

# **AR-B5800 Board**

**Intel® Core™2 LGA775 PROCESSOR**

**Networking Board**

## **User Manual**

**Manual Rev.: 1.2**

**Book Number: AR-B5800-2011.01.10**

## Revision

Version	Date	Author	Description
1.0	2011/2/22	Terry Lei	Initial release
1.1	2011/2/23	Terry Lei	2.2.1 Locations Modify Item24 Modify to SYSFAN3 Item27 Modify to SYSFAN1  2.3 Connector and Jumper Setting Item24 Modify to SYSFAN3 Item27 Modify to SYSFAN1
1.2	2011/3/15		1.1 Remove "Operating Temperature"

## Copyright 2011

### **All Rights Reserved.**

Manual's first edition: June 11,2011

For the purpose of improving reliability, design and function, the information in this document is subject to change without prior notice and does not represent a commitment on the part of the manufacturer.

In no event will the manufacturer be liable for direct, indirect, special, incidental, or consequential damages arising out of the use or inability to use the product or documentation, even if advised of the possibility of such damages.

This document contains proprietary information protected by copyright. All rights are reserved. No part of this Manual may be reproduced by any mechanical, electronic, or other means in any form without prior written permission of the manufacturer.

### **Trademarks**

AR-B5800 is a registered trademarks of Acrosser; IBM PC is a registered trademark of the International Business Machines Corporation; Pentium is a registered trademark of Intel Technologies Inc; Award is a registered trademark of Award Software International Inc; other product names mentioned herein are used for identification purposes only and may be trademarks and/or registered trademarks of their respective companies.

## Table of Contents

<b>1 Introduction .....</b>	<b>5</b>
1.1 Specifications .....	6
1.2 Package Contents .....	7
1.3 Block Diagram .....	8
<b>2 H/W Information .....</b>	<b>9</b>
2.1 Locations (Top side) .....	9
2.2 Connectors and Jumper Setting.....	11
2.3 Connector and Jumper Setting .....	13
<b>3 Bios setting.....</b>	<b>17</b>
3.1 Main Setup .....	18
3.2 Advanced Chipset Setup .....	19
3.3 PnP/PCI Setup .....	20
3.4 Peripherals Setup.....	21
3.5 PC Health Setup.....	23
3.6 Boot Setup .....	23
3.7 Exit Setup .....	24
3.8 BIOS Update.....	26
<b>4 Software Installation and Programming Guide</b>	<b>31</b>
4.1 Introduction.....	31
4.2 File Descriptions.....	33
4.3 API List and Descriptions .....	36

# 1 Introduction

AR-B5800 is designed for rack-mounted platform for networking appliance, e.g. VPN, SSL, UTM or firewall. With Intel advanced Core 2 Quad / Duo / Pentium / Celeron CPU, AR-B5800 is a powerful platform to satisfy different applications. By eight 10/100/1000Mbps LANs, the AR-B5800 is sufficient for the small to middle size business security solution.

AR-B5800 can install 2 x HDD for RAID 0/1 redundancy. Customers don't need to worry about data lost due to HDD defected problem. BIOS, GPIO and Jumper can control LAN bypass feature. It provides flexibility to access Internet by user setting. It also has standard PCIe x 8 slot. Customers can purchase suitable add-on card to meet their appliance.

Key features:

1. Support Intel Core 2 Quad or Core 2 Duo or Pentium or Celeron CPU
2. Intel G41 + ICH7R Chipset to support RAID 0/1 redundancy
3. DDRIII DIMM x 2, up to 4GB memory
4. Intel 82574L 10/100/1000Mbps x 6 + 82541PI 10/100/1000Mbps x 2
5. Two pairs LAN ports support bypass feature (LAN 1/2 + LAN 3/4)
6. LAN bypass can be controlled by BIOS, GPIO and Jumper
7. CF socket, 2.5" HDD x 2, SATA II interface x 2
8. Console, VGA (pinhead), USB 2.0 x 4 (2 x connectors, 2 x pin head)
9. Support boot from LAN, console redirection
10. Support standard PCIe x 8 slot for feature expansion

## 1.1 Specifications

- **CPU:** a LGA775 socket for Intel Core2 Processors in the 775-Land LGA package.
- **DMA channels:** 7.
- **Interrupt levels:** 16 (24 APIC interrupts).
- **Chipset:** Intel G41 express chipset 82G41 + 82801GR + W83627DHG-P.
- **Memory:** provides two 240-pin DIMM sockets to support DDRIII 1066 non-ECC DIMM. The memory capability can up to 2GB.
- **VGA Controller:** G41 GMCH integrated.
- **Analog Display Interface:** 10-pin box header, and resolution up to 2048x1536@75Hz.
- **Serial ATA Interface:** supports Two SATA devices, and data transfer rates up to 300MB/s per device.
- **Compact flash interface:** supports TYPE-II compact flash card with UDMA supported.
- **USB2.0 interface:** one stacked USB connector and two 10-pin pin header to support Six USB2.0 compatible devices. All resettable fuses protected.
- **Ethernet interface:** on-board six PCI express gigabit Ethernet controllers and two PCI gigabit Ethernet controllers to support eight LAN ports. They provide a standard IEEE 802.3 Ethernet interface for 1000BASE-T, 100BASE-TX, and 10BASE-T applications (802.3, 802.3u, and 802.3ab, respectively).
- **BYPASS function:** supports by ports LAN1 & LAN2 , LAN3 & LAN4 software programmable.
- **PCIE X8 interface:** One PCI Express x8 slot.
- **Serial ports (RS232):** One high-speed 16550 compatible UARTs ports with 16-byte send/receive FIFOs.  
COM1: RJ45 connector.
- **LCM interface:** a 7-pin pin header could be used to LCM for chassis' control panel.
- **General Purpose Input/Output:** 8-bit, 3.3V TTL level, bidirectional, and software programmable GPIOs.
- **WATCHDOG:** software programmable 1~255 second(s) / minute(s).
- **Power Consumption:** **+12V [5.3A], +5V [3.5A], +3.3V [1.8A], -12V [0.9A], +5Vsb [0.8A] typically.**

Test equipments list as below:

Main board: AR-B5800.

Processor: Intel Pentium Processor E6500 2.93GHz / FSB1066 / 2MB  
L2 cache / 45nm.

Memory: one Kingston KVR1333D3N9/2G.

One 3.5" HDD

OS: Windows XP SP3.

Processor was running at 100% loading.

Note: A proper power supply unit choice means that we should consider at least about

- a.) Protection of overload, short-circuit, and other safeties.
- b.) Summation of all devices' power requirements.
- c.) Thermal de-rating.

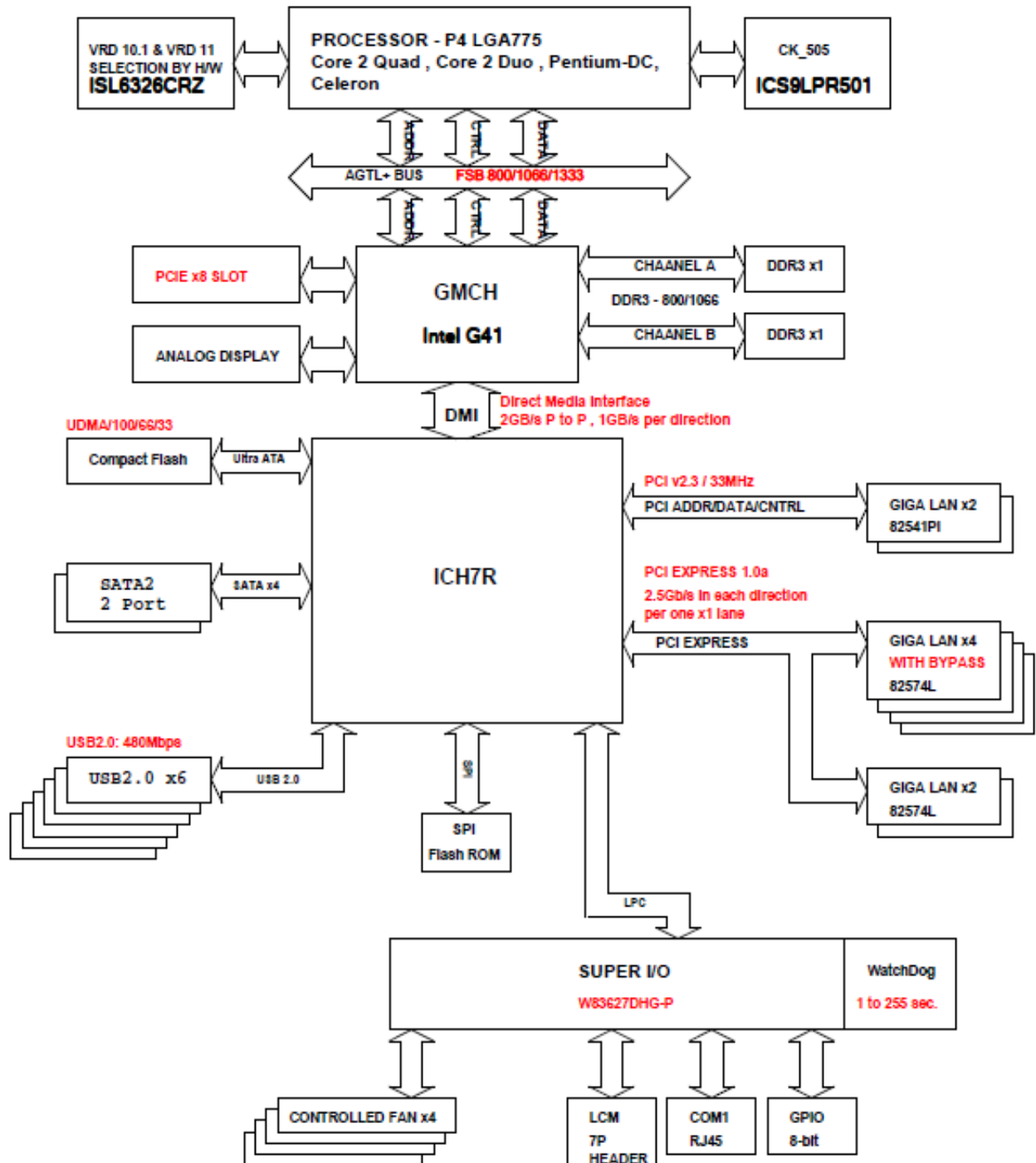
## 1.2 Package Contents

Check if the following items are included in the package.

In addition to this User's Manual, the AR-B5800 package includes the following items:

- A quick setup manual.
- One AR-B5800 networking board.
- One Software utility CD.
- One adaptable cable for COM1.
- One D-SUB-15 adaptable cable for VGA.
- One SATA cable.
- One USB adaptable cable. (Optional) (ACROSSER's P/N. [190030779-G](#))

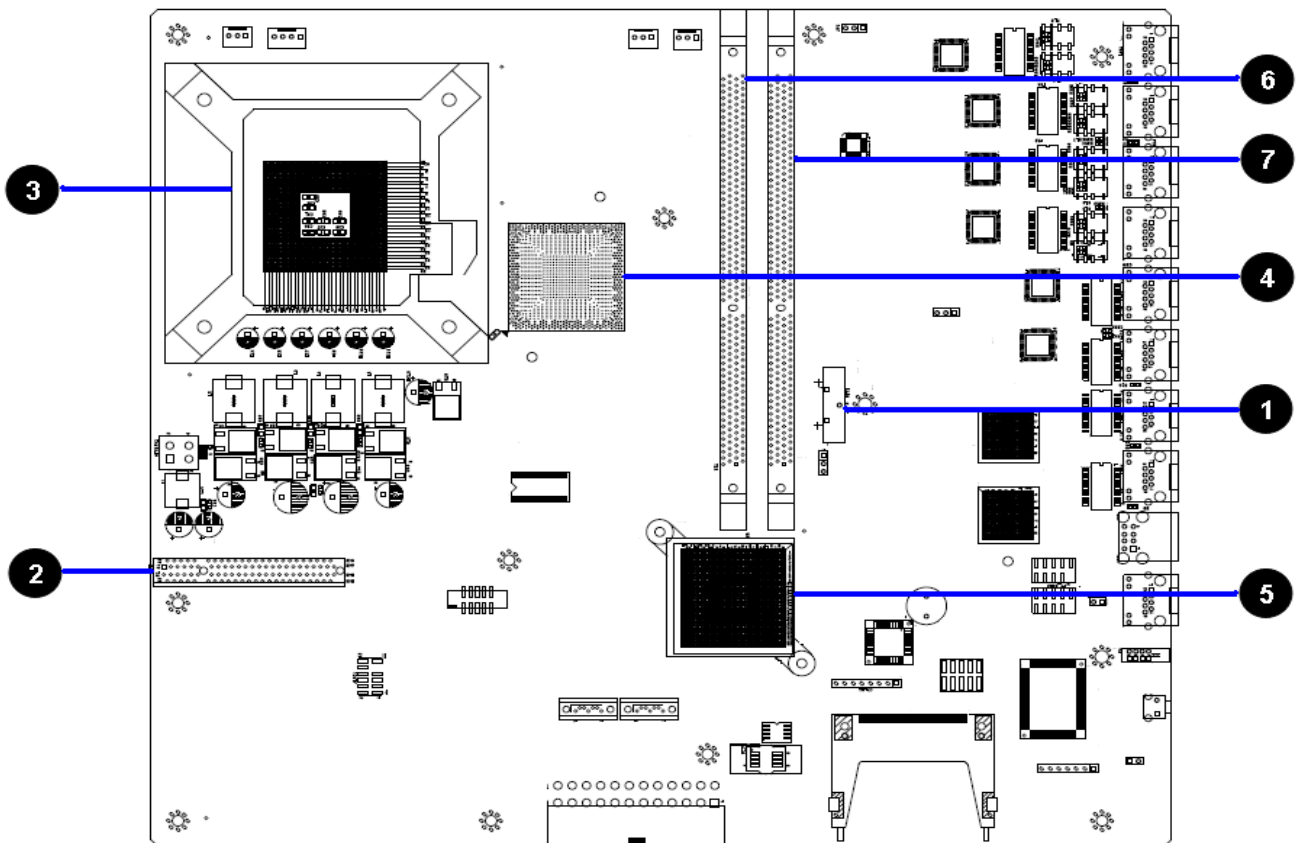
### 1.3 Block Diagram



## 2 H/W Information

This chapter describes the installation of AR-B5800. At first, it shows the Function diagram and the layout of AR-B5800. It then describes the unpacking information which you should read carefully, as well as the jumper/switch settings for the AR-B5800 configuration

### 2.1 Locations (Top side)

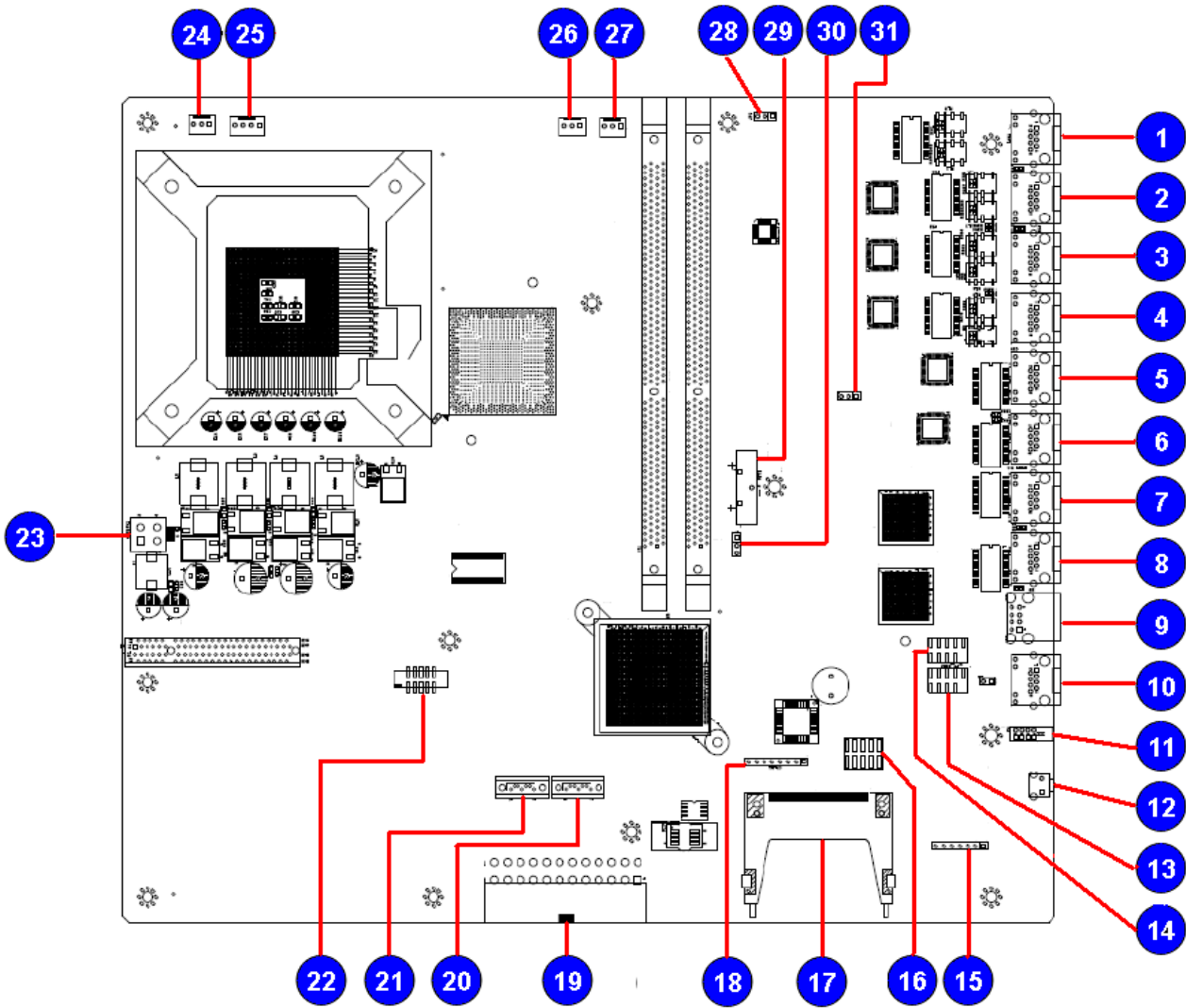


---

①	<b><i>RTC1</i></b> System RTC battery socket	⑤	<b><i>Intel ICH7R</i></b>
②	<b><i>PCIEXP1</i></b> PCI-Express X8 Slot	⑥	<b><i>DIMM1</i></b> 240-Pin DDR3 Socket
③	<b><i>LGA775 CPU Socket</i></b>	⑦	<b><i>DIMM2</i></b> 240-Pin DDR3 Socket
④	<b><i>Intel GMCH 82G41</i></b>		






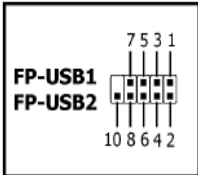
## 2.2 Connectors and Jumper Setting

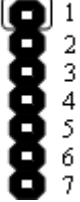

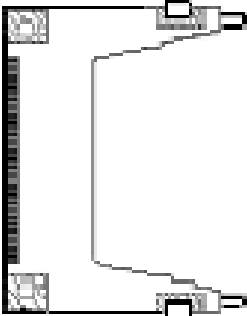
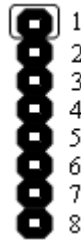
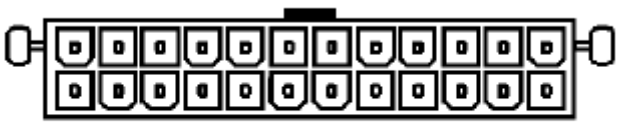
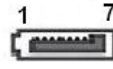
### 2.2.1 Locations (Top side)

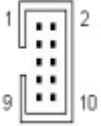
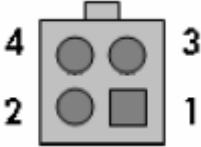









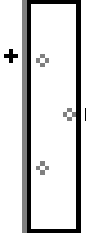






1	LAN1 LAN1 RJ45 Connector	14	FP_USB1 Internal USB2, USB3 connector.	27	SYSFAN1 System FAN Connector.
2	LAN2 LAN2 RJ45 Connector	15	LCM1 Pin Header for LCM	28	JP1 For LAN1/LAN2 Bypass Function Select.
3	LAN3 LAN3 RJ45 Connector	16	GP1 GPIO Header.	29	RTC1 CR2032 Battery Hold Connector.
4	LAN4 LAN4 RJ45 Connector.	17	CF1 CF CARD SOCKET.	30	CCMOS1 CMOS Memory Clearing Header
5	LAN5 LAN5 RJ45 Connector	18	CPLD1 For CPLD Firmware Update	31	JP4 For LAN3/LAN4 Bypass Function Select.
6	LAN6 LAN6 RJ45 Connector	19	ATXPWR1 ATX Power Supply input connector..		
7	LAN7 LAN7 RJ45 Connector.	20	SATA2 SATA device connector #2.		
8	LAN8 LAN8 RJ45 Connector.	21	SATA1 SATA device connector #1.		
9	USB1 Two USB ports (USB0, USB1) connector.	22	VGA1 VGA connector (2x5 Pin Header)		
10	COM1 RS232 Serial Port COM1.(RJ45)	23	ATX12V1 ATX12V Power Supply input Connector.		
11	LED1 4 in 1 LED for LAN Bypass, Power & HDD LED.	24	SYSFAN3 System FAN Connector.		
12	RST2 System Reset Switch.	25	CPUFAN1 CPU FAN Connector.		
13	FP_USB2 Internal USB4, USB5 connector.	26	SYSFAN2 System FAN Connector.		

## 2.3 Connector and Jumper Setting

1. LAN1 ~ LAN8 RJ45 Connector		9. USB1 Connector																									
	<p><b>LAN RJ45 Connector</b> Connects to Local Area Network.</p>		<p><b>External USB Connector</b> Connects to USB devices such as scanner, digital speakers, monitor, mouse, keyboard, hub, digital camera, joystick etc.</p>																								
10. COM1 ( RJ45 Connector )		11. LED1																									
	<p><b>COM Port RJ45 Connector</b></p>		<p><b>Green1: Power ON LED.</b> <b>Green2: HDD LED</b> <b>Yellow1: LAN3&amp;LAN4 Bypass LED.</b> <b>Yellow2: LAN1&amp;LAN2 Bypass LED.</b></p>																								
19. RST2		13,14. FP_USB1 & FP_USB2																									
	<p><b>Push this button to reset the system.</b></p>		<table border="1"> <thead> <tr> <th>Pin</th> <th>Pin Assignment</th> <th>Pin</th> <th>Pin Assignment</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>5VDUAL</td> <td>2</td> <td>5VDUAL</td> </tr> <tr> <td>3</td> <td>Data0 -</td> <td>4</td> <td>Data1 -</td> </tr> <tr> <td>5</td> <td>Data0 +</td> <td>6</td> <td>Data1 +</td> </tr> <tr> <td>7</td> <td>Ground</td> <td>8</td> <td>Ground</td> </tr> <tr> <td></td> <td></td> <td>10</td> <td>NC</td> </tr> </tbody> </table>	Pin	Pin Assignment	Pin	Pin Assignment	1	5VDUAL	2	5VDUAL	3	Data0 -	4	Data1 -	5	Data0 +	6	Data1 +	7	Ground	8	Ground			10	NC
Pin	Pin Assignment	Pin	Pin Assignment																								
1	5VDUAL	2	5VDUAL																								
3	Data0 -	4	Data1 -																								
5	Data0 +	6	Data1 +																								
7	Ground	8	Ground																								
		10	NC																								

15. LCM1 ( for LCM use )		16. GP1 ( GPIO Header )																																									
	<table border="1"> <thead> <tr> <th>Pin</th> <th>SIGNAL</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>Pull-High to VCC5</td> </tr> <tr> <td>2</td> <td>VCC5</td> </tr> <tr> <td>3</td> <td>SOUTB</td> </tr> <tr> <td>4</td> <td>SINB</td> </tr> <tr> <td>5</td> <td>RTSB#</td> </tr> <tr> <td>6</td> <td>CTSB#</td> </tr> <tr> <td>7</td> <td>GND</td> </tr> </tbody> </table>	Pin	SIGNAL	1	Pull-High to VCC5	2	VCC5	3	SOUTB	4	SINB	5	RTSB#	6	CTSB#	7	GND		<table border="1"> <thead> <tr> <th>PIN</th> <th>SIGNAL</th> <th>PIN</th> <th>SIGNAL</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>VCC_GP</td> <td>2</td> <td>GND</td> </tr> <tr> <td>3</td> <td>GP30</td> <td>4</td> <td>GP34</td> </tr> <tr> <td>5</td> <td>GP31</td> <td>6</td> <td>GP35</td> </tr> <tr> <td>7</td> <td>GP32</td> <td>8</td> <td>GP36</td> </tr> <tr> <td>9</td> <td>GP33</td> <td>10</td> <td>GP37</td> </tr> </tbody> </table>	PIN	SIGNAL	PIN	SIGNAL	1	VCC_GP	2	GND	3	GP30	4	GP34	5	GP31	6	GP35	7	GP32	8	GP36	9	GP33	10	GP37
Pin	SIGNAL																																										
1	Pull-High to VCC5																																										
2	VCC5																																										
3	SOUTB																																										
4	SINB																																										
5	RTSB#																																										
6	CTSB#																																										
7	GND																																										
PIN	SIGNAL	PIN	SIGNAL																																								
1	VCC_GP	2	GND																																								
3	GP30	4	GP34																																								
5	GP31	6	GP35																																								
7	GP32	8	GP36																																								
9	GP33	10	GP37																																								
17. CF1 ( CF CARD Socket )		18. CPLD1 ( for CPLD Firmware Update)																																									
			<table border="1"> <thead> <tr> <th>Pin</th> <th>SIGNAL</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>3VDUAL</td> </tr> <tr> <td>2</td> <td>G_TDO</td> </tr> <tr> <td>3</td> <td>G_TDI</td> </tr> <tr> <td>4</td> <td>NC</td> </tr> <tr> <td>5</td> <td>NC</td> </tr> <tr> <td>6</td> <td>G_TMS</td> </tr> <tr> <td>7</td> <td>GND</td> </tr> <tr> <td>8</td> <td>G_TCK</td> </tr> </tbody> </table>	Pin	SIGNAL	1	3VDUAL	2	G_TDO	3	G_TDI	4	NC	5	NC	6	G_TMS	7	GND	8	G_TCK																						
Pin	SIGNAL																																										
1	3VDUAL																																										
2	G_TDO																																										
3	G_TDI																																										
4	NC																																										
5	NC																																										
6	G_TMS																																										
7	GND																																										
8	G_TCK																																										
19. ATXPWR1 ( ATX Power Supply Input )		20,21. SATA1, SATA2 (SATA device connector #1 and #2).																																									
 <p style="text-align: center; color: blue;"><b>ATXPWR1</b></p>			<p><b>To connect SATA device:</b></p> <ol style="list-style-type: none"> <li>Attach either end of the signal cable to the SATA connector on motherboard. Attach the other end to the SATA device.</li> <li>Attach the SATA power cable to the SATA device and connect the other end from the power supply.</li> </ol>																																								

22. VGA1 (2x5pin 2mm Wafer).		23. ATX12V1. (ATX12V Power Input)																			
	VGA Wafer CONNECTOR .																				
24. SYSFAN3 (System FAN connector 3).		25. CPUFAN1 (CPU FAN connector).																			
	<table border="1"> <thead> <tr> <th>PIN</th> <th>SIGNAL</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>GND</td> </tr> <tr> <td>2</td> <td>+12V</td> </tr> <tr> <td>3</td> <td>Fan speed data</td> </tr> </tbody> </table>	PIN	SIGNAL	1	GND	2	+12V	3	Fan speed data		<table border="1"> <thead> <tr> <th>PIN</th> <th>SIGNAL</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>GND</td> </tr> <tr> <td>2</td> <td>+12V</td> </tr> <tr> <td>3</td> <td>Fan speed data</td> </tr> <tr> <td>4</td> <td>CPU FAN PWM Control</td> </tr> </tbody> </table> <p>ON/OFF controlled by CPU temperature setting of BIOS.</p>	PIN	SIGNAL	1	GND	2	+12V	3	Fan speed data	4	CPU FAN PWM Control
PIN	SIGNAL																				
1	GND																				
2	+12V																				
3	Fan speed data																				
PIN	SIGNAL																				
1	GND																				
2	+12V																				
3	Fan speed data																				
4	CPU FAN PWM Control																				
26. SYS_FAN2 (System FAN connector 2).		27. SYS_FAN1 (System FAN connector 1).																			
	<table border="1"> <thead> <tr> <th>PIN</th> <th>SIGNAL</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>GND</td> </tr> <tr> <td>2</td> <td>+12V</td> </tr> <tr> <td>3</td> <td>Fan speed data</td> </tr> </tbody> </table>	PIN	SIGNAL	1	GND	2	+12V	3	Fan speed data		<table border="1"> <thead> <tr> <th>PIN</th> <th>SIGNAL</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>GND</td> </tr> <tr> <td>2</td> <td>+12V</td> </tr> <tr> <td>3</td> <td>Fan speed data</td> </tr> </tbody> </table>	PIN	SIGNAL	1	GND	2	+12V	3	Fan speed data		
PIN	SIGNAL																				
1	GND																				
2	+12V																				
3	Fan speed data																				
PIN	SIGNAL																				
1	GND																				
2	+12V																				
3	Fan speed data																				

28. JP1 (For LAN1/LAN2 Bypass Function Select.).		29. RTC1	
	<div style="border: 1px solid black; padding: 5px; margin-bottom: 10px;"> <p>Normal (Default)  3 2 1</p> <p>By CPLD  3 2 1</p> <p style="color: blue;"><b>Bypass Setting</b></p> </div> <p><b>Pins 1 and 2 shorted</b> (Default): <b>Forced Normal.</b></p> <p><b>Pins 2 and 3 shorted:</b> Controlled By CPLD. <b>Otherwise : Forced Bypass</b></p>		<p><b>CMOS Backup Battery:</b></p> <p>An onboard battery saves the CMOS memory to keep the BIOS information stays on even after disconnected your system with power source. Nevertheless, this backup battery exhausts after some five years.</p> <p>Once the error message like “<b>CMOS BATTERY HAS FAILED</b>” or “<b>CMOS checksum error</b>” displays on monitor, this backup battery is no longer functional and has to be renewed</p>
30. CCMOS1.		31. JP4 (For LAN3/LAN4 Bypass Function Select.).	
	<p><b>Pins 1 and 2 shorted (Default):</b> Normal operation.</p> <p><b>Pins 2 and 3 shorted:</b> Clear CMOS memory.</p> <div style="border: 1px solid black; padding: 5px; margin-top: 10px;"> <p>Normal (Default)  3 2 1</p> <p>Clear CMOS  3 2 1</p> <p style="text-align: center;"><b>CCMOS1</b></p> </div>		<div style="border: 1px solid black; padding: 5px; margin-bottom: 10px;"> <p>Normal (Default)  3 2 1</p> <p>By CPLD  3 2 1</p> <p style="color: blue;"><b>Bypass Setting</b></p> </div> <p><b>Pins 1 and 2 shorted</b> (Default): <b>Forced Normal.</b></p> <p><b>Pins 2 and 3 shorted:</b> Controlled By CPLD. <b>Otherwise : Forced Bypass</b></p>

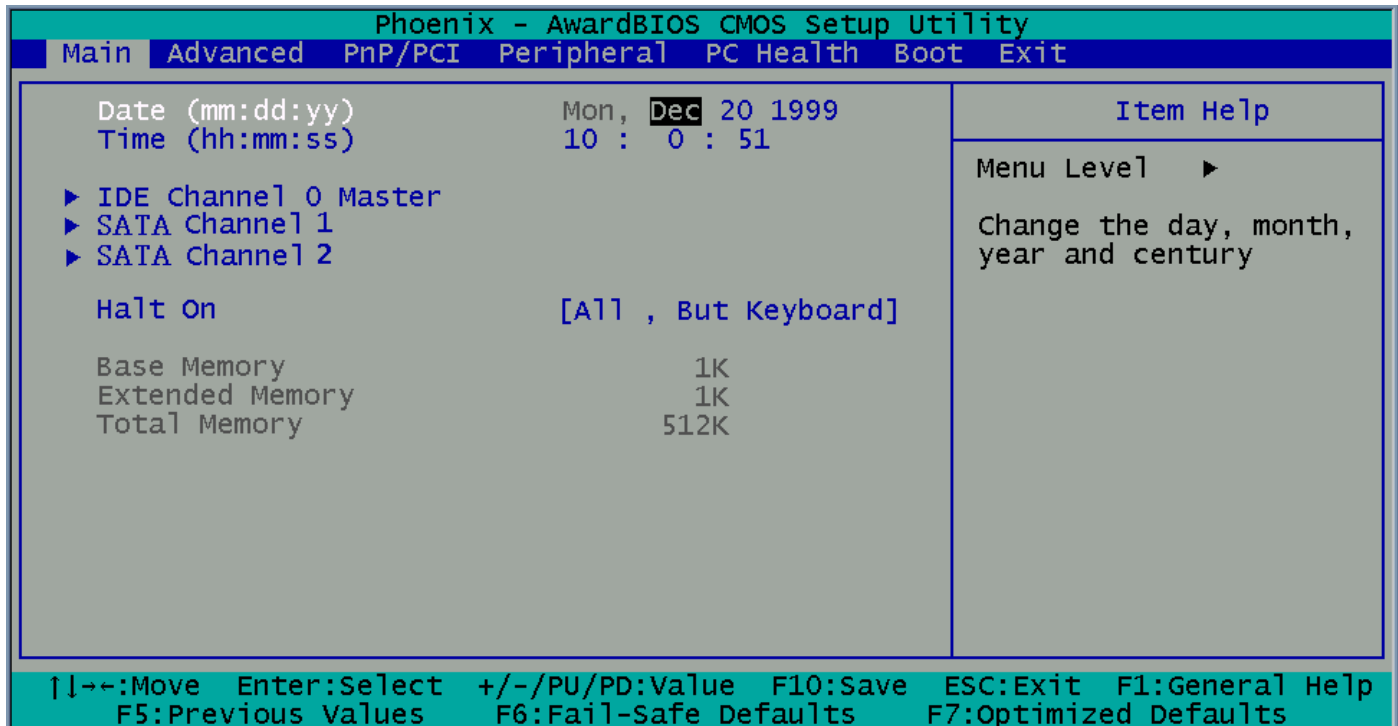
# 3 Bios setting

This chapter describes the BIOS menu displays and explains how to perform common tasks needed to get the system up and running. It also gives detailed explanation of the elements found in each of the BIOS menus. The following topics are covered:

- Main Setup
- Advanced Chipset Setup
- Peripherals Setup
- PnP/PCI Setup
- PC Health Setup
- Boot Setup
- Exit Setup

### 3.1 Main Setup

Once you enter the Award BIOS™ CMOS Setup Utility, the Main Menu will appear on the screen. Use the arrow keys to highlight the item and then use the <Pg Up> <Pg Dn> keys to select the desired value in each item.



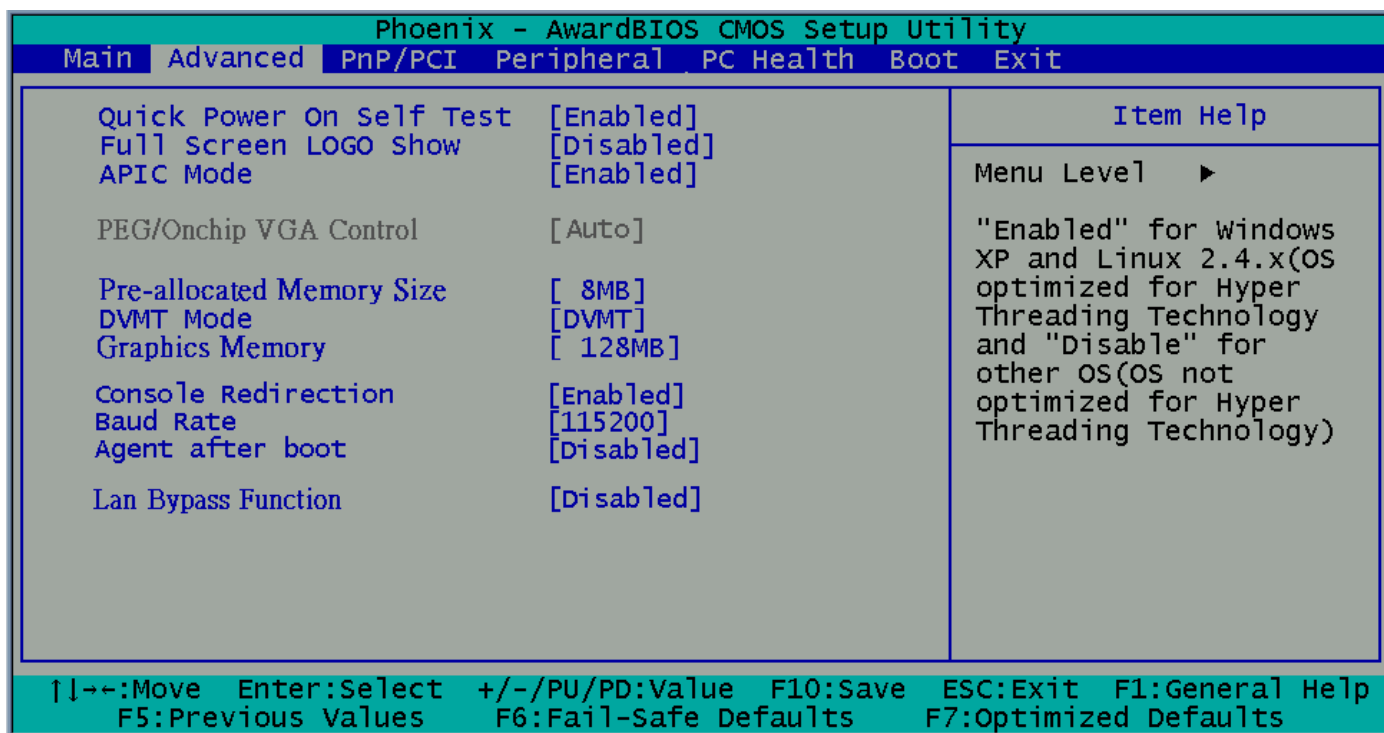
Note: The control keys are listed at the bottom of the menu. If you need any help with the item fields, you can press the <F1> key, and the relevant information will be displayed.

Option	Choice	Description
<b>Date Setup</b>	N/A	Set the system date. Note that the 'Day' automatically changes when you set the date.
<b>Time Setup</b>	N/A	Set the system time.
<b>IDE Channel 0 Master</b>	N/A	The onboard CF connectors provide one channel for connecting one CF CARD Only the BIOS will auto-detect the CF type.

<b>SATA Channel 1/2</b>	N/A	The onboard SATA connectors provide 1 channel for connecting one SATA hard disks, the BIOS will auto-detect the SATA type.
<b>Halt On</b>	All Errors, No Errors, All but keyboard.	Select the situation in which you want the BIOS to stop the POST process and notify you.

### 3.2 Advanced Chipset Setup

This section allows you to configure and improve your system and follows you to set up some system features according to your preference.

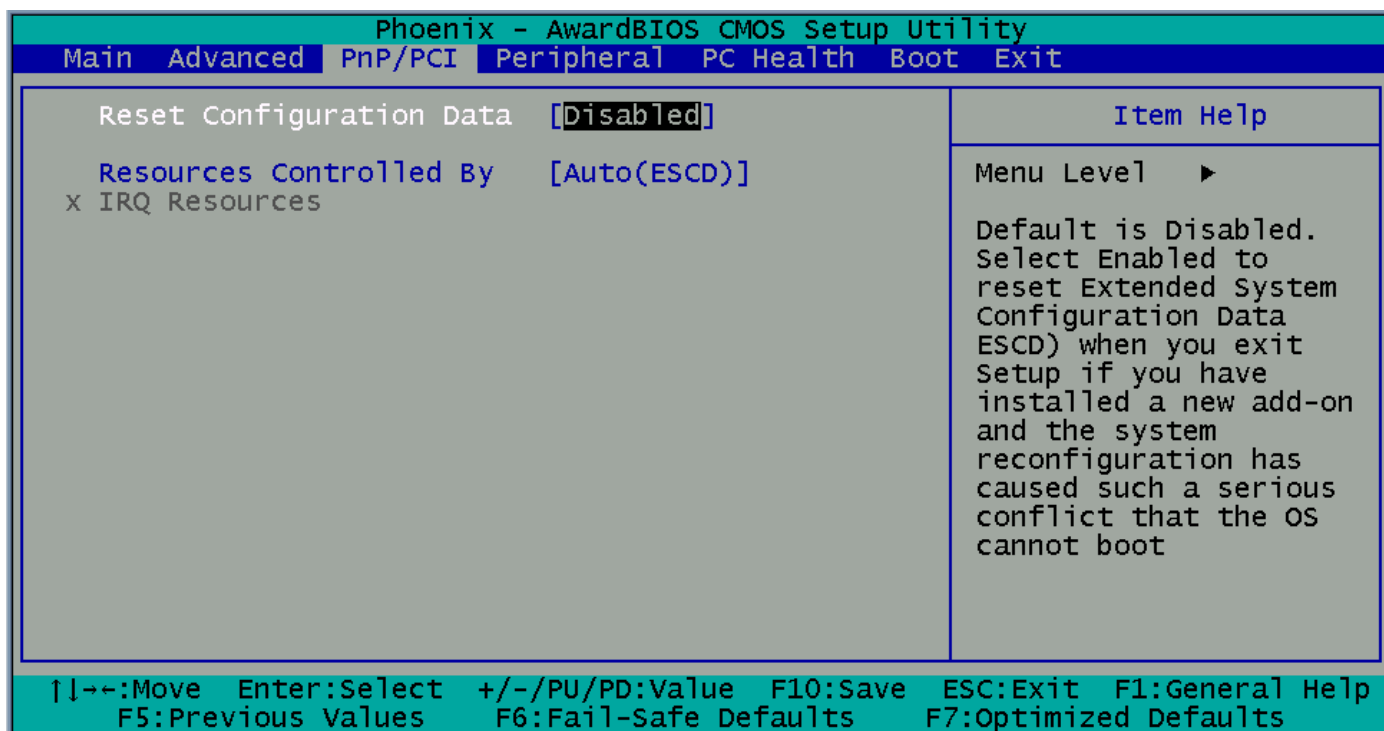


Option	Choice	Description
<b>Quick Power On Self Test</b>	Enabled Disabled	This category speeds up Power On Self Test (POST) after you have powered up the computer. If it is set to Enable, BIOS will shorten or skip some check items during POST.

<b>Full Screen Logo Show</b>	Enabled Disabled	Select Enabled to show the OEM full screen logo if you have add-in BIOS.
<b>APIC Mode</b>	Enabled Disabled	Advanced Programmable Interrupt Controller. This item [Enabled] for more system INTerrupts that AR-B5800 required.
<b>Pre-allocated Memory Size</b>	32Mb 64Mb 128Mb	This Item is for setting the Frame Buffer (Share system memory as display memory).
<b>DVMT Mode</b>	Enabled Disabled	Dynamic Video Memory Technology. [Enabled] for optimizing amount of memory is located for balanced graphics and system performance.
<b>Console Redirection</b>	Enabled Disabled	[Enabled] for user who want to remote control the system via serial port.
<b>Baud Rate</b>	115200(Max)	The baud rate of remote control machine should the same as the system for communication.
<b>LAN Bypass Function</b>	Enabled Disabled	For user Enable/Disable LAN Bypass Function !!.

### 3.3 PnP/PCI Setup

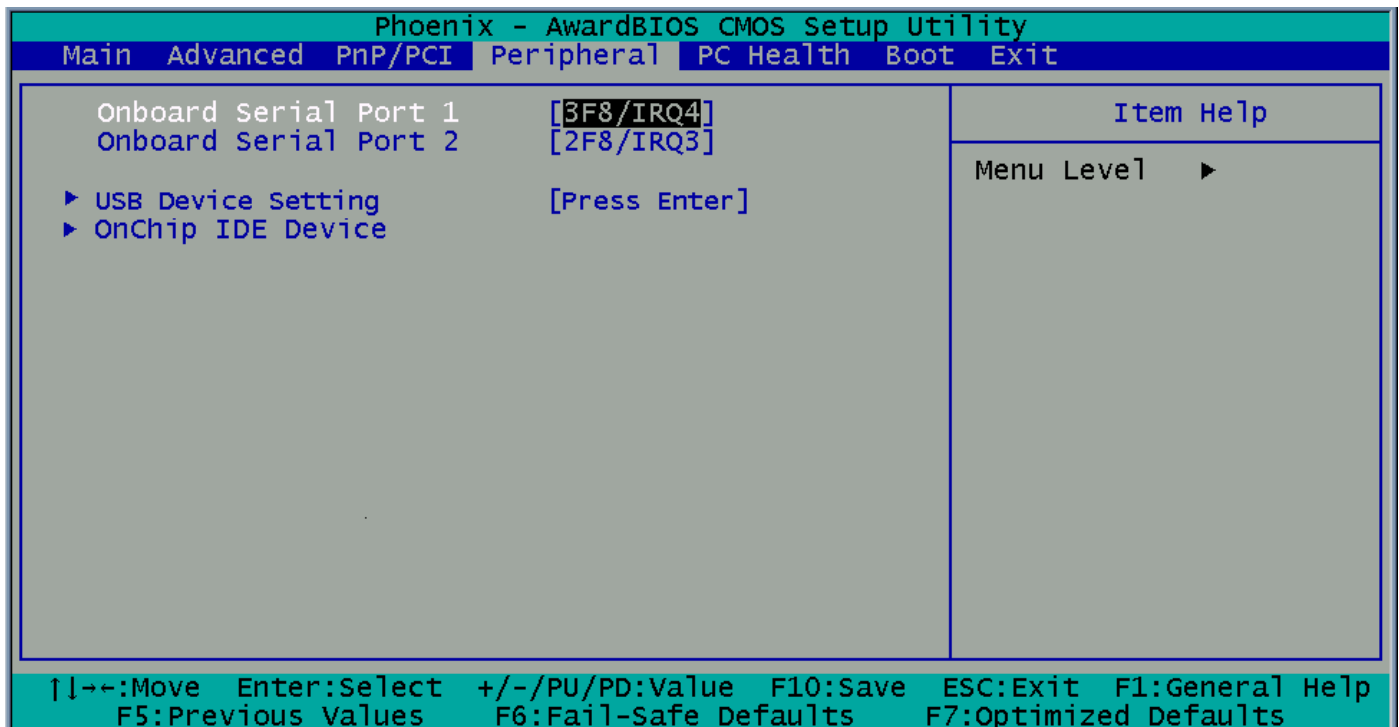
The option configures the PCI bus system. All PCI bus system on the system use INT#, thus all installed PCI cards must be set to this value.



Option	Choice	Description
<b>Reset Configuration Data</b>	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 and the system reconfiguration has caused such a serious conflict, then the operating system cannot boot.
<b>Resources Controlled By</b>	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," then you may choose specific resources by going into each of the submenus.
<b>IRQ Resources</b>	N/A	When resources are controlled manually, assign a type to each system interrupt, depending on the type of the device that uses the interrupt

### 3.4 Peripherals Setup

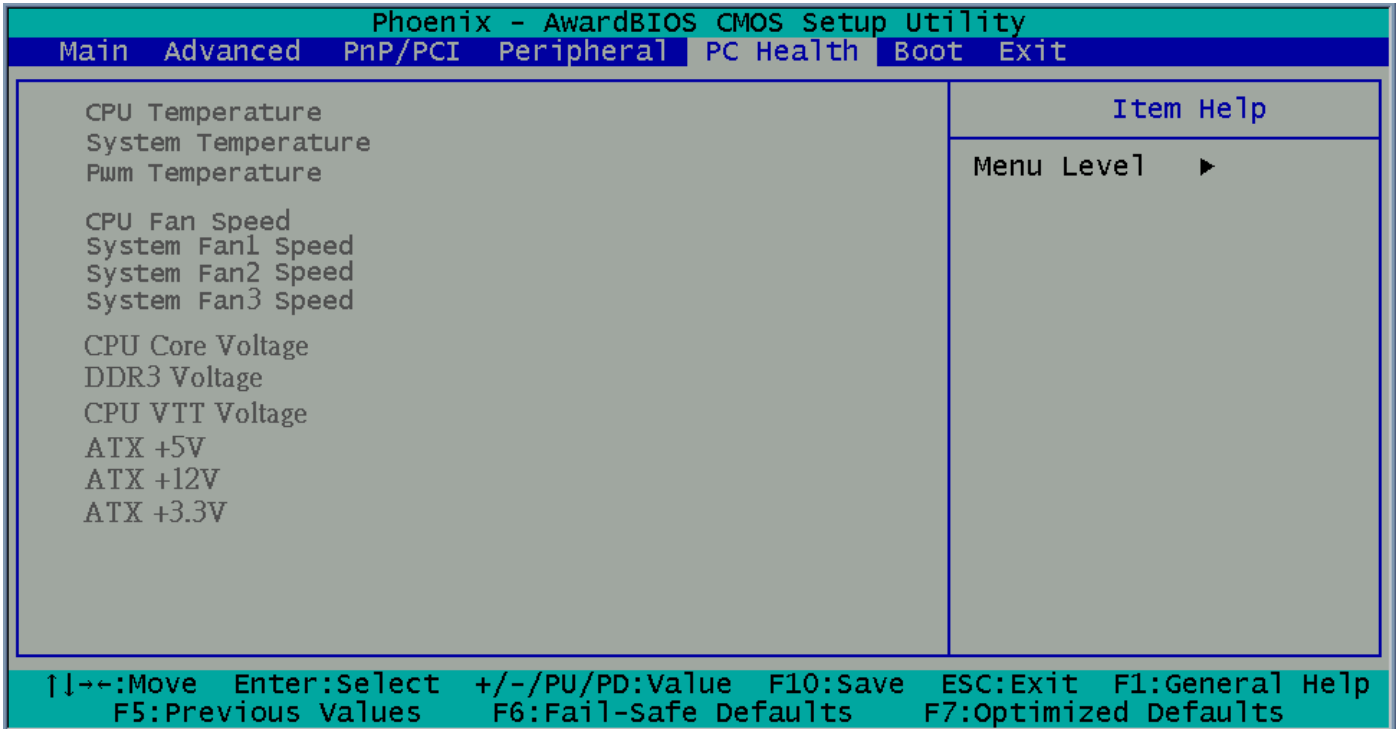
This option controls the configuration of the board's chipset. Control keys for this screen are the same as for the previous screen.



Option	Choice	Description
<b>Onboard Serial Port 1</b> <b>Onboard Serial Port 2</b>	Serial Port 1: 3F8 / IRQ4 Serial Port 2: 2F8 / IRQ3	Select an address and the corresponding interrupt for each serial port.
<b>USB Device Setting</b>	Enter to Select USB Device Setting	
<b>USB 1.0 Controller</b>	Enabled Disabled	Select Enabled if your system contains a Universal Serial Bus (USB) 1.0 controller and you have USB peripherals
<b>USB 2.0 Controller</b>	Enabled Disabled	Select Enabled if your system contains a Universal Serial Bus (USB) 2.0 controller and you have USB peripherals
<b>USB Operation Mode</b>	Full/Low Speed High Speed	Auto decide USB device operation mode. [High Speed]: If USB device was high speed device, then it operated on high speed mode. If USB device was full/Low speed device, then it operated on full/low speed mode. [Full/Low Speed]: All of USB device operated on full/low speed mode.
<b>USB Keyboard Function</b>	Enabled Disabled	[Enable] or [Disable] Legacy Support of USB Keyboard
<b>USB Storage Function</b>	Enabled Disabled	[Enable] or [Disable] Legacy Support of USB Storage
<b>On chip IDE DEVICE</b>	Enabled Disabled	The integrated peripheral controller contains an IDE interface with support for two IDE channels. Select Enabled to activate each channel separately.

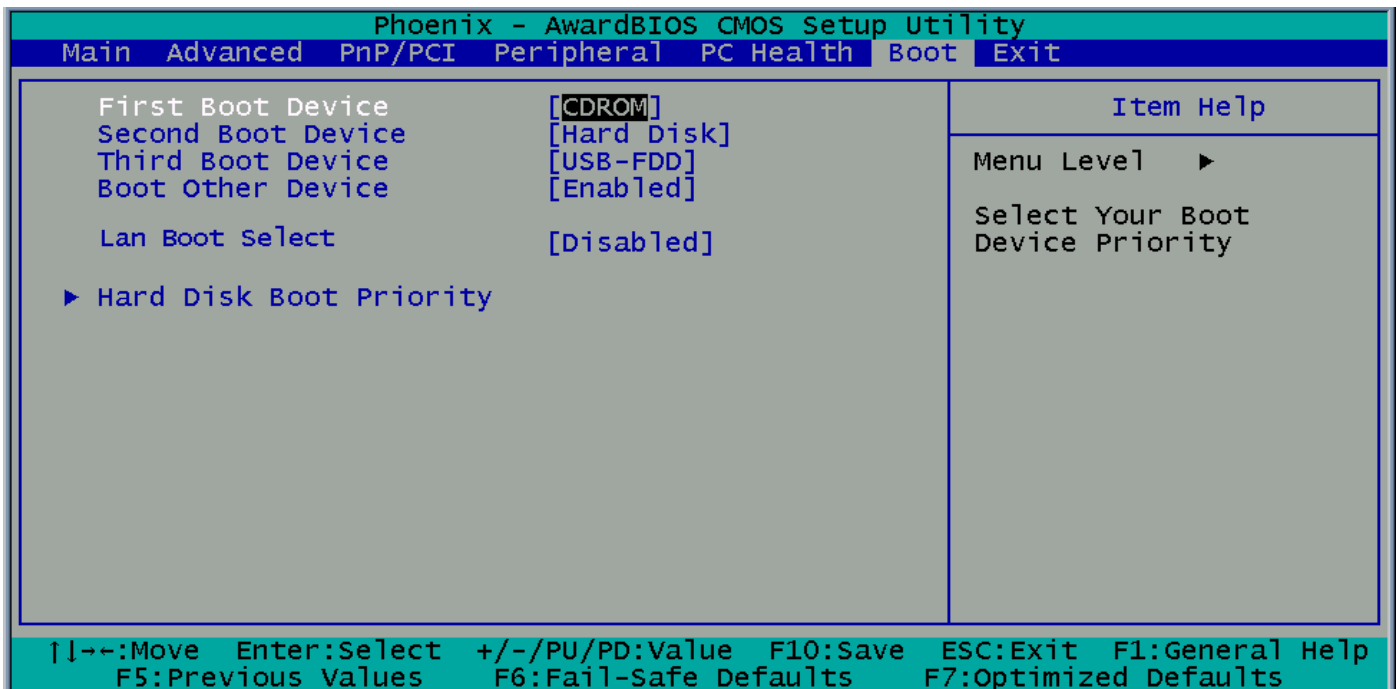
### 3.5 PC Health Setup

This section shows the parameters in determining the PC Health Status. These parameters include temperatures, fan speeds, and voltages.



### 3.6 Boot Setup

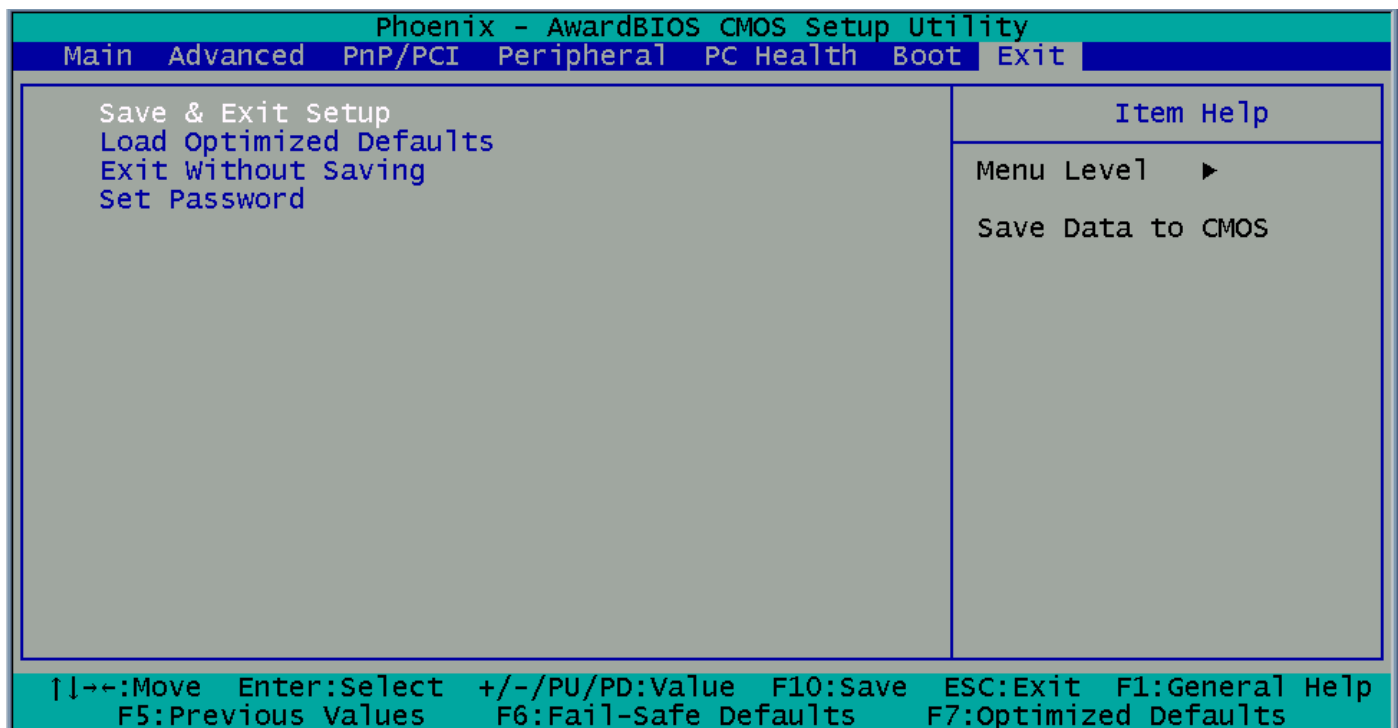
This section is used to exit the BIOS main menu. After making your changes, you can either save them or exit the BIOS menu and without saving the new values.



Option	Choice	Description
<b>First / Second / Third Boot Device/Other Boot Device</b>	Hard Disk CDROM USB-FDD USB-CDROM LAN Disabled	The BIOS attempts to load the operating system from the devices in the sequence selected in these items.
<b>LAN Boot Select</b>	Enabled Disabled	These fields allow the system to search for an OS from LAN
<b>Hard Disk Boot Priority</b>	N/A	These fields set the Boot Priority for each Hard Disk

### 3.7 Exit Setup

This section is used to configure exit mode.



Option	Choice	Description
<b>Save &amp; Exit Setup</b>	Pressing <Enter> on this item for confirmation:  <b>Save to CMOS and EXIT (Y/N)? Y</b>	Press “Y” to store 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
<b>Load Optimized Defaults</b>	When you press <Enter> on this item you get a confirmation dialog box with a message like this:  <b>Load Optimized Defaults (Y/N)? N</b>	Press ‘Y’ to load the default values that are factory-set for optimal-performance system operations.
<b>Exit Without Saving</b>	Pressing <Enter> on this item for confirmation:  <b>Quit without saving (Y/N)? Y</b>	This allows you to exit Setup without storing any changes in CMOS. The previous selections remain in effect. This shall exit the Setup utility and restart your computer.
<b>Set Password</b>	Pressing <Enter> on this item for confirmation:  <b>ENTER PASSWORD:</b>	When a password has been enabled, you will be prompted to enter your password every time you try to enter Setup. This prevents unauthorized persons from changing any part of your system configuration.  Type the password, up to eight characters in length, and press <Enter>. The password typed now will clear any previous password from the 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>

		<p>when you are prompted to enter the password. A message will confirm that the password will be disabled. Once the password is disabled, the system will boot and you can enter Setup freely.</p>
--	--	--

### 3.8 BIOS Update

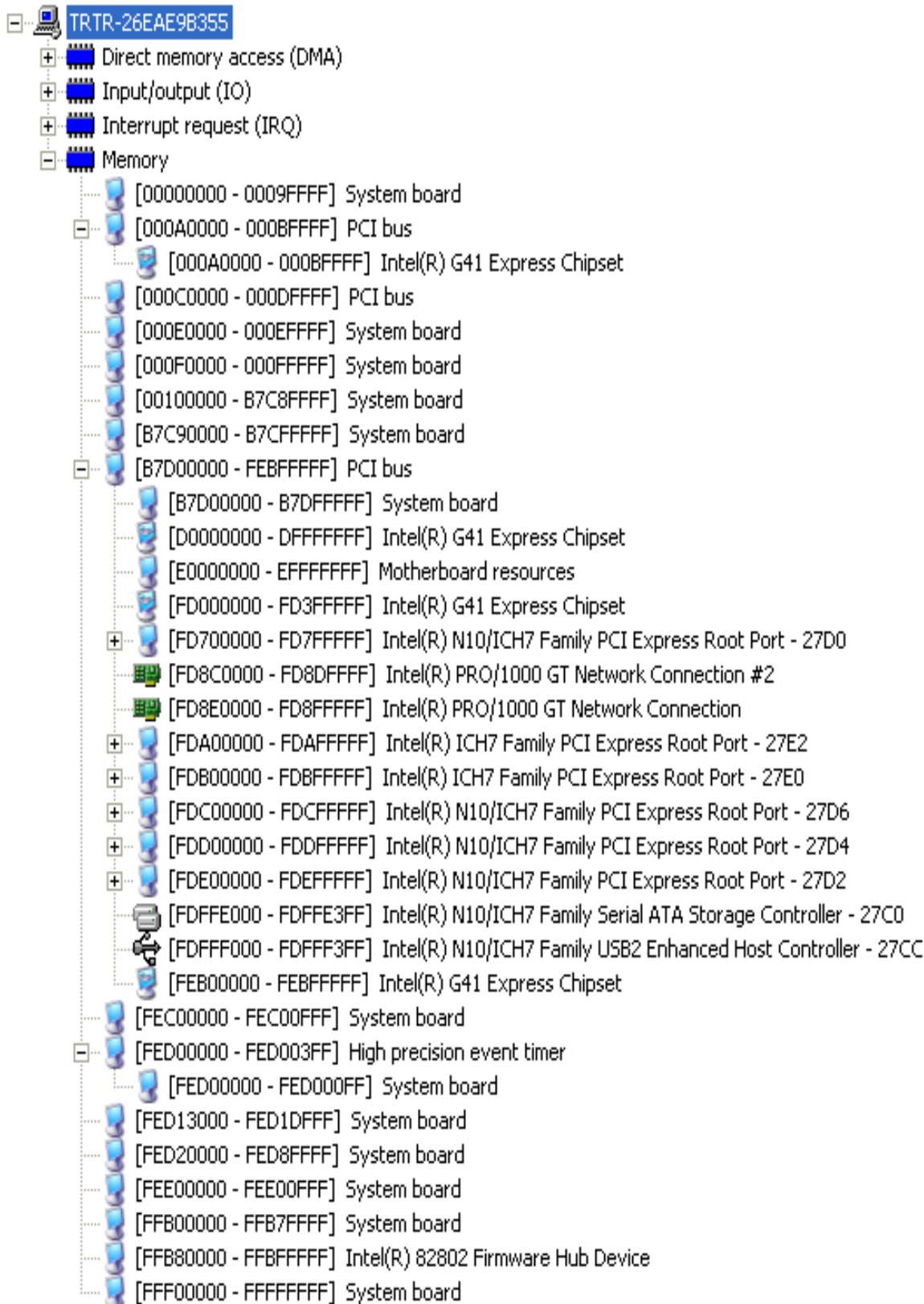
The BIOS program instructions are contained within computer chips called FLASH ROMs that are located on your system board. The chips can be electronically reprogrammed, allowing you to update your BIOS firmware without removing and installing chips.

The AR-B5800 provides the FLASH BIOS update function for you to easily to update BIOS.

Please follow these operating steps to update BIOS:

Step 1:	You must boot up system into MS-DOS first and please don't detect files CONFIG.SYS and AUTOEXEC.BAT.
Step 2:	In the MS-DOS mode, you should execute the AWDFLASH program to update BIOS.
Step 3:	Follow all messages then you can update BIOS smoothly.

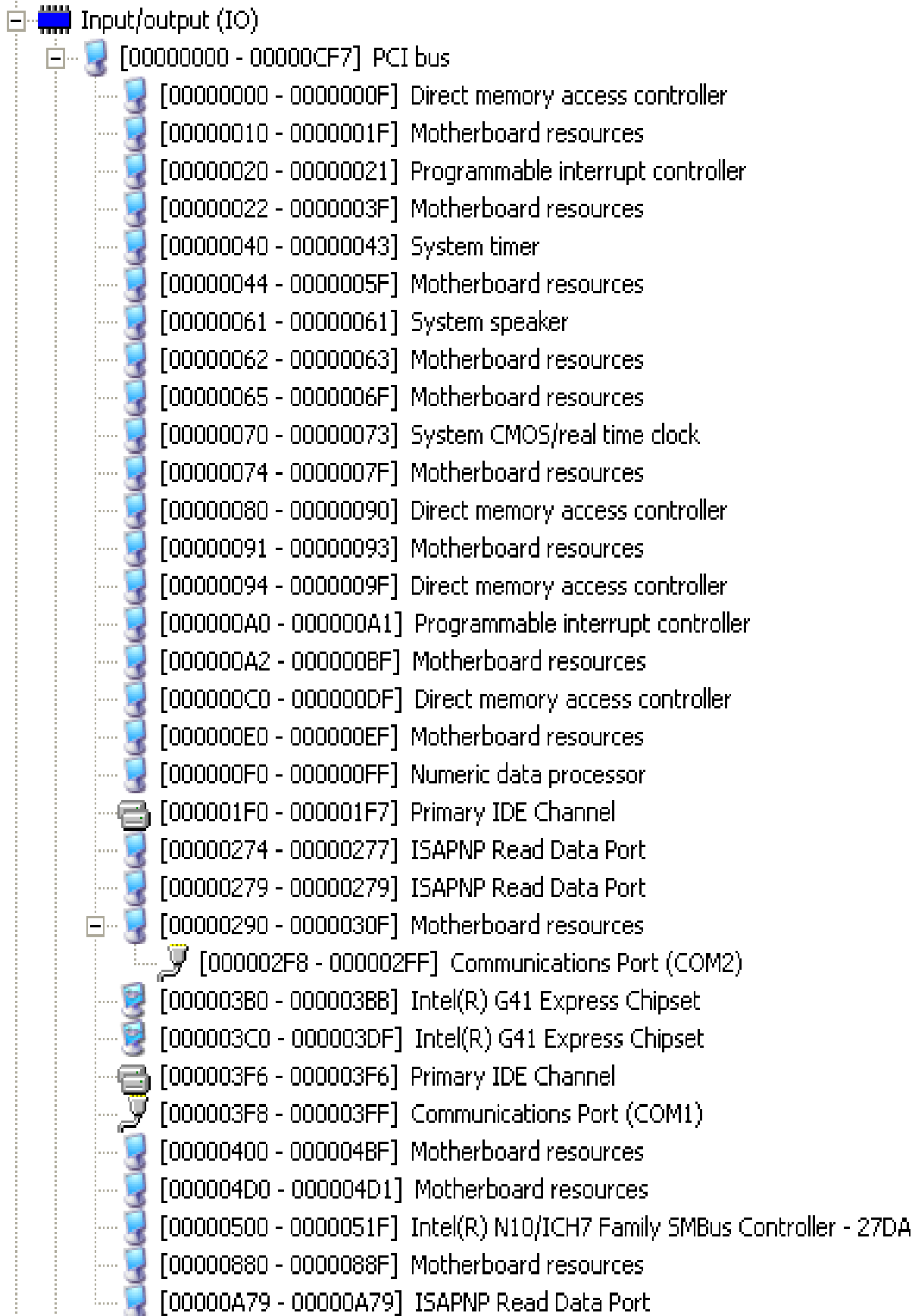
## APPENDIX A. MEMORY MAP



## APPEXDIX B. IRQ MAP

Interrupt request (IRQ)	
(ISA) 0	High precision event timer
(ISA) 3	Communications Port (COM2)
(ISA) 4	Communications Port (COM1)
(ISA) 8	High precision event timer
(ISA) 9	Microsoft ACPI-Compliant System
(ISA) 13	Numeric data processor
(ISA) 14	Primary IDE Channel
(PCI) 15	Intel(R) N10/ICH7 Family SMBus Controller - 27DA
(PCI) 16	Intel(R) 82574L Gigabit Network Connection #2
(PCI) 16	Intel(R) 82574L Gigabit Network Connection #5
(PCI) 16	Intel(R) G41 Express Chipset
(PCI) 16	Intel(R) ICH7 Family PCI Express Root Port - 27E0
(PCI) 16	Intel(R) N10/ICH7 Family PCI Express Root Port - 27D0
(PCI) 16	Intel(R) N10/ICH7 Family USB Universal Host Controller - 27CB
(PCI) 17	Intel(R) 82574L Gigabit Network Connection #3
(PCI) 17	Intel(R) 82574L Gigabit Network Connection #6
(PCI) 17	Intel(R) ICH7 Family PCI Express Root Port - 27E2
(PCI) 17	Intel(R) N10/ICH7 Family PCI Express Root Port - 27D2
(PCI) 18	Intel(R) 82574L Gigabit Network Connection
(PCI) 18	Intel(R) N10/ICH7 Family PCI Express Root Port - 27D4
(PCI) 18	Intel(R) N10/ICH7 Family USB Universal Host Controller - 27CA
(PCI) 19	Intel(R) 82574L Gigabit Network Connection #4
(PCI) 19	Intel(R) N10/ICH7 Family PCI Express Root Port - 27D6
(PCI) 19	Intel(R) N10/ICH7 Family Serial ATA Storage Controller - 27C0
(PCI) 19	Intel(R) N10/ICH7 Family USB Universal Host Controller - 27C9
(PCI) 20	Intel(R) PRO/1000 GT Network Connection #2
(PCI) 23	Intel(R) N10/ICH7 Family USB Universal Host Controller - 27C8
(PCI) 23	Intel(R) N10/ICH7 Family USB2 Enhanced Host Controller - 27CC
(PCI) 23	Intel(R) PRO/1000 GT Network Connection

## APPENDIX C. I/O PORT MAP



- [-] [00000D00 - 0000FFFF] PCI bus
  - [+] [00008E00 - 00008E3F] Intel(R) PRO/1000 GT Network Connection #2
  - [+] [00008F00 - 00008F3F] Intel(R) PRO/1000 GT Network Connection
  - [-] [00009000 - 00009FFF] Intel(R) N10/ICH7 Family PCI Express Root Port - 27D6
    - [+] [00009F00 - 00009F1F] Intel(R) 82574L Gigabit Network Connection #4
  - [-] [0000A000 - 0000AFFF] Intel(R) N10/ICH7 Family PCI Express Root Port - 27D4
    - [+] [0000AF00 - 0000AF1F] Intel(R) 82574L Gigabit Network Connection
  - [-] [0000B000 - 0000BFFF] Intel(R) N10/ICH7 Family PCI Express Root Port - 27D2
    - [+] [0000BF00 - 0000BF1F] Intel(R) 82574L Gigabit Network Connection #3
  - [-] [0000C000 - 0000CFFF] Intel(R) N10/ICH7 Family PCI Express Root Port - 27D0
    - [+] [0000CF00 - 0000CF1F] Intel(R) 82574L Gigabit Network Connection #2
  - [-] [0000D000 - 0000DFFF] Intel(R) ICH7 Family PCI Express Root Port - 27E2
    - [+] [0000DF00 - 0000DF1F] Intel(R) 82574L Gigabit Network Connection #6
  - [-] [0000E000 - 0000EFFF] Intel(R) ICH7 Family PCI Express Root Port - 27E0
    - [+] [0000EF00 - 0000EF1F] Intel(R) 82574L Gigabit Network Connection #5
  - [+] [0000F500 - 0000F50F] Intel(R) N10/ICH7 Family Serial ATA Storage Controller - 27C0
  - [+] [0000F600 - 0000F603] Intel(R) N10/ICH7 Family Serial ATA Storage Controller - 27C0
  - [+] [0000F700 - 0000F707] Intel(R) N10/ICH7 Family Serial ATA Storage Controller - 27C0
  - [+] [0000F800 - 0000F803] Intel(R) N10/ICH7 Family Serial ATA Storage Controller - 27C0
  - [+] [0000F900 - 0000F907] Intel(R) N10/ICH7 Family Serial ATA Storage Controller - 27C0
  - [+] [0000FA00 - 0000FA0F] Intel(R) ICH7 Family Ultra ATA Storage Controllers - 27DF
  - [+] [0000FB00 - 0000FB1F] Intel(R) N10/ICH7 Family USB Universal Host Controller - 27CB
  - [+] [0000FC00 - 0000FC1F] Intel(R) N10/ICH7 Family USB Universal Host Controller - 27CA
  - [+] [0000FD00 - 0000FD1F] Intel(R) N10/ICH7 Family USB Universal Host Controller - 27C9
  - [+] [0000FE00 - 0000FE1F] Intel(R) N10/ICH7 Family USB Universal Host Controller - 27C8
  - [+] [0000FF00 - 0000FF07] Intel(R) G41 Express Chipset

# 4 Software Installation and Programming Guide

## 4.1 Introduction

### LCD Control Module

#### Overview

The LCM (short for LCD Control Module) APIs provide interfaces to control the module. By invoking these APIs, programmers can implement the applications which have the functions listed below:

1. Clear LCD screen.
2. Turn on or off the cursor on the screen.
3. Move the cursor on the screen.
4. Turn on or off the text on the screen.
5. Get the identification of the pressed key of the LCM.
6. Show the text on the screen.

### GPIO and Watchdog

#### Overview

AR-B5800 provides both a GPIO interface and a Watchdog timer. Users can use the GPIO and Watchdog APIs to configure and to access the GPIO interface and the Watchdog timer. The GPIO has eight ports. Users can configure each pin as input or output respectively. The Watchdog timer can be set to 1~255 seconds. Setting the timer to zero disables the timer. The remaining seconds of the timer to reboot can be read from the timer.

In this GPIO and Watchdog package, we provides:

1. API source code.
2. GPIO and Watchdog test utility and the utility source code.

Here is the GPIO Mapping Table:

<b>Pin Name on SIO</b>	<b>I/O Direction</b>	<b>Signal Name in Source Code</b>
GP30	<b>Configurable</b>	<b>GPIO0</b>
GP31	<b>Configurable</b>	<b>GPIO1</b>
GP32	<b>Configurable</b>	<b>GPIO2</b>
GP33	<b>Configurable</b>	<b>GPIO3</b>

---

GP34	<b>Configurable</b>	<b>GPIO4</b>
GP35	<b>Configurable</b>	<b>GPIO5</b>
GP36	<b>Configurable</b>	<b>GPIO6</b>
GP37	<b>Configurable</b>	<b>GPIO7</b>

In the GPIO APIs, we use the signal names ‘GPIO0’, ‘GPIO1’...etc. to identify the GPIO ports.

## **LAN Bypass Subsystem**

### **Overview**

Two pairs of LAN ports on AR-B5800 implement the bypass function. Users can invoke the LAN Bypass APIs to control the bypass states of the LAN ports and set up the LAN Bypass Watchdog timer ( this watchdog timer is different from the system watchdog timer mentioned in the previous paragraph ).

### **AR-B5800 Library (Window platform only)**

The released code for Windows platform includes a folder called ‘AR-B5800’. In this folder, there are header files and source codes of all the APIs of LCM module, GPIO, Watchdog, and LAN Bypass functions. The source codes in this folder generate the API library ‘AR-B5800.lib’ and ‘AR-B5800.dll’. Users who want to invoke the APIs can include the ‘AR-B5800.h’ in their application source code and compile their application with the library ‘AR-B5800.lib’ or ‘AR-B5800.dll’.

Besides AR-B5800 libraries, there are WinIo32.sys, WinIo32.lib, and WinIo32.dll in this directory. These files are indispensable for LAN Bypass application, GPIO and Watchdog application. Please put these three files with LAN Bypass application, GPIO and Watchdog application in the same folder. The APIs invoke WinIo32.lib, WinIo32.dll, and WinIo32.sys implicitly.

## 4.2 File Descriptions

### LCD Control Module

On Linux platform:

1. `lcmdemo.c`

This file is the source code of the demo program. This program displays the user interface, processes user's input, and invokes LCM APIs to demonstrates the functions of LCM.

2. `lcm.c`

This file includes the hardware independent implementation of LCM APIs. All the APIs in this file invoke the hardware dependent functions 'InitSerialPort()', 'WriteSerial()', 'ReadSerial()' and 'CloseSerialPort()' for accessing the serial port

3. `lcm.h`

This file includes the declarations and macro definitions needed by `lcm.c`.

4. `serialport.c`

This file includes the hardware dependent implementation of 'InitSerialPort()', 'WriteSerial()', 'ReadSerial()' and 'CloseSerialPort()' for accessing the serial port.

5. `serialport.h`

This file includes the declarations and macro definitions needed by `serialport.c`.

6. `Makefile`

This is the instruction script for GNU make system.

On Window platform:

1. `LCM.cpp`

The source code of the LCM demo program.

2. `AR-B5800.h`

The header of the APIs.

3. `AR-B5800.lib` and `AR-B5800.dll`

The API libraries.

### GPIO and Watchdog

On Linux platform:

1. `sio_acce.c`  
The source code of the Watchdog and GPIO APIs for accessing the SuperIO.
2. `sio_acce.h`  
This file includes the declarations of the APIs and macro definitions.
3. `main.c`  
The source code of the utility.
4. Makefile

On Windows platform:

1. `GPIO_Watchdog.cpp`  
GPIO and Watchdog demo program source code.
2. `AR-B5800.h`  
The header file of the APIs.
3. `AR-B5800.lib` and `AR-B5800.dll`  
The API libraries.

### **LAN Bypass Subsystem**

On Linux platform:

1. `bypass.c`  
The source code of the APIs for setting up the bypass state.
2. `bypass.h`  
This file includes the declarations of the APIs and macro definitions.
3. `main.c`  
The source code of the utility.
4. Makefile

On Windows platform:

1. LAN\_Bypass.cpp  
LAN\_Bypass demo program source code.
  
2. AR-B5800.h  
The header file of the APIs.
  
3. AR-B5800.lib and AR-B5800.dll  
The API libraries.

## 4.3 API List and Descriptions

### Type Definitions

```
Typedef   char           i8;  
Typedef  unsigned char   u8;  
Typedef  short           i16;  
Typedef  unsigned short  u16;  
Typedef  unsigned long   u32;  
Typedef  int             i32;
```

### LCD Control Module

1. i32 clrscrLcm( void )

**Description:** Clear the screen of the LCM.

**Return value:** 0 after the screen is cleared.

2. i32 cursorLcm( bool mode )

**Description:** According to the argument 'mode', show the cursor on the LCM screen or eliminate the cursor on the LCM screen. The position of the cursor is unchanged.

mode = true, show the cursor.

mode = false, eliminate the cursor.

**Return value:** 0 after the cursor has been shown or eliminated.

3. i32 cursorActionLcm( i32 type)

**Description:** According to the argument 'type', move the cursor to the indicated position. The displayed text is not altered.

type = HOME, move the cursor to row 0, column 0.

type = MOVERIGHT, move the cursor to the column which is to the right of its original position if the original column < 15.

type = MOVELEFT, move the cursor to the column which is to the left of its original position if the original column > 0.

type = MOVEBACK, move the cursor to the column which is to the left of its original position and delete the character at the new position if the original column > 0.

**Return value:** 0 after the cursor is moved.

4. i32 displayLcm( bool mode )

**Description:** Show the text on the LCM screen or eliminate the text on the LCM screen. The content of

the text is not altered.

mode = true, show the text.

mode = false, eliminate the text.

**Return value:** 0 after the text has been shown or eliminated.

5. i32 getKeyLcm( void )

**Description:** Scan the LCM and return the identification of the pressed direction key.

**Return value:** 'UP' if the 'up' direction key is pressed.

'RIGHT' if the 'right' direction key is pressed.

'LEFT' if the 'left' direction key is pressed.

'DOWN' if the 'down' direction key is pressed.

'NONE' if none of the keys is pressed.

6. i32 getPositionLcm( i32 \*row, i32 \*column )

**Description:** Get the position of the cursor and write the coordinate to the memory pointed at by arguments 'row' and 'column'.

**Return value:** 0 if the request for the coordinate has been served.

7. i32 setPositionLcm( i32 row, i32 column )

**Description:** Set the position of the cursor according to the arguments 'row' and 'column'.

**Return value:** 0 after the position has been set.

-1 if the argument 'row' or 'column' meets any of the following conditions:

(1) row is not 0.

(2) row is not 1.

(3) column is less than 0.

(4) column is greater than 15.

8. i32 showLcm( i32 length, u8 \*info )

**Description:** Start from the current position of the cursor; print the text pointed at by 'info' to the LCM screen. The number of characters to be printed is at most 'length'. If the remaining columns available for printing the text is less than 'length', the number of the characters to be printed is:

16 – ( column number of the current position of the cursor ).

**Return value:** 0 after the text is printed.

## GPIO and Watchdog

### GPIO

#### 1. Syntax:

```
i32 setChDir( u8 val )
```

**Description:** Set the direction (Input/Output) of GPIO ports according to the parameter 'val'.

**Parameters:** The parameter 'val' is an unsigned character. Each bit of \*val corresponds to a GPIO port. Bit 0 corresponds to GPIO0. Bit 1 corresponds to GPIO1. Bit 2 corresponds to GPIO2, and so on. Setting a bit of 'val' as 0 configures the corresponding port as Output. Setting a bit of 'val' as 1 configures the corresponding port as Input.

**Return Value:** If the function gets the configuration successfully, it returns 0. If any error, it returns -1.

#### 2. Syntax:

```
i32 getChDir( u8 *val )
```

**Description:** Get the direction (Input/Output) of GPIO ports and put the configuration at \*val.

**Parameters:** The parameter 'val' points to an unsigned character. Each bit of \*val corresponds to a GPIO port. Bit 0 corresponds to GPIO0. Bit 1 corresponds to GPIO1. Bit 2 corresponds to GPIO2, and so on. A '0' bit at \*val indicates the corresponding port is an Output port. A '1' bit at \*val indicates the corresponding port is an Input port.

**Return Value:** If the function gets the configuration successfully, it returns 0. If any error, it returns -1.

#### 3. Syntax:

```
i32 getChLevel( u8 *val )
```

**Description:** Get the status value of GPIO ports 0~7 and put the value at \*val.

**Parameters:** The parameter 'val' points to an unsigned character. If a GPIO port is configured as an Output port, the bit at \*val which corresponds to this port indicates this port is outputting a '1' or '0'. If

a GPIO port is configured as an Input port, the corresponding bit at \*val is always '1'.

**Return Value:** If the function gets the values successfully, it returns 0. If any error, it returns -1.

#### 4. Syntax:

### **i32 setChLevel( u8 val )**

**Description:** Set the status bits of GPIO Output ports according to the variable 'val'. The status bits at the ports which are configured as input will not be affected.

**Parameters:** The parameter 'val' is an unsigned character. If a GPIO port is configured as an Output port, a '1' bit at 'val' directs the corresponding port to output a '1'. A '0' bit directs this port to output a '0'. If a GPIO ports is configured as an Input port, the setting to these input port is ignored.

**Return Value:** If the function sets the values successfully, it returns 0. If any error, it returns -1.

## Watchdog

#### 1. Syntax:

```
u8 getWtdTimer(void)
```

**Description:** This function read the value of the watchdog time counter and return it to the caller.

**Parameters:** None.

**Return Value:** This function return the value of the time counter and return it to the caller as an unsigned integer.

#### 2. Syntax:

```
void setWtdTimer( u8 val )
```

**Description:** This function sets the watchdog timer register to the value 'val' and starts to count down. The value could be 0 ~ 255. The unit is second. Setting the timer register to 0 disables the watchdog function and stops the countdown.

**Parameters:** The parameter 'val' is the value to set to watchdog timer register. The range is 0 ~ 255.

**Return Value:** None.

**LAN Bypass Subsystem**

1. void enableWdt(void)  
Enable Watchdog Timer. (This timer is different from the System Watchdog timer, which is configured by the API described in 6.2).
2. void disableWdt(void)  
Disable Watchdog Timer.
3. void reloadWdt(void)  
Reload Watchdog Timer.
4. void forceNormal(void)  
Force the port to become normal state.
5. void forceBypass(void)  
Force the port to become bypass state.
6. void setWdt4(void)  
Set the watchdog timer to 4 seconds.
7. void setWdt8(void)  
Set the watchdog timer to 8 seconds.
8. void setWdt16(void)  
Set the watchdog timer to 16 seconds.
9. void setWdt32(void)  
Set the watchdog timer to 32 seconds.