E28 Board

EPIC SBC supports Intel ATOM N270 Processor with Gigabit LAN / LVDS / TV out / 2COM / WIFI

User Manual

Manual Rev.: 1.0
Book Number: E28-2009.08.06

Revision

Version	Date	Author	Description
1.0	2009.08.06	chenyon	Initial release.

Copyright 2009

Trademarks

E28 is a registered trademarks of WTM; 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

1 INTRODUCTION

Welcome to the E28 Computer. The E28 is a Intel 945GSE chipset based platform designed for low power consumption and wide operating temperature. It supports the Atom N270 processor, while coming with a 533MHz Front Side Bus.

1.1 Specifications

- **Processor: on-board Intel Atom N270**
  - Single core and supports 2-Threads.
  - 1.6GHz core frequency.
  - 533MHz FSB.
  - 512KBs L2 cache.
  - 2.5W low TDP.
- **Chipset-North Bridge: Intel 945GSE**
  - One SODIMM socket supports DDRⅡ 533/400 SODIMM and capacity up to 2GBs max.
  - DVMT 3.0 supports 224MBs graphics memory max. (shared with system memory).
  - Analog display supports 400MHz/256-bit RAMDAC, resolution QXGA (2048x1536@75Hz).
  - 18-bit/36-bit LVDS supports Single/Dual channel LCD, resolution UXGA (1600x1200).
  - TV-OUT supports HDTV, 480P/720P/1080P/1080I.
- **Chipset-South Bridge: Intel 82801GM**
  - Two SATAⅡ connectors.
  - HD Audio Codec ALC662 supports 5.1 CH. audio output.
  - One PCI-e GbE controllers – RTL8111DL - support 1000/100/10 Mbps LANs.
  - Eight USB2.0 ports.
  - Supports +3.3V CompactFlash TypeⅡ card with Ultra-DMA mode 2/1/0.
  - Mini-PCIE supports One PCIE devices with PCIE Bus Master mode.
- **Super I/O: F1862FG**
  - Internal WatchDog, programmable 1~255 second(s)/minute(s).
  - 8 bits programmable bi-direction GPIOs, TTL-3.3V.
  - Two serial ports, supports RS232.
  - Two DC fan connectors, one supports ON/OFF control by system temperature.
  - Hardware monitor for voltage, fan speed and temperature.

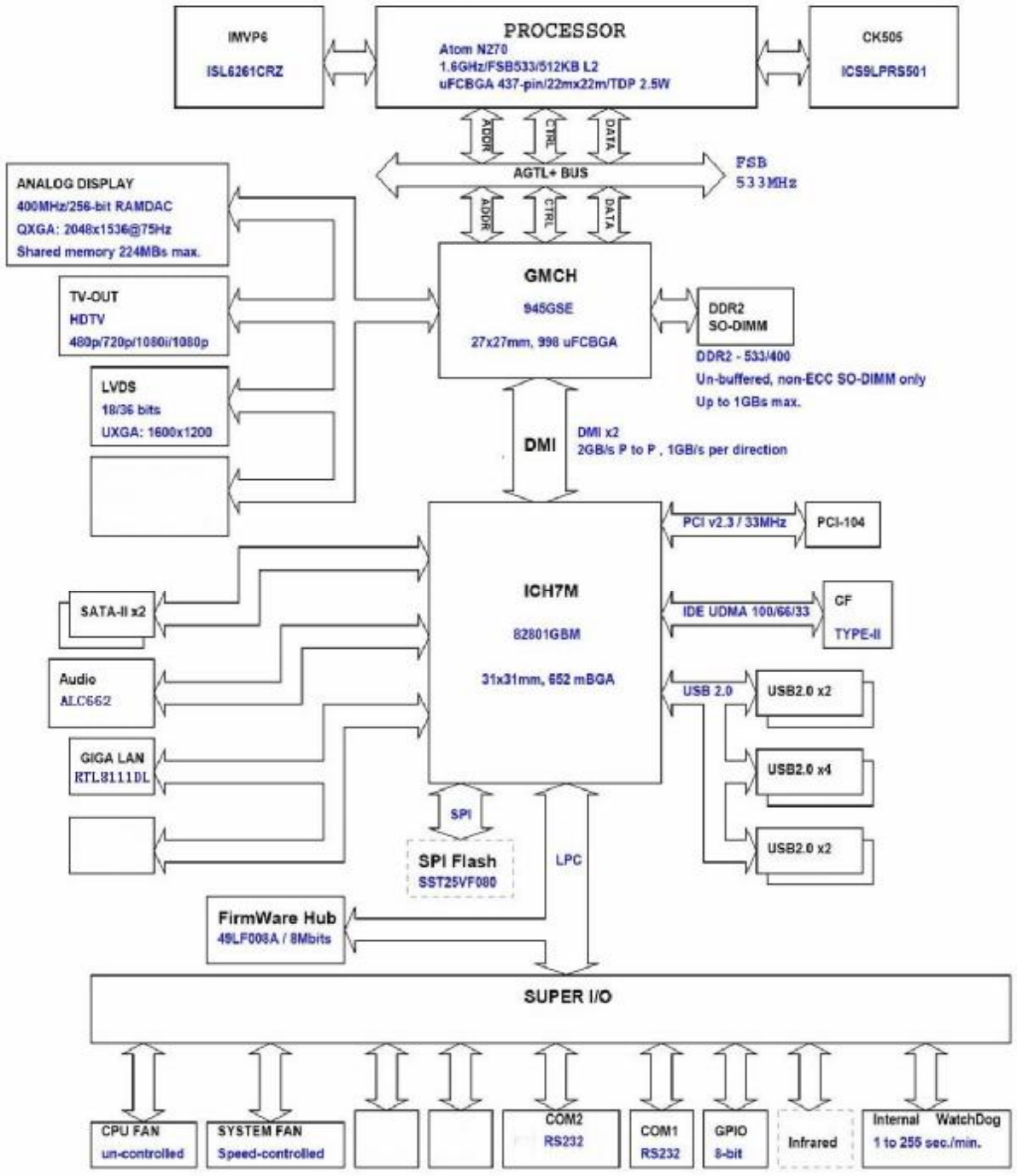
- **Others**
  - Power requirement: +12Vdc input only **(+12V@2.1A typically)**.
  - Operating temperature: -40~75℃ (-40~167℉). **[cold-start @ -20~75℃(-4~167℉)]**.
  - Storage temperature: -40~85℃ (-40~185℉).
  - Relative humidity: 0~90%@40℃ (104℉), non-condensing.
  - Dimension: 165 mm x 115 mm.

1.2 Package Contents

Check if the following items are included in the package.

- E28 EPIC SBC board
- Quick Manual
- Software Utility CD

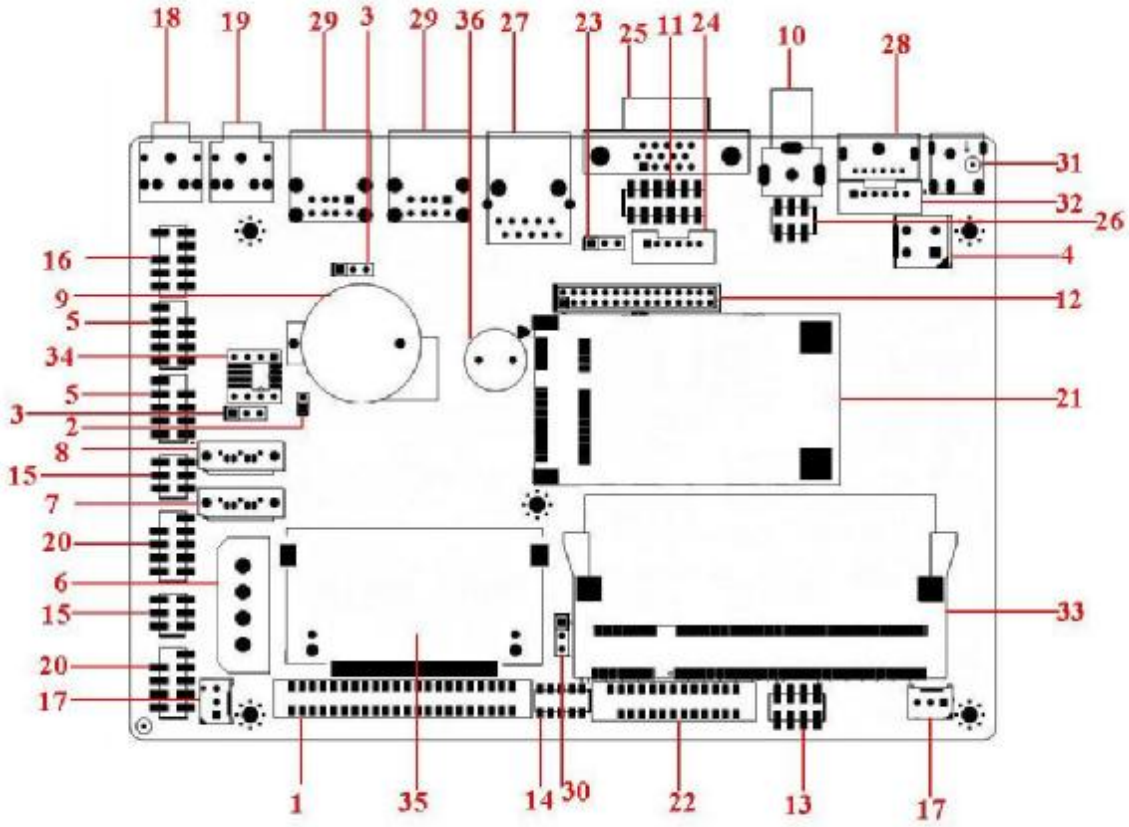
1.3 Block Diagram



2 H/W INFORMATION

This chapter describes the installation of E28. First, it shows the function diagram and the layout of E28. Then describes the unpacking information which you should read carefully, as well as the jumper/switch settings for the E28 configuration.

2.1 Locations (Top Side)



1	IDE1	12	LVDS1	23	JP2
2	JBAT	13	FP1	24	INVERTER1
3	JP4 JP5	14	GPIO1	25	VGA
4	ATX12V	15	JCOMP1 JCOMP2	26	TV_OUT
5	USB3 USB4	16	AUDIO2	27	LAN1
6	PWOUT1	17	CHAFAN SFAN1	28	PS1
7	SATA1	18	MIC1	29	USB1 USB2
8	SATA2	19	LINE_OUT1	30	JP3
9	BAT1	20	COM1 COM2	31	DC_JACK
10	AV	21	MINI_PCIE1	32	KB/MS2
11	VGA1	22	PARALLEL1		

33	SODIMM1
34	U2 U3
35	CF1

2.3 Connector and Jumper Setting

1. IDE1: IDE1 44pin connector.

IDE1 44pin connector.

2. JBAT: CMOS data retention/clear.

STATUS	SETTING
Open	CMOS data retention. (Default).
Short	CMOS data Clear

3. JP4 JP5: USB POWER connects

STATUS	SETTING
1-2	USB POWER ON DISABLE (Default)
2-3	USB POWER ON ENANLE

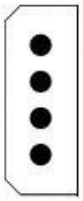
4. ATX12V: External +12V DC power input connector.

PIN	SETTING
1	GND
2	GND
3	+12V
4	+12V

5. USB3 USB4: Internal USB2.0 connector.

PIN	SETTING	PIN	SETTING
1	+5V	2	+5V
3	USB3-	4	USB4-
5	USB3+	6	USB4+
7	GND	8	GND
9	GND	10	GND

6. PWOUT1: Extra +12V and +5V DC power output connector (for SATA device).



PIN	SETTING
1	+12V
2	GND
3	GND
4	+5V

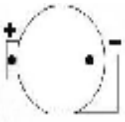
7. SATA1: SATA device connector #1.

SATA device connector #1.

8. SATA2: SATA device connector #2.

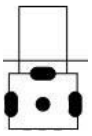
SATA device connector #2.

9. BAT1: CMOS battery holder.



CMOS battery holder.

10. AV: Infrared device connector.



AV device connector.

11. VGA1: Extra VGA signal connector.

PIN	SETTING	PIN	SETTING
1	RED	2	GND
3	GREED	4	GND
5	BULE	6	GND
7	HSYNC	8	GND
9	VSYNC	10	5V
11	DDC_CLK	12	DDC_DAT

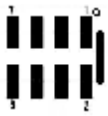
12. LVDS1: LCD panel (LVDS, 18-bit/36-bit) connector.

PIN	SETTING	PIN	SETTIN
1	VCC	2	VCC
3	VCC-	4	GND
5	GND	6	GND
7	RX00-	8	RX00+
9	RX01-	10	RX01+
11	RX02-	12	RX02+
13	GND	14	GND
15	RX0C-	16	RX0C+
17	RX03-	18	RX03+
19	RXE0-	20	RXE0+
21	RXE1-	22	RXE1+
23	RXE2-	24	RXE2+
25	GND	26	GND
27	RXEC-	28	RXEC+
29	RXE3-	30	RXE3+

E: Even for dual channel.

O: Odd for single channel

13. FP1: Front panel connector.



STATUS	SETTING
1-3 2-4	HD_LED. PW_LED
5-7	Reset
6-8	PW_BN

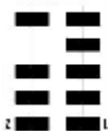
14. GPIO1: 8-bit TTL-3.3V GPIO connector.

PIN	SETTING	PIN	SETTING
1	GPIO [50]	2	GPIO [54]
3	GPIO [51]	4	GPIO [55]
5	GPIO [52]	6	GPIO [56]
7	GPIO [53]	8	GPIO [57]
9	GND	10	+3.3V

15. JCMOP1 JCOMP2: COM 9pin 5V or 12V selection.

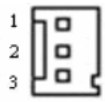
STATUS	SETTING
1-2	RS232. (Default)
3-4	12V.
5-6	5V.

16. AUDIO2: 5.1 channels Audio signal connector.



PIN	SETTING	PIN	SETTING
1	MIC2_L	2	NC
3	MIC2_R	4	GPIO
5	LINEOUT2-R	6	GND
7	SENSE_B	8	
9	LINEOUT2-L	10	GND

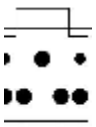
17. CHAFAN SFAN1: System DC fan connector.



PIN	SETTING
1	GND
2	+12V
3	Fan speed data

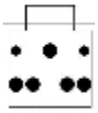
ON/OFF controlled by system temperature setti of BIOS.

18. MIC1: connector for MIC.



connector for MIC.

19. LINE_OUT1: connector for AUDIO OUT.



connector for AUDIO OUT

20. COM1 COM2: RS232 signa connector .

PIN	SETTING	PIN	SETTING
1	DCD	2	RX
3	TX	4	DTR
5	GND	6	DSR
7	RTS	8	CTS
9	RI	10	NC

21. MINI_PCIE1: connector for MINI_PCIE1.

connector for MINI_PCIE1

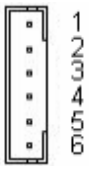
22. PARALLEL: LPT connector.

PIN	SETTING	PIN	SETTING
1	STB-	14	AFD-
2	PD0	15	ERR-
3	PD1	16	INIT-
4	PD2	17	SLIN-
5	PD3	18	GND
6	PD4	19	GND
7	PD5	20	GND
8	PD6	21	GND
9	PD7	22	GND
10	ACK-	23	GND
11	BUSY	24	GND
12	PE	25	GND
13	SLCT		

23. JP2: LCD panel driving voltage selection.

STATUS	SETTING
2-3	+3.3V for LCD panel. (Default).
1-2	+5V for LCD panel.

24. INVERTER1: LCD panel nverter



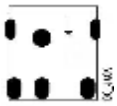
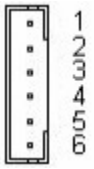
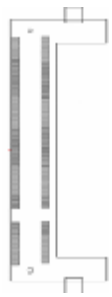
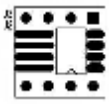
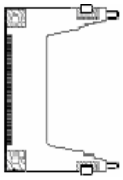
PIN	SETTING
1	GND
2	GND
3	BKL_CTL
4	BKL EN
5	12V
6	12V

25. VGA: D-SUB-15 female connector for VGA output.

D-SUB-15 female connector for VGA output.

26. TV-OUT: TV-out signal connector.

PIN	SETTING	PIN	SETTING
1	S-Y	2	GND.
3	GND	4	AV.
5	S-Pr	6	

27. LAN1: RJ45 connector for Gigabit Ethernet port #1.		28. PS1: KB+MS connector for port .		29. USB1 USB2: USB A-type stack connector for USB2.0 port,	
	RJ45 connector for Gigabit Ethernet port #1. Wake-On-LAN supported.		KB+MS connector for port .		Upper: Port #2. Lower: Port #1.
30. JP3: CF Master or Slave Selection.		**31. DC_JACK: power 12V input connector.**		**32. KB/MS2: KB+MS Extra connector.**	
	1-2: MASTER 2-3: SLAVE.		power 12V input connector.		
33. SODIMM1: 200-pin un-buffered DDR Ⅱ SODIMM socket.		**34. U2 U3: BIOS socket for flash EEPROM (system BIOS access).**		**35. CF1: Type Ⅱ compact flash card socket.**	
	Supports DDR Ⅱ 533/400MHz un-buffered and non-ECC SODIMM. Capacity is 2GBs max.		BIOS socket for flash EEPROM.		+3.3V CF card only and UDMA mode supported.

NOTE 1:

FP1: Front panel connector.

STATUS	SETTING
1-3 2-4	HD_LED PW_LED
5-7	Reset
6-8	PW_BN.

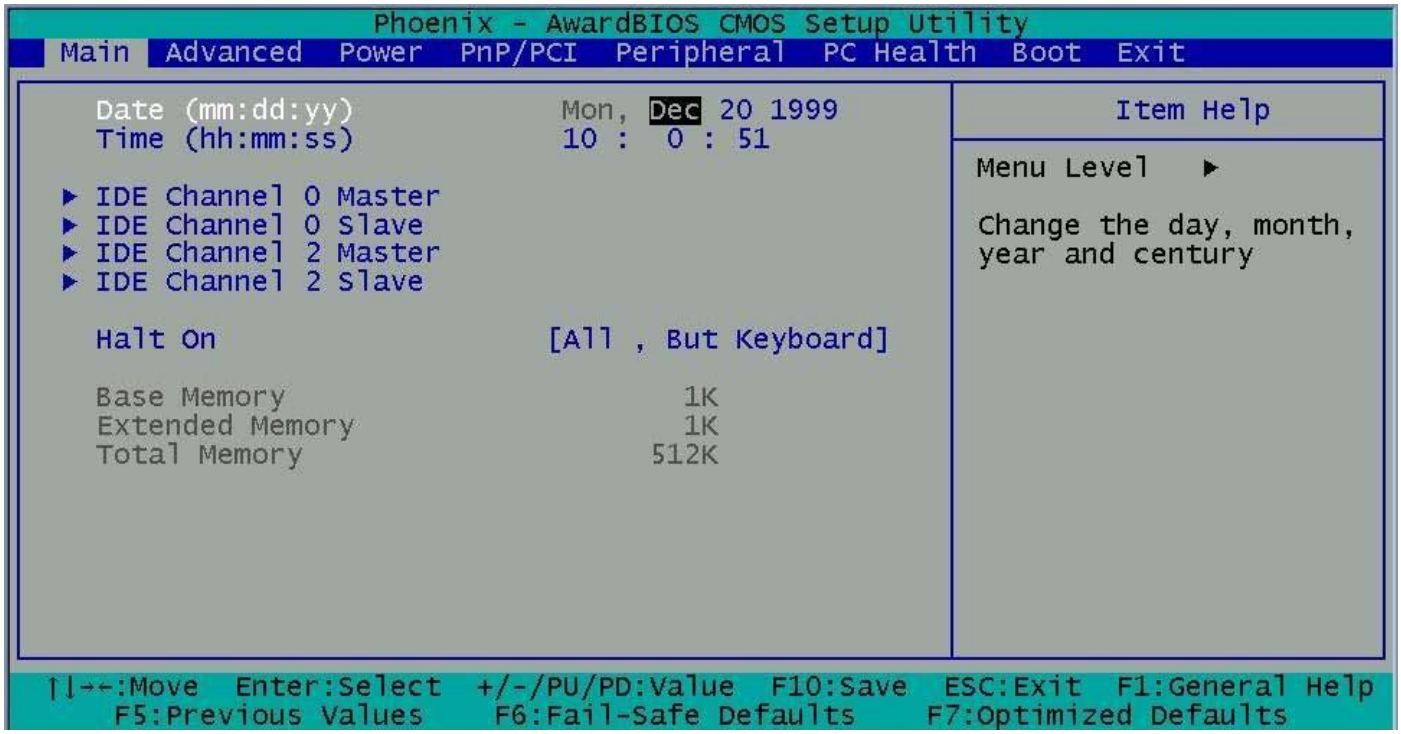
3 BIOS SETTING

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 menu displays. The following topics are covered:

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

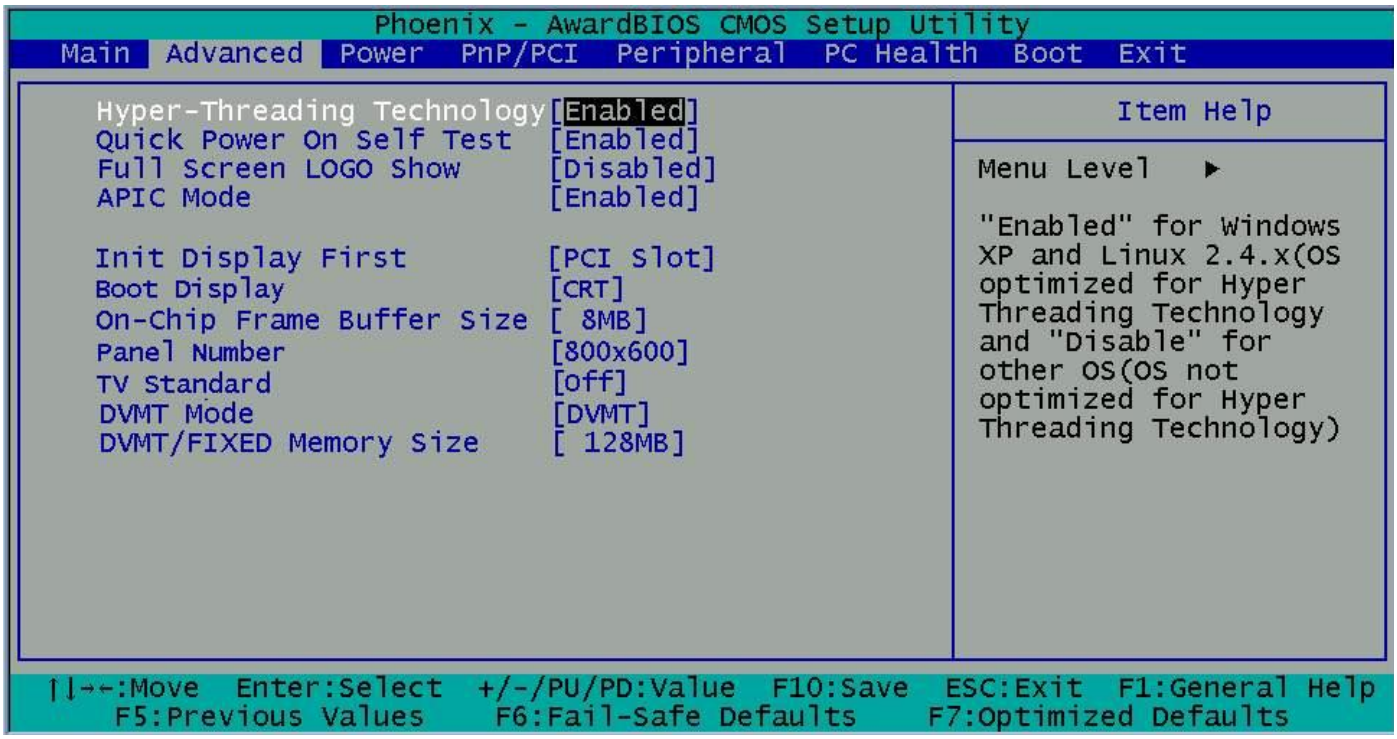
Once you enter the BIOS CMOS setup utility, you can use the control keys that listed at the bottom of the menu to select the desired value in each item.

3.1 Main Setup



Option	Choice	Description
Date Setup	N/A	To set the system date. Note that the 'Day' automatically changes when you set the date.
Time Setup	N/A	To set the system time.
IDE Channel 0 Master/Slave **IDE Channel 2 Master/Slave**	N/A	Press <Enter> to view the IDE device's information and related parameters.
Halt On	All Errors, No Errors, All, But keyboard.	To select the situation in which you want the BIOS to stop the POST process and notify you.

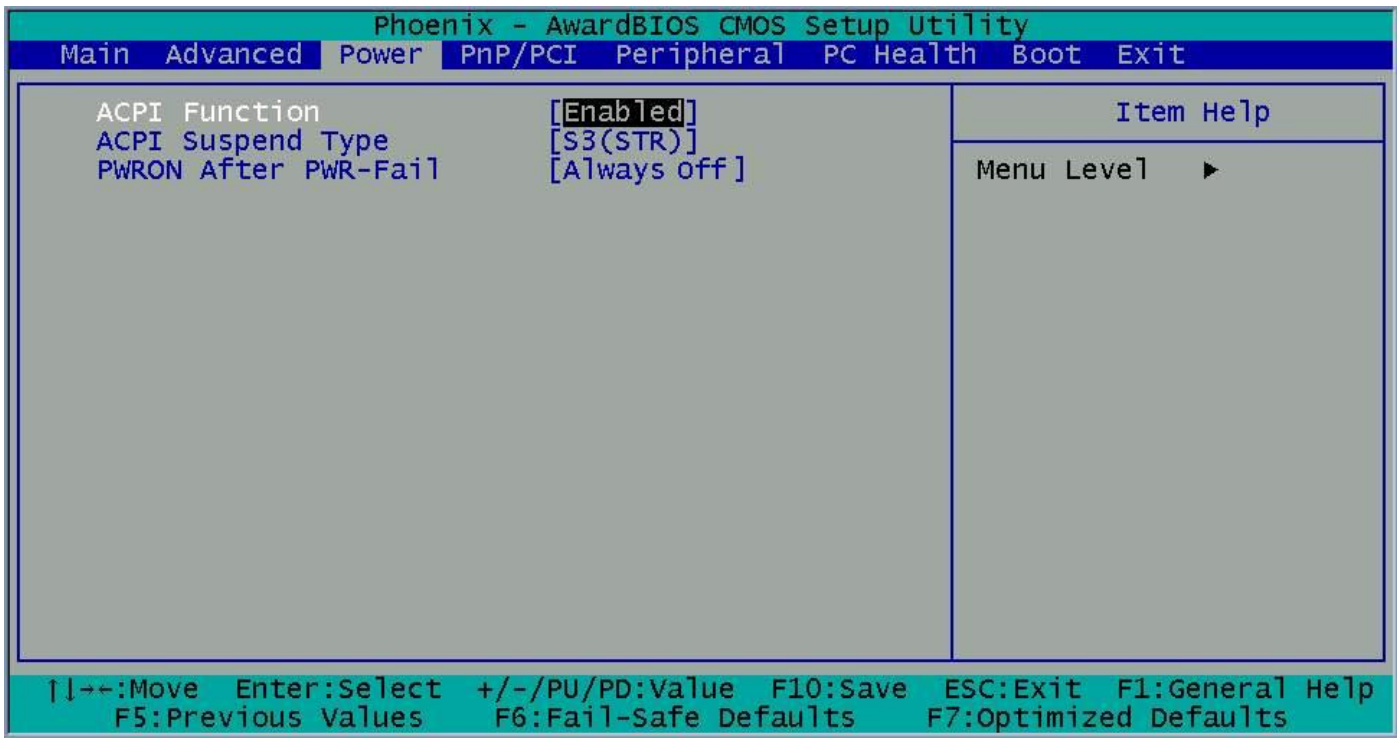
3.2 Advanced Setup



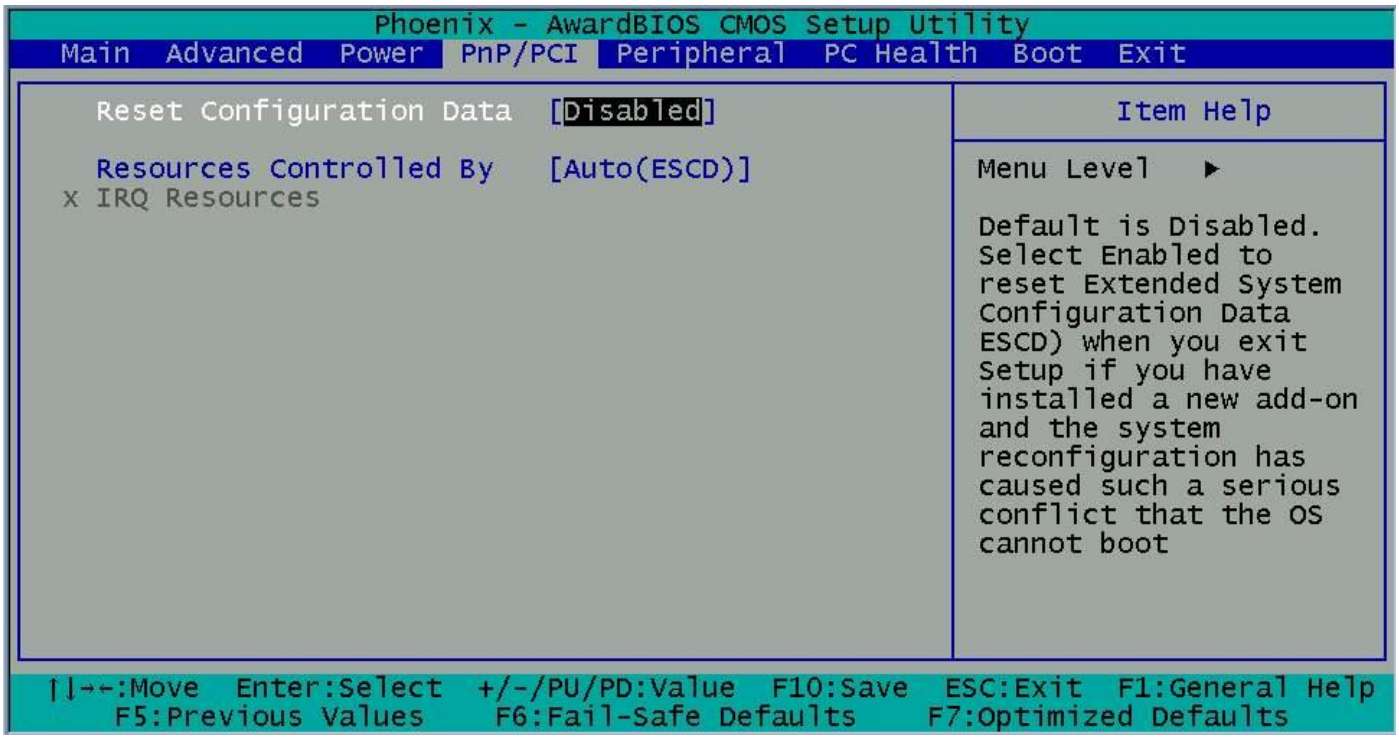
Option	Choice	Description
Quick Power On Self Test	Enabled Disabled	This category speeds up Power On Self Test (POST) after you have powered up the computer. If it is set to *Enabled*, BIOS will shorten or skip some check items during POST.
Full Screen Logo Show	Enabled Disabled	Select *Enabled* to show the OEM full screen logo if you have add-in BIOS.
Boot Display	CRT LVDS CRT+LVDS TV CRT+DVI	To set the display device.
Panel Type	800x600 1024x768	To set the LVDS panel resolution that you want.
DVMT mode	FIXED DVMT Both	To set the mode of Dynamic Video Memory Technology (DVMT).

DVMT/FIXED Memory Size	64MB 128MB 224MB	To set the shared memory size for DVMT.

3.3 Power Setup

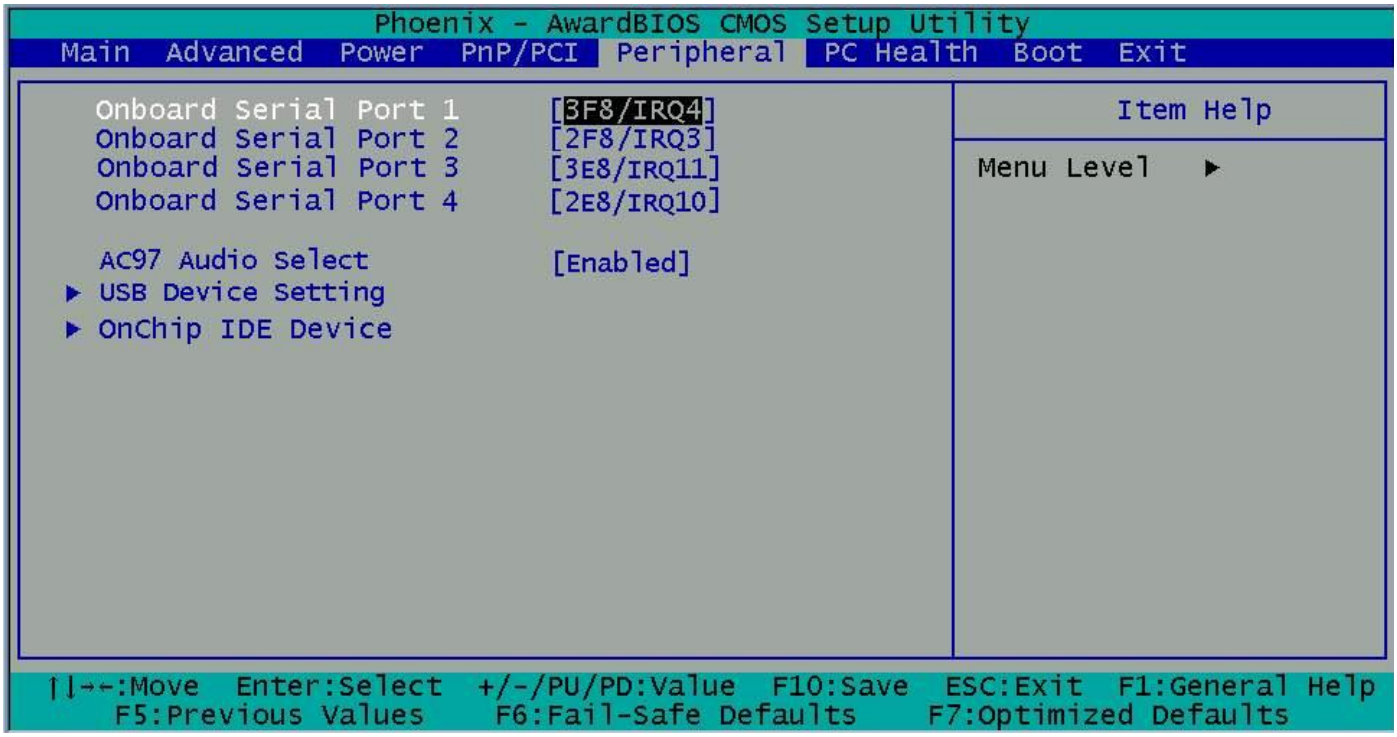


3.4 PnP/PCI Setup



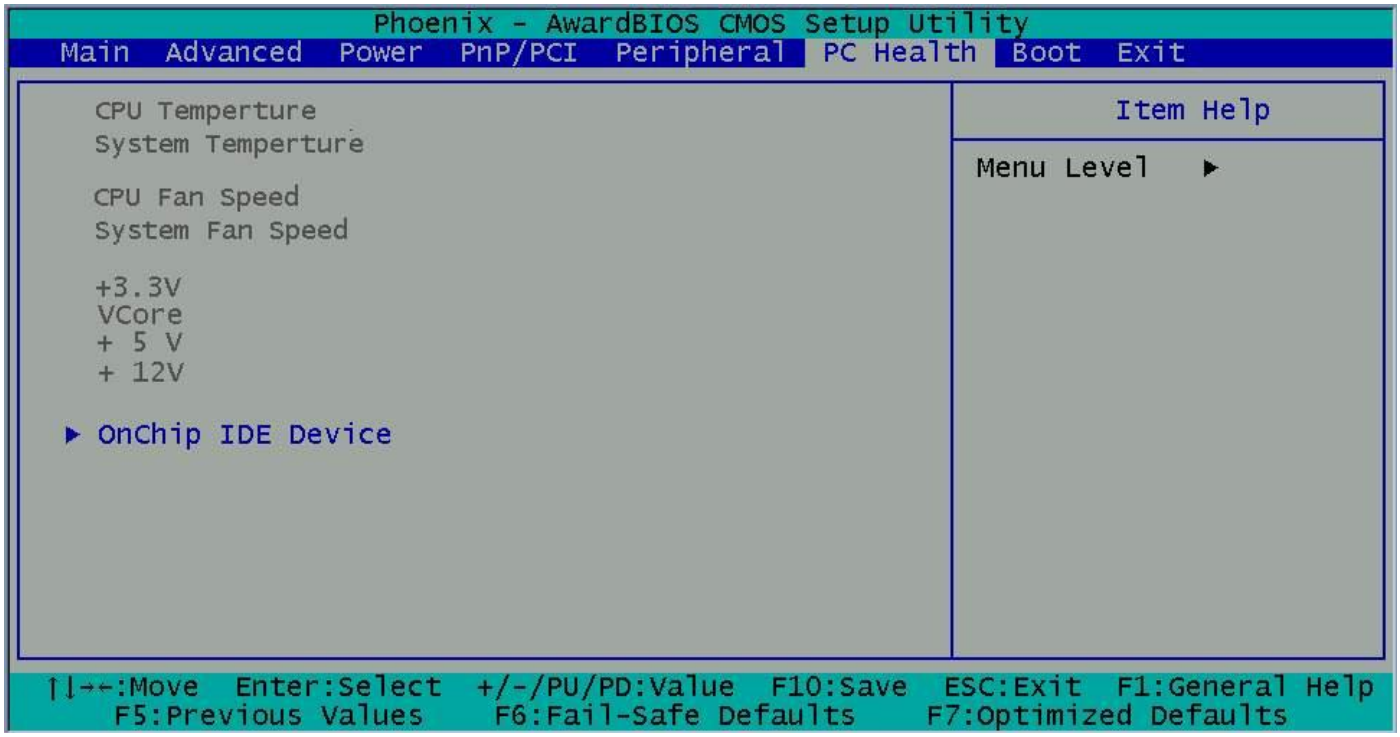
Option	Choice	Description
Reset Configuration Data	Enabled Disabled	Normally, you leave this field *Disabled*. Select *Enabled* to reset Extended System Configuration Data (ESCD) when you exit setup. If you have installed a new add-on and the system reconfiguration has caused such a serious conflict, then the operating system can not boot.
Resources 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,” then you may choose specific resources by going into each of the submenus.
IRQ Resources	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.

3.5 Peripherals Setup

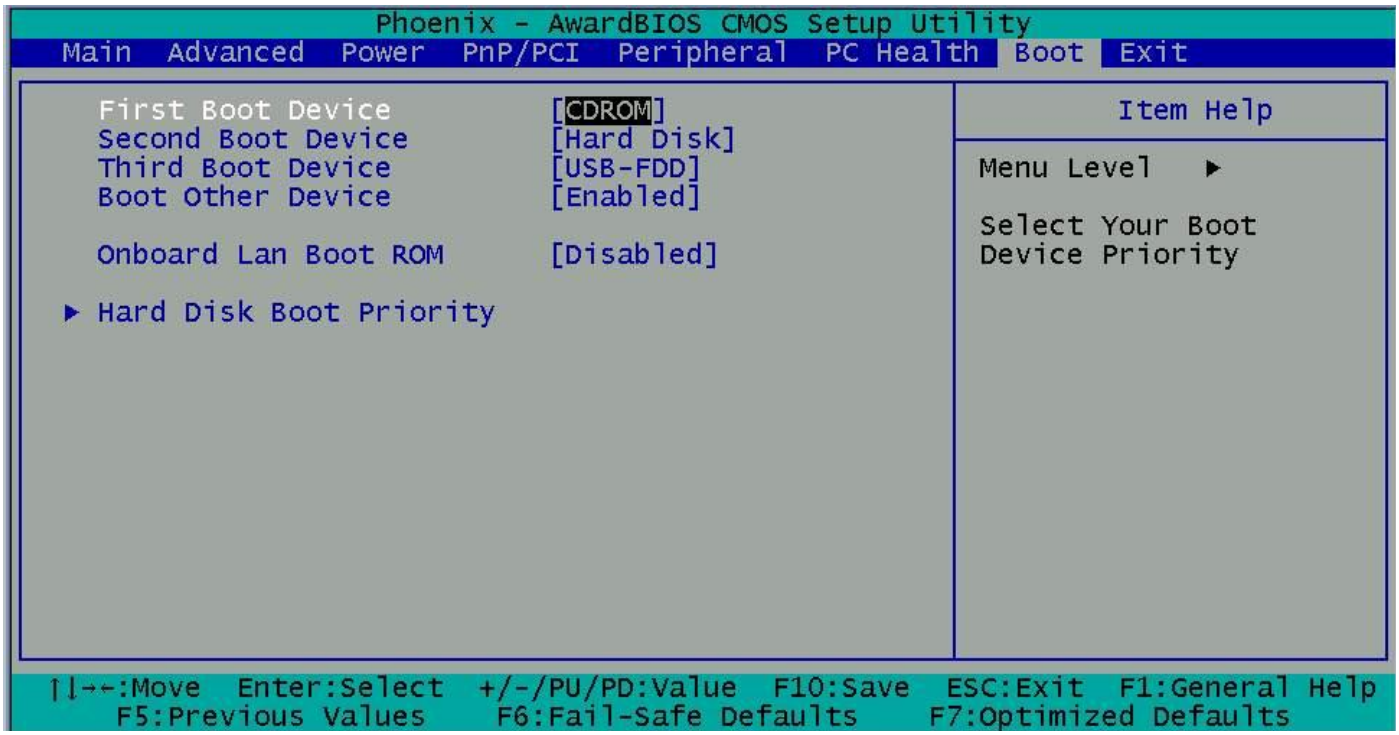


Option	Choice	Description
Onboard Serial Port 1 **Onboard Serial Port 2** **Onboard Serial Port 3** **Onboard Serial Port 4**	Serial Port 1: 3F8 / IRQ4 Serial Port 2: 2F8 / IRQ3 Serial Port 3: 3E8 / IRQ11 Serial Port 4: 2E8 / IRQ10	Select an address and the corresponding interrupt for each serial port.
AC97 Auido Select	Enabled Disabled	This item allows you to decide to enable/disable AC97 Audio.
USB Device setting	Press Enter	Press <Enter> to Enabled/Disabled USB controllers and view device's information.
OnChip IDE Device	N/A	Press <Enter> to Enabled/Disabled IDE/SATA controllers or set parameters.

3.6 PC Health Setup

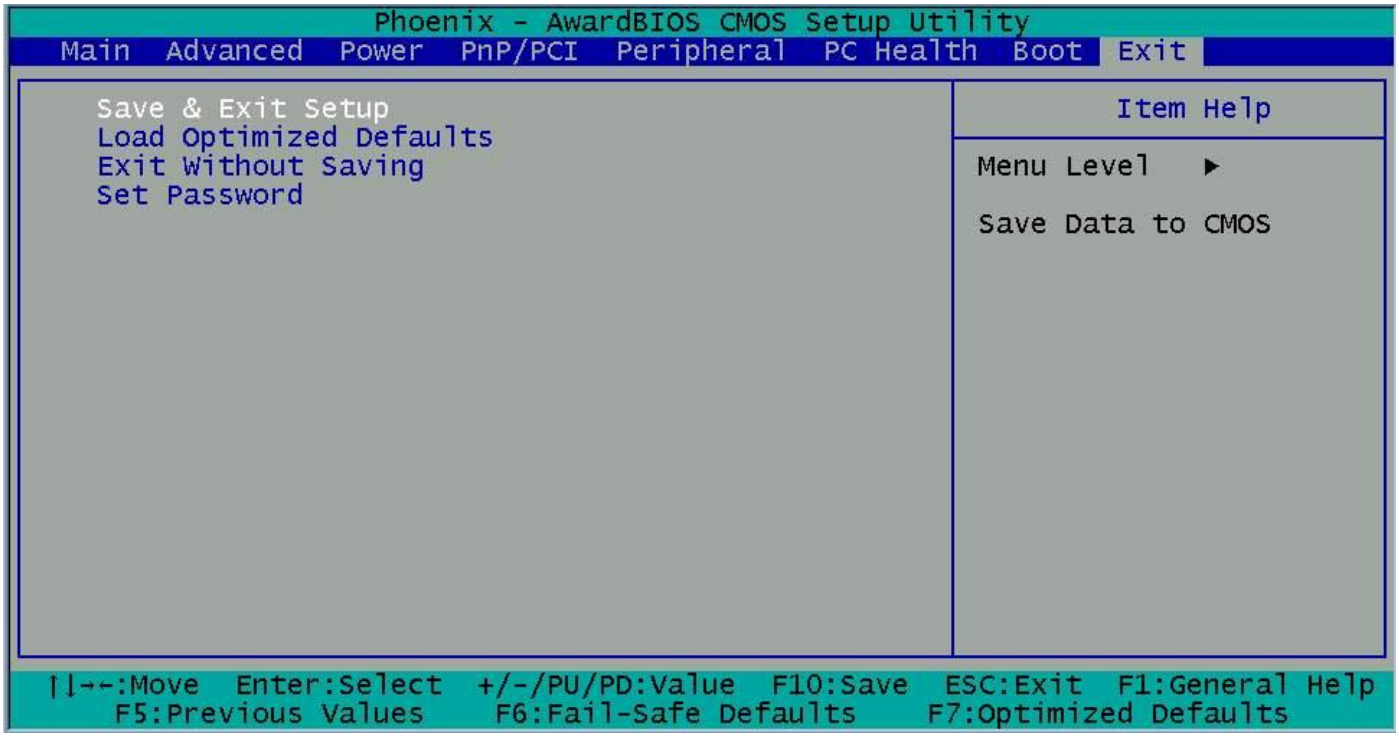


3.7 Boot Setup



Option	Choice	Description
First / Second / Third Boot Device/Other Boot Device	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.
Lan Boot Select	Disabled Lan-1 Lan-2	These fields allow the system to search for an OS from LAN.
Hard Disk Boot Priority	N/A	Press <Enter> to set the boot priority for each bootable device.

3.8 Exit Setup



Option	Choice	Description
Save & Exit Setup	Pressing <Enter> on this item for confirmation: Save to CMOS and EXIT (Y/N)? Y	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.
Load Optimized Defaults	When you press <Enter> on this item you get a confirmation dialog box with a message like this: Load Optimized Defaults (Y/N)? N	Press ‘Y’ to load the default values that are factory-set for optimal-performance system operations.

Exit Without Saving	Pressing <Enter> on this item for confirmation: Quit without saving (Y/N)? Y	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.
Set Password	Pressing <Enter> on this item for confirmation: ENTER PASSWORD:	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> 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.

4 BIOS REFRESHING, WATCHDOG AND GPIO PROGRAMMING

4.1 BIOS Refreshing

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 E28 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 could update BIOS smoothly.

4.2 WatchDog Programming

This section describes the usage of WatchDog. E28 integrated the WatchDog that enable user to reset the system after a time-out event. User can use a program to enable the WatchDog and program the timer in range of 1~255 second(s)/minute(s). Once user enables the WatchDog, the timer will start to count down to zero except trigger the timer by user's program continuously. After zeroize the timer (stop triggering), the WatchDog will generate a signal to reset the system. It can be used to prevent system crash or hang up. The WatchDog is disabled after reset and should be enabled by user's program.

Please refer to the following table to program WatchDog properly, and user could test WatchDog under 'Debug' program.

Address port: 2E and Data port: 2F	
C:>debug	To enter debug mode.
-o 2E 87 -o 2E 87	To enter configuration.
-o 2E 07	To point to Logical Device Number Reg.
-o 2F 07	To select logical device 7 (WatchDog)
-o 2E 30 -o 2F 01	To activate WatchDog.
-o 2E F5	Preparing to select the unit of timer equals minute or second.
-I 2F	To read the value of index "2F".
-o 2F xx	The value "xx" equals [(value of index "2F") OR (F7) or (FF)]. OR (F7): unit is second. OR (FF): unit is minute.
-o 2E F6	Preparing to set the WatchDog timer value.
-o 2F ##	The value "##" ranges between 01 ~ FF (1 ~ 255). 00: To disable WatchDog.
-o 2E FA	Preparing to set the WatchDog output signal.
-I 2F	To read the value of index "2F".
-o 2F xx	The value "xx" equals [(value of index "2F") OR (01)]. To issue signal WDTRST to reset system.
-o 2E F5	Preparing to start the WatchDog timer counting.
-I 2F	To read the value of index "2F".
-o 2F xx	The value "xx" equals [(value of index "2F") OR (20)]. To start timer counting.
-q	To quit debug mode

WatchDog demo program in Turbo C++ as following:

```
//==========================================================================
// Turbo C++ Version 3.0 Copyright(c) 1990, 1992 by Borland International,Inc.
//==========================================================================
// Describe : F81865 WatchDog timer test
//==========================================================================

//==========================================================================
//   Language include files
//==========================================================================
#include <conio.h>
#include <stdlib.h>
#include <stdio.h>
#include <dos.h>

//==========================================================================
//   Normal procedure
//==========================================================================
void Show_Help();

//==========================================================================
//   Main procedure
//==========================================================================
int main(int argc, char *argv[])
{
unsigned char IO_Port_Address=0x2E;
unsigned char Time;
int Temp;

if ( argc != 2 )
{   Show_Help();     return 1;        }

clrscr();

Time=atoi(argv[1]);

// Set Watchdog
```

```
outportb(IO_Port_Address,0x87);       // Enter Configuration
outportb(IO_Port_Address,0x87);

outportb(IO_Port_Address,0x07);       // Point to Logical Device Number Reg.
outportb(IO_Port_Address+1,0x07);   // Select logical device 7, (Watchdog Function)

outportb(IO_Port_Address,0x30);       // Device Active register
outportb(IO_Port_Address+1,0x01);

outportb(IO_Port_Address,0xF5);       // Select Watchdog count mode seconds or minutes
outportb(IO_Port_Address+1,inportb(IO_Port_Address+1)&0xF7);       // Default is second, bit3=0

outportb(IO_Port_Address,0xF5);       // Select Watchdog output mode
outportb(IO_Port_Address+1,inportb(IO_Port_Address+1)|0x10);       // Set to Pulse mode, bit4=1

outportb(IO_Port_Address,0xF6);       // Set Watchdog Timer Value
outportb(IO_Port_Address+1,Time);   // 0x00 to disable, max 0xFF

outportb(IO_Port_Address,0xFA);       // Set Watchdog Time out output via WDTRST
outportb(IO_Port_Address+1,inportb(IO_Port_Address+1)|0x01); // bit0=1

outportb(IO_Port_Address,0xF5);       // Start Watchdog Time counting
outportb(IO_Port_Address+1,inportb(IO_Port_Address+1)|0x20); // bit5=1

textcolor(YELLOW);
for(Temp=Time;Temp>0;Temp--)
{ outportb(IO_Port_Address,0xF6);
Time=inportb(IO_Port_Address+1);     // Read Watchdog Timer Value


gotoxy(20,10);
cprintf(">>> After %3d Second will reset the system. <<<",Time);

delay(1000);
}

textcolor(LIGHTRED);
gotoxy(18,10);
```

```
cprintf("If you can see this message, Reset system is Fail",Time);

return 1;
}

//===========================================================================
// Function : Show_Help()
// Input    : -
// Change   : -
// Return   : -
// Description   : Show Title string.
//===========================================================================
void Show_Help()
{
clrscr();
printf("WatchDog Test for F81865       \n\n");
printf("Sample:                         \n");
printf("            WDT.EXE 10          \n");
printf("( For 10 seconds to reset. )\n");
}
```

4.3 GPIO Programming

This section describes the usage of GPIOs. E28 integrated eight bits, TTL-3.3V, bidirectional, and software programmable GPIOs for user's application.

Address port: 2E and Data port: 2F								
GP##	GP57	GP56	GP55	GP54	GP53	GP52	GP51	GP50
Bit #	Bit 7 (MSB)	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0 (LSB)

GPIO demo program in Turbo C++ as following:

```
//=========================================================================
// Turbo C++ Version 3.0 Copyright(c) 1990, 1992 by Borland International,Inc.
//=========================================================================
// Describe : GPIO50~GPIO57 Test utility for F81865.
//=========================================================================

//=========================================================================
//   Language include files
//=========================================================================
#include <conio.h>
#include <stdio.h>

//=========================================================================
//   Normal procedure
//=========================================================================
void Show_Help();
void Show_Fail();
void Show_Pass();

//=========================================================================
//   Main procedure
//=========================================================================
int main(int argc)
{
```

```
  char *Model_Name="AR-B5432";
  unsigned char       IO_PORT_BASE=0x2E;   // DATA_PORT = IO_PORT_BASE + 1;
  unsigned char data;
  int result=0;

  if ( argc > 1 )
    {   Show_Help();     return 1;   }

  clrscr();
  textcolor(WHITE);
  gotoxy(1, 1);
cprintf("<>================================================================================<>");
  gotoxy(1, 2);   cprintf("|| F81865 GPIO Test Utility v1.0 Acrosser Technology Co., Ltd.                    ||");
  gotoxy(1, 3);
cprintf("<>================================================================================<>");
  gotoxy(1, 4);
cprintf("<>================================================================================<>");
  gotoxy(1, 5);   cprintf("|| Model Name    :                                                              ||");
  gotoxy(1, 6);   cprintf("|| SIO IO Base :                                                               ||");
  gotoxy(1, 7);
cprintf("<>================================================================================<>");

  // Show Got Parameter Informat
  textcolor(LIGHTGRAY);
  gotoxy(18,5);   cprintf("%s",Model_Name);
  gotoxy(18,6);   cprintf("%X",IO_PORT_BASE);

  // Enter F81865 Config
  outportb(IO_PORT_BASE,0x87);
  outportb(IO_PORT_BASE,0x87);

  // Set Multi-function Pins to GPIO
  outportb(IO_PORT_BASE,0x2A);
  outportb(IO_PORT_BASE+1,(inportb(IO_PORT_BASE+1) | 0x08));

  // Select GPIO Port device
  outportb(IO_PORT_BASE,0x07);
  outportb(IO_PORT_BASE+1,0x06);
```

```
// Set GPIO Port Active
outportb(IO_PORT_BASE,0x30);
outportb(IO_PORT_BASE+1,0x01);

// Set F81865 GPIO50~53 to Output, GPIO54~GPIO57 to Input
outportb(IO_PORT_BASE,0xA0);
outportb(IO_PORT_BASE+1,0x0F);

// Set F81865 GPIO50~53 to High
outportb(IO_PORT_BASE,0xA1);
outportb(IO_PORT_BASE+1,0x0F);
// Read F81865 GPIO54~57 Status, if not High error.
outportb(IO_PORT_BASE,0xA2);
data=inportb(IO_PORT_BASE+1)&0xF0;
if(data!=0xF0)
    result=1;
// Set F81865 GPIO50~53 to Low
outportb(IO_PORT_BASE,0xA1);
outportb(IO_PORT_BASE+1,0x00);
// Read F81865 GPIO54~57 Status, if not Low error.
outportb(IO_PORT_BASE,0xA2);
data=inportb(IO_PORT_BASE+1)&0xF0;
if(data!=0x00)
    result=1;

// Set F81865 GPIO50~53 to input, GPIO54~GPIO57 to Output
outportb(IO_PORT_BASE,0xA0);
outportb(IO_PORT_BASE+1,0xF0);

// Set F81865 GPIO54~57 to High
outportb(IO_PORT_BASE,0xA1);
outportb(IO_PORT_BASE+1,0xF0);
// Read F81865 GPIO50~53 Status, if not High error.
outportb(IO_PORT_BASE,0xA2);
data=inportb(IO_PORT_BASE+1)&0x0F;
if(data!=0x0F)
    result=1;
```

```
    // Set F81865 GPIO54~57 to Low
    outportb(IO_PORT_BASE,0xA1);
    outportb(IO_PORT_BASE+1,0x00);
    // Read F81865 GPIO50~53 Status, if not Low error.
    outportb(IO_PORT_BASE,0xA2);
    data=inportb(IO_PORT_BASE+1)&0x0F;
    if(data!=0x00)
        result=1;

    // Exit F81865 Config
    outportb(IO_PORT_BASE,0xAA);

    if(result)
      Show_Fail();
    else
      Show_Pass();

    return result;
}

//============================================================================
// Function : Show_Help()
// Input      : -
// Change  : -
// Return    : -
// Description    : Show Title string.
//============================================================================
void Show_Help()
{
    clrscr();
    printf("GPIO Test utility for F81865\n\n");
    printf("VCC                          GND     \n");
    printf("GPIO50     迁迁迁迁     GPIO54\n");
    printf("GPIO51     迁迁迁迁     GPIO55\n");
    printf("GPIO52     迁迁迁迁     GPIO56\n");
    printf("GPIO53     迁迁迁迁     GPIO57\n");

}
```

```
//=============================================================================
// Function: Show_Fail()
// Input     : -
// Change  : -
// Return    : -
// Description    : Show Fail Message.
//=============================================================================
void Show_Fail()
{
    textcolor(LIGHTRED);
    gotoxy(20,10);    cprintf("  ");
    gotoxy(20,11);    cprintf("  ");
    gotoxy(20,12);    cprintf("  ");
    gotoxy(20,13);    cprintf("  ");
    gotoxy(20,14);    cprintf("  ");
}

//=============================================================================
// Function: Show_Pass()
// Input     : -
// Change  : -
// Return    : -
// Description    : Show Pass Message.
//=============================================================================
void Show_Pass()
{
    textcolor(LIGHTGREEN);
    gotoxy(20,10);    cprintf("  ");
    gotoxy(20,11);    cprintf("  ");
    gotoxy(20,12);    cprintf("  ");
    gotoxy(20,13);    cprintf("  ");
    gotoxy(20,14);    cprintf("  ");
}
```

5 ELECTRICAL CHARACTERISTICS

5.1 Basic Electrical Characteristics Table

Electrical Characteristics					
	Parameter / Condition	Value			Unit
		Min.	Typ.	Max.	
+12V	External power input for system or +12Vdc power output (for SATA, LCD inverter, … etc.)	11.4	12.0	12.6	V
+5V	+5Vdc power output (for SATA, USB, DVI, … etc.)	4.75	5.0	5.25	V
+3.3V	+3.3Vdc power output (for LVDS, PCI-104, … etc.)	3.14	3.30	3.46	V
GPIO V_{IL}	GPIO's maximum Input LOW voltage	-	-	0.8	V
GPIO V_{IH}	GPIO's minimum input HIGH voltage	2.0	-	-	V
GPIO V_{OL}	GPIO's typical output LOW voltage	-	0	-	V
GPIO V_{OH}	GPIO's typical output HIGH voltage	-	3.3	-	V