □□ error.
OS Shutdown	Window has been shut down because of □□ error.
ALARM	ALARM has output because of □□ error.
LAMP	LAMP has output because of □□ error.

When a "+3.3V" error occurs and the buzzer sounds, two errors will be displayed, "+3.3V Error has occurred." and "Buzzer has sounded because of +3.3V error".

A.4 System Monitor/RAS Feature API-DLL

A.4.1 Operation Environment

The following information explains the Dynamic Link Libraries used by the System Monitor/RAS feature on a PL-X920 Series unit.

API-DLLs provide the interface for applications to access the System Monitor/ RAS feature (System Monitor/RAS Device Driver). Applications can use DLLs to access the following types of features.

1. Driver Version information
2. System Monitor feature status
3. Read out (Get) various monitoring parameters (voltage, fan, temperature)
4. System Monitor current data (voltage, fan, temperature)
5. Watchdog parameters
6. Alarm processing
7. General input processing
8. Reset (of PL unit)
9. Software data mirroring
10. Event handling

■ Compatible Operating Systems

The API-DLLs contained on the PL unit's CD-ROM are compatible with the following OS types.

- Mircrosoft Windows®95
- Microsoft Windows®98
- Microsoft WindowsNT®4.0
- Microsoft Windows®2000

Each OS must use its corresponding System Monitor/RAS Device Driver.

■ Compatible Languages

- Microsoft Visual C
- Microsoft Visual C++
- Microsoft Visual Basic

◆ Required Files

The following files are required when using DLLs. Each language requires its own set of files.

- Visual C

File Name	Description
PL_Iocif.h	Driver interface definition "include" file
PL_Ioc.LIB	Library definition file
PL_Ioc.dll	Dynamic link library file

- Visual C++

File Name	Description
PL_Iocif.h	Driver interface definition "include" file
PL_Iocall.h	CPL_Iocall class definition "include" file
PL_Ioctl.h	CPL_Ioctl class definition "include" file
PL_Ioc.LIB	Library definition file
PL_Ioc.dll	Dynamic Link library file
PL_SmiIoctl.h	CPL_SmiIocrlclass definition "include" file (used only with software mirroring feature)

* "#include header files should be "included" in the following order.

#include PL_Iocif.h

#include PL_Ioctl.h

PL_Iocall.h is automatically included, and does not need to be directly designated.

- Visual Basic

File Name	Description
PL_Ioc.bas	Driver interface definition file
PL_Ioc.LIB	Library definition file
PL_Ioc.dll	Dynamic link library file

■ Dynamic Link Library (DLL)

In order for an application to use PL_Ioc.dll, it should be copied to the following folder.

OS	Location
Windows95/Windows98	C:\Windows\System
WindowsNT4.0/Windows2000	C:\Winnt\System32

A.4.2 Class Contents

■ CPL_Ioctl Class

This class is used to set the parameters for device driver access using CPL_Ioctl class.

Key Word	Type	Variable Name	Description
public	HANDLE	m_Drvhandle	Device driver handle

■ CPL_Iocall Class

This uses the parameters set in CPL_Ioctl, and calls up DeviceIoControl (Driver Access function).

However, since this class succeeds CPL_Ioctl, it cannot be used directly.

Key Word	Type	Variable Name	Description
public	HANDLE	m_h	Device driver handle
public	LONG	m_long	Control code for action to perform
public	void *	m_ibp	Input data buffer address
public	ULONG	m_ibsize	Input data buffer size
public	void *	m_obp	Output data buffer address
public	ULONG	m_obsize	Output data buffer size
public	DWORD	m_retsize	Address for actual no. of output bytes
public	LPOVERLAPPED	m_ovlp	Address of overlap design

■ CPL_SmiIoctl Class

This class is used to set the parameters for device driver access using CPL_SmiIoctl class.

This class is only used when using the Software Mirroring driver.

Key Word	Type	Variable Name	Description
public	HANDLE	m_Drvhandle	Device driver handle

A.4.3 Visual C Functions

Function Name	Description
InitIoctl	Creates the CPL_Ioctl object
EndIoctl	Destroys the CPL_Ioctl object
GetDrvHandle	Gets the driver handle
CloseDrvHandle	Destroys the driver handle
GetDrvVersion	Gets the driver version
GetMonitorSetup	Gets the monitoring enabled/disabled setting
GetVoltParam	Gets the voltage monitoring parameter
GetCurrentVolt	Gets the current voltage value
GetFanParam	Gets the fan monitoring parameter
GetCurrentFan	Gets the current fan value
GetTempParam	Gets the temperature monitoring parameter
GetCurrentTemp	Gets the current temperature value
SetWdtCounter	Sets the watchdog timer counter
GetWdtCounter	Gets the watchdog timer counter
SetWdtMask	Sets warning masking in case of watchdog timer timeout
GetWdtMask	Gets warning masking in case of watchdog timer timeout
StartWdt	Starts the watchdog timer
StopWdt	Stops the watchdog timer
RestartWdt	Restarts the watchdog timer
RunningWdt	Gets the watchdog timer operation status
SetWarningOut	Sets warning output
GetWarningOut	Gets warning output
GetUniversalIn	Gets universal input
ClearUniversalIn	Clears the universal input latched status
SetUniversalInMask	Sets universal input masking
GetUniversalInMask	Gets universal input masking
SetResetMask	Sets reset-masking
GetResetMask	Gets reset-masking
SetIdeErr	Sets data mirroring (software) error
GetIdeErrHard	Gets data mirroring (hardware) error
GetEvent	Gets the error event
ClearEvent	Clears the error event
StartInsideBuzzer	Starts PL internal buzzer
StopInsideBuzzer	Stops PL internal buzzer
ChkInsideBuzzer	Checks PL internal buzzer status
GetWdtTimeout	Gets the timeout status of the watchdog timer
ClearWdtTimeout	Clears the timeout status of the watchdog timer
SetWarningDOUT	Sets the warning output DOUT
GetWarningDOUT	Gets the warning output DOUT
GetSmiDrvHandle	Gets Software Mirroring driver handle
CloseSmiDrvHandle	Destroys Software Mirroring driver handle
GetSmiAryStatus	Gets status of Software Mirroring Array
GetSmiDevStatus	Gets status of Software Mirroring Device
SetWdtResetMask	Sets the Reset Mask of the watchdog timer
GetWdtResetMask	Gets the Reset Mask of the watchdog timer

A.4.4 Visual C Function Specifications (Details)

InitIoctl

Call Format	void WINAPI InitIoctl(void)
Return Value	None
Arguments	None
Processing	Creates a CPL_Ioctl object. The object is not destroyed until the EndIoctl function is called.
Example	InitIoctl();

EndIoctl

Call Format	void WINAPI EndIoctl(void)
Return Value	None
Arguments	None
Processing	Destroys the object created using the InitIoctl function.
Example	EndIoctl();

GetDrvHandle

Call Format	int WINAPI GetDrvHandle(HANDLE * pHndl)
Return Value	0: Normal 1: Error
Arguments	(I/O) HANDLE *pHndl Pointer to the device driver handle
Processing	Gets the device driver handle to communicate with the device driver.

Example	int ret;
	HANDLE hndl;
	ret = GetDrvHandle(&hndl);

Note: **An error occurs if the System Monitor/RAS Device Driver is not running.**

CloseDrvHandle

Call Format	BOOL WINAPI CloseDrvHandle(void)
Return Value	TRUE: Normal FALSE: Error
Arguments	None
Processing	Destroys the device driver handle created using the GetDrvHandle function.

Example	BOOL ret;
	//Destroys the handle
	ret = CloseDrvHandle();

GetDrvVersion

Call Format BOOL WINAPI GetDrvVersion(int *pMajor, int *pMinor)

Return Value TRUE: Normal
FALSE: Error

Arguments (I/O) int *pMajor Pointer to version information (Major, 0 to 99).
(I/O) int *pMinor Pointer to version information (Minor, 0 to 99).

Processing Gets the driver's version information.

Example

```
BOOL     ret;
int      Major, Minor;
ret = GetDrvVersion( &Major, &Minor );
```

Note: **If the version is 1.10, then you will get**
Major: 1 (decimal)
Minor: 10 (decimal).

GetMonitorSetup

Call Format BOOL WINAPI GetMonitorSetup(int Selector, int *pSetup)

Return Value TRUE: Normal
FALSE: Error

Arguments (I) int Selector Parameters

MONITOR_VOLT_CPU	CPU core voltage
MONITOR_VOLT_P33	+3.3 V
MONITOR_VOLT_P50	+5.0 V
MONITOR_VOLT_P12	+12 V
MONITOR_VOLT_M12	-12 V
MONITOR_VOLT_M50	-5.0 V
MONITOR_TEMP_SYSTEM	System temperature
MONITOR_TEMP_CPU	CPU temperature
MONITOR_TEMP_OPT	Option temperature
MONITOR_FAN_CPU	CPU fan
MONITOR_FAN_POWER	Power fan
MONITOR_FAN_OPT	Option fan
MONITOR_VOLT_VIT	CPU core voltage 2

(I/O) int *pSetup Pointer to Data
0: Disabled
1: Enabled

Processing Gets the current monitoring status (enabled/disabled).

Example

```
BOOL     ret;
int      Setup;
// Gets the CPU core voltage setup status.
ret = GetMonitorSetup( MONITOR_VOLT_CPU, &Setup );
```

GetVoltParam

Call Format BOOL WINAPI GetVoltParam
(int Selector, int *pULimit, int *pLLimit)

Return Value TRUE: Normal
FALSE: Error

Arguments (I) int Selector Parameters

MONITOR_VOLT_CPU	CPU core voltage
MONITOR_VOLT_P33	+3.3 V
MONITOR_VOLT_P50	+5.0 V
MONITOR_VOLT_P12	+12 V
MONITOR_VOLT_M12	-12 V
MONITOR_VOLT_M50	-5.0 V
MONITOR_VOLT_VIT	CPU core voltage 2

(I/O) int *pULimit Pointer to upper-limit voltage value (Unit: mV)
(I/O) int *pLLimit Pointer to lower-limit voltage value (Unit: mV)

Processing Gets the voltage monitoring parameter.

Example

```
BOOL    ret;
int     ULimit, LLimit;
// Get the upper and lower-limit values of the CPU core
// voltage.
ret = GetVoltParam( MONITOR_VOLT_CPU, &ULimit, &LLimit );
```

Note: **Since the data taken from this function is shown in mV units, the following conversion is needed for use in (Volt) units:**

Data in Volt unit = Data in mV unit/1000

GetCurrentVolt

Call Format BOOL WINAPI GetCurrentVolt(int Selector, int *pData)

Return Value TRUE: Normal

FALSE: Error

Arguments (I) int Selector Parameters

MONITOR_VOLT_CPU	CPU core voltage
MONITOR_VOLT_P33	+3.3 V
MONITOR_VOLT_P50	+5.0 V
MONITOR_VOLT_P12	+12 V
MONITOR_VOLT_M12	-12 V
MONITOR_VOLT_M50	-5.0 V
MONITOR_VOLT_VIT	CPU core voltage 2

(I/O) int *pData Pointer to the voltage value (Unit: mV)

Processing Gets the current voltage value.

Example

```
BOOL     ret;
int      Data;
// Gets the CPU core voltage value.
ret = GetCurrentVolt( MONITOR_VOLT_CPU, &Data );
```

Note: Since the data taken from this function is in mV units, the following conversion is needed for use in (Volt) units:

Data in Volt unit = Data in mV unit/1000

GetFanParam

Call Format BOOL WINAPI GetFanParam (int Selector, int *pLLimit)

Return Value TRUE: Normal

FALSE: Error

Arguments (I) int Selector Parameters

MONITOR_FAN_CPU	CPU fan
MONITOR_FAN_POWER	Power fan
MONITOR_FAN_OPT	Option fan

(I/O) int *pLLimit Pointer to the lower-limit fan rotation speed (Unit: RPM)

(RPM: Revolutions Per Minute)

Processing Gets the fan monitoring parameter.

Example

```
BOOL     ret;
int      LLimit;
// Gets the lower-limit CPU fan rotation speed.
ret = GetFanParam( MONITOR_FAN_CPU, &LLimit );
```

GetCurrentFan

Call Format: BOOL WINAPI GetCurrentFan(int Selector, int *pData)

Return Value: TRUE: Normal
FALSE: Error

Arguments: (I) int Selector Parameters

MONITOR_FAN_CPU	CPU fan
MONITOR_FAN_POWER	Power fan
MONITOR_FAN_OPT	Option fan

(I/O) int *pData Pointer to the fan rotation speed
(Unit: RPM)
(RPM: Revolutions Per Minute)

Processing: Gets the current fan rotational speed.

Example:

```
BOOL    ret;
int     Data;
// Gets the CPU fan rotational speed.
ret = GetCurrentFan( MONITOR_FAN_CPU, &Data );
```

GetTempParam

Call Format: BOOL WINAPI GetTempParam(int Selector, int *pULimit)

Return Value: TRUE: Normal
FALSE: Error

Arguments: (I) int Selector Parameters

MONITOR_TEMP_SYSTEM	System temperature
MONITOR_TEMP_CPU	CPU temperature
MONITOR_TEMP_OPT	Option temperature

(I/O) int *pULimit Pointer to the upper-limit temperature
(Unit: Degrees Celsius)

Processing: Gets the temperature monitoring parameter.

Example:

```
BOOL    ret;
int     ULimit;
// Gets the system temperature upper-limit value.
ret = GetTempParam( MONITOR_TEMP_SYSTEM, &ULimit );
```

GetCurrentTemp

Call Format BOOL WINAPI GetCurrentTemp(int Selector, int *pData)

Return Value TRUE: Normal
FALSE: Error

Arguments (I) int Selector Parameters

MONITOR_TEMP_SYSTEM	System temperature
MONITOR_TEMP_CPU	CPU temperature
MONITOR_TEMP_OPT	Option temperature

(I/O) int *pData Pointer to the temperature
(Unit: Degrees Celsius)

Processing Gets the current temperature value.

Example

```
BOOL        ret;
int         Data;
// Gets the system temperature value.
ret = GetCurrentTemp( MONITOR_TEMP_SYSTEM, &Data );
```

SetWdtCounter

Call Format BOOL WINAPI SetWdtCounter(int Counter)

Return Value TRUE: Normal
FALSE: Error

Arguments (I) int Counter Sets to the watchdog timer's initial counter value (5 to 255) (Unit: Seconds)

Processing Sets the current watchdog timer's initial counter value.

Example

```
BOOL      ret;
// Sets the watchdog timer's initial counter value to 10 sec.
ret = SetWdtCounter( 10 );
```

GetWdtCounter

Call Format BOOL WINAPI GetWdtCounter(int *pCounter)

Return Value TRUE: Normal
FALSE: Error

Arguments (I/O) int *pCounter Pointer to the watchdog timer's initial counter value (Unit: Seconds)

Processing Gets the current watchdog timer's initial counter value.

Example

```
BOOL        ret;
int         Counter;
 ret = GetWdtCounter( &Counter );
```

SetWdtMask

Call Format	BOOL WINAPI SetWdtMask(int Selector, int Mask)
Return Value	TRUE: Normal FALSE: Error
Arguments	(I) int Selector Setting Item WARNING_LAMP LAMP WARNING_ALARM ALARM (I) int Mask Masking Information MASK_OFF Masking disabled MASK_ON Masking enabled
Processing	Sets masking for the warning output used when watchdog timer time-out occurs.

Example	BOOL ret;
	// Enables masking for the lamp output.
	ret = SetWdtMask(WARNING_LAMP, MASK_ON);
	// Disables masking for the alarm output.
	ret = SetWdtMask(WARNING_ALARM, MASK_OFF);

GetWdtMask

Call Format	BOOL WINAPI GetWdtMask(int Selector, int *pMask)
Return Value	TRUE: Normal FALSE: Error
Arguments	(I) int Selector Setting Item WARNING_LAMP LAMP WARNING_ALARM ALARM (I/O) int *pMask Pointer to Masking Information MASK_OFF Masking disabled MASK_ON Masking enabled
Processing	Gets the masking information used for warning output when watchdog timer time-out occurs.

Example	BOOL ret;
	int Mask;
	// Gets the masking information for the LAMP.
	ret = GetWdtMask(WARNING_LAMP, &Mask);
	// Gets the masking information for the alarm.
	ret = GetWdtMask(WARNING_ALARM, &Mask);

StartWdt

Call Format	BOOL WINAPI StartWdt(void)
Return Value	TRUE: Normal FALSE: Error
Arguments	None
Processing	Starts watchdog timer countdown.

Example	BOOL ret;
	ret = StartWdt();

StopWdt

Call Format	BOOL WINAPI StopWdt(void)
Return Value	TRUE: Normal FALSE: Error
Arguments	None
Processing	Stops watchdog timer countdown.

Example	BOOL ret;
	ret = StopWdt();

RestartWdt

Call Format	BOOL WINAPI RestartWdt(void)
Return Value	TRUE: Normal FALSE: Error
Arguments	None
Processing	Restarts watchdog timer countdown after resetting to the initial value.

Example	BOOL ret;
	ret = RestartWdt();

Note: **Processing does not occur if the watchdog timer is stopped.**

RunningWdt

Call Format	BOOL WINAPI RunningWdt(int *pRunFlag)		
Return Value	TRUE: Normal FALSE: Error		
Arguments	(I/O) int *pRunFlag	Pointer to Watchdog Timer Operation Status	
		WATCHDOG_STOP	Stopped
		WATCHDOG_COUNTDOWN	Countdown in progress
Processing	Gets the watchdog timer's operation status.		

Example	BOOL ret;
	int RunFlag;
	ret = RunningWdt(&RunFlag);		

SetWarningOut

Call Format BOOL WINAPI SetWarningOut(int Selector, int WarnOut)

Return Value TRUE: Normal
FALSE: Error

Arguments (I) int Selector Setting Item

WARNING_LAMP	LAMP
WARNING_ALARM	ALARM

(I) int WarnOut Output Status

OUTPUT_OFF	Output OFF
OUTPUT_ON	Output ON

Processing Sets setting item warning information (LAMP or ALARM).

Example

```
BOOL     ret;
// Sets the LAMP output status to ON.
ret = SetWarningOut( WARNING_LAMP, OUTPUT_ON );
// Sets the ALARM output status to OFF.
ret = SetWarningOut( WARNING_ALARM, OUTPUT_OFF );
```

GetWarningOut

Call Format BOOL WINAPI GetWarningOut(int Selector, int *pWarnOut)

Return Value TRUE: Normal
FALSE: Error

Arguments (I) int Selector Setting Item

WARNING_LAMP	LAMP
WARNING_ALARM	ALARM

(I/O) int *pWarnOut Pointer to Output Status

OUTPUT_OFF	Output OFF
OUTPUT_ON	Output ON

Processing Gets currently set item's warning status (LAMP or ALARM).

Example

```
BOOL       ret;
 int       WarnOut;
// Gets the LAMP output status.
ret = GetWarningOut( WARNING_LAMP, &WarnOut );
// Gets the ALARM output status.
ret = GetWarningOut( WARNING_ALARM, &WarnOut );
```

GetUniversalIn

Call Format	BOOL WINAPI GetUniversalIn(int Selector, int *pUniIn)
Return Value	TRUE: Normal FALSE: Error
Arguments	(I) int Selector Designated Port PORT_UNI0 Universal Input 0 PORT_UNI1 Universal Input 1 (I/O) int *pUniIn Pointer to Input Status INPUT_OFF Input OFF INPUT_ON Input ON
Processing	Gets the input status of the designated port (Universal Input 0, Universal Input 1).

Example

```
BOOL      ret;
int       UniIn;
// Get the input status of Universal Input 0.
ret = GetUniversalIn( PORT_UNI0, &UniIn );
// Get the input status of Universal Input 1.
ret = GetUniversalIn( PORT_UNI1, &UniIn );
```

ClearUniversalIn

Call Format	BOOL WINAPI ClearUniversalIn(int Selector)
Return Value	TRUE: Normal FALSE: Error
Arguments	(I) int Selector Designated Port PORT_UNI0 Universal Input 0 PORT_UNI1 Universal Input 1
Processing	Cancels the input status of the designated port (Universal Input 0, Universal Input 1).

Example

```
BOOL ret;
// Cancels the input status of Universal Input 0.
ret = ClearUniversalIn( PORT_UNI0 );
// Cancels the input status of Universal Input 1.
ret = ClearUniversalIn( PORT_UNI1 );
```

SetUniversalInMask

Call Format   BOOL WINAPI SetUniversalInMask(int Selector, int Mask)

Return Value   TRUE: Normal
FALSE: Error

Arguments   (I) int Selector   Designated Port
PORT_UNI0   Universal Input 0
PORT_UNI1   Universal Input 1
(I) int Mask   Masking Information
MASK_OFF   Masking disabled
MASK_ON   Masking enabled

Processing   Sets the masking information for the designated port (Universal Input 0, Universal Input 1).

Example

```
BOOL      ret;
// Disable masking for Universal Input 0.
ret = SetUniversalInMask( PORT_UNI0, MASK_OFF );
// Enable masking for Universal Input 1.
ret = SetUniversalInMask( PORT_UNI1, MASK_ON );
```

GetUniversalInMask

Call Format   BOOL WINAPI GetUniversalInMask(int Selector, int *pMask)

Return Value   TRUE: Normal
FALSE: Error

Arguments   (I) int Selector   Designated Port
PORT_UNI0   Universal Input 0
PORT_UNI1   Universal Input 1
(I/O) int *pMask   Pointer to Masking Information
MASK_OFF   Masking disabled
MASK_ON   Masking enabled

Processing   Gets the masking information for the designatd port (Universal Input 0, Universal Input 1).

Example

```
BOOL        ret;
int         Mask;
// Gets the masking information for Universal input 0.
ret = GetUniversalInMask( PORT_UNI0, &Mask );
// Gets the masking information for Universal input 1.
ret = GetUniversalInMask( PORT_UNI1, &Mask );
```

SetResetMask

Call Format	BOOL WINAPI SetResetMask(int Mask)		
Return Value	TRUE: Normal		
	FALSE: Error		
Arguments	(I) int Mask	Masking Information	
		MASK_OFF	Masking disabled
		MASK_ON	Masking enabled
Processing	Sets reset-masking.		
Example	BOOL ret;		
	// Disable reset-masking.		
	ret = SetResetMask(MASK_OFF);		

GetResetMask

Call Format	BOOL WINAPI GetResetMask(int *pMask)		
Return Value	TRUE: Normal		
	FALSE: Error		
Arguments	(I/O) int *pMask	Pointer to Masking Information	
		MASK_OFF	Masking disabled
		MASK_ON	Masking enabled
Processing	Gets the current reset-masking information.		
Example	BOOL ret;		
	int Mask;		
	ret = GetResetMask(&Mask);		

SetIdeErr

Call Format	BOOL WINAPI SetIdeErr(int IdeErr)		
Return Value	TRUE: Normal		
	FALSE: Error		
Arguments	(I) int IdeErr	Error Output Status	
		IDE_ERROR_OFF	Error Output OFF
		IDE_ERROR_ON	Error Output ON
Processing	Sets the software control used to create IDE eror output.		
Example	BOOL ret;		
	// Sets IDE error output to OFF.		
	ret = SetIdeErr(IDE_ERROR_OFF);		

GetIdeErrHard

Call Format	BOOL WINAPI GetIdeErrHard(int Selector, int *pIdeErr)
Return Value	TRUE: Normal FALSE: Error
Arguments	(I) int Selector Parameters IDE_ERROR_1 IDE_ERR1 IDE_ERROR_2 IDE_ERR2 (I/O) int *pIdeErr Pointer to error signal IDE_ERROR_OFF Normal IDE_ERROR_ON Error
Processing	Gets the current IDE error signal output by the hardware.

Example

```
BOOL        ret;
int         IdeErr;
// Gets the IDE_ERR1 signal
ret = GetIdeErrHard( IDE_ERROR_1, &IdeErr );
```

GetEvent

Call Format BOOL WINAPI GetEvent(int Selector, int *pEvent)

Return Value TRUE: Normal
FALSE: Error

Arguments (I) int Selector Parameters

EVENT_VOLT_CPU	CPU core voltage
EVENT_VOLT_P33	+3.3 V
EVENT_VOLT_P50	+5.0 V
EVENT_VOLT_P12	+12 V
EVENT_VOLT_M12	-12 V
EVENT_VOLT_M50	-5.0 V
EVENT_VOLT_VIT	CPU core voltage 2
EVENT_FAN_CPU	CPU FAN
EVENT_FAN_POWER	POWER FAN
EVENT_FAN_OPT	OPTION FAN
EVENT_TEMP_SYSTEM	SYSTEM temperature
EVENT_TEMP_CPU_OPT	CPU or option temperature
EVENT_UNI_IN0	Universal Input 0
EVENT_UNI_IN1	Universal Input 1
EVENT_WDT_TIMEOUT	Watchdog Timeout

(I/O) int *pEvent Pointer to Error Event Information

ERROR_EVENT_OFF	Without error event
ERROR_EVENT_ON	With error event

Processing Checks the machine for voltage, fan, and temperature errors, and the Universal Input information (event) and Watchdog Timeout information.

Example

```
BOOL      ret;
int       Event;
// Gets the error event information for the CPU core voltage.
ret = GetEvent( EVENT_VOLT_CPU, &Event );
```

ClearEvent

Call Format BOOL WINAPI ClearEvent(int Selector)

Return Value TRUE: Normal
FALSE: Error

Arguments (I) int Selector Parameters used for cancelling error events

EVENT_VOLT_CPU	CPU core voltage
EVENT_VOLT_P33	+3.3 V
EVENT_VOLT_P50	+5.0 V
EVENT_VOLT_P12	+12 V
EVENT_VOLT_M12	-12 V
EVENT_VOLT_M50	-5.0 V
EVENT_VOLT_VIT	CPU core voltage 2
EVENT_FAN_CPU	CPU FAN
EVENT_FAN_POWER	POWER FAN
EVENT_FAN_OPT	OPTION FAN
EVENT_TEMP_SYSTEM	SYSTEM temperature
EVENT_TEMP_CPU_OPT	CPU or option temperature
EVENT_UNI_IN0	Universal input 0
EVENT_UNI_IN1	Universal input 1
EVENT_WDT_TIMEOUT	Watchdog Timeout

Processing Cancels the error event.

Example

```
BOOL    ret;
// Cancels the CPU core voltage error event.
ret = ClearEvent( EVENT_VOLT_CPU );
```

StartInsideBuzzer

Call Format BOOL WINAPI StartInsideBuzzer (int hz, int ms)

Return Value BOOL TRUE: Normal
FALSE: Error

Arguments (I) int hz Buzzer frequency (Hz)
(I) int ms Buzzer length (ms)

Processing Starts the PL unit's internal buzzer, based on the designated frequency and length.

Example

```
BOOL    ret;
int     hz = 600;
int     ms = 1000;
// PL internal buzzer will sound at 600MHz for 1 second.
ret = StartInsideBuzzer ( hz, ms );
```

Note: **This feature cannot be used with a PL running WindowsNT 4.0 or Windows 2000, due to the use of Windows 95 and Windows 98 functions.**

StopInsideBuzzer

Call Format	BOOL WINAPI StopInsideBuzzer (void)
Return Value	BOOL TRUE: Normal FALSE: Error
Arguments	None.
Processing	Stops the PL unit's internal buzzer.

Example	BOOL ret;
	// Stops PL internal buzzer.
	ret = StopInsideBuzzer ( );

Note: **This feature cannot be used with a PL running Windows NT or Windows 2000, due to the use of Windows 95 and Windows 98 functions.**

ChkInsideBuzzer

Call Format	BOOL WINAPI ChkInsideBuzzer (int *BuzzerParam)
Return Value	BOOL TRUE: Normal FALSE: Error
Arguments	(I/O) int *BuzzerParam Pointer to Buzzer Condition BUZZER_ON Buzzer is ON BUZZER_OFF Buzzer is OFF
Processing	Checks the buzzer's operation status.

Example	BOOL ret;
	int BuzzerParam;
	// Checks buzzer status.
	ret = ChkInsideBuzzer (&BuzzerParam);

Note: **This feature cannot be used with a PL running WindowsNT 4.0 or Windows 2000, due to the use of Windows 95 and Windows 98 functions.**

GetWdtTimeout

Call Format	BOOL WINAPI GetWdtTimeout(int *pTimebuf)		
Return Value	TRUE: Normal		
	FALSE: Error		
Arguments	(I/O) int *pTimebuf	Pointer to Watchdog Timeout Status	
		TIMEOUT_OK	Not timeout
		TIMEOUT_ERROR	Timeout
Processing	Gets watchdog timeout status.		

Example

```
BOOL    ret;
int     Timebuf;
// Gets watchdog timeout status.
ret = GetWdtTimeout( &Timebuf );
```

ClearWdtTimeout

Call Format BOOL WINAPI ClearWdtTimeout(void)

Return Value TRUE: Normal
FALSE: Error

Arguments None

Processing Clears the watchdog timeout status.

Example

```
BOOL    ret;
// Clears the watchdog timeout status.
ret = GetWdtTimeout();
```

SetWarningDOUT

Call Format	BOOL WINAPI SetWarningDOUT(int WarningOut)		
Return Value	TRUE: Normal		
	FALSE: Error		
Arguments	(I) int WarningOut	Output status	
		OUTPUT_OFF	Output OFF
		OUTPUT_ON	Output ON
Processing	Sets DOUT warning status of current setting item.		

Example

```
BOOL    ret;
// Sets warning DOUT output status to OFF.
ret = SetWarningDOUT( OUTPUT_OFF );
```

GetWarningDOUT

Call Format	BOOL WINAPI GetWarningDOUT(int *pWarningOut)
Return Value	TRUE: Normal FALSE: Error
Arguments	(I/O) int *pWarningOut Pointer to Output Status OUTPUT_OFF Output OFF OUTPUT_ON Output ON
Processing	Gets DOUT warning status of current setting item.

Example	BOOL ret;
	int WarningOut;
	// Gets DOUT Output status.
	ret = GetWarningDOUT(&WarningOut);

GetSmiDrvHandle

Call Format	int WINAPI GetSmiDrvHandle(void)
Return Value	0: Normal 1: Error
Arguments	None
Processing	Gets Software Mirroring Device Driver Handle.

Example	int ret;
	ret = GetSmiDrvHandle();

Note: **When the Software Mirroring Device Driver is not operating, an error occurs.**

CloseSmiDrvHandle

Call Format	BOOL WINAPI CloseSmiDrvHandle(void)
Return Value	TRUE: Normal FALSE: Error
Arguments	None
Processing	Destroys the device driver handle created using the GetSmiDrvHandle function.

Example	BOOL ret;
	// Destroys the device driver handle created using the GetSmiDrvHandle function.
	ret = ClosetSmiDrvHandle();

GetSmiAryStatus

Call Format BOOL WINAPI GetSmiAryStatus(int *pStatus)

Return Value TRUE: Normal
FALSE: Error

Arguments (I/O) int *pStatus Pointor to Software Mirroring Disk Status

ARYSTAT_GOOD	Good
ARYSTAT_NOTEXIST	No output
ARYSTAT_UNCONFIG	Unconfigured
ARYSTAT_REBUILD	Rebuilding
ARYSTAT_REDUCE	Reduced
ARYSTAT_DEAD	Dead

Processing Gets SoftMirror Status

Example

```
BOOL        ret;
int         Status;
// Gets Software Mirroring Status.
ret = GetSmiAryStatus(&Status);
```

GetSmiDevStatus

Call Format BOOL WINAPI GetSmiDevStatus(int Id ,int *pType ,int *pStatus)

Return Value TRUE: Normal
FALSE: Error

Arguments (I) int Id Device ID
0 : Master HDD
1 : Slave HDD

(I/O) int* pType Device Type

ATADEVICE	ATA DEVICE
ATAPIDEVICE	CD-ROM
UNKNOWNDEVICE	Unknown DEVICE
NODEVICE	No DEVICE

(I/O) int* pStatus Device Status

DEVSTAT_GOOD	Good
DEVSTAT_NOTEXIST	No DEVICE
DEVSTAT_BROKEN	BROKEN

Processing Gets Software Mirroring Device Status

Example

```
BOOL    ret;
int    Id, Type, Status;
// Gets the device status
Id = 0;
ret = GetSmiDevStatus(Id ,&Type ,&Status);
```

SetWdtResetMask

Call Format BOOL WINAPI SetWdtResetMask(int Mask)

Return Value TRUE: Normal
FALSE: Error

Arguments (I/O) int Mask Masking Information
MASK_OFF Masking disabled
MASK_ON Masking enabled

Processing Sets the H/W reset mask used when WDT timeout occurs.

Example
```
BOOL    ret;
// Destroys the mask used for reset at WDT timeout.
ret = SetWdtResetMask( MASK_OFF );
```

GetWdtResetMask

Call Format BOOL WINAPI GetWdtResetMask(int *pMask)

Return Value TRUE: Normal
FALSE: Error

Arguments (I/O) int *pMask Pointer to Masking Information
MASK_OFF Masking disabled
MASK_ON Masking enabled

Processing Gets the H/W reset mask data used at WDT timeout

Example
```
BOOL      ret;
int       Mask;
// Gets the reset mask data used at WDT timeout
ret = GetWdtResetMask( &Mask );
```

A.4.5 Visual C++ Functions

Function Name	Description
GetDrvHandle	Gets the driver handle
CloseDrvHandle	Destroys the driver handle
GetDrvVersion	Gets the driver version
GetMonitorSetup	Gets the monitoring enabled/disabled setting
GetVoltParam	Gets the voltage monitoring parameter
GetCurrentVolt	Gets the current voltage value
GetFanParam	Gets the fan monitoring parameter
GetCurrentFan	Gets the current fan value
GetTempParam	Gets the temperature monitoring parameter
GetCurrentTemp	Gets the current temperature value
SetWdtCounter	Sets the watchdog timer counter
GetWdtCounter	Gets the watchdog timer counter
SetWdtMask	Sets warning masking in case of watchdog timer time-out
GetWdtMask	Gets warning masking in case of watchdog timer time-out
StartWdt	Starts the watchdog timer
StopWdt	Stops the watchdog timer
RestartWdt	Restarts the watchdog timer
RunningWdt	Gets the watchdog timer operation status
SetWarningOut	Sets warning output
GetWarningOut	Gets warning output
GetUniversalIn	Gets universal input
ClearUniversalIn	Clears the universal input latched status
SetUniversalInMask	Sets universal input masking
GetUniversalInMask	Gets universal input masking
SetResetMask	Sets reset-masking
GetResetMask	Gets reset-masking
SetIdeErr	Sets software mirroring error
GetIdeErrHard	Gets hardware mirroring error
GetEvent	Gets the error event
ClearEvent	Clears the error event
StartInsideBuzzer	Starts PL unit's internal buzzer
StopInsideBuzzer	Stops PL unit's internal buzzer
ChkInsideBuzzer	Checks PL unit's internal buzzer
GetWdtTimeout	Gets watchdog timeout status
ClearWdtTimeout	Clears the watchdog timeout status
SetWarningDOUT	Sets warning DOUT
GetWarningDOUT	Gets warning DOUT
GetSmiDrvHandle	Gets Software Mirroring driver handle
CloseSmiDrvHandle	Gets Software Mirroring Status
GetSmiAryStatus	Gets Software Mirroring feature Status
GetSmiDevStatus	Gets Software Mirroring Device Status
SetWdtResetMask	Sets Watchdog Timeout reset masking
GetWdtResetMask	Gets Watchdog Timeout reset masking

A.4.6 Visual C++ Function Specifications (Details)

GetDrvHandle

Call Format	int GetDrvHandle(void)
Return Value	0: Normal 1: Error
Arguments	None
Processing	Gets the device driver handle to communicate with the device driver. The handle Getsed is stored into the member variable m_handle.

Example 1	CPL_Ioctl m_Ioc;
	m_Ioc.GetDrvHandle();

Example 2	int ret;
	HANDLE hndl;
	ret = ::GetDrvHandle(&hndl);

Note: **An error occurs if the System Monitor/RAS Device Driver is not running.**

CloseDrvHandle

Call Format	BOOL CloseDrvHandle(void)
Return Value	TRUE: Normal FALSE: Error
Arguments	None
Processing	Destroys the device driver handle created using the GetDrvHandle function.

Example 1	CPL_Ioctl m_Ioc;
	BOOL ret;
	// Destroys the device driver handle.
	ret = m_Ioc.CloseDrvHandle();

Example 2	BOOL ret;
	// Destroys the device driver handle.
	ret = ::CloseDrvHandle();

GetDrvVersion

Call Format	BOOL GetDrvVersion(int *pMajor, int *pMinor)	
Return Value	TRUE: Normal FALSE: Error	
Arguments	(I/O) int *pMajor	Pointer to version information (Major, 0 to 99).
	(I/O) int *pMinor	Pointer to version information (Minor, 0 to 99).
Processing	Gets the driver's version information.	

Example 1

```
CPL_Ioctl  m_Ioc;
BOOL       ret;
int        Major, Minor;
ret = m_Ioc.GetDrvVersion( &Major, &Minor );
```

Example 2

```
BOOL       ret;
int        Major, Minor;
ret = ::GetDrvVersion( &Major, &Minor );
```

Note:

If the version is 1.10, then you will get

Major: 1 (decimal)

Minor: 10 (decimal).

GetMonitorSetup

Call Format BOOL GetMonitorSetup(int Selector, int *pSetup)

Return Value TRUE: Normal

FALSE: Error

Arguments (I) int Selector Parameters

MONITOR_VOLT_CPU	CPU core voltage
MONITOR_VOLT_P33	+3.3 V
MONITOR_VOLT_P50	+5.0 V
MONITOR_VOLT_P12	+12 V
MONITOR_VOLT_M12	-12 V
MONITOR_VOLT_M50	-5.0 V
MONITOR_TEMP_SYSTEM	System temperature
MONITOR_TEMP_CPU	CPU temperature
MONITOR_TEMP_OPT	Option temperature
MONITOR_FAN_CPU	CPU fan
MONITOR_FAN_POWER	Power fan
MONITOR_FAN_OPT	Option fan
MONITOR_VOLT_VIT	CPU core voltage 2

(I/O) int *pSetup Pointer to Getsed Data

0: Disabled

1: Enabled

Processing Gets the current monitoring enabled/disabled status.

Example 1

```
CPL_Ioctl       m_Ioc;
BOOL            ret;
int             Setup;
// Gets the CPU core voltage setup status.
ret = m_Ioc.GetMonitorSetup( MONITOR_VOLT_CPU, &Setup );
```

Example 2

```
BOOL            ret;
int             Setup;
// Get the CPU core voltage setup status.
ret = ::GetMonitorSetup( MONITOR_VOLT_CPU, &Setup );
```

GetVoltParam

Call Format BOOL GetVoltParam (int Selector, int *pULimit, int *pLLimit)

Return Value TRUE: Normal
FALSE: Error

Arguments (I) int Selector Parameters

MONITOR_VOLT_CPU	CPU core voltage
MONITOR_VOLT_P33	+3.3 V
MONITOR_VOLT_P50	+5.0 V
MONITOR_VOLT_P12	+12 V
MONITOR_VOLT_M12	-12 V
MONITOR_VOLT_M50	-5.0 V
MONITOR_VOLT_VIT	CPU core voltage 2

(I/O) int *pULimit Pointer to upper-limit voltage value (Unit: mV)
(I/O) int *pLLimit Pointer to lower-limit voltage value (Unit: mV)

Processing Gets the voltage monitoring parameter.

Example 1

```
CPL_Ioctl       m_Ioc;
BOOL            ret;
int             ULimit, LLimit;
// Get the upper and lower-limit values of the CPU core voltage.
ret = m_Ioc.GetVoltParam( MONITOR_VOLT_CPU,
 &ULimit, &LLimit );
```

Example 2

```
BOOL            ret;
int             ULimit, LLimit;
// Get the upper and lower-limit values of the CPU core voltage.
ret = ::GetVoltParam( MONITOR_VOLT_CPU, &ULimit, &LLimit );
```

Note:

Since the data taken from this function is shown in mV units, the following conversion is needed for use in (Volt) units:

Data in Volt unit = Data in mV unit/1000

GetCurrentVolt

Call Format BOOL GetCurrentVolt(int Selector, int *pData)

Return Value TRUE: Normal

FALSE: Error

Arguments (I) int Selector Parameters

MONITOR_VOLT_CPU	CPU core voltage
MONITOR_VOLT_P33	+3.3 V
MONITOR_VOLT_P50	+5.0 V
MONITOR_VOLT_P12	+12 V
MONITOR_VOLT_M12	-12 V
MONITOR_VOLT_M50	-5.0 V
MONITOR_VOLT_VIT	CPU core voltage 2

(I/O) int *pData Pointer to the voltage value (Unit: mV)

Processing Gets the current voltage value.

Example 1

```
CPL_Ioctl     m_Ioc;
BOOL          ret;
int           Data;
// Get the CPU core voltage value.
ret = m_Ioc.GetCurrentVolt( MONITOR_VOLT_CPU, &Data );
```

Example 2

```
BOOL          ret;
int           Data;
// Get the CPU core voltage value.
ret = ::GetCurrentVolt( MONITOR_VOLT_CPU, &Data );
```

Since the data taken from this function is shown in mV units, the following conversion is needed for use in (Volt) units:

Data in Volt unit = Data in mV unit/1000

GetFanParam

Call Format BOOL GetFanParam (int Selector, int *pLLimit)

Return Value TRUE: Normal
FALSE: Error

Arguments (I) int Selector Parameters

MONITOR_FAN_CPU	CPU fan
MONITOR_FAN_POWER	Power fan
MONITOR_FAN_OPT	Option fan

(I/O) int *pLLimit Pointer to the lower-limit fan rotation speed (Unit: RPM)
(RPM: Revolutions Per Minute)

Processing Gets the fan monitoring parameter.

Example 1

```
CPL_Ioctl       m_Ioc;
BOOL            ret;
int             LLimit;
// Get the lower-limit CPU fan rotational speed.
ret = m_Ioc.GetFanParam( MONITOR_FAN_CPU, &LLimit );
```

Example 2

```
BOOL            ret;
int             LLimit;
// Get the lower-limit CPU fan rotation speed.
ret = ::GetFanParam( MONITOR_FAN_CPU, &LLimit );
```

GetCurrentFan

Call Format	BOOL GetCurrentFan(int Selector, int *pData)		
Return Value	TRUE: Normal		
	FALSE: Error		
Arguments	(I) int Selector	Parameters	
		MONITOR_FAN_CPU	CPU fan
		MONITOR_FAN_POWER	Power fan
		MONITOR_FAN_OPT	Option fan
	(I/O) int *pData	Pointer to the fan rotation speed (Unit: RPM)	
		(RPM: Revolutions Per Minute)	
Processing	Gets the current fan rotation speed.		

Example 1

```
CPL_Ioctl    m_Ioc;
BOOL         ret;
int          Data;
// Get the CPU fan rotational speed.
ret = m_Ioc.GetCurrentFan( MONITOR_FAN_CPU, &Data );
```

Example 2

```
BOOL         ret;
int          Data;
// Get the CPU fan rotational speed.
ret = ::GetCurrentFan( MONITOR_FAN_CPU, &Data );
```

GetTempParam

Call Format BOOL GetTempParam(int Selector, int *pULimit)

Return Value TRUE: Normal
FALSE: Error

Arguments (I) int Selector Parameters

MONITOR_TEMP_SYSTEM	System temperature
MONITOR_TEMP_CPU	CPU temperature
MONITOR_TEMP_OPT	Option temperature

(I/O) int *pULimit Pointer to the upper-limit temperature
(Unit: Degrees Celsius)

Processing Gets the temperature monitoring parameter.

Example 1

```
CPL_Ioctl      m_Ioc;
BOOL           ret;
int            ULimit;
// Get the system temperature upper-limit value.
ret = m_Ioc.GetTempParam( MONITOR_TEMP_SYSTEM,
&ULimit );
```

Example 2

```
BOOL           ret;
int            ULimit;
ret = ::GetTempParam( MONITOR_TEMP_SYSTEM, &ULimit );
```

GetCurrentTemp

Call Format BOOL GetCurrentTemp(int Selector, int *pData)

Return Value TRUE: Normal
FALSE: Error

Arguments (I) int Selector Parameters

MONITOR_TEMP_SYSTEM	System temperature
MONITOR_TEMP_CPU	CPU temperature
MONITOR_TEMP_OPT	Option temperature

(I/O) int *pData Pointer to the temperature
(Unit: Degrees Celsius)

Processing Gets the current temperature value.

Example 1

```
CPL_Ioctl      m_Ioc;
BOOL           ret;
int            Data;
// Gets the system temperature value.
ret = m_Ioc.GetCurrentTemp( MONITOR_TEMP_SYSTEM, &Data );
```

Example 2

```
BOOL           ret;
int            Data;
// Gets the system temperature value.
ret = ::GetCurrentTemp( MONITOR_TEMP_SYSTEM, &Data );
```

SetWdtCounter

Call Format BOOL SetWdtCounter(int Counter)

Return Value TRUE: Normal
FALSE: Error

Arguments (I) int Counter Sets to the watchdog timer's initial (5 to 255) counter value (Unit: Seconds)

Processing Sets watchdog timer's initial counter value.

Example 1

```
CPL_Ioctl       m_Ioc;
BOOL            ret;
// Sets the Watchdog Timer's initial count value to 10.
ret = m_Ioc.SetWdtCounter( 10 );
```

Example 2

```
BOOL            ret;
// Sets the Watchdog Timer's initial count value to 10.
ret = ::SetWdtCounter( 10 );
```

GetWdtCounter

Call Format BOOL GetWdtCounter(int *pCounter)

Return Value TRUE: Normal
FALSE: Error

Arguments (I/O) int *pCounter Pointer to the watchdog timer's initial counter value (Unit: Seconds)

Processing Gets the current watchdog timer's initial counter value.

Example 1

```
CPL_Ioctl       m_Ioc;
BOOL            ret;
int             Counter;
ret = m_Ioc.GetWdtCounter( &Counter );
```

Example 2

```
BOOL            ret;
int             Counter;
ret = ::GetWdtCounter( &Counter );
```

SetWdtMask

Call Format BOOL SetWdtMask(int Selector, int Mask)

Return Value TRUE: Normal
FALSE: Error

Arguments (I) int Selector Setting Item

WARNING_LAMP	LAMP
WARNING_ALARM	ALARM

(I) int Mask Masking Information

MASK_OFF	Masking disabled
MASK_ON	Masking enabled

Processing Sets masking for the warning that is output when watchdog timer time-out occurs.

Example 1

```
CPL_Ioctl       m_Ioc;
BOOL            ret;
// Enable masking for LAMP output.
ret = m_Ioc.SetWdtMask( WARNING_LAMP, MASK_ON );
// Disable masking for ALARM output.
ret = m_Ioc.SetWdtMask( WARNING_ALARM, MASK_OFF );
```

Example 2

```
BOOL            ret;
// Enable masking for LAMP output.
ret = ::SetWdtMask( WARNING_LAMP, MASK_ON );
// Disable masking for ALARM output.
ret = ::SetWdtMask( WARNING_ALARM, MASK_OFF );
```

GetWdtMask

Call Format BOOL GetWdtMask(int Selector, int *pMask)

Return Value TRUE: Normal
FALSE: Error

Arguments (I) int Selector Setting Item

WARNING_LAMP	LAMP
WARNING_ALARM	ALARM

(I/O) int *pMask Pointer to Masking Information

MASK_OFF	Masking disabled
MASK_ON	Masking enabled

Processing Gets the masking information for warning output that is created when a watchdog timer time-out occurs.

Example 1

```
CPL_Ioctl       m_Ioc;
BOOL            ret;
int             Mask;
// Gets the LAMP masking information.
ret = m_Ioc.GetWdtMask( WARNING_LAMP, &Mask );
// Get the ALARM masking information.
ret = m_Ioc.GetWdtMask( WARNING_ALARM, &Mask );
```

Example 2

```
BOOL            ret;
int             Mask;
// Gets the LAMP0 masking information.
ret = ::GetWdtMask( WARNING_LAMP, &Mask );
//  Get the ALARM masking information.
ret = ::GetWdtMask( WARNING_ALARM, &Mask );
```

StartWdt

Call Format BOOL StartWdt(void)

Return Value TRUE: Normal
FALSE: Error

Arguments None

Processing Starts watchdog timer countdown.

Example 1

```
CPL_Ioctl       m_Ioc;
BOOL            ret;
ret = m_Ioc.StartWdt();
```

Example 2

```
BOOL            ret;
ret = ::StartWdt();
```

StopWdt

Call Format	BOOL StopWdt(void)	
Return Value	TRUE: Normal	
	FALSE: Error	
Arguments	None	
Processing	Stops watchdog timer countdown.	
Example 1	CPL_Ioctl	m_Ioc;
	BOOL	ret;
	ret = m_Ioc.StopWdt();	
Example 2	BOOL	ret;
	ret = ::StopWdt();	

RestartWdt

Call Format	BOOL RestartWdt(void)	
Return Value	TRUE: Normal	
	FALSE: Error	
Arguments	None	
Processing	Restarts watchdog timer countdown after resetting to the initial value.	
Example 1	CPL_Ioctl	m_Ioc;
	BOOL	ret;
	m_Ioc.RestartWdt();	
Example 2	BOOL	ret;
	ret = ::RestartWdt();	

Note: **Processing cannot be performed if the watchdog timer is stopped.**

RunningWdt

Call Format	BOOL RunningWdt(int *pRunFlag)		
Return Value	TRUE: Normal		
	FALSE: Error		
Arguments	(I/O) int *pRunFlag	Pointer to Watchdog Timer Operation Status	
	WATCHDOG_STOP	Stopped	
	WATCHDOG_COUNTDOWN	Countdown in progress	
Processing	Gets the watchdog timer's operation status.		
Example 1	CPL_Ioctl	m_Ioc;	
	BOOL	ret;	
	int	RunFlag;	
	ret = m_Ioc.RunningWdt(&RunFlag);		
Example 2	BOOL	ret;	
	int	RunFlag;	
	ret = ::RunningWdt(&RunFlag);		

SetWarningOut

Call Format	BOOL SetWarningOut(int Selector, int WarnOut)		
Return Value	TRUE: Normal		
	FALSE: Error		
Arguments	(I) int Selector	Setting Item	
		WARNING_LAMP	LAMP
		WARNING_ALARM	ALARM
	(I) int WarnOut	Output Status	
		OUTPUT_OFF	Output OFF
		OUTPUT_ON	Output ON
Processing	Sets the warning information for the set item (lamp or alarm).		
Example 1	CPL_Ioctl	m_Ioc;	
	BOOL	ret;	
	// Sets the LAMP output status to ON.		
	ret = m_Ioc.SetWarningOut(WARNING_LAMP, OUTPUT_ON);		
	// Sets the ALARMoutput status to OFF.		
	ret = m_Ioc.SetWarningOut(WARNING_ALARM, OUTPUT_OFF);		
Example 2	BOOL	ret;	
	// Sets the LAMP output status to ON.		
	ret = ::SetWarningOut(WARNING_LAMP, OUTPUT_ON);		
	// Sets the ALARM output status to OFF.		
	ret = ::SetWarningOut(WARNING_ALARM, OUTPUT_OFF);		

GetWarningOut

Call Format BOOL GetWarningOut(int Selector, int *pWarnOut)

Return Value TRUE: Normal

FALSE: Error

Arguments (I) int Selector Settting Item

WARNING_LAMP	LAMP
WARNING_ALARM	ALARM

(I/O) int *pWarnOut Pointer to Output Status

OUTPUT_OFF	Output OFF
OUTPUT_ON	Output ON

Processing Gets the warning status of the current set item (LAMP or ALARM).

Example 1

```
CPL_Ioctl        m_Ioc;
BOOL             ret;
int              WarnOut;
// Gets the LAMP output status.
ret = m_Ioc.GetWarningOut( WARNING_LAMP, &WarnOut );
// Gets the ALARM output status.
ret = m_Ioc.GetWarningOut( WARNING_ALARM, &WarnOut );
```

Example 2

```
BOOL       ret;
int        WarnOut;
// Gets the LAMP output status.
ret = ::GetWarningOut( WARNING_LAMP, &WarnOut );
// Gets the ALARM output status.
ret = ::GetWarningOut( WARNING_ALARM, &WarnOut );
```

GetUniversalIn

Call Format BOOL GetUniversalIn(int Selector, int *pUniIn)

Return Value TRUE: Normal

FALSE: Error

Arguments (I) int Selector Designated Port

PORT_UNI0	Universal Input 0
PORT_UNI1	Universal Input 1

(I/O) int *pUniIn Pointer to Input Status

INPUT_OFF	Input OFF
INPUT_ON	Input ON

Processing Gets the input status of the designated port (Universal Input 0, Universal Input 1).

Example1

```
CPL_Ioctl       m_Ioc;
BOOL            ret;
int             UniIn;
// Gets the input status of Universal Input 0.
ret = m_Ioc.GetUniversalIn( PORT_UNI0, &UniIn );
// Gets the input status of Universal Input 1.
ret = m_Ioc.GetUniversalIn( PORT_UNI1, &UniIn );
```

Example 2

```
BOOL            ret;
int             UniIn;
// Gets the input status of Universal Input 0.
ret = ::GetUniversalIn( PORT_UNI0, &UniIn );
// Gets the input status of Universal Input 1.
ret = ::GetUniversalIn( PORT_UNI1, &UniIn );
```

ClearUniversalIn

Call Format	BOOL ClearUniversalIn(int Selector)		
Return Value	TRUE: Normal		
	FALSE: Error		
Arguments	(I) int Selector	Designated Port	
		PORT_UNI0	Universal Input 0
		PORT_UNI1	Universal Input 1
Processing	Clears the input status of the designated port (Universal Input 0, Universal Input 1).		

Example 1

```
CPL_Ioctl        m_Ioc;
BOOL             ret;
// Cancels the output of Universal Input 0.
ret = m_Ioc.ClearUniversalIn( PORT_UNI0 );
// Cancels the output of Universal Input 1.
ret = m_Ioc.ClearUniversalIn( PORT_UNI1 );
```

Example 2

```
BOOL             ret;
// Cancels the output of Universal Input 0.
ret = ::ClearUniversalIn( PORT_UNI0 );
// Cancels the output of Universal Input 1.
ret = ::ClearUniversalIn( PORT_UNI1 );
```

SetUniversalInMask

Call Format	BOOL SetUniversalInMask(int Selector, int Mask)		
Return Value	TRUE: Normal		
	FALSE: Error		
Arguments	(I) int Selector	Designated Port	
		PORT_UNI0	Universal Input 0
		PORT_UNI1	Universal Input 1
	(I/O) int Mask	Masking Information	
		MASK_OFF	Masking disabled
		MASK_ON	Masking enabled
Processing	Sets the masking information for the designated port (Universal Input 0, Universal Input 1).		

Example 1

```
CPL_Ioctl          m_Ioc;
BOOL               ret;
// Disable masking for Universal Input 0.
ret = m_Ioc.SetUniversalInMask( PORT_UNI0, MASK_OFF );
// Enable masking for Universal Input 1.
ret = m_Ioc.SetUniversalInMask( PORT_UNI1, MASK_ON );
```

Example 2

```
BOOL                ret;
// Disable masking for Universal Input 0.
ret = ::SetUniversalInMask( PORT_UNI0, MASK_OFF );
// Enable masking for Universal Input 1.
ret = ::SetUniversalInMask( PORT_UNI1, MASK_ON );
```

GetUniversalInMask

Call Format	BOOL GetUniversalInMask(int Selector, int *pMask)	
Return Value	TRUE: Normal	
	FALSE: Error	
Arguments	(I) int Selector	Designated Port
		PORT_UNI0 Universal Input 0
		PORT_UNI1 Universal Input 1
	(I/O) int *pMask	Pointer to Masking Information
		MASK_OFF Masking disabled
		MASK_ON Masking enabled
Processing	Gets the masking information for the designated port (Universal Input 0, Universal Input 1).	

Example 1

```
CPL_Ioctl       m_Ioc;
BOOL            ret;
int             Mask;
// Gets the masking information for Universal input 0.
ret = m_Ioc.GetUniversalInMask( PORT_UNI0, &Mask );
// Gets the masking information for Universal input 1.
ret = m_Ioc.GetUniversalInMask( PORT_UNI1, &Mask );
```

Example 2

```
BOOL            ret;
int             Mask;
// Gets the masking information for Universal input 0.
ret = ::GetUniversalInMask( PORT_UNI0, &Mask );
// Gets the masking information for Universal input 1.
ret = ::GetUniversalInMask( PORT_UNI1, &Mask );
```

SetResetMask

Call Format	BOOL SetResetMask(int Mask)	
Return Value	TRUE: Normal	
	FALSE: Error	
Arguments	(I) int Mask	Masking Information
		MASK_OFF Masking disabled
		MASK_ON Masking enabled
Processing	Sets reset-masking.	

Example 1

```
CPL_Ioctl       m_Ioc;
BOOL            ret;
// Disable reset-masking.
ret = m_Ioc.SetResetMask( MASK_OFF );
```

Example 2

```
BOOL            ret;
// Disable reset-masking.
ret = ::SetResetMask( MASK_OFF );
```

GetResetMask

Call Format	BOOL GetResetMask(int *pMask)		
Return Value	TRUE: Normal		
	FALSE: Error		
Arguments	(I/O) int *pMask	Pointer to Masking Information	
		MASK_OFF	Masking disabled
		MASK_ON	Masking enabled
Processing	Gets the current reset-masking information.		
Example 1	CPL_Ioctl	m_Ioc;	
	BOOL	ret;	
	int	Mask;	
	ret = m_Ioc.GetResetMask(&Mask);		
Example 2	BOOL	ret;	
	int	Mask;	
	ret = ::GetResetMask(&Mask);		

SetIdeErr

Call Format	BOOL SetIdeErr(int IdeErr)		
Return Value	TRUE: Normal		
	FALSE: Error		
Arguments	(I) int IdeErr	Error Output Information	
		IDE_ERROR_OFF	Error Output OFF
		IDE_ERROR_ON	Error Output ON
Processing	Uses software control to set IDE error output.		
Example 1	CPL_Ioctl	m_Ioc;	
	BOOL	ret;	
	ret = m_Ioc.SetIdeErr(IDE_ERROR_OFF);		
Example 2	BOOL	ret;	
	ret = ::SetIdeErr(IDE_ERROR_OFF);		

GetIdeErrHard

Call Format	BOOL GetIdeErrHard(int Selector, int *pIdeErr)		
Return Value	TRUE: Normal		
	FALSE: Error		
Arguments	(I) int Selector	Parameters	
		IDE_ERROR_1	IDE_ERR1
		IDE_ERROR_2	IDE_ERR2
	(I/O) int *pIdeErr	Pointer to Output Status	
		IDE_ERROR_OFF	Normal
		IDE_ERROR_ON	Error
Processing	Gets hardware's current IDE error signal.		

Example 1

```
CPL_Ioctl      m_Ioc;
BOOL           ret;
int            IdeErr;
// Gets the IDE_ERR1 signal status
 ret = m_Ioc.GetIdeErrHard( IDE_ERROR_1, &IdeErr );
```

Example 2

```
BOOL           ret;
int            IdeErr;
// Gets the IDE_ERR1 signal status
ret = ::GetIdeErrHard( IDE_ERROR_1, &IdeErr );
```

GetEvent

Call Format BOOL GetEvent(int Selector, int *pEvent)

Return Value TRUE: Normal

FALSE: Error

Arguments (I) int Selector Parameter s

EVENT_VOLT_CPU	CPU core voltage
EVENT_VOLT_P33	+3.3 V
EVENT_VOLT_P50	+5.0 V
EVENT_VOLT_P12	+12 V
EVENT_VOLT_M12	-12 V
EVENT_VOLT_M50	-5.0 V
EVENT_VOLT_VIT	CPU core voltage 2
EVENT_FAN_CPU	CPU fan
EVENT_FAN_POWER	Power fan
EVENT_FAN_OPT	Option fan
EVENT_TEMP_SYSTEM	System temperature
EVENT_TEMP_CPU_OPT	CPU or option temperature
EVENT_UNI_IN0	Universal input 0
EVENT_UNI_IN1	Universal input 1
EVENT_WDT_TIMEOUT	Watchdog Timeout

(I/O) int *pEvent Pointer to Error Event Information

ERROR_EVENT_OFF	Without error event
ERROR_EVENT_ON	With error event

Processing Checks the machine for voltage, fan, and temperature errors, and the Universal Input information (event) and Watchdog Timeout error.

Example 1

```
CPL_Ioctl      m_Ioc;
BOOL           ret;
int            Event;
// Gets the error event information for the CPU core voltage.
ret = m_Ioc.GetEvent( EVENT_VOLT_CPU, &Event );
```

Example 2

```
BOOL    ret;
int     Event;
// Gets the error event information for the CPU core voltage.
ret = ::GetEvent( EVENT_VOLT_CPU, &Event );
```

ClearEvent

Call Format BOOL ClearEvent(int Selector)

Return Value TRUE: Normal

FALSE: Error

Arguments (I) int Selector Designated Parameters for ClearEvent

EVENT_VOLT_CPU	CPU core voltage
EVENT_VOLT_P33	+3.3 V
EVENT_VOLT_P50	+5.0 V
EVENT_VOLT_P12	+12 V
EVENT_VOLT_M12	-12 V
EVENT_VOLT_M50	-5.0 V
EVENT_VOLT_VIT	CPU core voltage 2
EVENT_FAN_CPU	CPU fan
EVENT_FAN_POWER	Power fan
EVENT_FAN_OPT	Option fan
EVENT_TEMP_SYSTEM	System temperature
EVENT_TEMP_CPU_OPT	CPU or option temperature
EVENT_UNI_IN0	Universal input 0
EVENT_UNI_IN1	Universal input 1
EVENT_WDT_TIMEOUT	Watchdog Timeout

Processing Cancels the error event.

Example 1

```
CPL_Ioctl       m_Ioc;
BOOL            ret;
// Cancels the error event for the CPU core voltage.
ret = m_Ioc.ClearEvent( EVENT_VOLT_CPU );
```

Example 2

```
BOOL            ret;
// Cancels the error event for the CPU core voltage.
ret = ::ClearEvent( EVENT_VOLT_CPU );
```

StartInsideBuzzer

Call Format	BOOL WINAPI StartInsideBuzzer (int hz, int ms)	
Return Value	BOOL	TRUE: Normal FALSE: Error
Arguments	(I) int hz	Buzzer frequency (Hz)
	(I) int ms	Buzzer length (ms)
Processing	Starts the PL unit's internal buzzer, based on the designated frequency and length.	

Example 1	BOOL ret;
	int hz = 600;
	int ms = 1000;
	// PL internal buzzer will sound at 600MHz for 1 second.
	ret = m_Ioc.StartInsideBuzzer (hz, ms);	

Example 2	BOOL ret;
	int hz = 600;
	int ms = 1000;
	// PL internal buzzer will sound at 600MHz for 1 second.
	ret = ::StartInsideBuzzer (hz, ms);	

Note: **This feature cannot be used with a PL running WindowsNT 4.0 or Windows 2000 due to the use of Windows 95 and Windows 98 functions.**

StopInsideBuzzer

Call Format	BOOL WINAPI StopInsideBuzzer (void)	
Return Value	BOOL	TRUE: Normal FALSE: Error
Arguments	None.	
Processing	Stops the PL unit's internal buzzer.	

Example 1	CPL_Ioctl m_loc;
	BOOL ret;
	// Stops PL internal buzzer.
	ret = m_Ioc.StopInsideBuzzer ( );	

Example 2	BOOL ret;
	// Stops PL internal buzzer.
	ret = ::StopInsideBuzzer ( );	

Note: **This feature cannot be used with a PL running WindowsNT 4.0 or Windows 2000 due to the use of Windows 95 and Windows 98 functions.**

ChkInsideBuzzer

Call Format BOOL WINAPI ChkInsideBuzzer (int *BuzzerParam)

Return Value BOOL TRUE: Normal
FALSE: Error

Arguments (I/O) int *BuzzerParam Buzzer Status Pointer

BUZZER_ON	Buzzer is ON
BUZZER_OFF	Buzzer is OFF

Processing Checks the buzzer's operation status.

Example 1

```
CPL_Ioctl     m_loc;
BOOL          ret;
int           BuzzerParam;
// Checks buzzer status.
ret = m_Ioc.ChkInsideBuzzer ( &BuzzerParam );
```

Example 2

```
BOOL   ret;
// Checks buzzer status.
ret = ::ChkInsideBuzzer ( &BuzzerParam );
```

Note: **This feature cannot be used with a PL unit running WindowsNT 4.0 or Windows 2000, due to the use of Windows 95 and Windows 98 functions.**

GetWdtTimeout

Call Format BOOL GetWdtTimeout(int *pTimebuf)

Return Value TRUE: Normal
FALSE: Error

Arguments (I/O) int *pTimebuf Pointer to Watchdog Status

TIMEOUT_OK	Not timeout
TIMEOUT_ERROR	Now timeout

Processing Gets watchdog timeout status.

Example 1

```
CPL_Ioctl     m_Ioc;
BOOL          ret;
int           Timebuf;
// Gets watchdog timeout status.
ret = Gm_Ioc.GetWdtTimeout( &Timebuf );
```

Example 2

```
BOOL   ret;
int    Timebuf;
// Gets watchdog timeout status.
ret = ::GetWdtTimeout( &Timebuf );
```

ClearWdtTimeout

Call Format	BOOL ClearWdtTimeout(void)	
Return Value	TRUE: Normal	
	FALSE: Error	
Arguments	None	
Processing	Clears the watchdog timeout status.	
Example 1	CPL_Ioctl	m_Ioc;
	BOOL	ret;
	// Clears the watchdog timeout status.	
	ret = m_Ioc.GetWdtTimeout();	
Example 2	BOOL	ret;
	// Clears the watchdog timeout status.	
	ret = ::GetWdtTimeout();	

SetWarningDOUT

Call Format	BOOL SetWarningDOUT(int WarningOut)		
Return Value	TRUE: Normal		
	FALSE: Error		
Arguments	(I) int WarningOut	Output status	
		OUTPUT_OFF	Output OFF
		OUTPUT_ON	Output ON
Processing	Sets alarm status of DOUT.		
Example 1	CPL_Ioctl	m_Ioc;	
	BOOL	ret;	
	// Sets DOUT output status to OFF.		
	ret = m_Ioc.SetWarningDOUT(OUTPUT_OFF);		
Example 2	BOOL	ret;	
	// Sets DOUT output status to OFF.		
	ret = ::SetWarningDOUT(OUTPUT_OFF);		

GetWarningDOUT

Call Format	BOOL GetWarningDOUT(int *pWarningOUT)
Return Value	TRUE: Normal FALSE: Error
Arguments	(I/O) int *pWarningOut Pointer to Output Status OUTPUT_OFF Output OFF OUTPUT_ON Output ON
Processing	Gets alarm status of DOUT.

Example 1	CPL_Ioctl m_Ioc;
	BOOL ret;
	int WarningOut;
	// Gets alarm status of DOUT.
	ret = m_Ioc.GetWarningDOUT(&WarningOut);

Example 2	BOOL ret;
	int WarningOut;
	// Gets alarm status of DOUT.
	ret = ::GetWarningDOUT(&WarningOut);

GetSmiDrvHandle

Call Format	int GetSmiDrvHandle(void)
Return Value	0: Normal Not 0 : Error
Arguments	NONE
Processing	Gets device driver handle for communication with Software Mirroring device driver.

Example 1	CPL_SmiIoctl m_SmiIoc;
	BOOL ret;
	// Gets Software Mirroring driver handle.
	ret = m_SmiIoc.GetSmiDrvHandle();

Example 2	BOOL ret;
	// Gets Software Mirroring driver handle.
	ret = ::GetSmiDrvHandle();

Note: **When the Software Mirroring Driver is not loaded, an error is returned.**

CloseSmiDrvHandle

Call Format	BOOL CloseSmiDrvHandle(void)
Return Value	True: Normal False: Error
Arguments	NONE
Processing	Destroys handle created in GetSmiDrvHandle.

Example 1	CPL_SmiIoctl m_SmiIoc;
	BOOL ret;
	// Destroys Software Mirroring driver handle.
	ret = m_SmiIoc.CloseSmiDrvHandle();

Example 2	BOOL ret;
	// Destroys Software Mirroring driver handle.
	ret = ::CloseSmiDrvHandle();

GetSmiAryStatus

Call Format	BOOL GetSmiAryStatus(int *pStatus)
Return Value	TRUE: Normal FALSE: Error
Arguments	(I/O) int *pStatus Pointer to Mirroring Status

ARYSTAT_GOOD	Good
ARYSTAT_NOTEXIST	Not output
ARYSTAT_UNCONFIG	Unconfigured
ARYSTAT_REBUILD	Rebuilding
ARYSTAT_REDUCE	Reduced
ARYSTAT_DEAD	Dead

Processing	Gets Software Mirroring status.

Example 1	CPL_SmiIoctl m_Smiloc;
	BOOL ret;
	int Status;
	// Gets Software Mirroring status.
	ret = m_Smiloc.GetSmiAryStatus(&Status);

Example 2	BOOL ret;
	int Status;
	// Gets Software Mirroring status.
	ret = ::GetSmiAryStatus(&Status);

GetSmiDevStatus

Call Format BOOL GetSmiDevStatus(int Id ,int *pType ,int *pStatus)

Return Value TRUE: Normal
FALSE: Error

Arguments

(I) int Id	Device ID	
	0 : Master HDD	
	1 : Slave HDD	
(I/O int* pType	Device Type	
	ATADEVICE	ATA DEVICE
	ATAPIDEVICE	CD-ROM
	UNKNOWNDEVICE	Unknown DEVICE
	NODEVICE	No DEVICE
(I/O) int* pStatus	Device Status	
	DEVSTAT_GOOD	Good
	DEVSTAT_NOTEXIST	No DEVICE
	DEVSTAT_BROKEN	BROKEN

Processing Gets Device Status of software mirroring.

Example 1

```
CPL_SmiIoctl    m_SmiIoc;
BOOL            ret;
int             Id, Type, Status;
// Gets device status.
Id = 0;
ret = m_SmiIoc.GetSmiDevStatus(ID ,&Type ,&Status);
```

Example 2

```
BOOL            ret;
int             Id, Type, Status;
// Gets device status.
Id = 0;
ret = ::GetSmiDevStatus(ID ,&Type ,&Status);
```

SetWdtResetMask

Call Format	BOOL SetWdtResetMask(int Mask)		
Return Value	TRUE: Normal		
	FALSE: Error		
Arguments	(I/O) int Mask	Masking Information	
		MASK_OFF	Masking disabled
		MASK_ON	Masking enabled
Processing	Sets WDT Timeout H/W reset-masking.		

Example 1	`CPL_Ioctl      m_Ioc;`
	`BOOL           ret;`
	`// Disables mask set during WDT timeout.`
	`ret = m_Ioc.SetWdtResetMask( MASK_OFF );`
Example 2	`BOOL           ret;`
	`// Disables mask set during WDT timeout.`
	`ret = ::SetWdtResetMask( MASK_OFF );`

GetWdtResetMask

Call Format	BOOL GetWdtResetMask(int *pMask)		
Return Value	TRUE: Normal		
	FALSE: Error		
Arguments	(I/O) int *pMask	Pointer to Masking Information	
		MASK_OFF	Masking disabled
		MASK_ON	Masking enabled
Processing	Gets the current WDT timeout H/W reset-masking information.		

Example 1	`CPL_Ioctl      m_Ioc;`
	`BOOL           ret;`
	`int            Mask;`
	`// Gets data of WDT timeout reset mask.`
	`ret = m_Ioc.GetWdtResetMask( &Mask );`
Example 2	`BOOL           ret;`
	`int            Mask;`
	`// Gets data of WDT timeout reset mask.`
	`ret = ::GetWdtResetMask( &Mask );`

A.4.7 Visual Basic Functions

Function Name	Description
InitIoctl	Creates a CPL_Ioctl object
EndIoctl	Destroys a CPL_Ioctl object
GetDrvHandle	Gets the driver handle
CloseDrvHandle	Destroys the driver handle
GetDrvVersion	Gets the driver version
GetMonitorSetup	Gets the enabled/disabled monitor settings
GetVoltParam	Gets the voltage monitoring parameters
GetCurrentVolt	Gets the current value of the voltage
GetFanParam	Gets the parameters for monitoring the FAN
GetCurrentFan	Gets the current value of the FAN
GetTempParam	Gets the parameters for monitoring the temperature
GetCurrentTemp	Gets the current value of the temperature
SetWdtCounter	Sets the value for the watchdog timer counter
GetWdtCounter	Gets the watchdog timer counter
SetWdtMask	Sets the watchdog timer counter time-out status warning mask
GetWdtMask	Gets the watchdog timer counter time-out status warning mask
StartWdt	Starts the watchdog timer
StopWdt	Stops the watchdog timer
RestartWdt	Restarts the watchdog timer
RunningWdt	Gets the watchdog status
SetWarningOut	Sets the warning output
GetWarningOut	Gets the warning output
GetUniversalIn	Gets the universal input
ClearUniversalIn	Clears the universal input latch
SetUniversalInMask	Sets the universal input mask
GetUniversalInMask	Gets the universal input mask
SetResetMask	Sets the reset mask
GetResetMask	Gets the reset mask
SetIdeErr	Sets the mirroring error (software error)
GetIdeErrHard	Gets the mirroring error (hardware error)
GetEvent	Gets an error event
ClearEvent	Clears an error event
StartInsideBuzzer	Starts PL internal buzzer
StopInsideBuzzer	Stops PL internal buzzer
ChkInsideBuzzer	Checks PL internal buzzer
GetWdtTimeout	Gets the time-out status of the watchdog timer
ClearWdtTimeout	Clear the time-out status of the watchdog timer
SetWarningDOUT	Sets the warning output DOUT
GetWarningDOUT	Gets the warning output DOUT
GetSemiDrvHandle	Gets Software Mirroring driver handle
CloseSmiDrvHandle	Destroys Software Mirroring driver handle
GetSmiAryStatus	Gets status of Software Mirroring Array
GetSmiDevStatus	Gets status of Software Mirroring Device
SetWdtResetMask	Sets the Reset Mask of the watchdog timer
GetWdtResetMask	Gets the Reset Mask of the watchdog timer

A.4.8 Visual Basic Function Specifications (Details)

InitIoctl

Call format	Declare Sub InitIoctl Lib "PL_Ioc.dll" ()
Return value	None
Argument	None
Processing	Creates a CPL_Ioctl object. The created object will not be released until the "EndIoctl" function is called.
Example	InitIoctl()

EndIoctl

Call format	Declare Sub EndIoctl Lib "PL_Ioc.dll" ()
Return value	None
Argument	None
Processing	Destroys the object created with the "InitIoctl" function.
Example	EndIoctl()

GetDrvHandle

Call format	Declare Function GetDrvHandle Lib "PL_Ioc.dll" (ByRef hndl As Long) As Long	
Return value	0: Normal 1: Error	
Argument	hndl As Long	Device driver handle (pass by reference)
Processing	Gets the device driver handle to exchange information with the device driver.	
Example	Dim ret As Long Dim hndl As Long ret = GetDrvHandle(hndl)	

Note: **An error will result if the system monitor/RAS device driver is not operating.**

CloseDrvHandle

Call format	Declare Function CloseDrvHandle Lib "PL_Ioc.dll"() As Long
Return value	Other than 0: Normal 0: Error
Argument	None
Processing	Destroys the handle acquired with the "GetDrvHandle" function.

Example	Dim ret As Long
	// Destroy handle
	ret = CloseDrvHandle()

GetDrvVersion

Call format	Declare Function GetDrvVersion Lib "PL_Ioc.dll"	
	(ByRef Major As Long, ByRef Minor As Long) As Long	
Return value	Other than 0: Normal	
	0: Error	
Argument	Major As Long	Version data (Major, 0 to 99) (pass by reference)
	Minor As Long	Version data (Major, 0 to 99) (pass by reference)
Processing	Gets the driver version.	
Example	Dim ret As Long	
	Dim Major As Long	
	Dim Minor As Long	
	ret = GetDrvVersion(Major, Minor)	

Note:

When the version is 1.10,

Major:1	**(Decimal)**
Minor:10	**(Decimal)**

GetMonitorSetup

Call format Declare Function GetMonitorSetup Lib "PL_Ioc.dll"
(ByVal Selector As Long, ByRef Setup As Long) As Long

Return value Other than 0: Normal
0: Error

Argument Selector As Long Parameters (pass by value)

MONITOR_VOLT_CPU	CPU Core voltage
MONITOR_VOLT_P33	+3.3V
MONITOR_VOLT_P50	+5.0V
MONITOR_VOLT_P12	+12V
MONITOR_VOLT_M12	-12V
MONITOR_VOLT_M50	-5.0V
MONITOR_VOLT_VIT	CPU Core voltage 2
MONITOR_TEMP_SYSTEM	SYSTEM Temp.
MONITOR_TEMP_CPU	CPU Temp.
MONITOR_TEMP_OPT	OPTION Temp.
MONITOR_FAN_CPU	CPU FAN
MONITOR_FAN_POWER	POWER FAN
MONITOR_FAN_OPT	OPTION FAN

Setup As Long Get data (pass by reference)
0:Disable
1:Enable

Processing Gets the current enabled/disabled monitor status.

Example

```
Dim ret As Long
Dim Setup As Long
// Get the setup status of the CPU core voltage
ret = GetMonitorSetup(MONITOR_VOLT_CPU, Setup )
```

GetVoltParam

Call format Declare Function GetVoltParam Lib "PL_Ioc.dll"

(ByVal Selector As Long, ByRef ULimit As Long, ByRef LLimit As Long) As Long

Return value Other than 0: Normal

0: Error

Argument Selector As Long Parameters (pass by value)

MONITOR_VOLT_CPU	CPU Core voltage
MONITOR_VOLT_P33	+3.3V
MONITOR_VOLT_P50	+5.0V
MONITOR_VOLT_P12	+12V
MONITOR_VOLT_M12	-12V
MONITOR_VOLT_M50	-5.0V
MONITOR_VOLT_VIT	CPU Core voltage 2

Ulimit As Long Voltage value upper limit (unit: mV)
(pass by reference)

LLimit As Long Voltage value lower limit (unit: mV)
(pass by reference)

Processing Gets the voltag monitoring parameter.

Example

```
Dim ret As Long
Dim ULimit  As Long
Dim LLimit As Long
// Get the upper/lower limit of the CPU core voltage value
ret = GetVoltParam( MONITOR_VOLT_CPU, ULimit, LLimit )
```

Note:

Since the data received from this function is in mV units, the following conversion is needed for use in (Volt) units:

Data in Volt unit = Data in mV unit/1000

GetCurrentVolt

Call format Declare Function GetCurrentVolt Lib "PL_Ioc.dll"
(ByVal Selector As Long, ByRef Data As Long) As Long

Return value Other than 0: Normal
0: Error

Argument Selector As Long Parameters (pass by value)

MONITOR_VOLT_CPU	CPU Core voltage
MONITOR_VOLT_P33	+3.3V
MONITOR_VOLT_P50	+5.0V
MONITOR_VOLT_P12	+12V
MONITOR_VOLT_M12	-12V
MONITOR_VOLT_M50	-5.0V
MONITOR_VOLT_VIT	CPU Core voltage 2

Data As Long Voltage value (unit: mV) (pass by reference)

Processing Gets the current voltage value.

Example Dim ret As Long
Dim Data As Long
// Get the CPU core voltage value.
ret = GetCurrentVolt(MONITOR_VOLT_CPU, Data)

Note:

Since the data received from this function is in mV units, the following conversion is needed for use in (Volt) units:

Data in Volt unit = Data in mV unit/1000

GetFanParam

Call format Declare Function GetFanParam Lib "PL_Ioc.dll"
(ByVal Selector As Long, ByRef LLimit As Long) As Long

Return value Other than 0: Normal
0: Error

Argument Selector As Long Parameters (pass by reference)

MONITOR_FAN_CPU	CPU FAN
MONITOR_FAN_POWER	POWER FAN
MONITOR_FAN_OPT	OPTION FAN

LLimit As Long CPU FAN revolution lower limit value
(unit: RPM) (pass by value)
(RPM: revolutions per minute)

Processing Gets the parameter for monitoring the FAN.

Example Dim ret As Long
Dim LLimit As Long
// Get the CPU FAN lower limit rpm value
ret = GetFanParam(MONITOR_FAN_CPU, LLimit)

GetCurrentFan

Call format Declare Function GetCurrentFan Lib "PL_Ioc.dll"
(ByVal Selector As Long, ByRef Data As Long) As Long

Return value Other than 0: Normal
0: Error

Argument Selector As Long Parameters (pass by value)

MONITOR_FAN_CPU	CPU FAN
MONITOR_FAN_POWER	POWER FAN
MONITOR_FAN_OPT	OPTION FAN

Data As Long CPU FAN revolution lower limit value (unit: RPM) (pass by reference)
(RPM: revolutions per minute)

Processing Gets the current FAN rpm.

Example

```
Dim ret As Long
Dim Data As Long
// Get the number of revolutions of the CPU FAN
ret = GetCurrentFan( MONITOR_FAN_CPU, Data )
```

GetTempParam

Call format Declare Function GetTempParam Lib "PL_Ioc.dll"
(ByVal Selector As Long, ByRef ULimit As Long) As Long

Return value Other than 0: Normal
0: Error

Argument Selector As Long Get parameter (pass by value)

MONITOR_TEMP_SYSTEM	SYSTEM temp.
MONITOR_TEMP_CPU	CPU temp.
MONITOR_TEMP_OPT	OPTION temp.

Ulimit As Long Temperature upper limit (unit: °C) (pass by reference)

Processing Gets the parameter for monitoring the temperature.

Example

```
Dim ret As Long
Dim ULimit As Long
// Gets the upper limit of SYSTEM temperature
ret = GetTempParam( MONITOR_TEMP_SYSTEM, ULimit )
```

GetCurrentTemp

Call format	Declare Function GetCurrentTemp Lib "PL_Ioc.dll" (ByVal Selector As Long, ByRef Data As Long) As Long	
Return value	Other than 0: Normal 0: Error	
Argument	Selector As Long	Parameters (pass by value)
	MONITOR_TEMP_SYSTEM	SYSTEM temp.
	MONITOR_TEMP_CPU	CPU temp.
	MONITOR_TEMP_OPT	OPTION temp.
	Data As Long	Temperature value (unit: ^{o}C) (pass by reference)
Processing	Gets the current temperature value.	

Example	Dim ret As Long
	Dim Data As Long
	// Get the current value of SYSTEM temperature
	ret = GetCurrentTemp(MONITOR_TEMP_SYSTEM, Data)	

SetWdtCounter

Call format	Declare Function SetWdtCounter Lib "PL_Ioc.dll" (ByVal Counter As Long) As Long
Return value	Other than 0: Normal 0: Error
Argument	Counter As Long The initial counter value of the watchdog timer (pass by value) (5 to 255) (unit: second)
Processing	Sets the initial counter value for the watchdog timer.

Example	Dim ret As Long
	// Sets the initial counter value for the watchdog timer to 10 seconds
	ret = SetWdtCounter(10)

GetWdtCounter

Call format	Declare Function GetWdtCounter Lib "PL_Ioc.dll" (ByVal Counter As Long) As Long
Return value	Other than 0: Normal 0: Error
Argument	Counter As Long The initial counter value of the watchdog timer (pass by value) (unit: second)
Processing	Gets the initial counter value of the current watchdog timer.
Example	Dim ret As Long Dim Counter As Long ret = GetWdtCounter(Counter)

SetWdtMask

Call format	Declare Function SetWdtMask Lib "PL_Ioc.dll"		
	(ByVal Selector As Long, ByVal Mask As Long) As Long		
Return value	Other than 0: Normal		
	0: Error		
Argument	Selector As Long	Setup items (pass by value)	
		WARNING_LAMP	LAMP
		WARNING_ALARM	ALARM
	Mask As Long	Mask data (pass by value)	
		MASK_OFF	Release mask
		MASK_ON	Mask
Processing	Sets the warning mask to be output when a watchdog timer time-out occurs.		

Example

```
Dim ret As Long
// Mask the LAMP output
ret = SetWdtMask( WARNING_LAMP, MASK_ON )
// Release the mask for the ALARM output
ret = SetWdtMask( WARNING_ALARM, MASK_OFF )
```

GetWdtMask

Call format	Declare Function GetWdtMask Lib "PL_Ioc.dll"		
	(ByVal Selector As Long, ByRef Mask As Long) As Long		
Return value	Other than 0: Normal		
	0: Error		
Argument	Selector As Long	Setup items (pass by reference)	
		WARNING_LAMP	LAMP
		WARNING_ALARM	ALARM
	Mask As Long	(pass by reference)	
		MASK_OFF	Release the mask
		MASK_ON	Mask
Processing	Gets the WDT timeout warning output mask data.		

Example

```
Dim ret As Long
Dim Mask As Long
// Gets LAMP mask data
ret = GetWdtMask( WARNING_LAMP, Mask )
// Gets ALARM mask data
ret = GetWdtMask( WARNING_ALARM, Mask )
```

StartWdt

Call format	Declare Function StartWdt Lib "PL_Ioc.dll" () As Long
Return value	Other than 0: Normal 0: Error
Argument	None
Processing	Stops the WDT countdown.
Example	Dim ret As Long ret = StartWdt()

StopWdt

Call format	Declare Function StopWdt Lib "PL_Ioc.dll" () As Long
Return value	Other than 0: Normal 0: Error
Argument	None
Processing	Stops the WDT countdown.
Example	Dim ret As Long ret = StopWdt()

RestartWdt

Call format	Declare Function RestartWdt Lib "PL_Ioc.dll" () As Long
Return value	Other than 0: Normal 0: Error
Argument	None
Processing	Resets the initial value of the watchdog timer to the default value, and restarts the countdown.
Example	Dim ret As Long ret = RestartWdt()

Note: **Processing cannot be performed if the watchdog timer is stopped.**

RunningWdt

Call format	Declare Function RunningWdt Lib "PL_Ioc.dll" (ByRef RunFlag As Long) As Long
Return value	Other than 0: Normal 0: Error
Argument	RunFlag As Long Operating status of the watchdog timer (pass by reference) WATCHDOG_STOP Stopped WATCHDOG_COUNTDOWN Counting down
Processing	Gets the operating status of the watchdog timer.
Example	Dim ret As Long Dim RunFlag As Long ret = RunningWdt(RunFlag)

SetWarningOut

Call format	Declare Function SetWarningOut Lib "PL_Ioc.dll"		
	(ByVal Selector As Long, ByVal WarnOut As Long) As Long		
Return value	Other than 0: Normal		
	0: Error		
Argument	Selector As Long	Setting items (pass by value)	
		WARNING_LAMP	LAMP
		WARNING_ALARM	ALARM
	WarnOut As Long	Output condition (pass by value)	
		OUTPUT_OFF	Output OFF
		OUTPUT_ON	Output ON
Processing	Sets warning data for the setup items (LAMP and ALARM).		
Example	Dim ret As Long		
	// Set the output status of the LAMP to ON		
	ret = SetWarningOut(WARNING_LAMP, OUTPUT_ON)		
	// Set the output status of the ALARM to OFF		
	ret = SetWarningOut(WARNING_ALARM, OUTPUT_OFF)		

GetWarningOut

Call format	Declare Function GetWarningOut Lib "PL_Ioc.dll"		
	(ByVal Selector As Long, ByRef WarnOut As Long) As Long		
Return value	Other than 0: Normal		
	0: Error		
Argument	Selector As Long	Setting items (pass by value)	
		WARNING_LAMP	LAMP
		WARNING_ALARM	ALARM
	WarnOut As Long	Output condition (pass by value)	
		OUTPUT_OFF	Output OFF
		OUTPUT_ON	Output ON
Processing	Gets the current warning status of the setup items (LAMP and ALARM).		
Example	Dim ret As Long		
	Dim WarnOut As Long		
	// Gets the output status of the LAMP		
	ret = GetWarningOut(WARNING_LAMP, WarnOut)		
	// Get the output status of the ALARM		
	ret = GetWarningOut(WARNING_ALARM, WarnOut)		

GetUniversalIn

Call format	Declare Function GetUniversalIn Lib "PL_Ioc.dll" (ByVal Selector As Long, ByRef UniIn As Long) As Long		
Return value	Other than 0: Normal 0: Error		
Argument	Selector As Long	Designated port (pass by value)	
		PORT_UNI0	Universal Input 0
		PORT_UNI1	Universal Input 1
	UniIn As Long	Input status (pass by reference)	
		INPUT_OFF	No input
		INPUT_ON	Input
Processing	Gets the input status of the designated port (Universal Input 0 and Universal Input 1).		

Example	Dim ret As Long
	Dim UniIn As Long
	// Get the input status of the Universal Input 0
	ret = GetUniversalIn(PORT_UNI0, UniIn)
	// Get the input status of the Universal Input 1
	ret = GetUniversalIn(PORT_UNI1, UniIn)		

ClearUniversalIn

Call format	Declare Function ClearUniversalIn Lib "PL_Ioc.dll" (ByVal Selector As Long) As Long		
Return value	Other than 0: Normal 0: Error		
Argument	Selector As Long	Designated port (pass by value)	
		PORT_UNI0	Universal Input 0
		PORT_UNI1	Universal Input 1
Processing	Clears the input status of the designated port (Universal Input 0 and Universal Input 1).		

Example	Dim ret As Long
	// Clear the input status of Universal Input 0
	ret = ClearUniversalIn(PORT_UNI0)
	// Clear the input status of Universal Input 1
	ret = ClearUniversalIn(PORT_UNI1)		

SetUniversalInMask

Call format	Declare Function SetUniversalInMask Lib "PL_Ioc.dll" (ByVal Selector As Long, ByVal Mask As Long) As Long		
Return value	Other than 0: Normal 0: Error		
Argument	Selector As Long	Designated port (pass by value)	
		PORT_UNI0	Universal Input 0
		PORT_UNI1	Universal Input 1
	Mask As Long	Mask data (pass by value)	
		MASK_OFF	Clear mask
		MASK_ON	Mask
Processing	Sets the masking information of the designated ports (Universal Input 0 and Universal Input 1).		

Example

```
Dim ret As Long
//  Release the masking for Universal Input 0
ret = SetUniversalInMask( PORT_UNI0, MASK_OFF )
// Mask Universal Input 1
ret = SetUniversalInMask( PORT_UNI1, MASK_ON )
```

GetUniversalInMask

Call format	Declare Function GetUniversalInMask Lib "PL_Ioc.dll" (ByVal Selector As Long, ByRef Mask As Long) As Long		
Return value	Other than 0: Normal 0: Error		
Argument	Selector As Long	Designated port (pass by value)	
		PORT_UNI0	Universal Input 0
		PORT_UNI1	Universal Input 1
	Mask As Long	Mask data (pass by reference)	
		MASK_OFF	Release mask
		MASK_ON	Mask
Processing	Gets the masking information of the subject ports (Universal Input 0 and Universal Input 1).		

Example

```
Dim ret As Long
Dim Mask As Long
// Get the masking information for Universal Input 0
ret = GetUniversalInMask( PORT_UNI0, Mask )
// Get the masking information for Universal Input 1
ret = GetUniversalInMask( PORT_UNI1, Mask )
```

SetResetMask

Call format Declare Function SetResetMask Lib "PL_Ioc.dll"
(ByVal Mask As Long) As Long

Return value Other than 0: Normal
0: Error

Argument Mask As Long Mask data (pass by value)
MASK_OFF Release mask
MASK_ON Mask

Processing Sets the reset mask.

Example Dim ret As Long
// Releases the reset mask
ret = SetResetMask(MASK_OFF)

GetResetMask

Call format Declare Function GetResetMask Lib "PL_Ioc.dll"
(ByRef Mask As Long) As Long

Return value Other than 0: Normal
0: Error

Argument Mask As Long Mask data (pass by reference)
MASK_OFF Release mask
MASK_ON Mask

Processing Gets the current reset mask information.

Example Dim ret As Long
Dim Mask As Long
ret = GetResetMask(Mask)

SetIdeErr

Call format Declare Function SetIdeErr Lib "PL_Ioc.dll"
(ByVal IdeErr As Long) As Long

Return value Other than 0: Normal
0: Error

Argument IdeErr As Long Error output data (pass by reference)
IDE_ERROR_OFF Disables error output
IDE_ERROR_ON Enables error output

Processing Uses software control to set the IDE error output.

Example Dim ret As Long
// Set the system to disable the IDE error output
ret = SetIdeErr(IDE_ERROR_OFF)

GetIdeErrHard

Call format	Declare Function GetIdeErrHard Lib "PL_Ioc.dll" (ByVal Selector As Long, ByRef IdeErr As Long) As Long		
Return value	Other than 0: Normal 0: Error		
Argument	Selector As Long	Get parameter (pass by value)	
		IDE_ERROR_1	IDE_ERR1
		IDE_ERROR_2	IDE_ERR2
	IdeErr As Long	Error signal (pass by reference)	
		IDE_ERROR_OFF	Normal
		IDE_ERROR_ON	Error
Processing	Gets the current IDE error signal output by the hardware.		

Example	Dim ret As Long
	Dim IdeErr As Long
	// Gets the IDE ERR1 signal
	ret = GetIdeErrHard(IDE_ERROR_1, IdeErr)		

GetEvent

Call format Declare Function GetEvent Lib "PL_Ioc.dll"
(ByVal Selector As Long, ByRef Event As Long) As Long

Return value Other than 0: Normal
0: Error

Argument Selector As Long Parameters (pass by value)

EVENT_VOLT_CPU	CPU core voltage
EVENT_VOLT_P33	+3.3V
EVENT_VOLT_P50	+5.0V
EVENT_VOLT_P12	+12V
EVENT_VOLT_M12	-12V
EVENT_VOLT_M50	-5.0V
MONITOR_VOLT_VIT	CPU Core voltage 2
EVENT_FAN_CPU	CPU FAN
EVENT_FAN_POWER	POWER FAN
EVENT_FAN_OPT	OPTION FAN
EVENT_TEMP_SYSTEM	SYSTEM temp.
EVENT_TEMP_CPU_OPT	CPU or OPTION temp.
EVENT_UNI_IN0	Universal Input 0
EVENT_UNI_IN1	Universal Input 1
EVENT_WDT_TIMEOUT	Watchdog Timeout

Event As Long Error event data (pass by reference)

ERROR_EVENT_OFF	No error event
ERROR_EVENT_ON	Error event

Processing Checks for the irregularities in the machine voltage, FAN, and temperature, Universal Input function (event) data, and WatchDog Timeout data.

Example

```
Dim ret As Long
Dim Event As Long
// Gets the error event data of the CPU core voltage
ret = GetEvent( EVENT_VOLT_CPU, Event )
```

ClearEvent

Call format Declare Function ClearEvent Lib "PL_Ioc.dll"
(ByVal Selector As Long) As Long

Return value Other than 0: Normal
0: Error

Argument Selector As Long Designated error event cancel parameters (pass by value)

EVENT_VOLT_CPU	CPU core voltage
EVENT_VOLT_P33	+3.3V
EVENT_VOLT_P50	+5.0V
EVENT_VOLT_P12	+12V
EVENT_VOLT_M12	-12V
EVENT_VOLT_M50	-5.0V
MONITOR_VOLT_VIT	CPU Core voltage 2
EVENT_FAN_CPU	CPU FAN
EVENT_FAN_POWER	POWER FAN
EVENT_FAN_OPT	OPTION FAN
EVENT_TEMP_SYSTEM	System temp.
EVENT_TEMP_CPU_OPT	CPU or OPTION temp.
EVENT_UNI_IN0	Universal Input0
EVENT_UNI_IN1	Universal Input1
EVENT_WDT_TIMEOUT	Watchdog Timeout

Processing Cancels the error event.

Example

```
Dim ret As Long
// Cancels the error event.
ret = ClearEvent( EVENT_VOLT_CPU )
```

StartInsideBuzzer

Call format Declare Function StartInsideBuzzer Lib "PL_Ioc.dll"
(ByVal hz As Long, ByVal ms As Long) As Long

Return value Other than 0: Normal
0: Error

Argument

hz As Long	Buzzer frequency (pass by value)
ms As Long	Buzzer sound period (pass by value)

Processing Triggers the internal buzzer to sound at the specified frequency and for the specified period.

Example1

```
Dim ret As Long
Dim hz As Long
Dim ms As Long
// Sound the buzzer for 1 second at 600 Hz
hz = 600
ms = 1000
ret = StartInsideBuzzer( hz, ms )
```

Note: **This feature cannot be used with a PL running WindowsNT 4.0 or Windows 2000 due to the use of Windows 95 and Windows 98 functions.**

StopInsideBuzzer

Call format Declare Function StopInsideBuzzer Lib "PL_Ioc.dll" () As Long

Return value Other than 0: Normal
0: Error

Argument None

Processing Stops the internal buzzer.

Example

```
Dim ret As Long
// Stops the internal Buzzer.
ret = StopInsideBuzzer()
```

Note: **This feature cannot be used with a PL running WindowsNT 4.0 or Windows 2000 due to the use of Windows 95 and Windows 98 functions.**

ChkInsideBuzzer

Call format	Declare Function ChkInsideBuzzer Lib "PL_Ioc.dll" (ByRef buff As Long) As Long
Return value	Other than 0: Normal 0: Error
Argument	BuzzerParam As Long Buzzer status (pass by reference) BUZZER_ON Buzzer is ON BUZZER_OFF Buzzer is OFF
Processing	Checks for the ON/OFF status of the internal buzzer.

Example	Dim ret As Long
	Dim BuzzerParam As Long
	// Checks the buzzer status
	ret = ChkInsideBuzzer(BuzzerParam)

This feature cannot be used with a PL running WindowsNT 4.0 or Windows 2000 due to the use of Windows 95 and Windows 98 functions.

GetWdtTimeout

Call format	Declare Function GetWdtTimeout Lib "L_Ioc.dll" (ByRef Timebuf As Long) As Long
Return value	Other than 0: Normal 0: Error
Argument	Timebuf As Long WDT status (pass by reference)
Processing	Gets the watchdog timeout status.

Example	Dim ret As Long
	Dim Timebuf As Long
	// Gets the timeout status of the watchdog.
	ret = GetWdtTimeout(Timebuf)

ClearWdtTimeout

Call format	Declare Function ClearWdtTimeout Lib "PL_Ioc.dll () As Long
Return value	Other than 0: Normal 0: Error
Argument	None
Processing	Clears the timeout status of the watchdog.

Example	Dim ret As Long
	// Clear the timeout status of the watchdog.
	ret = ClearWdtTimeout()

SetWarningDOUT

Call format Declare Function SetWarningDOUT Lib "PL_Ioc.dll"
(ByVal WarningOut As Long) As Long

Return value Other than 0: Normal
0: Error

Argument WarningOut As Long Output status (pass by value)

OUTPUT_OFF	Output OFF
OUTPUT_ON	Output ON

Processing Sets the warning status of the current setup item (DOUT).

Example Dim ret As Long
// Set the output status of DOUT to OFF.
ret = SetWarningDOUT(OUTPUT_OFF)

GetWarningDOUT

Call format Declare Function GetWarningDOUT Lib "PL_Ioc.dll"
(ByRef WarningOut As Long) As Long

Return value Other than 0: Normal
0: Error

Argument WarningOut As Long Output status (pass by reference)

OUTPUT_OFF	Output OFF
OUTPUT_ON	Output ON

Processing Gets the warning status of the current setup item (DOUT).

Example1 Dim ret As Long
Dim WarningOut As Long
ret = GetWarningDOUT(WarningOut)

GetSmiDrvHandle

Call format Declare Function GetSmiDrvHandle Lib "PL_Ioc.dll" () As Long

Return value 0: Normal
1: Error

Argument None

Processing Gets the device driver handle to exchange information with the software mirroring device driver.

Example1 Dim ret As Long
ret = GetSmiDrvHandle()

Note: **An error will occur if the software mirroring device driver is not running.**

CloseSmiDrvHandle

Call format Declare Function CloseSmiDrvHandle Lib "PL_Ioc.dll" () As Long

Return value Other than 0: Normal

0: Error

Argument None

Processing Destroys the handle acquired with the "GetSmiDrvHandle" function.

Example

```
Dim ret As Long
// Destroys the handle.
ret = CloseSmiDrvHandle()
```

GetSmiAryStatus

Call format Declare Function GetSmiAryStatus Lib "PL_Ioc.dll"
(ByRef Status As Long) As Long

Return value Other than 0: Normal

0: Error

Argument Status As Long Software mirroring status (pass by reference)

ARYSTAT_GOOD	Normal
ARYSTAT_NOTEXIST	No output
ARYSTAT_UNCONFIG	Not configured
ARYSTAT_REBUILD	Being rebuilt
ARYSTAT_REDUCE	Being reduced
ARYSTAT_DEAD	Mirror status destroyed

Processing Gets the status of the software mirroring feature.

Example

```
Dim ret As Long
Dim Status As Long
// Get the status of the software mirroring feature.
ret = GetSmiAryStatus( Status )
```

GetSmiDevStatus

Call format: Declare Function GetSmiDevStatus Lib "PL_Ioc.dll"
(ByVal Id As Long, ByRef Type As Long, ByRef Status As Long) As Long

Return value: Other than 0: Normal
0: Error

Argument: Id As Long — Device ID (pass by value)
0 : Master HDD
1 : Slave HDD

Type As Long — Device type (pass by reference)

ATADEVICE	ATA type device
ATAPIDEVICE	CD-ROM
UNKNOWNDEVICE	Unknown device.
NODEVICE	No device

Status As Long — Device status (pass by reference)

DEVSTAT_GOOD	Normal
DEVSTAT_NOTEXIST	Not connected
DEVSTAT_BROKEN	Device failure

Processing: Gets the device status of the software mirroring feature.

Example:

```
Dim ret As Long
Dim Id As Long
Dim Type As Long
Dim Status As Long
// Gets the device status of the software mirroring feature.
Id = 0
ret = GetSmiDevStatus( Id, Type, Status )
```

SetWdtResetMask

Call format: Declare Function SetWdtResetMask Lib "PL_Ioc.dll"
(ByVal Mask As Long) As Long

Return value: Other than 0: Normal
0: Error

Argument: Mask As Long — Mask data (pass by value)

MASK_OFF	Release mask
MASK_ON	Mask

Processing: Sets the H/W reset mask for the WDT timeout.

Example:

```
Dim ret As Long
// Releases the reset masking used when WDT timeout occurs.
ret = SetWdtResetMask( MASK_OFF )
```

GetWdtResetMask

Call format	Declare Function GetWdtResetMask Lib "PL_Ioc.dll" (ByRef Mask As Long) As Long		
Return value	Other than 0: Normal 0: Error		
Argument	Mask As Long	Mask data (pass by reference)	
		MASK_OFF	Releases mask
		MASK_ON	Mask
Processing	Gets the WDT timeout's H/W reset mask data.		

Example	Dim ret As Long
	Dim Mask As Long
	// Get the H/W reset mask data for the WDT timeout.
	ret = GetWdtResetMask(Mask)		

A.5 Backlight Control API-DLL

A.5.1 Operation Environment

The following information explains the Dynamic Link Libraries used by the backlight control feature on a PL-X920 Series unit.

API-DLLs provide the interface for applications to access the System Monitor/ RAS feature (System Monitor/RAS Device Driver). Applications can use DLLs to access the following feature.

1. Backlight Control ON/OFF

■ Compatible Operating Systems

The API-DLLs contained on the PL unit's CD-ROM are compatible with the following OS types.

- Mircrosoft Windows®95
- Microsoft Windows®98
- Microsoft WindowsNT®4.0
- Microsoft Windows®2000

Each OS must use its corresponding Backlight Control Device.

■ Compatible Languages

- Microsoft Visual C
- Microsoft Visual C++
- Microsoft Visual Basic

◆ Required Files

The following files are required when using DLLs. Each language requires its own set of files.

- Visual C

File Name	Description
PL_BLIocif.h	Driver interface definition "include" file
PL_BLIoc.LIB	Library definition file
PL_BLIoc.dll	Dynamic link library file

- Visual C++

File Name	Description
PL_BLIocif.h	Driver interface definition "include" file
PL_BLIocall.h	CPL_BLIocall class definition "include" file
PL_BLIoctl.h	CPL_BLIoctl class definition "include" file
PL_BLIoc.LIB	Library definition file
PL_BLIoc.dll	Dynamic Link library file

* "#include header files should be "included" in the following order.

```
#include PL_BLIocif.h
#include PL_BLIoctl.h
```

PL_BLIocall.h is automatically included, and does not need to be directly designated.

- Visual Basic

File Name	Description
PL_BLIoc.bas	Driver interface definition file
PL_BLIoc.LIB	Library definition file
PL_BLIoc.dll	Dynamic link library file

■ Dynamic Link Library (DLL)

In order for an application to use PL_BLIoc.dll, it should be copied to the following folder.

OS	Location
Windows95/Windows98	C:\Windows\System
WindowsNT4.0/Windows2000	C:\Winnt\System32

A.5.2 Class Contents

■ CPL_BLIoctl Class

This class is used to set the parameters for device driver access using CPL_BLIoctl class.

Key Word	Type	Variable Name	Description
public	HANDLE	m_Drvhandle	Device driver handle

■ CPL_BLIocal Class

This uses the parameters set in CPL_BLIoctl, and calls up DeviceIoControl (Driver Access function).

However, since this class succeeds CPL_BLIoctl, it cannot be used directly.

Key Word	Type	Variable Name	Description
public	HANDLE	m_h	Device driver handle
public	LONG	m_long	Control code for action to perform
public	void *	m_ibp	Input data buffer address
public	ULONG	m_ibsize	Input data buffer size
public	void *	m_obp	Output data buffer address
public	ULONG	m_obsize	Output data buffer size
public	DWORD	m_retsize	Address for actual no. of output bytes
public	LPOVERLAPPED	m_ovlp	Address of overlap design

A.5.3 Visual C Functions

Function Name	Description
InitBLIoctl	Creates the CPL_BLIoctl object
EndBLIoctl	Destroys the CPL_BLIoctl object
GetBLDrvHandle	Gets the driver handle
GetBLDrvVersion	Gets the driver version
SetBLControl	Sets the backlight control values
GetBLControl	Gets the backlight control settings

A.5.4 Visual C Function Specifications (Details)

InitBLIoctl

Call Format void WINAPI InitBLIoctl(void)

Return Value None

Arguments None

Processing Creates a CPL_BLIoctl object. The object once created is not destroyed until the EndBLIoctl function is called.

Example InitBLIoctl();

EndBLIoctl

Call Format void WINAPI EndBLIoctl(void)

Return Value None

Arguments None

Processing Destroys the object created using the InitBLIoctl function.

Example EndBLIoctl();

GetBLDrvHandle

Call Format int WINAPI GetBLDrvHandle(HANDLE * pHndl)

Return Value 0: Normal
1: Error

Arguments (I/O) HANDLE *pHndl Pointer to the device driver handle

Processing Gets the device driver handle to communicate with the device driver.

Example
```
int       ret;
HANDLE    hndl;
ret = GetBLDrvHandle( &hndl );
```

Note: **An error will occur if the Backlight Control Device Driver is not running.**

GetBLDrvVersion

Call Format	BOOL WINAPI GetBLDrvVersion (int *pMajor, int *pMinor)
Return Value	TRUE: Normal FALSE: Error
Arguments	(I/O) int *pMajor Pointer to version information (Major, 0 to 99). (I/O)int *pMinor Pointer to version information (Minor, 0 to 99).
Processing	Gets the driver's version information.

Example	BOOL ret;
	int Major, Minor;
	ret = GetBLDrvVersion(&Major, &Minor);

If the version is 1.10, then you will get

Major: 1 (decimal)

Minor: 10 (decimal).

SetBLControl

Call Format	BOOL WINAPI SetBLControl (int BLFlag)
Return Value	TRUE: Normal FALSE: Error
Arguments	(I) int BLF flag Setting Parameters BACKLIGHT_OFF Backlight OFF BACKLIGHT_ON Backlight ON
Processing	Sets the backlight ON/OFF.

Example	BOOL ret;
	// Turns the backlight control ON.
	ret = SetBLControl(BACKLIGHT_ON);

GetBLControl

Call Format	BOOL WINAPI GetBLControl (int pBLFlag)
Return Value	TRUE: Normal FALSE: Error
Arguments	(I/O) int *pBLFlag Pointer to backlight condition BACKLIGHT_OFF Backlight OFF BACKLIGHT_ON Backlight ON
Processing	Gets the backlight control (settings) condition.

Example	BOOL ret;
	int BLFlag;
	// Gets the backlight control (settings) condition.
	ret = GetBLControl(&BLFlag);

A.5.5 Visual C++ Functions

Function Name	Description
GetBLDrvHandle	Gets the driver handle
GetBLDrvVersion	Gets the driver version
SetBLControl	Sets the backlight control values
GetBLControl	Gets the backlight control settings

A.5.6 Visual C++ Function Specifications (Details)

GetBLDrvHandle

Call Format int GetBLDrvHandle(void)

Return Value 0: Normal
1: Error

Arguments None

Processing Gets the device driver handle to communicate with the device driver. The handle obtained is stored in the member variable m_handle.

Example 1 CPL_BLIoctl m_BLIoc;
m_BLIoc.GetBLDrvHandle();

Example 2 int ret;
HANDLE hndl;
ret = ::GetBLDrvHandle(&hndl);

Note: **An error will occur if the Backlight Control Device Driver is not running.**

GetBLDrvVersion

Call Format	BOOL GetBLDrvVersion(int *pMajor, int *pMinor)
Return Value	TRUE: Normal FALSE: Error
Arguments	(I/O) int *pMajor Pointer to version information (Major, 0 to 99). (I/O)int *pMinor Pointer to version information (Minor, 0 to 99).
Processing	Gets the driver's version information.

Example 1	CPL_BLIoctl m_BLIoc;
	BOOL ret;
	int Major, Minor;
	ret = m_BLIoc.GetBLDrvVersion(&Major, &Minor);

Example 2	BOOL ret;
	int Major, Minor;
	ret = ::GetBLDrvVersion(&Major, &Minor);

Note:

If the version is 1.10, then you will get
Major: 1 (decimal)
Minor: 10 (decimal).

SetBLControl

Call Format	BOOL SetBLControl (int BLFlag)
Return Value	TRUE: Normal FALSE: Error
Arguments	(I) int BLFlag Setting Parameters BACKLIGHT_OFF Backlight OFF BACKLIGHT_ON Backlight ON
Processing	Sets the backlight ON/OFF.

Example 1	CPL_BLIoc m_BLIoc;
	BOOL ret;
	// Turns the backlight control ON.
	ret = m_BLIoc.SetBLControl(BACKLIGHT_ON)

Example 2	BOOL ret;
	// Turns the backlight control ON.
	ret = ::SetBLControl(BACKLIGHT_ON);

GetBLControl

Call Format	BOOL GetBLControl (int *pBLFlag)
Return Value	TRUE: Normal FALSE: Error
Arguments	(I/O) int *pBLFlag Pointer to backlight condition BACKLIGHT_OFF Backlight OFF BACKLIGHT_ON Backlight ON
Processing	Gets the backlight control (settings) status.

Example 1

```
CPL_BLIoc m_BLIoc;
BOOL      ret;
int       BLFlag;
// Gets the backlight control condition.
ret = m_BLIoc.GetBLControl( &BLFlag );
```

Example 2

```
BOOL      ret;
int       BLFlag;
// Turns the backlight control ON.
ret = ::GetBLControl( &BLFlag );
```

A.5.7 Visual Basic Functions

Function Name	Description
InitBLIoctl	Creates the CPL_Ioctl object
EndBLIoctl	Destroys the CPL_Ioctl object
GetBLDrvHandle	Gets the driver handle
GetBLDrvVersion	Gets the driver version
SetBLControl	Sets the backlight control values
GetBLControl	Gets the backlight control settings

A.5.8 Visual Basic Function Specifications (Details)

InitBLIoctl

Call Format Declare Sub InitBLIoctl Lib "PL_BLIoc.dll" ()

Return Value None

Arguments None

Processing Creates a CPL_BLIoctl object. The object once created is not destroyed until the EndBLIoctl function is called.

Example Call InitBLIoctl

EndBLIoctl

Call Format Declare Sub EndBLIoctl Lib "PL_BLIoc.dll" ()

Return Value None

Arguments None

Processing Destroys the object created using the InitBLIoctl function.

Example Call EndBLIoctl

GetBLDrvHandle

Call Format Declare Function GetBLDrvHandle Lib "PL_BLIoc.dll" (ByRef hndl As Long) As Long

Return Value 0: Normal
1: Error

Arguments hndl As Long Pointer to device driver handle (pass by reference)

Processing Gets the device driver handle to communicate with the device driver.

Example Dim ret As Long
Dim hndl As Long
ret = GetBLDrvHandle(hndl)

Note: **An error will occur if the Backlight Control Device Driver is not running.**

GetBLDrvVersion

Call Format   Declare Function GetBLDrvVersion Lib "PL_BLIoc.dll"
(ByRef Major As Long, ByRef Minor As Long) As Long

Return Value   Other than 0: Normal
0: Error

Arguments   Major As Long   Pointer to version information (Major, 0 to 99) (pass by reference)
Minor As Long   Pointer to version information (Minor, 0 to 99) (pass by reference)

Processing   Gets the driver's version information.

Example   Dim ret As Long
Dim Major As Long
Dim Minor As Long
ret = GetBLDrvVersion(Major, Minor)

Note:

If the version is 1.10, then you will get
Major: 1 (decimal)
Minor: 10 (decimal).

SetBLControl

Call Format   Declare Function SetBLControl Lib "PL_BLIoc.dll"
(ByVal BLFlag As Long) As Long

Return Value   Other than 0: Normal
0: Error

Arguments   BLFlag As Long   Setting Parameters (pass by value)
BACKLIGHT_OFF   Backlight OFF
BACKLIGHT_ON   Backlight ON

Processing   Sets the backlight ON/OFF.

Example   Dim ret As Long;
// Turns the backlight control ON.
ret = SetBLControl(BACKLIGHT_ON)

GetBLControl

Call Format	Declare Function GetBLControl Lib "PL_BLIoc.dll" (ByRef BLFlag As Long) As Long
Return Value	Other than 0: Normal 0: Error
Arguments	BLFlag As Long Pointer to backlight status (pass by reference) BACKLIGHT_OFF Backlight OFF BACKLIGHT_ON Backlight ON
Processing	Gets the backlight control (settings) status.

Example	Dim ret As Long
	Dim BLFlag As Long
	// Gets the backlight control (settings) condition.
	ret = GetBLControl(BLFlag);