NAR-7070
Communication Appliance

User's Manual

Revision: 020

Portwell Inc.
6851 Mowry Ave. Newark, CA 94560
TELr: 510-790-9192 FAX: 510-790-9191
http://www.portwell.com
Email: info@mail.portwell.com

Item NO: B8980700

Table of Contents

Chapter 1 Introduction

1.1 About This Manual

This manual describes all required information for setting up and using the NAR-7070.

NAR-7070 provides the essential components for delivering optimal performance and functionality in the high-end communications appliance market segment. This manual should familiarize you with NAR-7070 operations and functions. NAR-7070 has seven on-board Ethernet to serve communication appliances, such as Firewall, which needs seven LAN ports to connect external network (internet), demilitarized zone and internal network.

Feature of NAR-7070 includes:

- The most advanced Communication Appliance built on Intel® , latest Netburst™ micro architecture and Hyper-Threading technology
- High computing power of dual Intel® Xeon™ processors
- 64bit Gigabit Ethernet provides high performance networking capacity
- Intel® E7501A chipset with 533MHz PSB
- User-friendly LCD control panel
- Comprehensive thermal solution for 2U platform
- Full-length PCI-X slot support
- Standard PMC connecter support
- 2G PC2100 DDR RAM, upgradeable to 8GB
- Two IDE hard disk drives
- 100V ~ 240V , 8A ~ 4A ,Auto-range
- CE NO:C332606
- FCC NO:F332606

1.2 Manual Organization

The manual describes how to configure your NAR-7070 system to meet various operating requirements. It is divided into three chapters, with each chapter addressing a basic concept and operation of this whole system.

Chapter 1: Introduction. This section briefly talks about how this document is organized. It includes some guidelines for users who do not want to read through everything, but still helps you find what you need.

Chapter 2: Hardware Configuration Setting and Installation. This chapter shows how the hardware is put together, including detailed information. It shows the definitions and locations of Jumpers and Connectors that you can easily configure your system. Descriptions on how to properly mount the CPU and main memory are also included to help you get a safe installation. Reading this chapter will teach you how to set up NAR-7060.

Chapter 3: Operation Information. This section gives you illustrations and more information on the system architecture and how its performance can be maximized.

Chapter 4: BIOS operation information.

Any updates to this manual, technical clarification and answers to frequently asked questions would be posted on the web site: ***http://www.portwell.com***

1.3 Technical Support Information

Users may find helpful tips or related information on Portwell's web site: ***http://www.portwell.com***. A direct contact to Portwell's technical person is also available. For further support, users may also contact Portwell's headquarter in Taipei or your local distributors.

Chapter 2 Getting Started

This section describes how the hardware installation and system settings should be done.

2.1 Included Hardware

The following hardware is included in your kit:

- **PPAP-3723L Communication Appliance System Board**
- **One serial port Null MODEM cable**
- **One LCD Modules**

2.2 Before You Begin

To prevent damage to any system board, it is important to handle it with care. The following measures are generally sufficient to protect your equipment from static electricity discharge:

When handling the board, use a grounded wrist strap designed for static discharge elimination and touch a grounded metal object before removing the board from the antistatic bag. Handle the board by its edges only; do not touch its components, peripheral chips, memory modules or gold contacts.

When handling processor chips or memory modules, avoid touching their pins or gold edge fingers. Put the value communications appliance system board and peripherals back into the antistatic bag when they are not in use or not installed in the chassis.

Some circuitry on the system board can continue operating even though the power is switched off. Under no circumstances should the Lithium coin cell used to power the real-time clock be allowed to be shorted. The coin cell can heat under these conditions and present a burn hazard.

WARNING!

1. **"CAUTION: DANGER OF EXPLOSION IF BATTERY IS INCORRECTLY REPLACED. REPLACE ONLY WITH SAME OR EQUIVALENT TYPE RECOMMENDED BY THE MANUFACTURER. DISCARD USED BATTERIES ACCORDING TO THE MANUFACTURER'S INSTRUCTIONS"**

2. **This guide is for technically qualified personnel who have experience installing and configuring system boards Disconnect the system board power supply from its power source before you connect/disconnect cables or install/remove any system board components. Failure to do this can result in personnel injury or equipment damage.**
3. **Avoid short-circuiting the lithium battery; this can cause it to superheat and cause burns if touched.**
4. **Do not operate the processor without a thermal solution. Damage to the processor**

can occur in seconds.

5. **Do not block air vents. Minimum 1/2-inch clearance required.**
6. **Please switch off the power, before you install/remove any system components. It can avoid occurring any damages.**

2.3 The Chassis

The system is integrated in a customized 2U chassis (***Fig. 2-1, Fig. 2-2***). On the front panel you will find 4-push-button LCD module and seven Ethernet, a COM port and a POWER button and RESET button. The back panel has two USB ports and two system FAN.

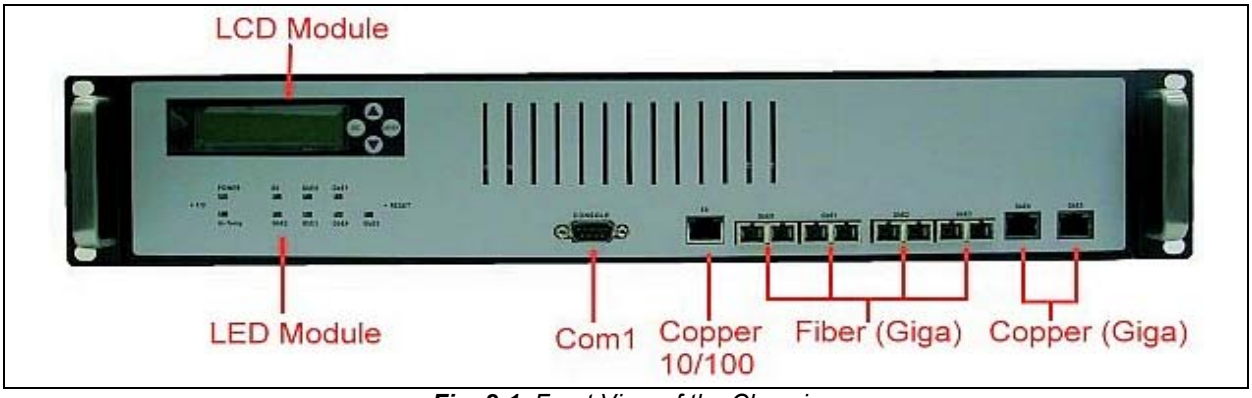


Fig. 2-1 *Front View of the Chassis*

Fig. 2-2 *Rear View of the Chassis*

2.4 Opening the Chassis

1. Screws out from cover (***Fig. 2-3***), slide the cover backwards and pull the rear edge upwards. (***Fig. 2-4***)

Fig. 2-3 Screws out from cover

Fig. 2-4 Slide the cover backwards and pull the rear edge upwards

2. The top cover (***Fig. 2-5***) can be removed from the base stand (***Fig. 2-6***)

Fig. 2-5 The top cover

Fig. 2-6 The base stand

2.5 Installing or Removing a SODIMM

Follow these steps to upgrade RAM module:

1. Install the system memory by push the latches on each side of the DIMM socket down. Align the memory module with the socket. Press memory module firmly down until it is sealed correctly. The socket latches are levered upwards and latch on to the edges of the DIMM. (***Fig. 2-7, 2-8***) (Slot 2 and 4 or slot 1 and 3 must be populated simultaneously)

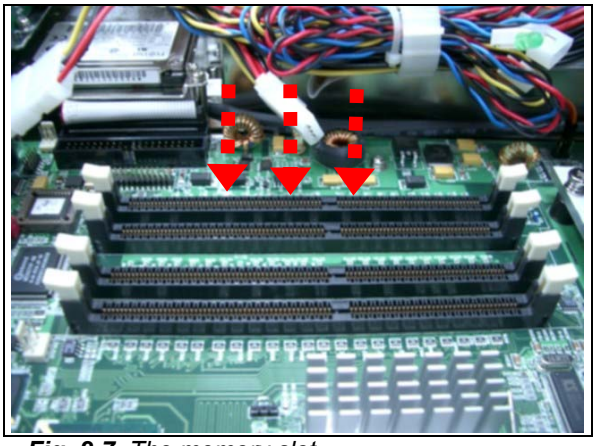


Fig. 2-7 The memory slot

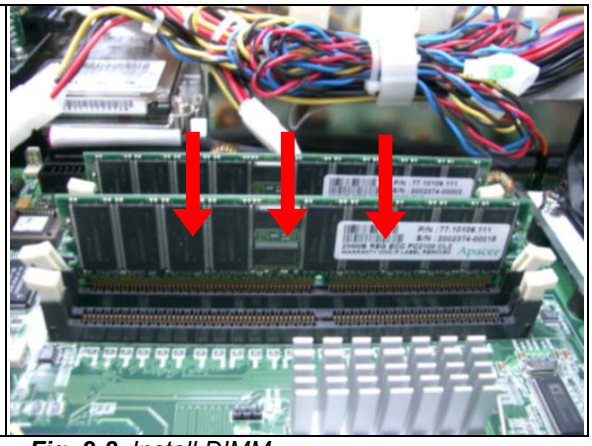


Fig. 2-8 Install DIMM

2. Push the latches on each side of the DIMM socket down to eject the DIMM (***Fig. 2-9***)

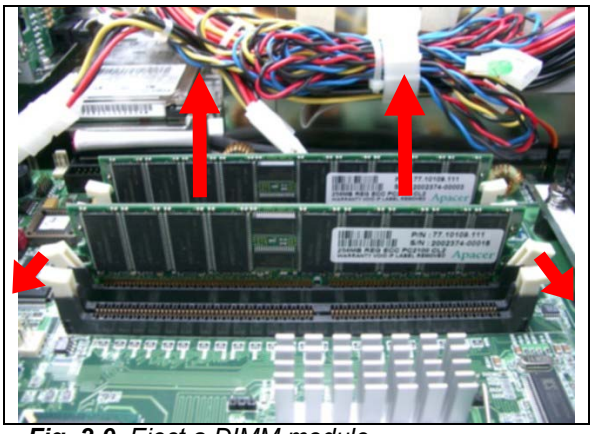


Fig. 2-9 *Eject a DIMM module*

2.6 Remove and Install CPU

3. Loosen and then take off the screws on the heat sink (***Fig. 2-10***).
4. Remove the heat sink and CPU space.(***Fig. 2-11***).

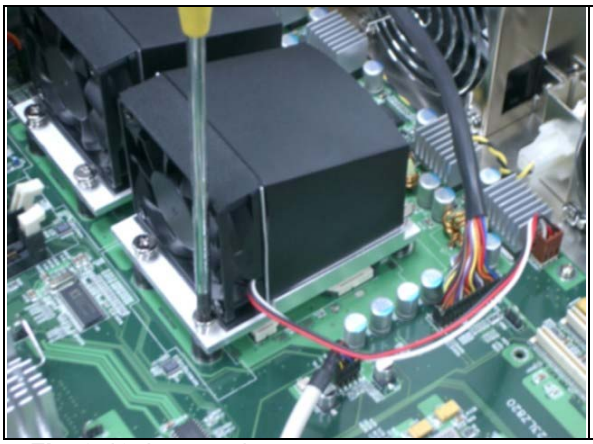


Fig. 2-10 *Loosen the screw*

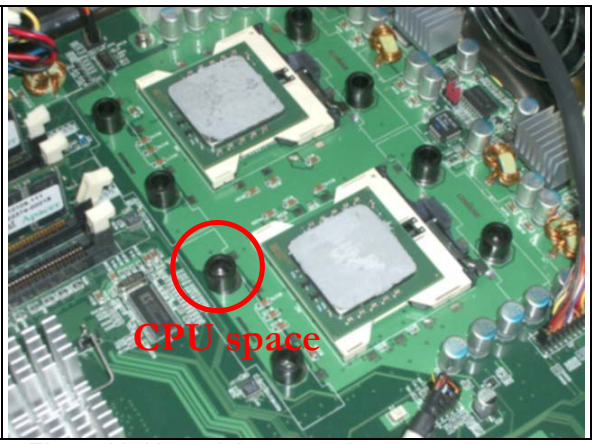


Fig. 2-11 *Heat sink removed*

5. Loosen the CPU socket (***Fig 2-12***)
6. Take the CPU chip out of the CPU socket (***Fig 2-13***)

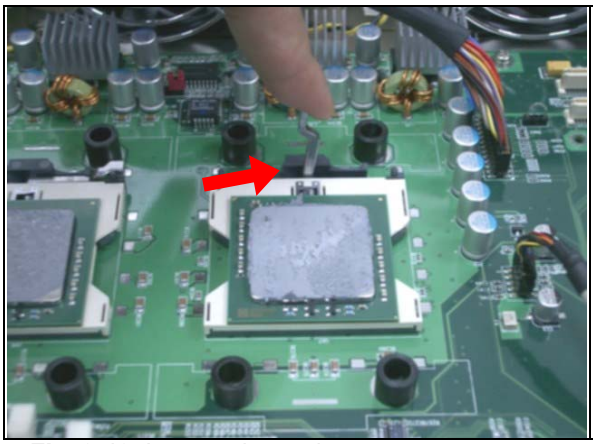


Fig. 2-12 *Loosen the socket*

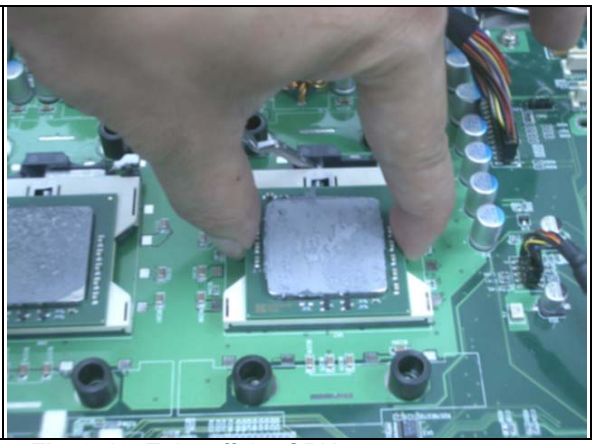


Fig. 2-13 *Take off the CPU*

7. Install CPU in opposite order as above

2.7 Remove and Install Battery

8. Press the metal clip down to eject the button battery (***Fig. 2-16***).
9. Replace a new battery by pressing it with fingertip to restore the battery (***Fig. 2-17***).

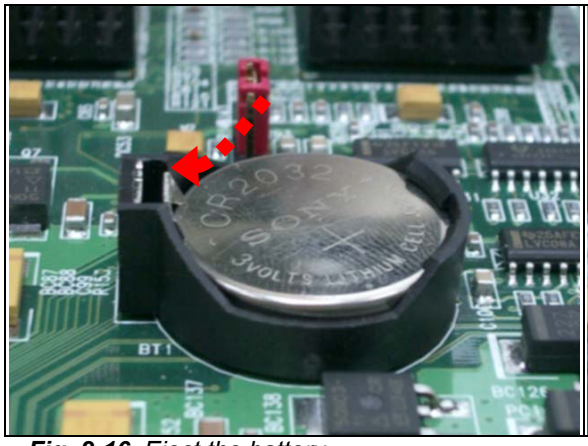


Fig. 2-16 *Eject the battery*

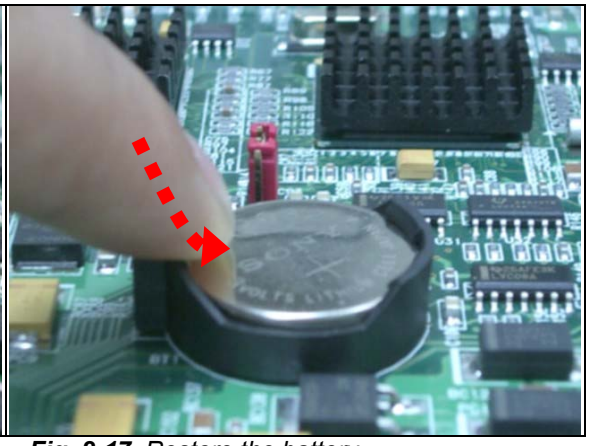


Fig. 2-17 *Restore the battery*

2.8 Remove and Install HDD

The system has an internal drive bay for one 2.5" hard disk drive. If the HDD did not pre-installed, follow the steps below:

Before a HDD can be installed onto NAR-7070.

10. Remove HDD bracket (***Fig. 2-18***) install HDD into the HDD bracket.

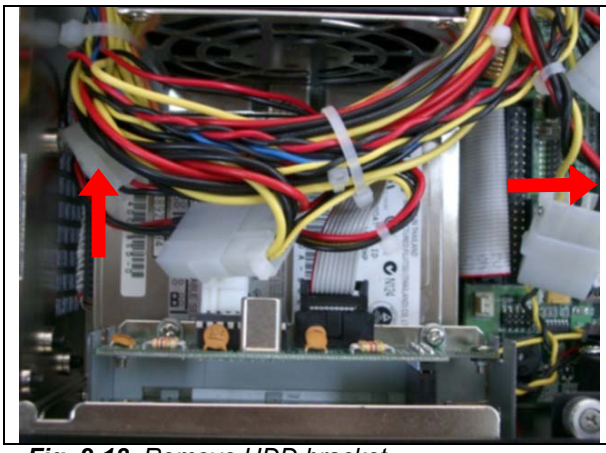


Fig. 2-18 *Remove HDD bracket*

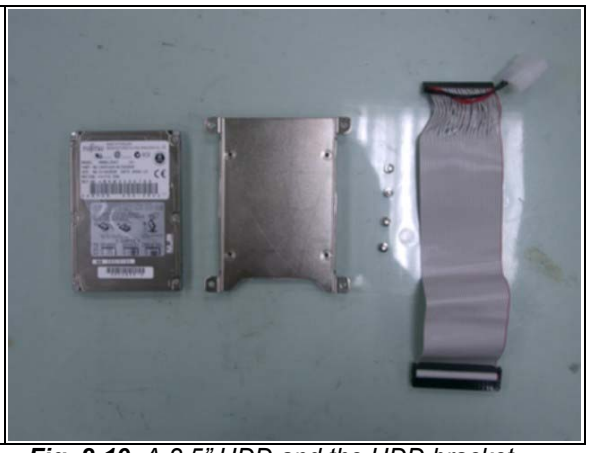


Fig. 2-19 *A 2.5" HDD and the HDD bracket*

11. Fasten the both screws to lock HDD and bracket together (***Fig. 2-20***).

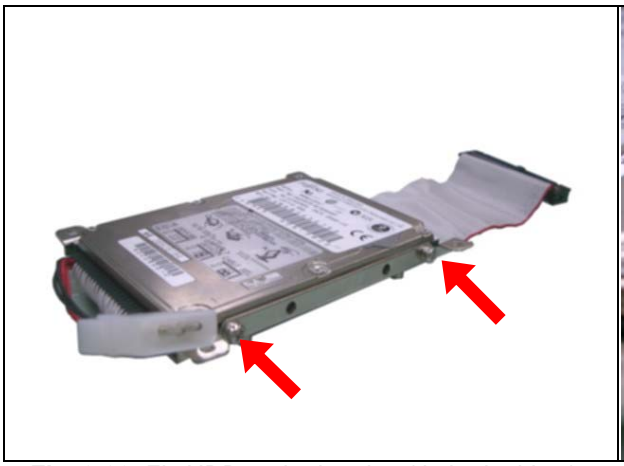


Fig. 2-20 Fix HDD to the bracket (in both sides)

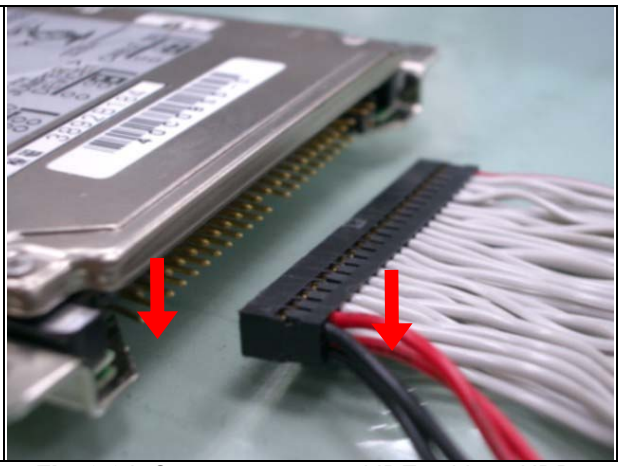


Fig. 2-21 Connect power and IDE cable to HDD

12. Connect the IDE cable and power connector to HDD (***Fig. 2-21***).
13. Fasten both screws back to lock HDD onto chassis (***Fig. 2-22***).

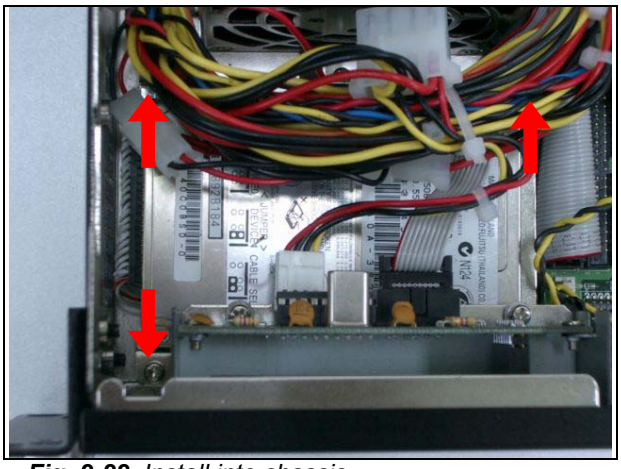


Fig. 2-22 Install into chassis

2.9 Remove and Install PCI-X Riser card

Two PCI-X slots are available in NAR-7070. Follow the steps below for installation:

14. The PCI-X Riser card is located on the left of the board. (***Fig. 2-23***)
15. To remove PCI-X Riser card, loosen and pull up the thumbscrews. (***Fig. 2-24, 2-25***)

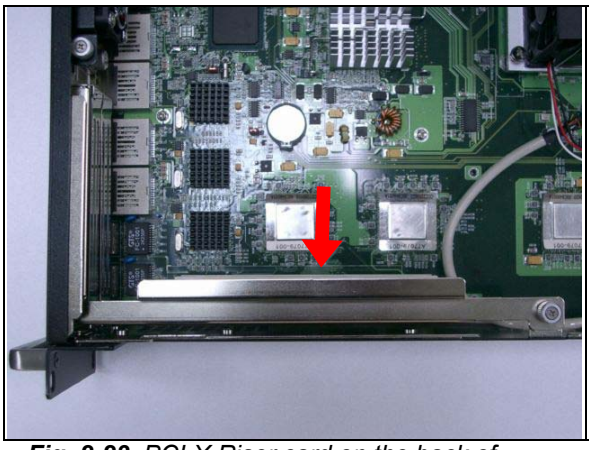


Fig. 2-23 PCI-X Riser card on the back of PPAP-3723L

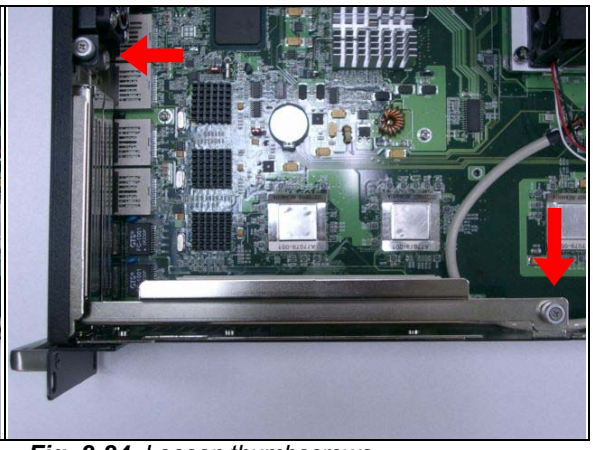


Fig. 2-24 Loosen thumbscrews.

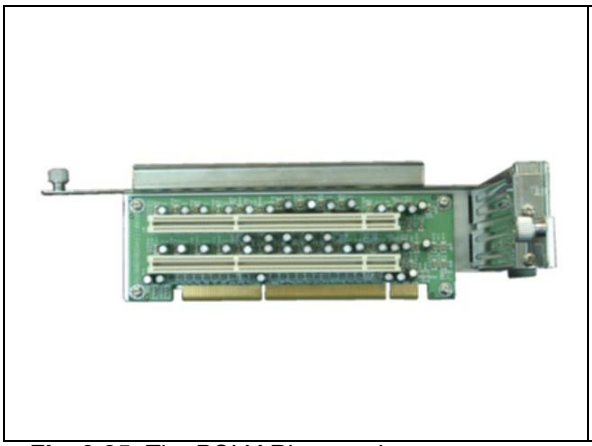


Fig. 2-25 The PCI-X Riser card

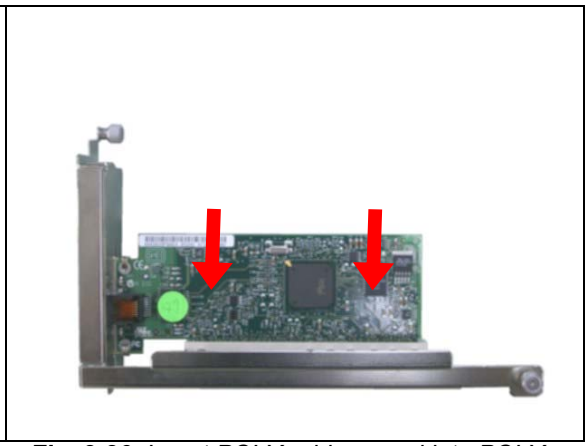


Fig. 2-26 Insert PCI-X add-on card into PCI-X Riser slot

16. Insert PCI-X card according to direction of arrow and tighten the thumbscrews. (***Fig. 2-26***)
17. Lock the PCI-X Riser card in position by a screw. (***Fig. 2-27***)

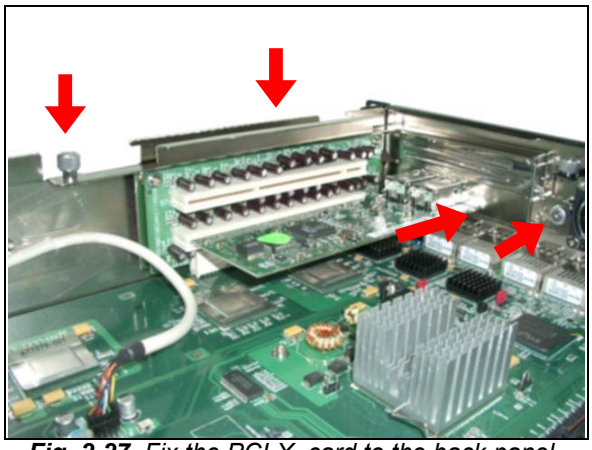


Fig. 2-27 Fix the PCI-X card to the back panel

2.10 Remove and Install LED cable & LED board

Follow the steps below to install or remove the LED cable and board:
1. LED board is under LCD module (Fig. 2-28),remove 4 screws and disconnect LED cable.
2. Disconnect LED cable from main board (Fig. 2-29)

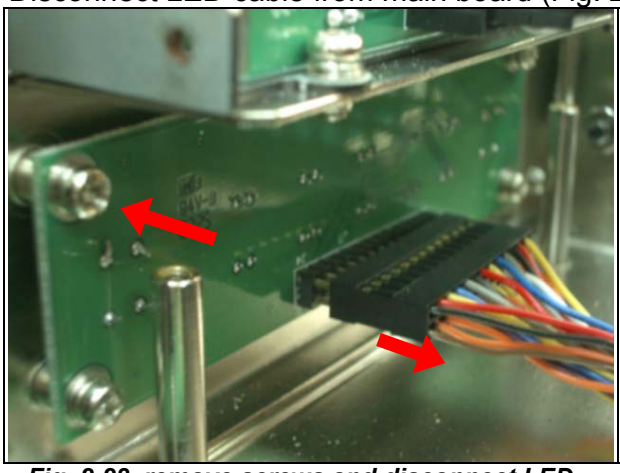


Fig. 2-28 remove screws and disconnect LED cable

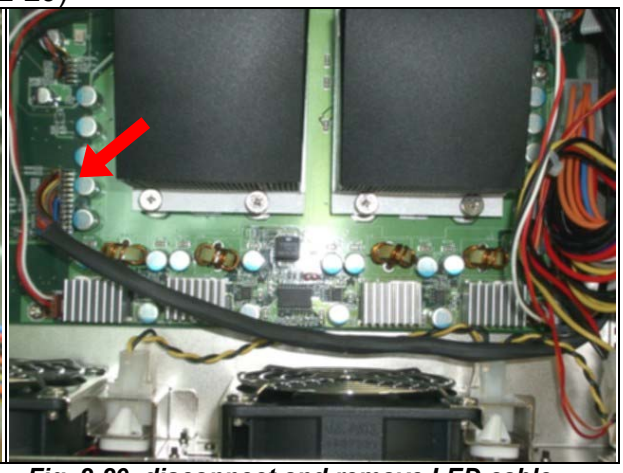


Fig. 2-29 disconnect and remove LED cable

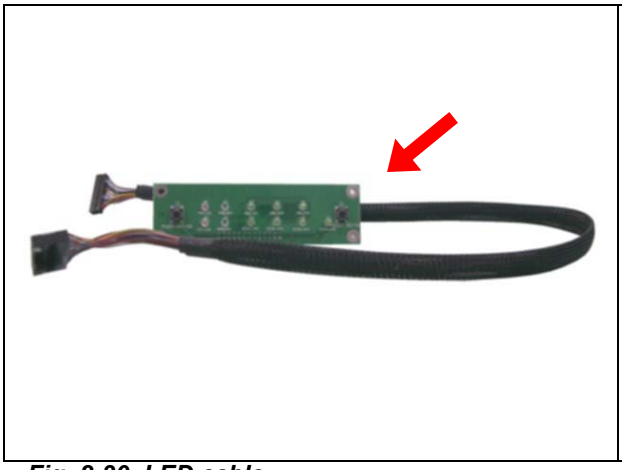


Fig. 2-30 LED cable

3. Reverse the steps to install LED board and cable.

2.10 Remove and Install System FAN

Follow the steps below to remove system FAN:
1. Remove two screws on the system fan module, disconnect the power plug.(Fig. 2-31,2-32)

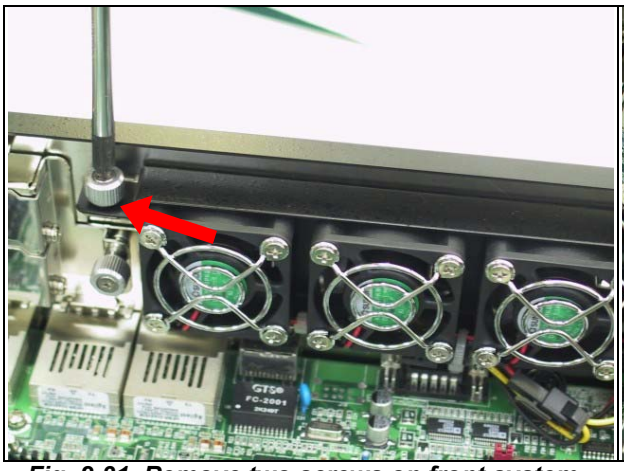


Fig. 2-31 Remove two screws on front system fan module

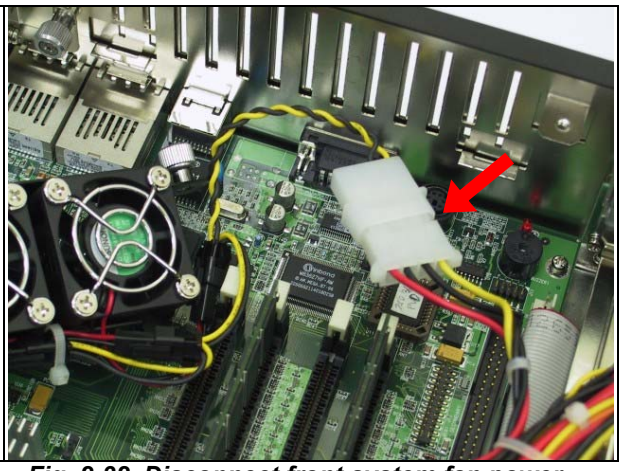


Fig. 2-32 Disconnect front system fan power

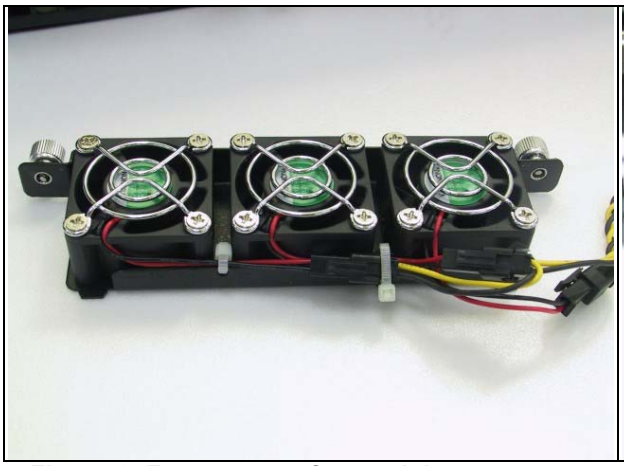


Fig. 2-33 Front system fan module

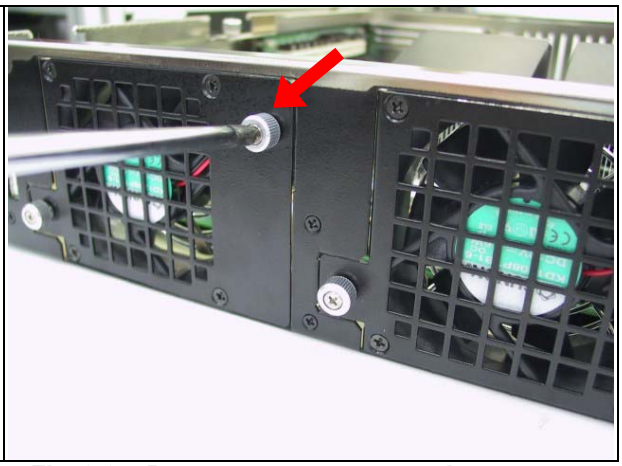


Fig. 2-34 Remove two screws to take out rear system fans

2. Remove two screws on rear system fan module to take out rear.(Fig. 2-33,2-34)

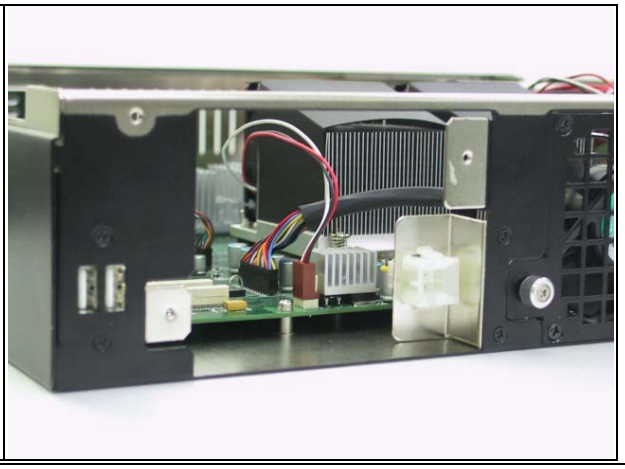


Fig. 2-35 Rear fan system module *Fig. 2-36 Rear fan system removed*

3. Reverse the above steps to install front and rear system fan modules.

2.10 Remove and Install LCD module & LCD cable

Follow the steps below to remove LCD module:
1. Remove two screws, LCD signal cable and power cord as shown. Push the LCD module toward inside the system. (Fig. 2-37,2-38,2-39)

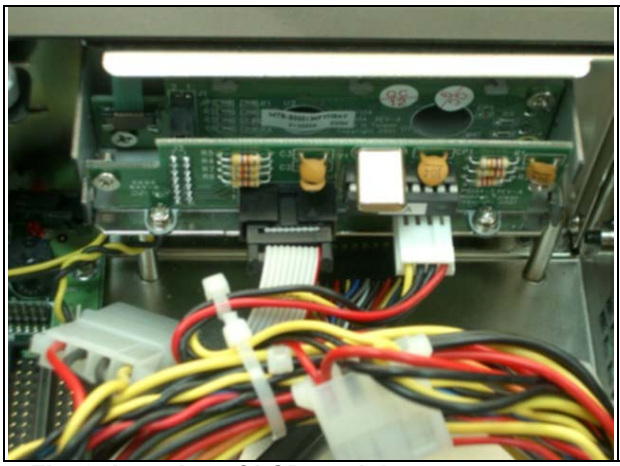


Fig. 2- Location of LCD module

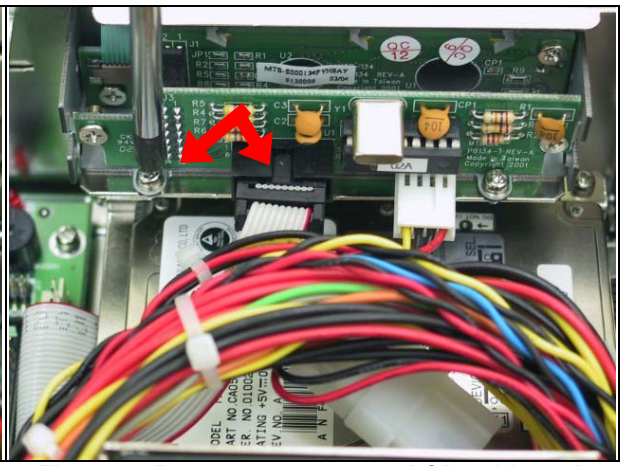


Fig. 2-38 Remove two screws, LCD cable and power cord as shown

Fig. 2-39 LCD module

2.10 Product Specifications

Model:	**NAR-7070**
Processor:	• Dual Intel® Xeon™ Processors (1.6 GHz – 2.8 GHz) with 512KB L2 Cache
Memory:	• 512MB PC1600 DDR RAM module, upgradeable to 8GB
BIOS:	• Award system BIOS with 512KB flash ROM
I/O Ports	• Six Gigabit Ethernet ports • One 10/100 BASE-T Ethernet ports • One RS-232 system console • Two USB ports
Storage Device	• One 2.5" HDD
Expansion	• Two 64bit PCI-X slot
LCD Panel	• 2X16 LCD module with Four-button keypad
LED	• LED indicators for power, HDD and Ethernet ports
Power	• 350W ATX PSU
Cooling	• Two 7cm FAN for CPU cooling • Two 8cm and three 4cm system fans
Operating Environment	• Temperature: 0℃ to 45℃ • Humidity: 5% to 95% RH
Dimension	• 431.0(W) x 408.0(D) x 88.0(H) mm • 17.00"(W) x 16"(D) x 3.46"(H)
Safety	• CE/FCC • LVDs

2.11 Hardware Configuration Setting

This section gives the definitions and shows the positions of jumpers, headers and connectors. All of the configuration jumpers on PPAP-3723 are in the proper position. The default settings set by factory are marked with a star (★).

Jumpers

In general, jumpers on PPAP-3723 system board are used to select options for certain features. Some of the jumpers are configurable for system enhancement. The others are for testing purpose only and should not be altered. To select any option, cover the jumper cap over (Short) or remove (NC) it from the jumper pins according to the following

instructions. Here N/C stands for “Not Connected”. (Please refer to ***Fig. 2-28*** for detailed jumper positions.)

Jumper Setting Table (JP1-JP9)

JP1	POWER ON CONTROL	Default Setting
1-2	POWER BUTTON POWER ON	
2-3	AUTO POWER ON	★

JP2	POWER SWITCH Pin Definition for LED Board Connector	Default Setting
1-2	POWER ON SWITCH	
2-3	Application Set Default SWITCH	★

JP3	CMOS Clear Jumper	Default Setting
1-2	Normal	★
2-3	Clear CMOS	

JP4	PMC Power Control	Default Setting
Short	3.3V PMC PCI	★
N/C	Non 3.3V PMC PCI	

JP5	PMC Power Control	Default Setting
Short	5V PCM PCI	
N/C	Non 5V PMC PCI	★

JP6	Manufacture Fix	Default Setting
1-2	Default	★
2-3	Non	

JP7	FAN Power	Default Setting
1-2	12V	★
2-3	5V	

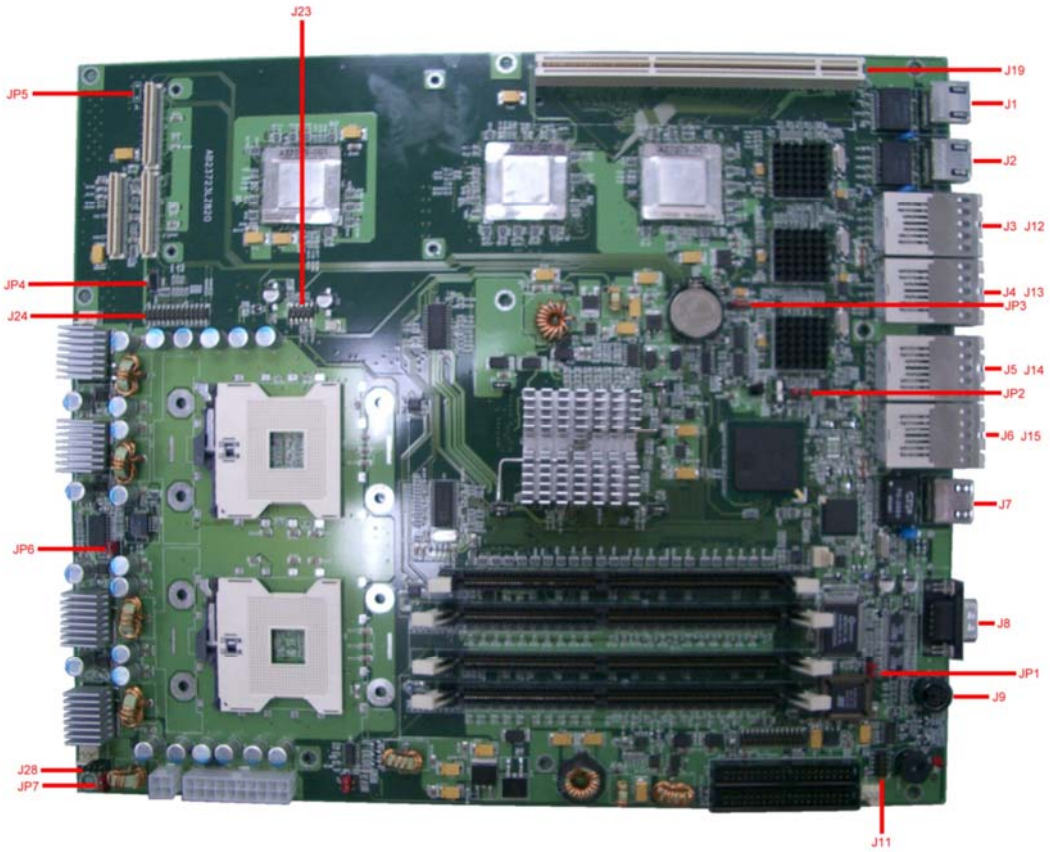


Fig. 2-28 Jumper Position

Connectors

Devices are connected through these connectors which includes IDE, COM Port etc…

Connector	Function	Remark
J1	Gigabit RJ45 Connector	
J2	Gigabit RJ45 Connector	
J3	Gigabit RJ45 Connector (Colay with J12)	
J4	Gigabit RJ45 Connector (Colay with J13)	
J5	Gigabit RJ45 Connector (Colay with J14)	
J6	Gigabit RJ45 Connector (Colay with J15)	
J7	10/100M bit RJ45 Connector	
J8	D Type COM1 Connector	
J9	PS/2 KB/MOUSE Connector	
J10	USB Stackup Connector	
J11	Reserve for Debugging	
J12	Gigabit Fiber Connector (Colay with J3)	
J13	Gigabit Fiber Connector (Colay with J4)	

J14	Gigabit Fiber Connector (Colay with J5)	
J15	Gigabit Fiber Connector (Colay with J6)	
J17	Hooker for MCH Heatsink	
J18	Hooker for MCH Heatsink	
J19	Vertical PCI-X Slot	
J20	Hooker for MCH Heatsink	
J21	Hooker for MCH Heatsink	
J22	Customized function header 1-2:Reserve 3-4:Intruder Detection Header(3-4 ON is the Default Setting) 5-6:Application Set Deffault Header	
J23	USB Header	
J24	LED Board Connector	
J25	64 bit PMC PCI Connector	
J26	64 bit PMC PCI Connector	
J27	64 bit PMC PCI Connector	
J28	COM 2 Header	

2.12 Install a Different Processor

Install CPU

1. Lift the handling lever of CPU socket outwards and upwards to the other end.
2. Align the processor pins with holes on the socket. Make sure that the notched corner or dot mark (pin 1) of the CPU corresponds to the socket's bevel end. Then press the CPU gently until it fits into place. If this operation is not easy or smooth, don't do it forcibly. You need to check and rebuild the CPU pin uniformly.
3. Push down the lever to lock processor chip into the socket.
4. Follow the installation guide of cooling fan or heat sink to mount it on CPU surface and lock it on the socket 604.
5. Be sure to follow particular CPU speed and voltage type to adjust the jumper settings properly for all boards.

Remove CPU

1. Unlock the cooling fan first.
2. Lift the lever of CPU socket outwards and upwards to the other end.
3. Carefully lift up the existing CPU to remove it from the socket.
4. Follow the steps of CPU installation to change to another one or place handling bar to close the opened socket.

Configure Processor Speed

Enter BIOS browser to select Frequency/Voltage Control, and then change CPU Clock Ratio to be 21X.

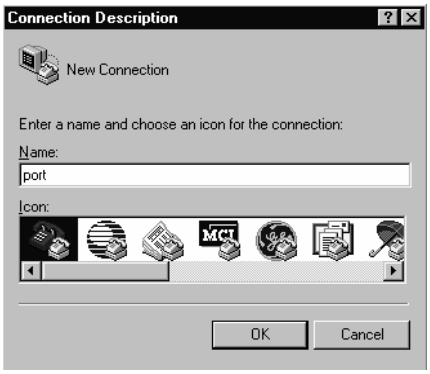
2.13 Connect to the console

Connection Using Hyper Terminal

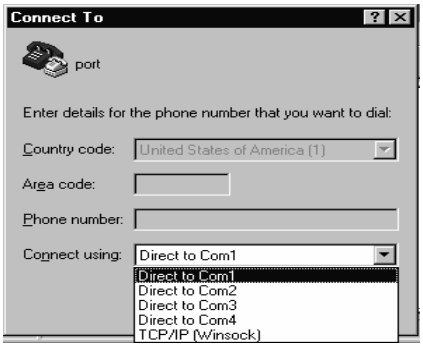
If users use a headless NAR-7070, which has no mouse/keyboard and VGA output connected to it, the console may be used to communicate with NAR-7070.

To access NAR-7070 via the console, Hyper Terminal is one of the choices. Follow the steps below for the setup:

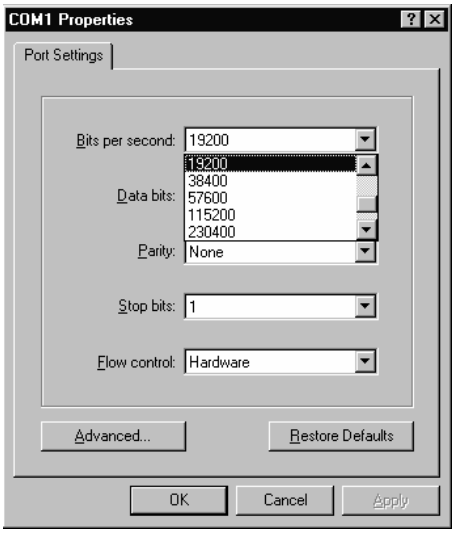
1. Execute HyperTerminal under C:\Program Files\Accessories\HyperTerminal
2. Enter a name to create new dial



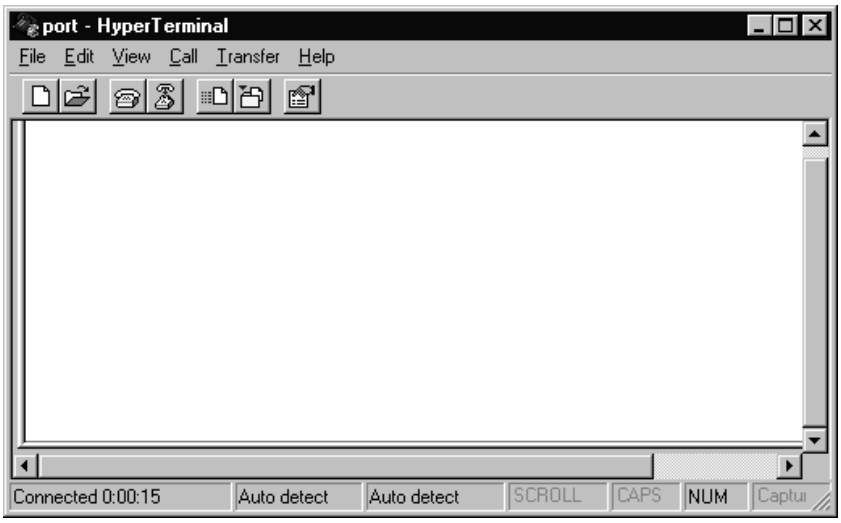
3. For the connection settings, make it Direct to Com1.



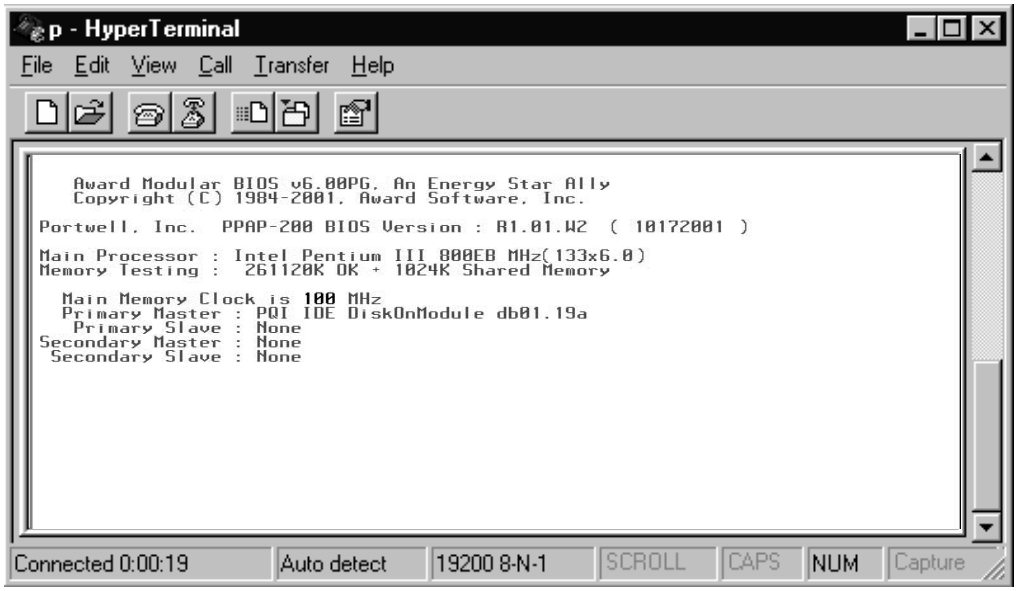
4. Please make the port settings to Baud rate 19200, Parity None, Data bits 8, Stop bits 1



5. Turn on the power of NAR-7060, after following screen was shown



6. You can then see the boot up information of NAR-7060



7. This is the end of this section. If the terminal did not port correctly, please check the previous steps.

Chapter 3 Operation Guide

3.1 Brief Guide of PPAP-3720

PPAP-3723 is a Communication Appliance computing board based on Intel® E7501 chipset technology. PPAP-3721 has Seven on-board Ethernet to serve communication appliances, such as Firewall, which needs three Ethernet ports to connect external network (internet), demilitarized zone and internal network. Different I/O management policies can be applied respectively to individual network to achieve the highest security level. Two built-in PCI-X slot permits further expansion for WAN connection, backup connection or even customized function card. The target market segment is communication appliance including Virtual Private Network, Load Balancing, Quality of Service, Intrusion Detection, Virus Detection, Firewall and Voice Over IP.

This PPAP-3723 system board is eligible with Intel® Xeon processors, and 184-pin DDR DIMM up to 8GB DDR RAM. The enhanced on-board PCI IDE interface supports 4 drives up to PIO mode 4 timing and Ultra DMA/100 synchronous mode feature. The on-board super I/O chipset integrates two serial ports driven by two high performance 16C550-compatible UARTs to provide 16-byte send/receive FIFOs. Besides, the two Universal Serial Bus ports provide high-speed data communication between peripherals and PC.

The on-board flash ROM is used to make the BIOS update easier. The high precision Real Time Clock/Calendar is built to support Y2K for accurate scheduling and storing configuration information. All of these features make PPAP-3723 excellent in stand-alone applications.

If any of these items is damaged or missing, please contact your vendor and save all packing materials for future replacement and maintenance.

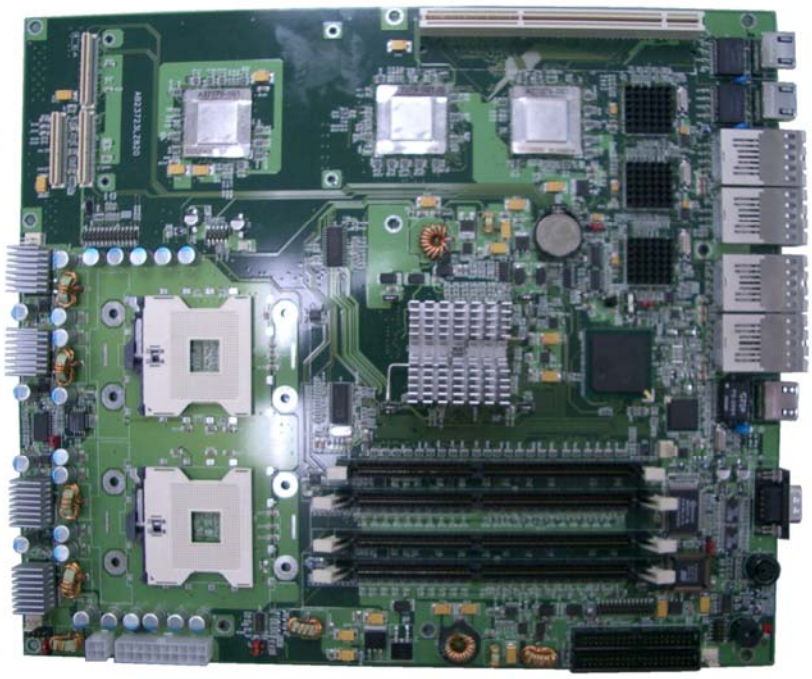


Fig. 3-1 PPAP-3723 Board

3.2 System Architecture

The following illustration of block diagram will show you how PPAP-3723 gives you a highly integrated system solution. The most up-to-date system architecture of PPAP-3723 includes two main VLSI chips. It contains E7501MCH and ICH3 to support Xeon processor, DDR DIMM, PCI bus interface, USB port, SMBus communication, and Ultra DMA/100 IDE Master. The on-board super I/O chip Winbond W83627HF supports two UARTs, FDC, parallel port and hardware monitoring.

PPAP-3723 has built-in Socket 603/604 to support Intel Xeon processor for cost-effective and high performance application. However.

The E7501 MCH provides a completely integrated solution for the system controller and data path components in a Xeon processor system. It provides optimized 64-bit DDR RAM interface.

The ICH3 provides a highly integrated multifunction for the best industry applications. It supports 2-channel dedicated Ultra ATA/33/66/100 IDE master interface, Universal Serial Bus **(USB)** controllers and one 64-bit PCI bus interface.

All detailed operating relations are shown in ***Fig. 3-2*** .(PPAP-3723 System Block Diagram)

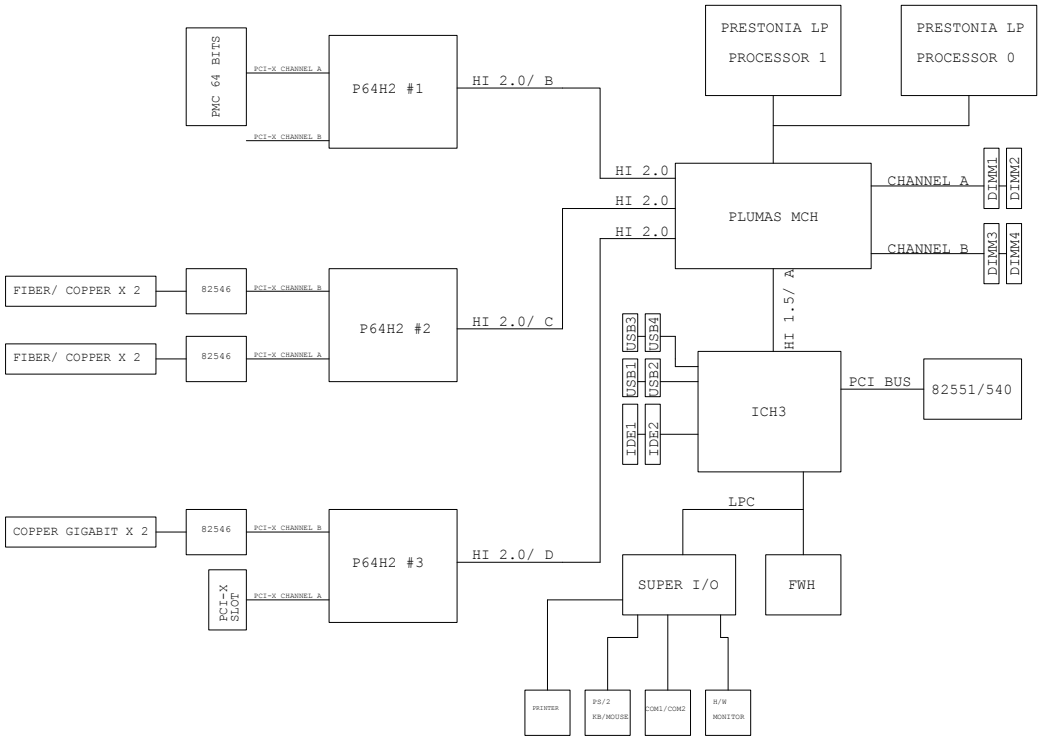


Fig. 3-2 PPAP-3723 E7501 Block Diagram

Chapter 4 BIOS Setup Information

4.1 Entering Setup

NAR-7060 is equipped with the AWARD BIOS stored in Flash ROM. This BIOS has a built-in Setup program that allows users to modify the basic system configuration easily. This type of information is stored in CMOS RAM so that it is retained during power-off periods. When system is turned on, NAR-7060 communicates with peripheral devices and check its hardware resources against the configuration information stored in the CMOS memory. If any error is detected, or the CMOS parameters need to be initialiged, the diagnostic program will prompt the user to enter the SETUP program. Some errors are significant enough to abort the start-up.

Turn on or reboot the computer. When the message “Hit <DEL> if you want to run SETUP” appears, press <Del> key immediately to enter BIOS setup program.

If the message disappears before you respond, but you still wish to enter Setup, please restart the system to try “COLD START” again by turning it OFF and then ON, or touch the "RESET" button. You may also restart from “WARM START” by pressing <Ctrl>, <Alt>, and <Delete> keys simultaneously. If you do not press the keys at the right time and the system will not boot, an error message will be displayed and you will again be asked to,

Press <F1> to Run SETUP or Resume

In BIOS setup, you can use the keyboard to choose among options or modify the system parameters to match the options with your system. The table below will show you all of keystroke functions in BIOS setup.

Keys to navigate within setup menu

Key	Functions
Up Arrow	Move to the previous item
Down Arrow	Move to the next item
Left Arrow	Move to the item on the left (menu bar)
Right Arrow	Move to the item on the right (menu bar)
Move Enter	Move to the item you desired
PgUp key	Increase the numeric value or make changes
PgDn key	Decrease the numeric value or make changes
+ key	Increase the numeric value or make changes
- key	Decrease the numeric value or make changes
Esc key	Main Menu -- Quit and not save changes into CMOS Status Page Setup Menu and Option Page Setup Menu -- Exit current page and return to Main Menu
F1 key	General help on Setup navigation keys
F5 key	Load previous values from CMOS
F6 key	Load the fail-safe defaults from BIOS default table
F7 key	Load the optimized defaults
F10 key	Save all the CMOS changes and exit

4.2 Main Menu

Once you enter NAR-7060 AWARD BIOS CMOS Setup Utility, you should start with the Main Menu. The Main Menu allows you to select from eleven setup functions and two exit choices. Use arrow keys to switch among items and press <Enter> key to accept or bring up the sub-menu.

Phoenix – Award WorkstationBIOS CMOS Setup Utility

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 Saving
Esc : Quit	↑ ↓ ← → : Select Item
F10 : Save & Exit Setup	(Shift)F2 : Change Color
Time, Date, Hard Disk Type….	

NOTE : It is strongly recommended to reload the Optimized Default Setting if CMOS is lost or BIOS is updated.

4.3 Standard CMOS Feature

This setup page includes all the items in a standard compatible BIOS. Use the arrow keys to highlight the item and then use the <PgUp>/<PgDn> or <+>/<-> keys to select the value or number you want in each item and press <Enter> key to certify it.

Follow command keys in CMOS Setup table to change **Date**, **Time**, **Drive type**, and **Boot Sector Virus Protection Status**.

■ **Screen shot**

Phoenix – Award WorkstationBIOS CMOS Setup Utility
Standard CMOS Features

```
Date(mm:dd:yy): Wed, Jan 17 2001                    | Item Help
Time(hh:mm:ss): 16:51:13                            |
                                                    | Menu Level
IDE Primary Master        [Seagate ST340011a]       |
IDE Primary Slave         [None]                    | Change the day,
IDE Secondary Master      [None]                    | month, year and
IDE Secondary Slave       [None]                    | century

Video                 [EGA/VGA]
Halt On               [All,But Keyboard]

Base Memory:                    640K
Extended Memory:                1047552K
Total Memory:                   1048576K

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

■ **Menu seletions**

Item	Options	Description
Date	Mm:dd:yy	Set the system date. Note that the 'Day' automatically changes when you set the date
Time	Hh:mm:ss	Set the system time
IDE Primary Master	-	Press [Enter] to enter Primary Master IDE configuration
IDE Primary Slave	-	
IDE Secondary Master	-	
IDE Secondary Slave	-	
Video	EGA/VGA CGA 40 CGA 80 MONO	Select the default video device
Halt On	All Errors No Errors All, but Keyboard All, but Diskette All, but Disk/Key	Select the situation in which you want the BIOS to stop the POST process and notify you

Item	Options	Description
Base Memory	N/A	Displays the amount of conventional memory detected during boot up
Extended Memory	N/A	Displays the amount of extended memory detected during boot up
Total Memory	N/A	Displays the total memory available in the system

- **IDE Primary Master Screen shot**

Phoenix – Award WorkstationBIOS CMOS Setup Utility
IDE Primary Master

IDE HDD Auto-Detection		[Press Enter]	Item Help
IDE Primary Master	[Auto]		Menu Level
Access Mode	[Auto]		
Capacity		0 MB	
Cylinder		0	
Head		0	
Precomp		0	
Landing Zone		0	
Sector		0	

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

- **Menu seletions**

Item	Options	Description
IDE HDD Auto-Detection	-	Auto-detection HDD type
IDE Primary Master	None Auto Manual	Select HDD dectection mode
Access Mode	CHS LBA Large Auto	Select HDD access mode
Capacity	-	Number of capacity
Cylinder	-	Number of cylinders
Head	-	Number of heads
Precomp	-	Write precomp
Landing Zone	-	Landing zone
Sector	-	Number of sector

4.4 Advanced BIOS Feature

This section allows you to configure your system for basic operation. You have the opportunity to select the system's default speed, boot-up sequence, keyboard operation, security.

- **Screen shot**

Phoenix – Award WorkstationBIOS CMOS Setup Utility
Advanced BIOS Features

Virus Warning	[Disabled]	Item Help
CPU L1 & L2 Cache	[Enabled]	
CPU Hyper-Threading	[Enabled]	
Quick Power On Self Test	[Enabled]	Menu Level
First Boot Device	[USB-FDD]	
Second Boot Device	[HDD-0]	
Boot Other Device	[Enabled]	
Boot Up NumLock Status	[On]	
Gate A20 Option	[Fast]	
Typematic Rate Setting	[Disabled]	
Typematic Rate (Chars/Sec)	6	
Typematic Delay (Msec)	250	
Security Option	[Setup]	
MPS Version Control For OS	[1.4]	
OS Select For DRAM > 64MB	[Non-OS2]	
Console Redirection	[Enabled]	
Baud Rate	**[19200]**	
Agent Connect via	**[NULL]**	
Agent wait time(min)	**[1]**	
Agent after boot	**[Disabled]**	
Report No FDD For WIN 95	**[No]**	

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

- **Menu seletions**

Item	Options	Description
Virus Warning	Enabled Disabled	Allows you to choose the VIRUS warning feature for IDE Hard Disk boot sector protection. If this function is enabled and someone attempt to write data into this area , BIOS will show a warning message on screen and alarm beep
CPU L1 & L2 Cache	Enabled Disabled	These two categories speed up memory access. However, it depends on CPU/chipset design.
CPU Hyper-Threading	Enabled Disabled	Enabled will allow one physical CPU emulate dual virtual processors.
Quick Power On Self Test	Enabled Disabled	Allows the system to skip certain tests while booting. This will speed up system boot.
First Boot Device	Floppy LS120 HDD-0 SCSI CDROM	Select Your First Boot Device Priority.

	HDD-1 HDD-2 HDD-3 ZIP100 USB-FDD USB-ZIP USB-CDROM USB-HDD LAN Disabled.	
Second Boot Device	Floppy LS120 HDD-0 SCSI CDROM HDD-1 HDD-2 HDD-3 ZIP100 USB-FDD USB-ZIP USB-CDROM USB-HDD LAN Disabled.	Select Your Second Boot Device Priority.
Boot Other Device	Enabled Disabled	Select Your Boot Device Priority.
Boot Up NumLock Status	On Off	Select power on state for NumLock.
Gate A20 Option	Normal Fast	This entry allows you to select how the gate A20 is handled. The gate A20 is a device used to address memory above 1 Mbytes. Initially, the gate A20 was handled via a pin on the keyboard. Today, while keyboards still provide this support, it is more common, and much faster, for the system chipset to provide support for gate A20.
Typematic Rate Setting	Enabled Disabled	Key strokes repeat at a rate determined by the keyboard controller. When enabled, the typematic rate and typematic delay can be selected.
Typematic Rate (Chars/Sec)	6 8 10 12 15 20 24 30	Sets the number of times a second to repeat a key stroke when you hold the key down.
Typematic Delay (Msec)	250 500 750 1000	Sets the delay time after the key is held down before it begins to repeat the keystroke.
Security Option	System Setup	Select whether the password is required every time the system boots or only when you enter setup.
MPS Version Control For OS	1.1 1.4	Multiprocessor spec revision the BIOS support.
OS Select For DRAM >	Non-OS2	Select the OS2 only if you are running OS/2 operating

64MB	OS2	system with greater than 64MB of RAM on the system.
Console Redirection	Enabled Disabled	Enabled – Attempt to redirect console via COM port. Disabled – Attempt to redirect console when keyboard absent.
Baud Rate	9600 19200 38400 57600 115200	Specify Baud Rate of console redirection
Agent Connect via	NULL	Connection modes: NULL – Direct connection agent wait time.
Agent wait time(min)	1 2 4 8	Timeout for connection
Agent after boot	Enabled Disabled	Keep Agent running after OS boot
Report No FDD For WIN 95	Yes No	

4.5 Advanced Chipset Features

This section allows you to configure the system based on the specific features of the Intel E7500 chipset. This chipset manages bus speeds and access to system memory resources, such as DDR RAM and the external cache. It must be stated that these items should never need to be altered. The default settings have been chosen because they provide the best operating conditions for your system. The only time you might consider making any changes would be if you discovered that data was being lost while using your system.

- **Screen shot**

Phoenix – Award WorkstationBIOS CMOS Setup Utility
Advanced Chipset Features

DRAM Timing Control	**[Press Enter]**	Item Help
System BIOS Cacheable	**[Enabled]**	
Video BIOS Cacheable	**[Disabled]**	Menu Level
Memory Hole At 15M-16M	**[Disabled]**	
Delayed Transaction	**[Enabled]**	

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

This chipset settings deal with CPU access to dynamic random access memory (DRAM). The default timings have been carefully chosen and should only be altered if data is being lost.The proper memory modules combination should follow user's manual.

- **Menu seletions**

Item	Options	Description
DRAM Timing Control	-	
System BIOS Cacheable	Enabled Disabled	Selecting Enabled allows caching of the system BIOS ROM at F0000h-FFFFFh, resulting in better system performance. However, if any program writes to this memory area, a system error may result.
Video BIOS Cacheable	Enabled Disabled	Enabled will speed up video BIOS cord access.
Memory Hole At 15M-16M	Enabled Disabled	In order to improve performance, certain space in memory is reserved for ISA cards. This memory must be mapped into the memory space below 16MB.
Delayed Transaction	Enabled Disabled	PCI bus option.

■ **Screen shot**

Phoenix – Award WorkstationBIOS CMOS Setup Utility
DRAM Timing Control

Memory Type	**Register , ECC**	Item Help
Memory Frequency For	**DDR200**	
DRAM Timing Configure	**[By SPD]**	Menu Level
CAS Latency Time	**2**	
Active to Precharge Delay	**5**	
DRAM RAS# to CAS# Delay	**2**	
DRAM RAS# Precharge	**2**	

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

■ **Menu seletions**

Item	Options	Description
Memory Type	-	
Memory Frequency For	-	
DRAM Timing Configure	Manual By SPD	Manualfactur don't recommend change default manu.
CAS Latency Time	1.5 2 2.5	Manualfactur don't recommend change default manu.
Active to Precharge Delay	7 6 5	Manualfactur don't recommend change default manu.
DRAM RAS# to CAS# Delay	3 2	Manualfactur don't recommend change default manu.
DRAM RAS# Precharge	3 2	Manualfactur don't recommend change default manu.

4.6 Integrated Peripherals

■ **Screen shot**

Phoenix – Award WorkstationBIOS CMOS Setup Utility
Integrated Peripherals

OnChip IDE Device	**[Press Enter]**	Item Help
OnBoard Device	**[Press Enter]**	
Onboard I/O Chip Setup	**[Press Enter]**	Menu Level

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

■ **Menu seletions**

Item	Options	Description
OnChip IDE Device	-	Press [Enter] to onchip IDE device configuration.
Onboard Device	-	Press [Enter] to onboard USB device configuration.
Onboard I/O Chip Setup	-	Press [Enter] to onboard I/O device configuration.

■ **Screen shot**

Phoenix – Award WorkstationBIOS CMOS Setup Utility
Integrated Peripherals

IDE HDD Block Mode	[Enabled]	Item Help
On-Chip Primary PCI IDE	[Enabled]	
IDE Primary Master PIO	[Auto]	Menu Level
IDE Primary Salve PIO	[Auto]	
IDE Primary Master UDMA	[Auto]	If your IDE hard drive supports block mode select Enabled for automatic detection of the optimal number of block read/writes per sector the drive can support
IDE Primary Slave UDMA	[Auto]	
On-Chip Secondary PCI IDE	[Enabled]	
IDE Secondary Master PIO	[Auto]	
IDE Secondary Slave PIO	[Auto]	
IDE Secondary Master UDMA	[Auto]	
IDE Secondary Slave UDMA	[Auto]	

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

■ **Menu seletions**

Item	Options	Description
IDE HDD Block Mode	Enabled Disabled	This item allows you to enable/disable IDE HDD Block Mode. The function is to collect the data that is nearby the one being read and leave them in the system buffer. Buffered data can be used with faster transmission rate so as to enhance system performance.
On-Chip Primary/Secondary PCI IDE	Enabled Disabled	The chipset contains a PCI IDE interface with support for two IDE channels. Select Enabled to activate the primary IDE interface. Select Disabled to deactivate this interface
IDE Primar/Secondary Master/Slave PIO	Auto Mode 0 Mode 1 Mode 2 Mode 3 Mode 4	The four IDE PIO (Programmed Input/Output) fields let you set a PIO mode (0-4) for each of the four IDE devices that the onboard IDE interface supports. Modes 0 through 4 provide successively increased performance. In Auto mode, the system automatically determines the best mode for each device.
IDE Primary/Secondary Master/Slave UDMA	Auto Disabled	For UDMA hard disk.

■ **Screen shot**

Phoenix – Award WorkstationBIOS CMOS Setup Utility
Onboard Device

USB Controller **USB Keyboard Support**	**[Enabled]** **[Disabled]**	Item Help Menu Level

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

■ **Menu seletions**

Item	Options	Description
USB Controller	Enabled Disabled	
USB Keyboard Support	Enabled Disabled	This item allows you to enable USB keyboard function under POST, BIOS setup menu, DOS, or Windows-NT with no USB driver loaded

■ **Screen shot**

Phoenix – Award WorkstationBIOS CMOS Setup Utility
Onboard I/O Chip Setup

Onboard Serial Port 1	**[3F8/IRQ4]**	Item Help
Onboard Serial Port 2	**[2F8/IRQ3]**	
PWRON After PWR-Fail	**[off]**	Menu Level

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

■ **Menu seletions**

Item	Options	Description
Onboard Serial Port 1/Port 2	3F8/IRQ4 2E8/IRQ3 3E8/IRQ4 2F8/IRQ3 Disabled Auto	Select an address and corresponding interrupt for the first and second serial ports.
PWRON After PWR-Fail	Off On Former-Sts	This option define the state while power resume after power lose. Off: the system will stay off affer power resume. On: the system will stay on affer power resume. Former-sts: the system will stay system former-sts affer power resume.

4.7 Power Management Setup

The Power Management Setup allows you to configure you system to most effectively save energy while operating in a manner consistent with your own style of computer use.

- **Screen shot**

Phoenix – Award WorkstationBIOS CMOS Setup Utility
Power Management Setup

ACPI Function	**[Enabled]**	Item Help
Power Management	**[User Define]**	
Video Off Method	**[DPMS]**	Menu Level
Video Off In Suspend	**[Yes]**	
Suspend Type	**[Stop Grant]**	
MODEM Use IRQ	**[3]**	
Suspend Mode	**[Disabled]**	
HDD Power Down	**[Disabled]**	
Soft-Off by PWR-BTTN	**[Instant-Off]**	
Power On by Ring	**[Enabled]**	
Resume by Alarm	**[Disabled]**	
Date(of Month) Alarm	**0**	
Time(hh:mm:ss) Alarm	**0 : 0 : 0**	
**** Reload Global Timer Events ****		
Primary IDE 0	**[Disabled]**	
Primary IDE 1	**[Disabled]**	
Secondary IDE 0	**[Disabled]**	
Secondary IDE 1	**[Disabled]**	
FDD,COM,LPT Port	**[Disabled]**	
PCI PIRQ[A-D]#	**[Disabled]**	

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

- **Menu seletions**

Item	Options	Description
ACPI Function	Enabled Disabled	
Power Management	User Define Min Saving Max Saving	This category allows you to select the type (or degree) of power saving and is directly related to "HDD Power Down", "Suspend Mode". There are three selections for Power Management, three of which have fixed mode settings.
Video Off Method	Blank Screen V/H SYNC+Blank DPMS	
Video Off In Suspend	Yes No	
Suspend Type	Stop Grant PwrOn Suspend	
MODEM Use IRQ	NA 3	

	4 5 7 9 10 11	
Suspend Mode	Disabled 1 Min 2 Min 4 Min 8 Min 12 Min 20 Min 30 Min 40 Min 1 Hour	When enabled and after the set time of system inactivity, all devices except the CPU will be shut off.
HDD Power Down	Disabled 1 Min – 15 Min	
Soft-Off by PWR-BTTN	Instant-Off Delay 4 Sec	This item allows users to set the time to remove the power after the power button is pressed.
PowerON by Ring	Enabled Disabled	When select "Enabled", a system that is at soft-off mode will be alert to Wake-On-Lan or Wake-On-Modem signal.
Resume by Alarm	Enabled Disabled	
Date(of Month) Alarm	0 - 31	
Time(hh:mm:ss) Alarm	Time	
Primary IDE 0/IDE 1	Enabled Disabled	
Secondary IDE 0/IDE 1	Enabled Disabled	
FDD,COM,LPT Port	Enabled Disabled	
PCI PIRQ[A-D]#	Enabled Disabled	

4.8 PnP/PCI Configuration Setup

This section describes configuring the PCI bus system. PCI, or Personal Computer Interconnect, is a system which allows I/O devices to operate at speeds nearing the speed the CPU itself uses when communicating with its own special components.

This section covers some very technical items and it is strongly recommended that only experienced users should make any changes to the default settings.

- **Screen shot**

Phoenix – Award WorkstationBIOS CMOS Setup Utility
PnP/PCI Configurations

Reset Configuration Data	**[Disabled]**	Item Help
Resources Controlled By	**[Auto(ESCD)]**	
IRQ Resources	**Press Enter**	Menu Level
PCI/VGA Palette Snoop	**[Disabled]**	Default is Disabled. Select Enabled to reset Extended System Configuration Data (ESCD) when you exit 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 : Modify F10 : Save ESC : Quit
F1:General Help
F5 : Previous Values F6 : Fail-Safe Defaults F7:Optimized Defaults

- **Menu seletions**

Item	Options	Description
Reset Configuration Data	Enabled Disabled	Normally, you leave this field Disabled. Select Enabled to reset Extended System Configuration Data (ESCD) when you exit Setup if you have installed a new add-on card and the system reconfiguration has caused such a serious conflict that the operating system can not boot.
Resource Controlled By	Auto(ESCD) Manual	The Award Plug and Play BIOS has the capacity to automatically configure all of the boot and Plug and Play compatible devices. However, this capability means absolutely nothing unless you are using a Plug and Play operating system such as Windows®95. If you set this field to "manual" choose specific resources by going into each of the sub menu that follows this field (a sub menu is preceded by a "➢").
IRQ Resources		
PCI/VGA Palette Snoop	Enabled Disabled	

■ **Screen shot**

Phoenix – Award WorkstationBIOS CMOS Setup Utility
IRQ Resources

IRQ-3 assigned to	**[PCI Device]**	Item Help
IRQ-4 assigned to	**[PCI Device]**	
IRQ-5 assigned to	**[PCI Device]**	Menu Level
IRQ-7 assigned to	**[PCI Device]**	
IRQ-9 assigned to	**[PCI Device]**	
IRQ-10 assigned to	**[PCI Device]**	
IRQ-11 assigned to	**[PCI Device]**	
IRQ-12 assigned to	**[PCI Device]**	
IRQ-14 assigned to	**[PCI Device]**	
IRQ-15 assigned to	**[PCI Device]**	

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

■ **Menu seletions**

Item	Options	Description
IRQ-3 –IRQ-15 assigned to	PCI Device Reserved	Legacy ISA for devices 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

4.9 PC Health Status

■ **Screen shot**

Phoenix – Award WorkstationBIOS CMOS Setup Utility
PC Health Status

CPU 1 TEMP	**37℃ / 98℉**	Item Help
CPU 2 TEMP	**37℃ / 98℉**	
IN0(V)	**1.44 V**	Menu Level
IN1(V)	**2.48 V**	
IN2(V)	**3.36 V**	
+ 5 V	**4 . 9 7 V**	
+12 V	**11 . 97 V**	
-12 V	**-12 . 11 V**	
- 5 V	**- 5 . 14 V**	
VBAT(V)	**3 . 4 0 V**	
5VSB	**5 . 0 4 V**	

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

4.10 Frequency/Voltage Control

■ **Screen shot**

Phoenix – Award WorkstationBIOS CMOS Setup Utility
Frequency/Voltage Control

CPU Clock Ratio **[20X]**	Item Help
	Menu Level

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

■ **Menu seletions**

Item	Options	Description
CPU Clock Ratio	16 – 20	

4.11 Default Menu

Selecting "Defaults" from the main menu shows you two options which are described below

Load Fail-Safe Defaults
When you press <Enter> on this item you get a confirmation dialog box with a message similar to:

Load Fail-Safe Defaults (Y/N) ? N

Pressing 'Y' loads the BIOS default values for the most stable, minimal-performance system operations.

Load Optimized Defaults
When you press <Enter> on this item you get a confirmation dialog box with a message similar to:

Load Optimized Defaults (Y/N) ? N

Pressing 'Y' loads the default values that are factory settings for optimal performance system operations.

4.12 Setup Supervisor Password

You can set either supervisor or user password, or both of then. The differences between are:

supervisor password : can enter and change the options of the setup menus.

ENTER PASSWORD

Type the password, up to eight characters in length, and press <Enter>. The password typed now will clear any previously entered password from CMOS memory. You will be asked to confirm the password. Type the password again and press <Enter>. You may also press <Esc> to abort the selection and not enter a password.

To disable a password, just press <Enter> when you are prompted to enter the password. A message will confirm the password will be disabled. Once the password is disabled, the system will boot and you can enter Setup freely.

PASSWORD DISABLED

When a password has been enabled, you will be prompted to enter it every time you try to enter Setup. This prevents an unauthorized person from changing any part of your system configuration.

Additionally, when a password is enabled, you can also require the BIOS to request a password every time your system is rebooted. This would prevent unauthorized use of your computer.

You determine when the password is required within the BIOS Features Setup Menu and its Security option (see Section 3). If the Security option is set to "System", the password will be required both at boot and at entry to Setup. If set to "Setup", prompting only occurs when trying to enter Setup.

4.13 Exiting Seleting

Save & Exit Setup

Pressing <Enter> on this item asks for confirmation:

Save to CMOS and EXIT (Y/N)? Y

Pressing "Y" stores the selections made in the menus in CMOS – a special section of memory that stays on after you turn your system off. The next time you boot your computer, the BIOS configures your system according to the Setup selections stored in CMOS. After saving the values the system is restarted again.

Exit Without Saving

Pressing <Enter> on this item asks for confirmation:

Quit without saving (Y/N)? Y

This allows you to exit Setup without storing in CMOS any change. The previous selections remain in effect. This exits the Setup utility and restarts your computer.

4.14 POST Messages

During the Power On Self-Test (POST), if the BIOS detects an error requiring you to do something to fix, it will either sound a beep code or display a message. If a message is displayed, it will be accompanied by:

PRESS F1 TO CONTINUE, CTRL-ALT-ESC OR DEL TO ENTER SETUP

POST Beep

Currently there are two kinds of beep codes in BIOS. This code indicates that a video error has occurred and the BIOS cannot initialize the video screen to display any additional information. This beep code consists of a single long beep followed by two short beeps. The other code indicates that your DRAM error has occurred. This beep code consists of a single long beep repeatedly.

Error Messages

One or more of the following messages may be displayed if the BIOS detects an error during the POST. This list includes messages for both the ISA and the EISA BIOS.

CMOS BATTERY HAS FAILED

CMOS battery is no longer functional. It should be replaced.

CMOS CHECKSUM ERROR

Checksum of CMOS is incorrect. This can indicate that CMOS has become corrupt. This error may have been caused by a weak battery. Check the battery and replace if necessary.

DISK BOOT FAILURE, INSERT SYSTEM DISK AND PRESS ENTER

No boot device was found. This could mean that either a boot drive was not detected or the drive does not contain proper system boot files. Insert a system disk into Drive A: and press <Enter>. If you assumed the system would boot from the hard drive, make sure the controller is inserted correctly and all cables are properly attached. Also be sure the disk is formatted as a boot device. Then reboot the system.

DISKETTE DRIVES OR TYPES MISMATCH ERROR - RUN SETUP

Type of diskette drive installed in the system is different from the CMOS definition. Run Setup to reconfigure the drive type correctly.

DISPLAY SWITCH IS SET INCORRECTLY

Display switch on the motherboard can be set to either monochrome or color. This indicates the switch is set to a different setting than indicated in Setup. Determine which setting is correct, and then either turn off the system and change the jumper, or enter Setup and change the VIDEO selection.

DISPLAY TYPE HAS CHANGED SINCE LAST BOOT

Since last powering off the system, the display adapter has been changed. You must configure the system for the new display type.

EISA Configuration Checksum Error PLEASE RUN EISA

CONFIGURATION UTILITY

The EISA non-volatile RAM checksum is incorrect or cannot correctly read the EISA slot. This can indicate either the EISA non-volatile memory has become corrupt or the slot has been configured incorrectly. Also be sure the card is installed firmly in the slot.

EISA Configuration Is Not Complete

PLEASE RUN EISA CONFIGURATION UTILITY

The slot configuration information stored in the EISA non-volatile memory is incomplete.

Note: When either of these errors appear, the system will boot in ISA mode, which allows you to run the EISA Configuration Utility.

ERROR ENCOUNTERED INITIALIZING HARD DRIVE

Hard drive cannot be initialized. Be sure the adapter is installed correctly and all cables are correctly and firmly attached. Also be sure the correct hard drive type is selected in Setup.

ERROR INITIALIZING HARD DISK CONTROLLER

Cannot initialize controller. Make sure the cord is correctly and firmly installed in the bus. Be sure the correct hard drive type is selected in Setup. Also check to see if any jumper needs to be set correctly on the hard drive.

FLOPPY DISK CNTRLR ERROR OR NO CNTRLR PRESENT

Cannot find or initialize the floppy drive controller. make sure the controller is installed correctly and firmly. If there are no floppy drives installed, be sure the Diskette Drive selection in Setup is set to NONE.

Invalid EISA Configuration

PLEASE RUN EISA CONFIGURATION UTILITY

The non-volatile memory containing EISA configuration information was programmed incorrectly or has become corrupt. Re-run EISA configuration utility to correctly program the memory.

NOTE: When this error appears, the system will boot in ISA mode, which allows you to run the EISA Configuration Utility.

KEYBOARD ERROR OR NO KEYBOARD PRESENT

Cannot initialize the keyboard. Make sure the keyboard is attached correctly and no keys are being pressed during the boot.

If you are purposely configuring the system without a keyboard, set the error halt condition in Setup to HALT ON ALL, BUT KEYBOARD. This will cause the BIOS to ignore the missing keyboard and continue the boot.

Memory Address Error at ...

Indicates a memory address error at a specific location. You can use this location along with the memory map for your system to find and replace the bad memory chips.

Memory parity Error at ...

Indicates a memory parity error at a specific location. You can use this location along with the memory map for your system to find and replace the bad memory chips.

MEMORY SIZE HAS CHANGED SINCE LAST BOOT

Memory has been added or removed since the last boot. In EISA mode use Configuration Utility to reconfigure the memory configuration. In ISA mode enter Setup and enter the new memory size in the memory fields.

Memory Verify Error at ...

Indicates an error verifying a value already written to memory. Use the location along with your system's memory map to locate the bad chip.

OFFENDING ADDRESS NOT FOUND

This message is used in conjunction with the I/O CHANNEL CHECK and RAM PARITY ERROR messages when the segment that has caused the problem cannot be isolated.

OFFENDING SEGMENT:

This message is used in conjunction with the I/O CHANNEL CHECK and RAM PARITY ERROR messages when the segment that has caused the problem has been isolated.

PRESS A KEY TO REBOOT

This will be displayed at the bottom screen when an error occurs that requires you to reboot. Press any key and the system will reboot.

PRESS F1 TO DISABLE NMI, F2 TO REBOOT

When BIOS detects a Non-maskable Interrupt condition during boot, this will allow you to disable the NMI and continue to boot, or you can reboot the system with the NMI enabled.

RAM PARITY ERROR - CHECKING FOR SEGMENT ...

Indicates a parity error in Random Access Memory.

Should Be Empty But EISA Board Found
PLEASE RUN EISA CONFIGURATION UTILITY

A valid board ID was found in a slot that was configured as having no board ID.

NOTE; When this error appears, the system will boot in ISA mode, which allows you to run the EISA Configuration Utility.

Should Have EISA Board But Not Found
PLEASE RUN EISA CONFIGURATION UTILITY

The board installed is not responding to the ID request, or no board ID has been found in the indicated slot.

NOTE: When this error appears, the system will boot in ISA mode, which allows you to run the EISA Configuration Utility.

Slot Not Empty

Indicates that a slot designated as empty by the EISA Configuration Utility actually contains a board.

NOTE: When this error appears, the system will boot in ISA mode, which allows you to run the EISA Configuration Utility.

SYSTEM HALTED, (CTRL-ALT-DEL) TO REBOOT ...

Indicates the present boot attempt has been aborted and the system must be rebooted. Press and hold down the CTRL and ALT keys and press DEL.

Wrong Board In Slot
PLEASE RUN EISA CONFIGURATION UTILITY

The board ID does not match the ID stored in the EISA non-volatile memory.

NOTE: When this error appears, the system will boot in ISA mode, which allows you to run the EISA Configuration Utility.

FLOPPY DISK(S) fail (80) → Unable to reset floppy subsystem.

FLOPPY DISK(S) fail (40) → Floppy Type dismatch.

Hard Disk(s) fail (80) → HDD reset failed

Hard Disk(s) fail (40) → HDD controller diagnostics failed.

Hard Disk(s) fail (20) → HDD initialization error.

Hard Disk(s) fail (10) → Unable to recalibrate fixed disk.

Hard Disk(s) fail (08) → Sector Verify failed.

Keyboard is locked out - Unlock the key.

Keyboard error or no keyboard present.

Cannot initialize the keyboard. Make sure the keyboard is attached correctly and no keys are being pressed during the boot.

BIOS ROM checksum error - System halted.

The checksum of ROM address F0000H-FFFFFH is bad.

Memory test fail.

BIOS reports the memory test fail if the onboard memory is tested error.

4.15 BIOS POST Check Point List

AWARDBIOS provides all IBM standard Power On Self Test (POST) routines as well as enhanced AWARDBIOS POST routines. The POST routines support CPU internal diagnostics. The POST checkpoint codes are accessible via the Manufacturing Test Port (I/O port 80h).

Whenever a recoverable error occurs during the POST, the system BIOS will display an error message describing the message and explaining the problem in detail so that the problem can be corrected.

During the POST, the BIOS signals a checkpoint by issuing one code to I/O address 80H. This code can be used to establish how far the BIOS has executed through the power-on sequence and what test is currently being performed. This is done to help troubleshoot faulty system board.

If the BIOS detects a terminal error condition, it will halt the POST process and attempt to display the checkpoint code written to port 80H. If the system hangs before the BIOS detects the terminal error, the value at port 80H will be the last

test performed. In this case, the terminal error cannot be displayed on the screen. The following POST checkpoint codes are valid for all AWARDBIOS products with a core BIOS date of 07/15/95 version 6.27 (Enhanced).

Code	Description
CFh	Test CMOS R/W functionality.
C0h	Early chipset initialization: -Disable shadow RAM -Disable L2 cache (socket 7 or below) -Program basic chipset registers
C1h	Detect memory -Auto-detection of DRAM size, type and ECC. -Auto-detection of L2 cache (socket 7 or below)
C3h	Expand compressed BIOS code to DRAM
C5h	Call chipset hook to copy BIOS back to E000 & F000 shadow RAM.
0h1	Expand the Xgroup codes locating in physical address 1000:0
02h	Reserved
03h	Initial Superio_Early_Init switch.
04h	Reserved
05h	1. Blank out screen 2. Clear CMOS error flag
06h	Reserved
07h	1. Clear 8042 interface 2. Initialize 8042 self-test

Code	Description
08h	1. Test special keyboard controller for Winbond 977 series Super I/O chips. 2. Enable keyboard interface.
09h	Reserved
0Ah	Disable PS/2 mouse interface (optional). Auto detect ports for keyboard & mouse followed by a port & interface swap (optional). Reset keyboard for Winbond 977 series Super I/O chips.
0Bh	Reserved
0Ch	Reserved
0Dh	Reserved
0Eh	Test F000h segment shadow to see whether it is R/W-able or not. If test fails, keep beeping the speaker.
0Fh	Reserved
10h	Auto detect flash type to load appropriate flash R/W codes into the run time area in F000 for ESCD & DMI support.
11h	Reserved
12h	Use walking 1's algorithm to check out interface in CMOS circuitry. Also set real-time clock power status, and then check for override.
13h	Reserved
14h	Program chipset default values into chipset. Chipset default values are MODBINable by OEM customers.
15h	Reeserved
16h	Initial Early_Init_Onboard_Generator switch.
17h	Reserved
18h	Detect CPU information including brand, SMI type (Cyrix or Intel) and CPU level (586 or 686).
19h	Reserved
1Ah	Reserved
1Bh	Initial interrupts vector table. If no special specified, all H/W interrupts are directed to SPURIOUS_INT_HDLR & S/W interrupts to SPURIOUS_soft_HDLR.
1Ch	Reserved
1Dh	Initial EARLY_PM_INIT switch.
1Eh	Reserved
1Fh	Load keyboard matrix (notebook platform)
20h	Reserved
21h	HPM initialization (notebook platform)
22h	Reserved
23h	1. Check validity of RTC value: e.g. a value of 5Ah is an invalid value for RTC minute. 2. Load CMOS settings into BIOS stack. If CMOS checksum fails, use default value instead. 3. Prepare BIOS resource map for PCI & PnP use. If ESCD is valid, take into consideration of the ESCD's legacy information. 4. Onboard clock generator initialization. Disable respective clock resource to empty PCI & DIMM slots. 5. Early PCI initialization: -Enumerate PCI bus number -Assign memory & I/O resource -Search for a valid VGA device & VGA BIOS, and put it into C000:0.

Code	Description
24h	Reserved
25h	Reserved
26h	Reserved
27h	Initialize INT 09 buffer
28h	Reserved
29h	Program CPU internal MTRR (P6 & PII) for 0-640K memory address. Initialize the APIC for Pentium class CPU. Program early chipset according to CMOS setup. Example: onboard IDE controller. Measure CPU speed. Invoke video BIOS.
2Ah	Reserved
2Bh	Reserved
2Ch	Reserved
2Dh	Initialize multi-language. Put information on screen display, including Award title, CPU type, CPU speed ….
2Eh	Reserved
2Fh	Reserved
30h	Reserved
31h	Reserved
32h	Reserved
33h	Reset keyboard except Winbond 977 series Super I/O chips.
34h	Reserved
35h	Reserved
36h	Reserved
37h	Reserved
38h	Reserved
39h	Reserved
3Ah	Reserved
3Bh	Reserved
3Ch	Test 8254
3Dh	Reserved
3Eh	Test 8259 interrupt mask bits for channel 1.
3Fh	Reserved
40h	Test 8259 interrupt mask bits for channel 2.
41h	Reserved
42h	Reserved
43h	Test 8259 functionality.
44h	Reserved
45h	Reserved
46h	Reserved
47h	Initialize EISA slot
48h	Reserved
49h	1. Calculate total memory by testing the last double word of each 64K page. 2. Program writes allocation for AMD K5 CPU.
4Ah	Reserved
4Bh	Reserved
4Ch	Reserved
4Dh	Reserved
4Eh	1. Program MTRR of M1 CPU 2. Initialize L2 cache for P6 class CPU & program CPU with proper cacheable range.

Code	Description
	3. Initialize the APIC for P6 class CPU. 4. On MP platform, adjust the cacheable range to smaller one in case the cacheable ranges between each CPU are not identical.
4Fh	Reserved
50h	Initialize USB
51h	Reserved
52h	Test all memory (clear all extended memory to 0)
53h	Reserved
54h	Reserved
55h	Display number of processors (multi-processor platform)
56h	Reserved
57h	1. Display PnP logo 2. Early ISA PnP initialization -Assign CSN to every ISA PnP device.
58h	Reserved
59h	Initialize the combined Trend Anti-Virus code.
5Ah	Reserved
5Bh	(Optional Feature) Show message for entering AWDFLASH.EXE from FDD (optional)
5Ch	Reserved
5Dh	1. Initialize Init_Onboard_Super_IO switch. 2. Initialize Init_Onbaord_AUDIO switch.
5Eh	Reserved
5Fh	Reserved
60h	Okay to enter Setup utility; i.e. not until this POST stage can users enter the CMOS setup utility.
61h	Reserved
62h	Reserved
63h	Reserved
64h	Reserved
65h	Initialize PS/2 Mouse
66h	Reserved
67h	Prepare memory size information for function call: INT 15h ax=E820h
68h	Reserved
69h	Turn on L2 cache
6Ah	Reserved
6Bh	Program chipset registers according to items described in Setup & Auto-configuration table.
6Ch	Reserved
6Dh	1. Assign resources to all ISA PnP devices. 2. Auto assign ports to onboard COM ports if the corresponding item in Setup is set to "AUTO".
6Eh	Reserved
6Fh	1. Initialize floppy controller 2. Set up floppy related fields in 40:hardware.
70h	Reserved
71h	Reserved
72h	Reserved

Code	Description
73h	(Optional Feature) Enter AWDFLASH.EXE if : -AWDFLASH is found in floppy drive. -ALT+F2 is pressed
74h	Reserved
75h	Detect & install all IDE devices: HDD, LS120, ZIP, CDROM…..
76h	Reserved
77h	Detect serial ports & parallel ports.
78h	Reserved
79h	Reserved
7Ah	Detect & install co-processor
7Bh	Reserved
7Ch	Reserved
7Dh	Reserved
7Eh	Reserved
7Fh	1. Switch back to text mode if full screen logo is supported. -If errors occur, report errors & wait for keys -If no errors occur or F1 key is pressed to continue: ◆Clear EPA or customization logo.
80h	Reserved
81h	Reserved
82h	1. Call chipset power management hook. 2. Recover the text fond used by EPA logo (not for full screen logo) 3. If password is set, ask for password.
83h	Save all data in stack back to CMOS
84h	Initialize ISA PnP boot devices
85h	1. USB final Initialization 2. NET PC: Build SYSID structure 3. Switch screen back to text mode 4. Set up ACPI table at top of memory. 5. Invoke ISA adapter ROMs 6. Assign IRQs to PCI devices 7. Initialize APM 8. Clear noise of IRQs.
86h	Reserved
87h	Reserved
88h	Reserved
89h	Reserved
90h	Reserved
91h	Reserved
92h	Reserved
93h	Read HDD boot sector information for Trend Anti-Virus code
94h	1. Enable L2 cache 2. Program boot up speed 3. Chipset final initialization. 4. Power management final initialization 5. Clear screen & display summary table 6. Program K6 write allocation 7. Program P6 class write combining

Code	Description
95h	1. Program daylight saving 2. Update keyboard LED & typematic rate
96h	1. Build MP table 2. Build & update ESCD 3. Set CMOS century to 20h or 19h 4. Load CMOS time into DOS timer tick 5. Build MSIRQ routing table.
FFh	Boot attempt (INT 19h)

4.16 Flash BIOS Utility

Utilize AWARD Flash BIOS programming utility to update on-board BIOS for the future new BIOS version. Please contact your technical window to get this utility if necessary.

NOTE : Remark or delete any installed Memory Management Utility (such as HIMEM.SYS, EMM386.EXE, QEMM.EXE, …, etc.) in the CONFIG.SYS files before running Flash programming utility.

Chapter 5 Appendix

5.1 Watch Dog Timer Sample Code

Watch Dog Timer is a special function; the user can monitor and control the system via software or hardware implementation. If the implementation does not respond in seconds, the system will be rebooted automatically. With this mechanism, the lost or damage can be minimized, when there is not monitoring personnel onsite.

Following list are PPAP-3720 Watch Dog Timer sample Code, this is for reference only:

```
/DG
 * PPAP-3720 Watch Dog Sample:
 *
 * Copyright (C) 2001 Portwell Inc.
 * Copyright (C) 1998,2000,2001,2002,2003. Chris Chiu
 *
 *   This program is free software; you can redistribute it and/or modify
 *   it under the terms of the GNU General Public License as published by
 *   the Free Software Foundation; either version 2 of the License, or
 *   (at your option) any later version.
 *
 *   This program is distributed in the hope that it will be useful,
 *   but WITHOUT ANY WARRANTY; without even the implied warranty of
 *   MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
 *   GNU General Public License for more details.
 *
 *   You should have received a copy of the GNU General Public License
 *   along with this program; if not, write to the Free Software
 *   Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307, USA.
 */


#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>
#include <errno.h>
```

```
#include <fcntl.h>
#include <sys/time.h>
#include <sys/types.h>
#include <sys/stat.h>

#include <asm/io.h> /* linux-specific */

#ifdef __GLIBC__
#  include <sys/perm.h>
#endif

unsigned int read_port(unsigned int port,int size)
{
   static int iopldone = 0;

   unsigned int val=0;

   if (port > 1024) {
         if (!iopldone && iopl(3)) {
            fprintf(stderr, " iopl(): %s\n", strerror(errno));
            return 0;
         }
         iopldone++;
   } else if (ioperm(port,size,1)) {
         fprintf(stderr, " ioperm(%x): %s\n", port, strerror(errno));
         return 0;
   }

   if (size == 4) {
         val=inl(port);
#ifdef DEBUG
         printf("Read_port:(0x%04x)=>0x%08x\n", port, val);
#endif
   } else if (size == 2) {
         val=inw(port);
#ifdef DEBUG
```

```
        printf("Read_port:(0x%04x)=>0x%04x\n", port, val);
#endif
    } else {
        val=inb(port);
#ifdef DEBUG
        printf("Read_port:(0x%04x)=>0x%02x\n", port, val);
#endif
    }
    return(val);
}


static int write_port(unsigned int port, unsigned int val, int size)
{
    static int iopldone = 0;

#ifdef DEBUG
    printf("Write_Port(0x%04x)<=0x%x\n", port, val);
#endif

    if (port > 1024) {
        if (!iopldone && iopl(3)) {
            fprintf(stderr, "iopl(): %s\n", strerror(errno));
            return 1;
        }
        iopldone++;
    } else if (ioperm(port,size,1)) {
        fprintf(stderr, "ioperm(%x): %s\n", port, strerror(errno));
        return 1;
    }

    if (size == 4)
        outl(val, port);
    else if (size == 2)
        outw(val&0xffff, port);
    else
```

```
            outb(val&0xff, port);
    return 0;
}

unsigned int TIMEOUT = 0x2;

void enable_wdt ( ) {

   unsigned int tmpa,tmpb=0;

   write_port(0x2e,0x87,1);
   write_port(0x2e,0x87,1);
   write_port(0x2e,0x2b,1);
   tmpa = read_port(0x2f,1);
   tmpa = tmpa & 0xef;
   write_port(0x2e,0x2b,1);
   write_port(0x2f,tmpa,1);
   write_port(0x2e,0x7,1);
   write_port(0x2f,0x8,1);
   write_port(0x2e,0xf5,1);
   tmpb = read_port(0x2f,1);
   //tmpb = tmpb | 0x8; //Select minute
   tmpb = tmpb | 0xf7; //Select second
   write_port(0x2e,0xf5,1);
   write_port(0x2f,tmpb,1);
   write_port(0x2e,0xf6,1);
   write_port(0x2f,TIMEOUT,1);
}
int main(int argc, char **argv)
{
 enable_wdt();
}
```

5.2 Reset To Default Sample Code

Reset To Default can provide user the flexibility of utilizing power button interface to be programmed into alternative function as user program requires, greatly enhanced user convenience.

Below is the sample code for Reset To Default, for reference only

```
; For  PPAP-3723,ROBO-8820 RESET to Default  testing
; By  Frank Hsu ,  07/25/2003
;
;       Reset to default status  can be read from  ICH3_GPI6.
;               After Power On reset, GPI6 = low ( 0 )
;    If Reset to Default (RST2DF) Button  pressed ( Triggered )
;     ,then GPI6  will  be  latch  to high ( 1 ).
;
;    RST2DF  register  can  be  cleared  by  ICH3_GPO19.
;    Write  a  pulse  timing ( High1_low_high2 )  to  clear  RST2DF to 0.
;    High1 : output  GPO19  high , and  keep  10 us.
;    Low   : output  GPO19  low  , and  keep  10 us.
;    High2 : output  GPo19  high  again , and  keep  high  always.
;
;
;  Programming Guide :
;  PG_Step1 : Enable ACPI IO port assignment and get PMBASE, then save to
;          EBX_Bit[31..16]
;
;          First : GPI_ROUT bit[13,12] P [0,0] : Let GPI6 not evoke SCI.
;                Write GPI_Rout bit[13,12] to [0,0] for no effect on GPI6
;            ( B0:D31:F0:Offset_B8h-Bit[13,12]P[0,0] , no SCI event evoked)
;
;          Second: Enabe ACPI IO port by  setting ACPI_CNTL bit4
;                B0:D31:F0:Offset_44h_bit4P1
;          Third : Get PMBASE  ( ACPI I/O port BAR ) and
;                save to EBX_bit[31..16].
;                PMBASE=:B0:D31:F0:Offset[40..43h]
;                Let Bit0 = 0.( PCI_BAR bit0 returns 1 for a IO_BAR )
;
;
;
;  PG_Step2 : Enable GPIO IO function and get GPIOBASE, then save to
;          ECX_Bit[31..16]
;
;          How  to  program GPIO19 ( Output  only , i.e. GPO19 )
;          ---------------------------------------------------
;          Get GPIOBASE =: B0:D31:F0:Offset[58..5Bh] ;(and let bit0 = 0 )
;             GPIO_CNTL =: B0:D31:F0:Offset_5Ch_bit4P1 ;Enable ICH3 GPIO
;
;          GPIO19
;             GP_LVL   (=:(GPIOBASE + 0Ch))_bit19P[0/1]; Write value 0/1
;          ---------------------------------------------------
;
;          How  to  read GPI6
;          =====================================================
;          GPI6 status MUST NOT be inverted First.
;          GPI_INV (=GPIOBASE+2Ch)-bit6P0. ( GPI6 not inverted )
;
;          Get GPI6 status from GPE1_STS (=PMBASE+2Ch)-bit6
;            0 = low , 1= high level
;          =====================================================

        .MODEL tiny
        .386
        .STACK 200h
        .data
```

```
PROMP1 DB'PORTWELL PPAP-3723,ROBO-8820,8820RSTD.exe, V1.00 07-25-2003,All rights reserved.$'


PROMP1_1 DB '  For  PPAP-3723, ROBO-8820 Reset-to-Default test .',13,10,'$'
PROMP_2_CR_LF  db  0Dh, 0Ah,0Dh, 0Ah, '$'
PROMP_Str1 db '  Reset-To-Default status latched by a F/F. ',0dh,0ah,'$'
PROMP_Str2 db '      This status bit = 0 ---> Normal.      ',0dh,0ah,'$'
PROMP_Str3 db '      This status bit = 1 ---> RST2DF button has been pressed.',0dh,0ah,'$'
PROMP_Str4 db '      This status bit can be read by ICH3_GPI6, ',0dh,0ah,'$'
PROMP_Str5 db '        and can be cleared by an ICH_GPO19  High1-Low-High2 pulse.',0dh,0ah,'$'
PROMP_Str6 db '                                          ',0dh,0ah,'$'
PROMP_Str7 db '        High1 =   30us  High  level          ',0dh,0ah,'$'
PROMP_Str8 db '        Low   =   30us  Low   level          ',0dh,0ah,'$'
PROMP_Str9 db '        High2 =   High  level again and no level change from now on.',0dh,0ah,'$'
PROMP_StrA db '                                          ',0dh,0ah,'$'
PROMP_rst2df db 0dh,0ah,' Press the Reset-to-Default button and then release it for the test NOW!$'
PROMP_anykey db 0dh,0ah,' Ready ? If yes , then Press any key to start test ..... $'

PROMP_err1 db 0dh,0ah,' ***** "Reset-to-Default F/F Initialization"  Failed. *****',0dh,0ah,'$'
PROMP_err1_1 db ' ( This may be a H/W error or Reset-to-Default button has ever been
pressed ! )',0dh,0ah,'$'
PROMP_err2 db 0dh,0ah,' ***** "Reset-to-Default event latched by F/F "  Failed. *****',0dh,0ah,'$'
PROMP_err3 db 0dh,0ah,' ***** "Clear Reset-to-Default F/F status "  Failed. *****',0dh,0ah,'$'
PROMP_TEST_OK db ' <<..... PPAP-3723/ROBO-8820 RESET-TO-DEFAULT test OK .....>>',0dh,0ah,'$'
PROMP_TEST_fail db ' <<*****  PPAP-3723/ROBO-8820 RESET-TO-DEFAULT test FAIL *****>>',0dh,0ah,'$'
PROMP_Qkey db 0dh,0ah,'Press "Q" key to stop test and return to DOS; or other key to go on next test.$'

GP_INV_OFFSET         db 2Ch  ; The offset value from GPIOBASE
GPE1_STS_OFFSET       db 2Ch  ; The offset value from PMBASE
GP_LVL_OFFSET         db 0Ch  ; The offset value from GPIOBASE
; EBX_bit[31..16]  save  PMBASE    ( B0:D31:F0:Offset[40..43h])
; ECX_bit[31..16]  save  GPIOBASE  ( B0:D31:F0:Offset[58..5Bh] )
        .code
programstart:
        mov ax,@data
        mov ds,ax

        lea dx,PROMP_2_CR_LF
        mov ah,09h
        int 21h

        lea dx,PROMP1
        mov ah,09h
        int 21h

        lea dx,PROMP_2_CR_LF
        mov ah,09h
        int 21h

        lea dx,PROMP1_1
        mov ah,09h
        int 21h

        lea dx,PROMP_2_CR_LF
        mov ah,09h
        int 21h

        lea dx,PROMP_Str1
        mov ah,09h
        int 21h
        lea dx,PROMP_Str2
        mov ah,09h
        int 21h
        lea dx,PROMP_Str3
        mov ah,09h
        int 21h
```

```
        lea dx,PROMP_Str4
        mov ah,09h
        int 21h
        lea dx,PROMP_Str5
        mov ah,09h
        int 21h
        lea dx,PROMP_Str6
        mov ah,09h
        int 21h
        lea dx,PROMP_Str7
        mov ah,09h
        int 21h
        lea dx,PROMP_Str8
        mov ah,09h
        int 21h
        lea dx,PROMP_Str9
        mov ah,09h
        int 21h
        lea dx,PROMP_StrA
        mov ah,09h
        int 21h



        mov edx,00000000h ; Error flag in EDX_BIT[16..18], 0=ok, 1=failed

;  PG_Step1 : Enable ACPI IO port assignment and get PMBASE, then save to
;           EBX_Bit[31..16]
;
;           First : GPI_ROUT bit[13,12] P [0,0] : Let GPI6 not evoke SCI.
;                 Write GPI_Rout bit[13,12] to [0,0] for no effect on GPI6
;             ( B0:D31:F0:Offset_B8h-Bit[13,12]P[0,0] , no SCI event evoked)
;
;           Second: Enabe ACPI IO port by  setting ACPI_CNTL bit4
;                 B0:D31:F0:Offset_44h_bit4P1
;           Third : Get PMBASE  ( ACPI I/O port BAR ) and
;                 save to EBX_bit[31..16].
;                 PMBASE=:B0:D31:F0:Offset[40..43h]
;                 Let Bit0 = 0.( PCI_BAR bit0 returns 1 for a IO_BAR )
;
;



 ;    ------------------------------------------------- 1_start
 ; Get  PMBASE  and   save  to  EBX_bit[31..16]
 ; Let GPI6 GPI_ROUT to [0,0] , i.e. not evoke SCI  in S0.

        mov  dx,0CF8h      ; PCI Config Read
        mov  eax,8000F8B8h ; B0:D31:F0:Offset_B8h
        out  dx,eax
        mov  dx,0CFCh
        in   eax,dx

        and  ah,0CFh       ; bit[13,12] set to [0,0] to let GPI6 not
        out  dx,eax        ; evoke SCI event


        mov  dx,0CF8h      ; PCI Config Read
        mov  eax,8000F844h ; B0:D31:F0:Offset_44h
        out  dx,eax
        mov  dx,0CFCh
        in   eax,dx

        or   al,10h        ; bit 4 set to 1 to enable PMBASE
        out  dx,eax        ;
```

```
        mov  dx,0CF8h      ; Get PMBASE
        mov  eax,8000F840h ; B0:D31:F0:Offset_40h
        out  dx,eax
        mov  dx,0CFCh
        in   eax,dx

        and  al,0feh       ; bit0 cleared to 0.
        rol  eax,10h
        mov  ebx,eax       ; Save PMBASE to EBX[31..16]
 ;   ------------------------------------------------- 1_end



;  PG_Step2 : Enable GPIO IO function and get GPIOBASE, then save to
;           ECX_Bit[31..16]
;
;           How  to  program GPO19
;           ------------------------------------------------------
;           Get GPIOBASE =: B0:D31:F0:Offset[58..5Bh] ;(and let bit0 = 0 )
;              GPIO_CNTL =: B0:D31:F0:Offset_5Ch_bit4P1 ;Enable ich3 GPIO
;
;           GPO19
;              GP_LVL   (=:(GPIOBASE + 0Ch))_bit19P[0/1]; Write value 0/1
;           ------------------------------------------------------
;
;           How  to  read GPI6
;           ======================================================
;           GPI6 status must NOT be inverted First.
;           GPI_INV (=GPIOBASE+2Ch)-bit6P0. ( GPI6 not inverted )
;
;           Get GPI6 status from GPE1_STS (=PMBASE+2Ch)-bit6
;            0 = low , 1= high level
;           ======================================================


 ;   ================================================== 2_start
 ; Get  GPIOBASE  Base  Address , and  save  to  ECX_bit[31..16]

        mov  dx,0CF8h
        mov  eax,8000F85Ch ; B0:D31:F0:Offset_5Ch
        out  dx,eax
        mov  dx,0CFCh
        in   eax,dx

        or   al,10h        ; 5Ch_Bit4P1  to  Enable  GPIO
        out  dx,eax

        mov  dx,0CF8h      ; Get GPIOBASE
        mov  eax,8000F858h ; B0:D31:F0:Offset_58h
        out  dx,eax
        mov  dx,0CFCh
        in   eax,dx

        and  al,0feh       ; bit 0 cleared to 0.
        rol  eax,10h
        mov  ecx,eax       ; Save GPIOBASE to ECX[31..16]


 ; Get  GPIOBASE  Base  Address , and  save  to  ECX_bit[31..16]


;  Testing  way :
;  --- t1
;  Read GPI6 first , GPI6=0  ? if yes,pass ; if no, failed
;
;  --- t2
;  RST2DF button pressed and released , read GPI6 ,GPI6 = 1 ? if yes, pass ; if no, failed
```

```
;
;  --- t3
;  Clear RST2DF status to 0 ,read GPI6 ,GPI6 = 0 ? if yes, pass ; if no, failed
;------------------------------------------------------------------- t_start
          rol  ecx,10h  ; Restore GPIOBASE from ECX[31..16] to ECX[15..0]

  ; make sure GPO19 = 1   start  ( RST2DF  F/F  no  cleared  by  GPO19 )
       xor  bx,bx

       mov  bl,GP_LVL_OFFSET      ;  Write GPO19 1
       mov  dx,cx                 ;
       add  dx,bx                 ;
       add  dx,02h                ; point to GPIO[16..23]  register
       in   al,dx                 ;  read  first

       call  IODELAY ;  io delay

       or   al,08h                ; bit3 ---> GPO19
       out  dx,al                 ; output GPO19 1
  ; make sure GPO19 = 1   end
; ============================= MUST  DO ==========================Start
   ; GP_INV bit6 MUST Program 0 for  GPI6  state  not  inverted.  start

       xor  bx,bx

       mov  bl,GP_INV_OFFSET      ;  Not invert GPI6 status
       mov  dx,cx                 ;
       add  dx,bx                 ;   bit6 ---> GPI6
       in   al,dx                 ;  read  first

       call  IODELAY ;  io delay

       and  al,0BFh               ; mask bit6  and  write 0
       out  dx,al

   ; GP_INV bit6 MUST Program 0 for  GPI6  state  not  inverted.  end
; ============================= MUST  DO ==========================End



;  ---- t1 start
;  GPI6 , read its  status , initialization  will  be  0 .
;
;          How  to  read  GPI6
;          ---------------------------------------------------
;          PMBASE has been stored in EBX[31..16].
;          Get GPI6 status from GPE1_STS (=PMBASE+2Ch)-bit6
;            0 = low , 1= high level
;          ---------------------------------------------------

    call READ_GPI6_TO_AL

    and al,40h  ; mask bit6
    cmp al,00h

       je  next_test1 ; okay , go on test
;       jz  next_test1 ; okay , go on test

   ; no , error  message  display

       lea dx,promp_err1
       mov ah,09h
       int 21h
       lea dx,promp_err1_1
       mov ah,09h
       int 21h
```

```
        ror  edx,10h    ; error  falg EDX_Bit16 , 1 --> Error  happened
        or   dl,01h
        rol  edx,10h

        call  KB_Wait

;  ---- t1 end

        next_test1 :

;  ---- t2 start
        lea dx,promp_rst2df
        mov ah,09h
        int 21h

        lea dx,promp_anykey
        mov ah,09h
        int 21h


        xor al,al          ; halt for ready?  Any  key  pressed to go on.
 WAIT_KB_0:
        mov ah,1
        int 21h

        cmp al,0
        je  WAIT_KB_0


        lea dx,PROMP_2_CR_LF
        mov ah,09h
        int 21h


;  test  RST2DF  button  pressed

     call  READ_GPI6_TO_AL

    and al,40h  ; mask bit6
    cmp al,40h

        je  next_test2 ; okay , go on test

   ; no , error  message  display

        lea dx,promp_err2
        mov ah,09h
        int 21h

        ror  edx,10h    ; error  falg EDX_Bit17 , 1 --> Error  happened
        or   dl,02h
        rol  edx,10h

        call  KB_Wait

;  ---- t2 end

        next_test2 :

;  ---- t3 start    ,Clear  RST2DF  F/F
;  GPO19  write 1,0,1
;

  ; =========   Write GPO19 1-0-1   start

        xor  bx,bx
```

```
        mov  bl,GP_LVL_OFFSET       ;  Write GPO19 1
        mov  dx,cx                  ;
        add  dx,bx                  ;
        add  dx,02h                 ; point to GPIO[16..23]  register
        in   al,dx                  ;  read  first

        call  IODELAY ;  io delay

        or   al,08h                 ;
        out  dx,al                  ; output GPO19 1 first

       call   FIXDELAY              ;  30 us delay

        in   al,dx                  ; output GPO19 0  then

        call  IODELAY

        and  al,0F7h
        out  dx,al

       call   FIXDELAY              ; 30 us delay

        in   al,dx                  ;  output  GPO19  high  finally

        call  IODELAY

        or   al,08h
        out  dx,al
  ; =========   Write GPO19 1-0-1   end


        call  READ_GPI6_TO_AL    ; check  RST2DF F/F

        and   al,40h             ; mask Bit6
        cmp   al,00h
        je    test_end           ; okay , then end

    ;   jz    test_end           ; okay , then end

   ; no , error  message  display

        lea dx,promp_err3
        mov ah,09h
        int 21h

        ror  edx,10h    ; error  falg EDX_Bit18 , 1 --> Error  happened
        or   dl,04h
        rol  edx,10h

;  ---- t3 end

   test_end :

        ror edx,10h   ; check error  flag
        cmp dl,00h
        je  test_ok

 test_fail :

        lea dx,PROMP_2_CR_LF
        mov ah,09h
        int 21h

        lea dx,promp_TEST_fail
        mov ah,09h
        int 21h
```

```
        jmp return_to_dos

  test_ok :
        lea dx,promp_TEST_OK
        mov ah,09h
        int 21h

        ror  ecx,10h              ;   ECX[15..0]  to  ECX[31..16]
                                  ;   Restore  GPIOBASE to ECX[31..16]
return_to_dos :
        mov  ah,4ch  ; Return  to  DOS
        int  21h

;-------------------------------------------------------------- t_end
  ;  ================================================== 2_end


IODELAY  PROC    near
        push  ax
        push  dx

        mov   dx,0edh
        in    al,dx
        jmp   $+2
        mov   dx,0edh
        in    al,dx

        pop   dx
        pop   ax
        ret
IODELAY  ENDP


KB_wait  PROC    near

        push ax
        push bx
        push cx
        push dx

        lea dx,PROMP_Qkey
        mov ah,9                ; Display "Q" key prompt
        int 21h

        xor al,al
 WAIT_KB:
        mov ah,1
        int 21h

        cmp al,0
        je  WAIT_KB

        cmp al,51h   ; "Q"  pressed ?
        je  test_fail
        cmp al,71h   ; "q"  pressed ?
        jne call_return
        jmp test_fail  ;

 call_return :

        lea dx,PROMP_2_CR_LF
        mov ah,09h
        int 21h

        pop dx
        pop cx
        pop bx
```

```
        pop ax

        ret

KB_wait  ENDP


READ_GPI6_TO_AL  PROC    near
    push bx
    push dx

    xor bx,bx

    rol ebx,10h  ;  restore PMBASE from EBX_bit[31..16] to EBX_bit[15..0]
    mov dx,bx
    ror ebx,10h  ;  save PMBASE to EBX_Bit[31..16]
    mov bl,GPE1_STS_OFFSET
    add dx,bx
    in  al,dx

    call  IODELAY ;  io delay


 ; MUST  to  do  write 1 to clear  GPE1_STS_bit6  to  0  FIRST due to the
 ; access ( 0/1 ) . This register is R/WC , and will be set
 ; at any time when GPI signal is high.
 ;

    and al,40h  ; mask bit6
    or  al,40h  ; WC
    out dx,al   ; Write bit6 to 0 first.

    call   IODELAY ; io delay

    in  al,dx   ; read GPI6 again

    pop  dx
    pop  bx
    ret
READ_GPI6_TO_AL  ENDP

;---------------------------------------------------------------;
;                       FIXED_DELAY                             ;
;---------------------------------------------------------------;
;       Input : (CX) count of 15 microseconds to wait           ;
;               STACK PRESENT                                   ;
;       Output: NONE                                            ;
;       CX=2 , 15us x 2 = 30 us                                 ;
;                                                               ;
; This routine is called to wait for 15 microseconds * count in ;
; (CX), then return.  Gives a programmed software delay.        ;
;---------------------------------------------------------------;
 FIXDELAY  PROC    near
        push cx
        push dx
        push ax
        pushf

        mov  cx,02h

       mov     dx,61h
       in      al,dx               ;
       jmp     $+2
       jmp     $+2
       and     al,00010000b                ;
       mov     ah,al                       ;
 fixed_delay_1:
```

```
        in      al,dx               ;
        jmp $+2
        jmp $+2
        and     al,00010000b            ;
        cmp     al,ah                   ;
        jz      short fixed_delay_1     ;
        mov     ah,al                   ;
        loop    short fixed_delay_1     ;

        popf
        pop  ax                         ;
        pop  dx
        pop  cx

        ret
    FIXDELAY  ENDP


END programstart
```

Chapter 6 EZIO-100

6.1 About EZIO-100

Proprietary keypad and LCD display interfaces are implemented in traditional computing system design, but they are usually different from system to system. The main purpose to roll this module out is to provide an easier man-machine interface for those computing systems regarding application friendly operation as a "must."

The design goals of this interface are:

- **A single interface for those applications where both LCD display and keypad are required.**
- **This interface should be available in every computing system.**
- **The communication implementation should be OS independent.**

Our solution is to use "Serial port" as the interface for both LCD display and keypad. A simple protocol is further defined so that applications can directly communicate with this module no matter what the Operating System is.

WARNING!

THE LCD DRIVER ICS ARE MADE OF CMOS PROCESS, DAMAGED BY STATIC CHARGE VERY EASILY. MAKE SURE THE USER IS GROUNDED WHEN HANDLING THE LCD.

6.2 Features

- **Ideal user interface for communication appliance**
- **No driver required; OS independent**
- **Alphanumeric characters display support**
- **Four key pads can be customized for different applications**
- **Easy system installation and operation**
- **Clearly display system status**
- **Single interface to SBC or M/B**

6.3 Mechanical Specification

Module Size (mm):	• 101.6(W) x 26.0(H) x 30.6(D) (max.)
Display Format:	• 16 characters x 2 lines
Character Size:	• 3.0 x 5.23 mm

6.4 General Specification

- **General Specification**

Display Resolution:	• 16 characters x 2 lines
Dimensional Outline (mm):	• 101.6(W) x 26.0(H) x 30.6(D) (max.)
Function Key:	• Four operation keys (up, down, enter and ESC)
Display Icon:	• Eight self-defined icons
Interface:	• RS-232

- **Absolute Maximum Rating**

Item	*Normal Temperature*			
	Operating		Storage	
	Max.	Min.	Max.	Min.
Ambient Temperature	0ºC	+50ºC	-20ºC	+70ºC
Humidity (w/o condensation)	Note 2, 4		Note 3, 5	

6.5 Product Outlook

6.6 Interface Pin Assignment

There are only two connectors in this module, as shown in ***Figure 6-1***: power connector and Serial Port connector. The power source into this module is 5 volt only. There are only three pins used in the Serial Port interface (***Figure 6-2***).

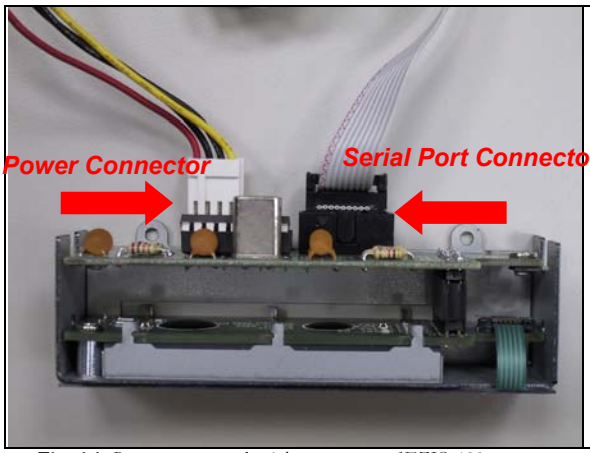
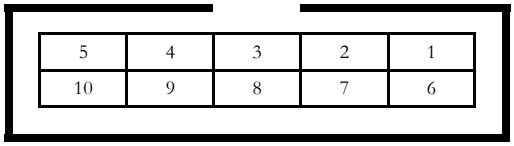


Fig. 6-1 *Power connector and serial port connector of EZIO-100*



Pin 2: TxD Pin 3 : RxD Pin 5 : Ground

Fig. 6-2 *Pin assignment*

In other words, the Serial Port is defined as DCE. Therefore, we can use a straight-through cable to connect it to the Serial Port of most of the computers, defined as DTE.

(1) Interface Pin Assignment

PIN NO.	PIN OUT	Description
1	NC	No connector
2	RXD	RS232 Data
3	TXD	RS232 Data
4	NC	No connector
5	V_{SS}	Ground
6	NC	No connector
7	NC	No connector
8	NC	No connector
9	NC	No connector
9	NC	No connector

(2) Power

PIN NO.	PIN OUT	Description
1	NC	No connector
2	GND	Power GND
3	GND	Power GND
4	+5V	Power VCC (+5V)

6.7 EZIO Function Command

First, all versions (00A, 01A, 02A) of EZIO can use those commands. Only the 02A version of EZIO firmware that adds “FE 28” & “FE 37” command can control start of HEX & End of HEX.

EZIO is an intelligent device, which will display those data received from RS-232 port and reply key pressing status to polling command from RS-232 port. Both commands and data go thru RS-232 ports. To distinguish between data and commands, the LCD/key-pad module recognizes a command prefix, 254 (Hex 0FE). The byte following “254” will be processed as a command. For example, to

clear the screen, send the command prefix (254) followed by the LCD clear-screen code (1). The valid data range is shown as the following table:

Valid data range	*Displayed characters*
0-7	Customized icon 0-7
48-57 (30-39 Hex)	0-9
65-90 (41-5A Hex)	A-Z
97-122 (61-7A Hex)	a-z

To get the key pressing status, a "read key" command can be issued to this module, which will check the key-pressing status and reply accordingly. The following are the commands and corresponding Decimal/Hex values:

	Functions/commands	*Decimal/Hex*	*Comment*
1.	Start Of HEX	40/28	Only for 02A
2.	End Of HEX	55/37	Only for 02A
3.	Clear screen	1/01	
4.	Home cursor	2/02	
5.	Read key	6/06	See note 1
6.	Blank display (retaining data)	8/08	
7.	Hide cursor & display blanked characters	12/0C	
8.	Turn on (blinking block cursor)	13/0D	
9.	Show underline cursor	14/0E	
10.	Move cursor 1 character left	16/10	
11.	Move cursor 1 character right	20/14	
12.	Scroll 1 character left	24/18	
13.	Scroll 1 character right	28/1C	
14.	Set display address (position the cursor) location	128 (Hex080)+ Location	See note 2
15.	Set character-generator address	64 (Hex 040)+ address	See note 3

Note 1: Upon receiving the "read key" command from host computer, the LCD/keypad module will check the status of every key and reply with status command accordingly. The replied message from LCD/key-pad module consists of a header and a status byte. The header byte is 253 (Hex0FD). The high nibble (with the most significant bit) of the status byte is always "4" and the low nibble (with the least significant bit) of the status byte is used to indicate key pressing status of the keypad module. This nibble will be "F" (of four 1s), if no key pressed while the "read key" received. "0" will be used to indicate key pressing status of corresponding key. There are four keys in this module – upper arrow, down arrow, enter (ENT), and escape (ESC). The relationship between the function key, corresponding status bit and status byte is shown as the table below.

Function key	*Corresponding status bit*	*Status byte*
Escape	The fourth bit of lower nibble (the least significant bit) (1110)	4E (H)
Up arrow	The third bit of lower nibble (1101)	4D (H)
Enter	The second bit of lower nibble (1011)	4B (H)
Down arrow	The first bit of lower nibble (0111)	47 (H)

More than one key can be pressed at the same time so that there may be more than one "0"s in the low nibble of status byte. For example, if Up and Down arrow keys are pressed at the same time while "read key" command received, the replied status will be "Hex045".

Note 2: This command can be used to place the cursor at any location. The corresponding address for each character on the screen is as follows:

For 16×2 Display Address

Character	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Location (Address)	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
	40	41	42	43	44	45	46	47	48	49	4A	4B	4C	4D	4E	4F

The addresses of characters at the same row are continuous, so moving cursor commands can be applied to shift the cursor position back and forth. However, the addresses of characters between upper and lower row are discontinuous. To change cursor position between upper row and lower row, this command will be applied.

Note 3: This command can be used to create customized icon. The starting address is 64 and every character will take 8 bytes to create a 5(W) x 7(H) resolution picture, as shown below:

CG RAM MAPPING

CG RAM Address (5 High … 0 Low)							Character Patterns (CG RAM data) (7 High … 0 Low)								
5	4	3	2	1	0		7	6	5	4	3	2	1	0	
0	0	0	0	0	0		*	*	*	0	1	1	0	0	
			0	0	1					1	0	0	1	0	
			0	1	0					0	0	1	0	0	←Character Pattern
			0	1	1					0	1	0	0	0	
			1	0	0					1	1	1	1	0	
			1	0	1					0	0	0	0	0	
			1	1	0					0	0	0	0	0	
			1	1	1					0	0	0	0	0	←Cursor
0	0	1	0	0	0		*	*	*	1	1	1	1	1	
			0	0	1					1	0	0	0	1	
			0	1	0					1	0	1	0	1	←Character Pattern
			0	1	1					1	0	1	1	1	
			1	0	0					1	0	1	0	1	
			1	0	1					1	0	0	0	1	
			1	1	0					1	1	1	1	1	
			1	1	1					0	0	0	0	0	←Cursor
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	
1	1	1	0	0	0		*	*	*	1	1	1	1	1	
			0	0	1					1	0	0	0	1	
			0	1	0					1	1	1	0	1	←Character Pattern
			0	1	1					1	0	0	0	1	
			1	0	0					1	0	1	1	1	
			1	0	1					1	0	0	0	1	
			1	1	0					1	1	1	1	1	
			1	1	1					0	0	0	0	0	←Cursor

To show the customized icon, simply send the data between "0" to "7" to this module.

For example, this module will display the customized icon at location 64 to 71 upon receiving data "0"; white it will display the customized icon at location 72 to 79 upon receiving data "1".

Watchdog timer is also built in the module. This module will reset itself and send out "reset packet" (0FDH, 0EH) thereafter.

The input must be a standard RS-232 or inverted TTL signal. The RS-232 setting should be:

- *Baud rate: 2400 bps*
- *Parity: None*
- *Data bits: 8*
- *Stop bit: 1*

What follows is the default setup after LCD module initiated:

- *2-line display mode; every character is 5 x 8 dots.*
- *Display on; cursor off; cursor blink off.*
- *Display will be cleared.*

- *Shift right for entry mode.*
- *Set address counter to “00”(cursor position to 0)*
- *In entry mode.*

6.8 Character Generator ROM (CGROM)

Lower 4 bits \ Upper 4 bits	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
0000	CG RAM (1)															
0001	CG RAM (2)															
0010	CG RAM (3)															
0011	CG RAM (4)															
0100	CG RAM (5)															
0101	CG RAM (6)															
0110	CG RAM (7)															
0111	CG RAM (8)															
1000	CG RAM (1)															
1001	CG RAM (2)															
1010	CG RAM (3)															
1011	CG RAM (4)															
1100	CG RAM (5)															
1101	CG RAM (6)															
1110	CG RAM (7)															
1111	CG RAM (8)															

6.9 Sample Code

```
/* ************************************
 * EZIO RS232 LCD Control Sample Program
 * ************************************
* ***************************************************************************
 * Company:          Portwell Inc.
 * Date:             4/16/2003
 * Program:          02A.c
 * Version:          1.02
 * Compile:          Linux GNU C
 * Purpose:          Direct access to EZIO LCD, the program will display
 *                         messages according to the control button. The current
 *                         version only has the following function:
 *
 *                         1: display welcome message
 *                         2: display UP message if "scroll up" button is pressed
 *                         3: display ENTER message if "ENTER" button is pressed
 *                         4: display ESC message if "ESC" button is pressed
 *                         5: display DOWN message if "scroll down" button is pressed
*
 * Program Overview:
 *
 *              - Parameters:
 *                      fd          : a file name for open() method, here represents the com port
 *                      Cmd         : command prefix
 *                      cls         : clear command
 *                      init        : initialize command
 *                      blank       : display blank screen
 *                      stopsend    : stop input/output
 *                      home        : move cursor to initial position
 *                      readkey     : set to read from EZIO
 *                      hide        : hide cursor & display blanked characters
 *                      movel       : move cursor one character left
 *                      mover       : move cursor one character right
 *                      turn        : turn on blinking-block cursor
 *                      show        : turn on underline cursor
 *                      scl         : scroll cursor one character left
 *                      scr         : scroll cursor one character right
 *                      setdis      : set character-generator address
 *
*               - Procedure:
 *                      1. The program sets up the environment, i.e. com port settings.
 *                      2. The main function MUST call init() twice to initialize EZIO
 *                         before any communication.
 *                      3. For executing any command, the command prefix, Cmd, MUST be
 *                         called be command. So all command contains two parts, eg.
 *                         to initialize the sequence of HEX number is 0xFE, 0x25.
 *                      4. After clear screen and display welcome message, ReadKey()
 *                         method must be call to advise EZIO for reading data.
 *                      5. A pooling method is implemented to get input from EZIO while
 *                         any button is pressed.
 *
 *              - NOTE: This program is a sample program provided " AS IS" with NO
 *                          warranty.
 *
 * Copyright (c) Portwell, Inc. All Rights Reserved.
 *
 * ***************************************************************************/
```

```
#include <sys/stat.h>
#include <fcntl.h>
#include <unistd.h>
#include <stdlib.h>

static int fd;

void SetEnvironment () {
  system("stty ispeed 2400 < /dev/ttyS1");
  system("stty raw < /dev/ttyS1");
}

int Cmd = 254;  /* EZIO Command */
int cls = 1;    /* Clear screen */
void Cls () {
  write(fd,&Cmd,1);
  write(fd,&cls,1);
}

int init = 0x28;
void Init () {
  write(fd,&Cmd,1);
  write(fd,&init,1);
}

int stopsend = 0x37;
void StopSend () {
  write(fd,&Cmd,1);
  write(fd,&init,1);
}

int home = 2    ;   /* Home cursor */
void Home () {
  write(fd,&Cmd,1);
  write(fd,&home,1);
}

int readkey = 6           ;   /* Read key */
void ReadKey () {
  write(fd,&Cmd,1);
  write(fd,&readkey,1);
}

int blank = 8    ;   /* Blank display */
void Blank () {
  write(fd,&Cmd,1);
  write(fd,&blank,1);
}

int hide = 12    ;   /* Hide cursor & display blanked characters */
void Hide () {
  write(fd,&Cmd,1);
  write(fd,&hide,1);
}

int turn = 13    ;   /* Turn On (blinking block cursor) */
void TurnOn () {
  write(fd,&Cmd,1);
  write(fd,&turn,1);
}
```

```
int show = 14  ;   /* Show underline cursor */
void Show () {
  write(fd,&Cmd,1);
  write(fd,&show,1);
}

int movel = 16 ;   /* Move cursor 1 character left */
void MoveL () {
  write(fd,&Cmd,1);
  write(fd,&movel,1);
}

int mover = 20 ;   /* Move cursor 1 character right */
void MoveR () {
  write(fd,&Cmd,1);
  write(fd,&mover,1);
}

int scl = 24;         /* Scroll cursor 1 character left */
void ScrollL(){
  write(fd,&Cmd,1);
  write(fd,&scl,1);
}

int scr = 28;       /* Scroll cursor 1 character right */
void ScrollR(){
  write(fd,&Cmd,1);
  write(fd,&scr,1);
}
int setdis =  64;/* Command */
void SetDis(){
  write(fd,&Cmd,1);
  write(fd,&setdis,1);

 }


/* Add or Change Show Message here */
char mes1[] = "Portwell EZIO";
char mes2[] = "**************";
char mes3[] = "Up is selected";
char mes4[] = "Down is selected";
char mes5[] = "Enter is selected";
char mes6[] = "ESC is selected";
char nul[] = "                                    ";

int a,b;
void ShowMessage (char *str1 , char *str2) {
   a = strlen(str1);
   b = 40 - a;
   write(fd,str1,a);
   write(fd,nul,b);
   write(fd,str2,strlen(str2));
}

int main () {

  SetEnvironment();    /* Set RAW mode */

  fd = open("/dev/ttyS1" ,O_RDWR);/** Open Serial port (COM2) */
```

```
        Init(); /* Initialize EZIO twice */
      Init();

        Cls();   /* Clear screen */
        ShowMessage(mes1,mes2);

  while (1) {
    int res;
    char buf[255];

    SetDis();
    ReadKey(); /* sub-routine to send "read key" command */
    res = read(fd,buf,255); /* read response from EZIO */

    switch(buf[1]) {       /* Switch the Read command */

        case 0x4D :    /* Up Botton was received */
                Cls();
              ShowMessage(mes1,mes3); /** display "Portwell EZIO" */
              break;              /** display "Up is selected */


      case 0x47 :       /** Down Botton was received */
              Cls();
              ShowMessage(mes1,mes4);  /** display "Portwell EZIO" */
              break;              /** display "Down is selected" */

      case 0x4B :       /** Enter Botton was received */
              Cls();
              ShowMessage(mes1,mes5);  /** display "Portwell EZIO" */
              break;              /** display "Enter is selected" */

      case 0x4E :       /** Escape Botton was received */
              Cls();
              ShowMessage(mes1,mes6);  /** display "Portwell EZIO" */
              break;              /** display "Escape is selected */
     }

  }

}
```