AR-B8172 Board

ISA CPU card with DM&P CPU Vortex86DX

User Manual

Manual Rev.: 1.0
Book Number: AR-B8172-2011.03.15

Revision

Version	Date	Author	Description
0.1	2011/03/11	Roger Nan	Fisrt Release
1.0	2011/03/15	Roger_Nan	Draft

Copyright 2011

Trademarks

AR-B8172 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

1 Introduction 5
1.1 Specifications 6
1.2 Package Contents 7
1.3 Block Diagram 8
2 H/W Information 9
2.1 Locations (Top side) 9
2.2 Connectors and Jumper Setting 10
2.3 Connector and Jumper Setting Table 12
3 Bios Setting 17
3.1 Main Setup 18
3.2 Advanced Setup 19
3.3 PCIPnP 21
3.4 Boot 23
3.5 Security 25
3.6 Chipset Setup 26
3.7 Exit 28
4 BIOS REFRESHING, WATCHDOG AND GPIO PROGRAMMING 29
4.1 BIOS Refreshing 29
4.2 WATCHDOG Programming 29
4.3 GPIO Programming 32
5 ELECTRICAL CHARACTERISTICS 36
5.1 Basic Electrical Characteristics Table 36

1 INTRODUCTION

AR-B8172 is based on a DM&P Vortex chip with 800MHz and L2 256KB cache. AR-B8172 equips 256MB DDR2-333MHz on-board memory, 512KB SRAM with battery backup, 1 x CF socket, 1 x IDE connector, 1 x FDD connector, 1 x Parallel port, 1 x PC/104, 4 x USB 2.0, 1 x RS-232, 1 x RS-232/422/485, 1 x VGA, 16-bit GPIO, 1 x 10/100MBbit LAN and 1 x Golden Finger for ISA Bus.

AR-B8172 has XGI Z9s 2D Graphic Core with 64MB independent graphic memory. AR-B8172 support 1600 x 1200 @32bit VGA resolution fulfilled general 2D application demands.

AR-B8172 is the best choice ISA card for industrial SBC of factory automation environment.

1.1 Specifications

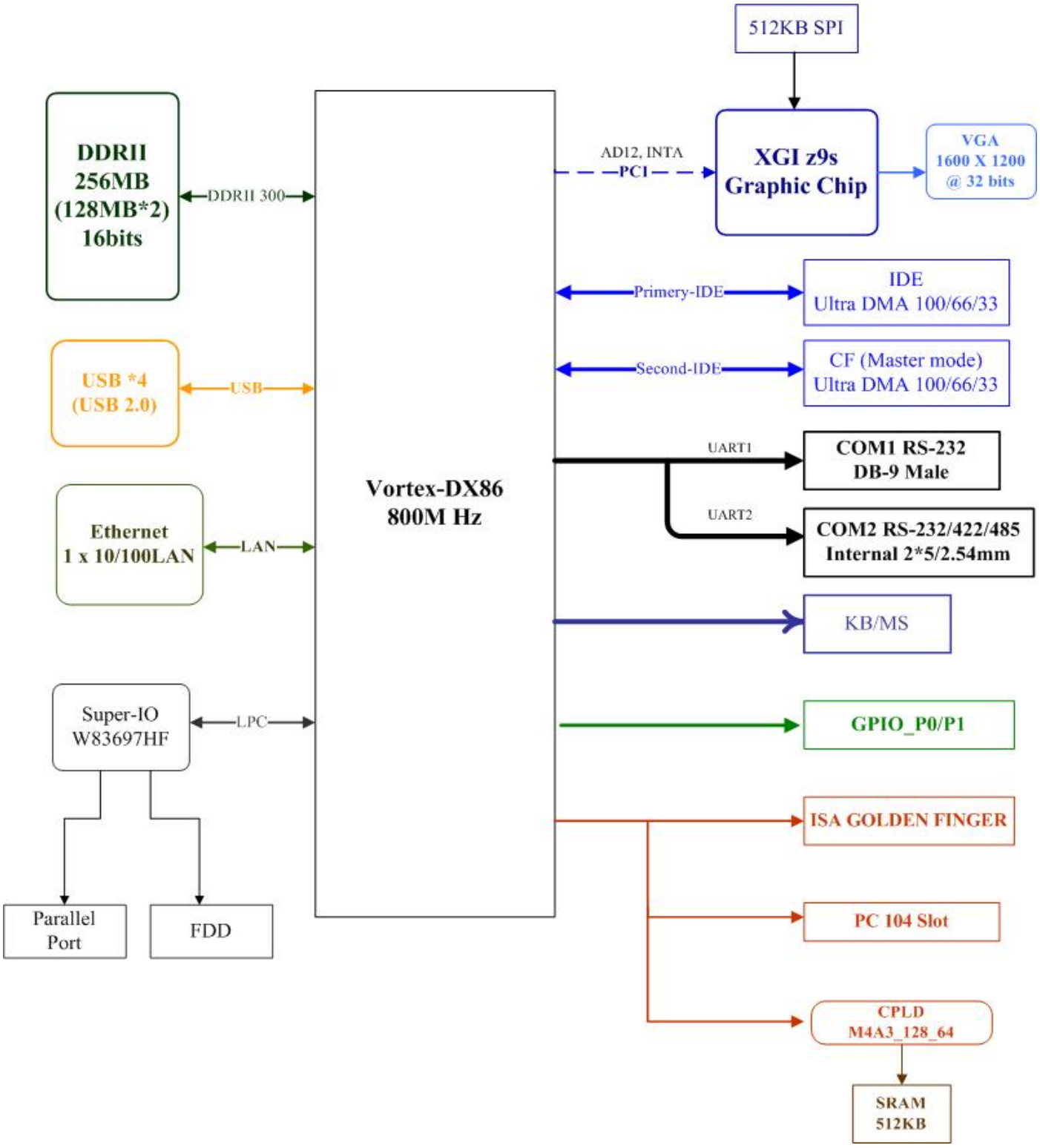
- AR-B8172, ISA bus CPU card with DM&P CPU Vortex86DX.
- On-board fanless DM&P Vortex86DX 800MHz / L1 32KB, L2: 256KB included in CPU.
- AMI BIOS Core-8 / On chip SPI Flash 2MBits built-in.
- Default On-board DDR2 256MB (128Mb*8*2) support DDR2 clock up to 333MHz.
- Interfaces with 4 through holes, follow AR-B1479A.
- Support AR-B1047 for 1 device at a time (not support 2 device at the same time).
- XGI Z9s Display Chipset / Video Memory: 64MB (max up to 64MB) / CRT: Up to 1600x1200 @ 32bits.
- RS-232 port: COM1: RS-232, COM2: RS-232/422/485 (internal) / LPT port x1 (Internal) / USB2.0 port x4 (4x internal).
- Enhanced IDE interface x1 (Ultra DMA 100/66/33) (Default: Secondary IDE*) / Supports Ultra DMA 33/66/100 for 40pins connector.
- CF socket x1 (Default: Secondary IDE/Master).
- 1 x 10/100 LAN (Built-in R6040) With RJ-45 (90 degree) connector build-in LED.
- 1x 3.5” Floppy driver connector (internal pin header connector).
- 1x PC/104 slot.
- 512KB SRAM (ISA interface).
- GPIO: 16 bits. Use GPIO_P0 & GPIO_P1 with interrupt support (input / output). Group to 2x pin header connectors (GPIO1, GPIO2).
- RTC / Watchdog: Software programmable from 1~256 seconds.

1.2 Package Contents

Check if the following items are included in the package.

- Quick Manual
- AR-B8172
- 1 x Software Utility CD

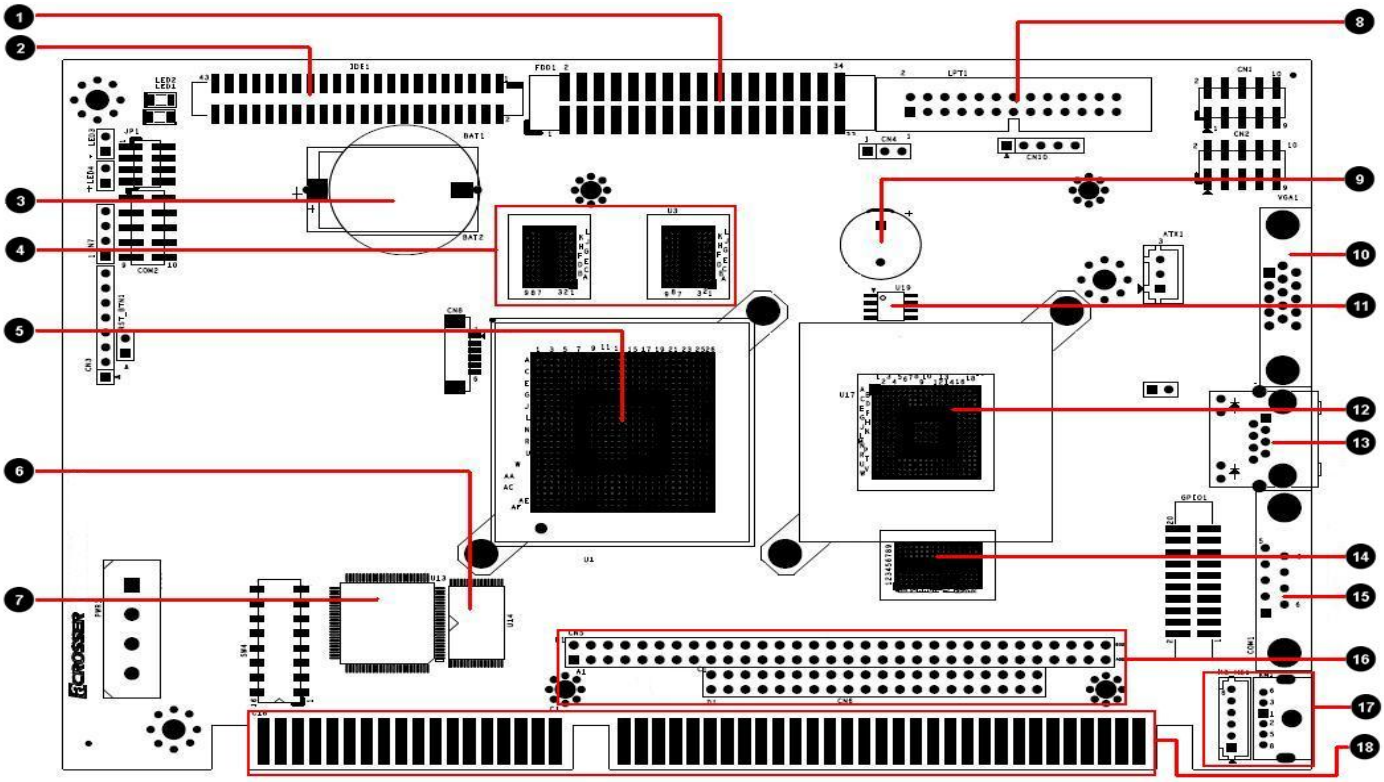
1.3 Block Diagram



2 H/W INFORMATION

This chapter describes the installation of AR-B8172. At first, it shows the function diagram and the layout of AR-B8172. It then describes the unpacking information which you should understand, as well as the jumper/switch settings for the AR-B8172 configuration

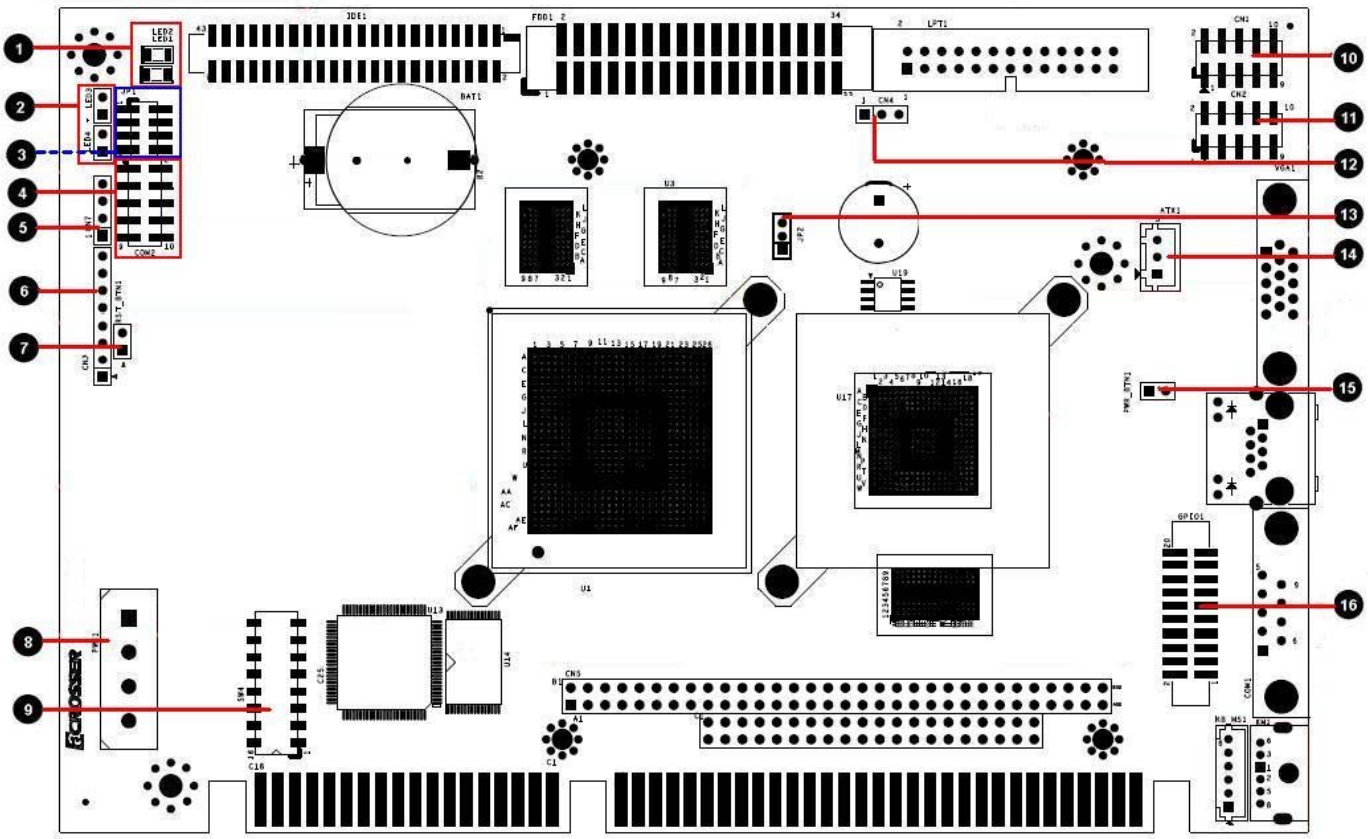
2.1 Locations (Top side)



1	**FDD1** FLOPPY CONNECTOR	8	**LPT1** PRINTER PORT	15	**COM1** RS232 DB9 CONNECTOR
2	**IDE1** 44PIN IDE CONNECTOR	9	**BZ1** BUZZER	16	**CN5/CN6** PC104 CONNECTOR
3	**BAT1** BATTERY for RTC/SRAM	10	**VGA1** VGA DB15 CONNECTOR	17	**KM1 (KB_MS1)** Keyboard/Mouse CONNECTOR
4	**U2, U3** System DDR2 RAM 128MBX2	11	**U19** VGA SPI FLASH 512KB	18	**ISA1** ISA BUS
5	**U1** CPU Vortex DX 800MHz	12	**U17** GPU XGI z9s		
6	**U14** SRAM 512KB	13	**LAN1** Ethernet 10/100		
7	**U13** CPLD(SRAM Controller)	14	**U18** GPU DDR2 64MB		

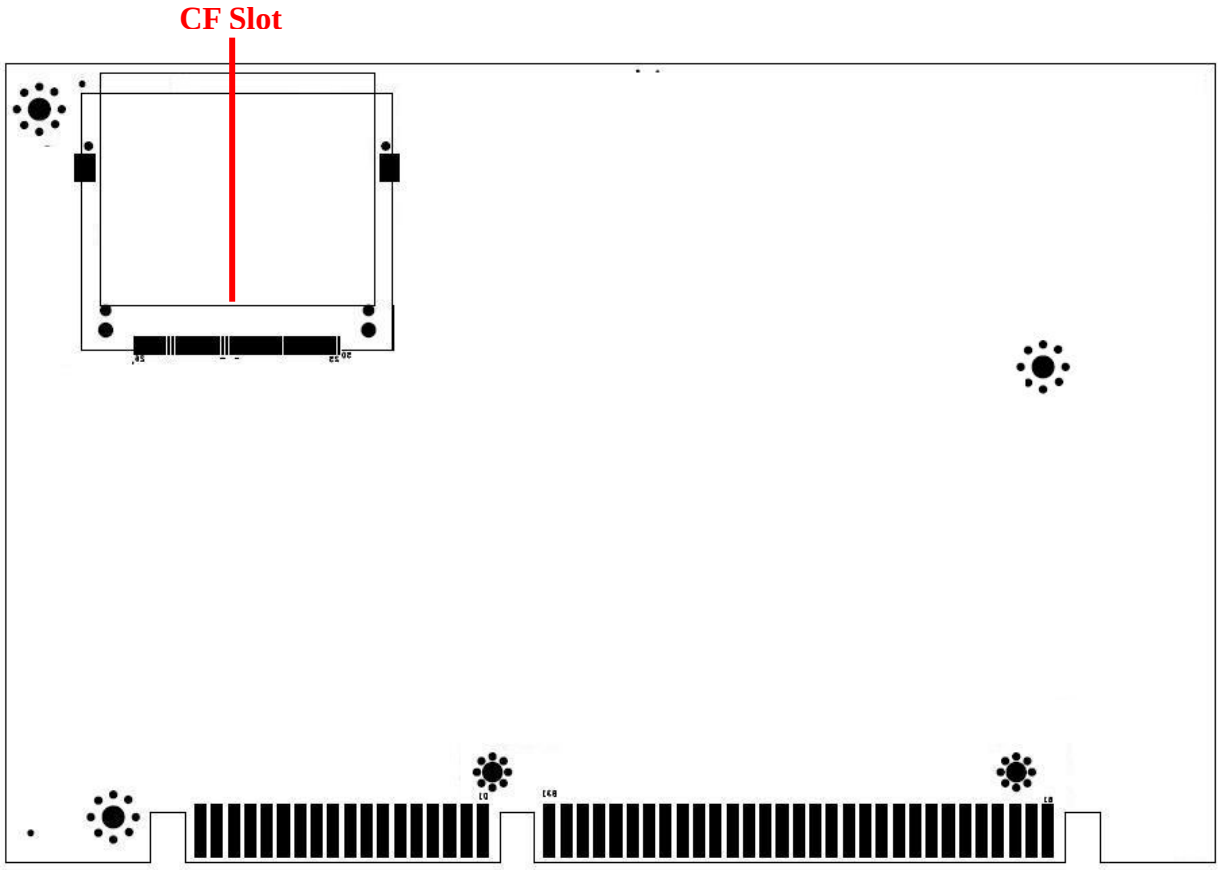
2.2 Connectors and Jumper Setting

2.2.1 Locations (Top side)



1	**LED1/LED2** System/HDD Status LED (On board)	7	**RST_BTN1** System Reset Button	13	**JP2** Clear CMOS data
2	**LED3/LED4** System/HDD Status LED Connector	8	**PWR1** AT Power Input	14	**ATX1** ATX Power Supply Connector
3	**JP1** RS-232/422/485 Selection with RS-485 termination resistor	9	**SW4** SRAM Address Selection Switch	15	**ATX_BTN1** ATX Power-ON Button
4	**COM2** RS-232 Connector	10	**CN1** USB 0/1 Connector	16	**GPIO1** GPIO 0/1 Connector
5	**CN7** RS-422/485 Connector (Signal shared from COM2)	11	**CN2** USB 2/3 Connector		
6	**CN3** Reserved	12	**CN4** Buzzer Connector (from BZ1)		

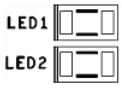
2.2.2 Locations (Bottom Side)



2.3 Connector and Jumper Setting Table

2.1 LED1/LED2 (On-board)

On-board Power/IDE (CF) status LED.



	FUNCTION
LED1 (Green)	POWER LED
LED2 (Yellow)	IDE/CF LED

2.2 LED3/LED4

Power/IDE (CF) status LED external 2.54mm 1×2PIN connector.

	FUNCTION
LED4 (POWER LED)	PIN1 : VCC+ PIN2 : LED Signal
LED3 (IDE/CF LED)	PIN1 : VCC+ PIN2 : LED Signal

2.3 JP1

COM2 RS-232/42/485 selection and RS-485 termination resistor.

JUMPER	FUNCTION
1-2 (default)	RS232
3-4	RS422
5-6	RS485
7-8	120Ω TERMINATION ENABLE

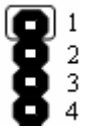
1.4 COM2

COM2 RS-232 2.54mm 2×5PIN Connector

PIN	SIGNAL	PIN	SIGNAL
1	DCD	**2**	DSR
3	RXD	**4**	RTS
5	TXD	**6**	CTS
7	DTR	**8**	RI
9	GND	**10**	N.C

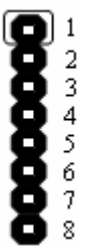
2.5 CN7

RS422/485 connector (signal from COM2).**NOTE1**



PIN	SIGNAL	
	RS485	**RS422**
1	485D+	422TX+
2	485D-	422TX-
3	N.C	422RX+
4	N.C	422RX-

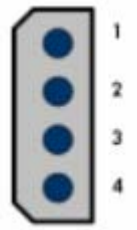
2.6 CN3 (reserved)



2.7 RST_BTN1
System reset button (2.54mm 1×2PIN connector)

RST_BTN1	FUNCTION
SHORT	RESET
OPEN	NORMAL

2.8 PWR1
AT power input



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

2.9 SW4
SRAM address selection switch

PIN	FUCTION	PIN	FUCTION
1	SRAMSEG_SEL0	5	CPLD_AI
2	SRAMSEG_SEL1	6	CPLD_AI
3	CPLD_ADR0	7	ENA_SR
4	CPLD_ADR1	8	Reserv

2.10-11 CN1/CN2
USB 0/1 & 3/4 connector

PIN	SIGNAL	PIN	SIGNAL
1	+5V	2	+5V
3	D0-	4	D1-
5	D0+	6	D1+
7	GND	8	GND
9	GND	10	GND

2.12 CN4
External buzzer connector (signal shared from BZ1)

PIN	FUNCTION
1	VCC
2	BZ-
3	BZ-

2.13 JP2
Clear CMOS data

JUMPER	FUNCTION
1-2 (default)	NORMAL
2-3	Clear CMOS

2.14 ATX1
ATX power supply connector

PIN	SIGNAL
1	PSON
2	5VSB
3	+5V

1. For ATX power supply, please connect to ATX1 connector to power supply.
2. For AT power, shorted PIN 2-3 directly then turn-on power supply to boot-up

2.15 ATX_BTN1
ATX power-on button

RST_BTN1	FUNCTION
SHORT	ATX power button
OPEN	Normal

2.16 GPIO1
GPIO 0/1 connector

PIN	FUNCTION	PIN	FUNCTION
1	GPIO00	2	GPIO07
3	GPIO01	4	GPIO06
5	GPIO02	6	GPIO05
7	GPIO03	8	GPIO04
9	GND	10	+3.3V
11	GPIO10	12	GPIO17
13	GPIO11	14	GPIO16
15	GPIO12	16	GPIO15
17	GPIO13	18	GPIO14
19	GND	20	+3.3V

- **SRAM Access Configuration:**

1. Enable SRAM access: SW4.PIN7 is "OFF" to Enable SRAM. If PIN7 is "On" then SRAM access will be disable.

SW4	Logical description
OFF	H
ON	L

2. SRAM Address Selection:

SRAM Start Address	SRAMSEG_SEL0	SRAMSEG_SEL1
D000:0000	L	L
D400:0000	H	H
D800:0000	H	L
DC00:0000	L	H

2. Device IO Address Selection:

Device IO Address	CPLD_ADR3	CPLD_ADR2	CPLD_ADR1	CPLD_ADR0
200h	L	L	L	L
214h	L	L	L	H
2A4h	L	L	H	L
2F4h	L	L	H	H
2C0h	L	H	L	L
2C4h	L	H	L	H
2D0h	L	H	H	L
2D4h	L	H	H	H
2E0h	H	L	L	L
2E4h	H	L	H	L

3. RAM is 512KB with 32 banks, each bank is 16KB size.

※ Others mode not descript on the above tables are reserved.

- **NOTE1:**

COM2 RS-232 signals is shared with RS-422/485 (connector is CN7). RS-485 supports 10 nodes with two 120ohm termination resisters:

Example 1. 10 node, RS-485 network with two 120 Ω termination resistors

Each RS-485 node has a load impedance of 12KΩ. 10 nodes in parallel give a load of 1200 Ω. Additionally, the two 120 Ω termination resistors result in another 60 Ω load, for a total load of 57 Ω. Clearly the termination resistors are responsible for a majority of the loading. In order to maintain at least 200mV between the B and A line, we need a bias current of 3.5 mA to flow through the load. To create this bias from a 5V supply a total series resistance of 1428 Ω or less is required. Subtract the 57 Ω that is already a part of the load, and we are left with 1371 Ω. Placing half of this value as a pullup to 5V and half as a pulldown to ground gives a maximum bias resistor value of 685Ω for each of the two biasing resistors.

3 BIOS SETTING

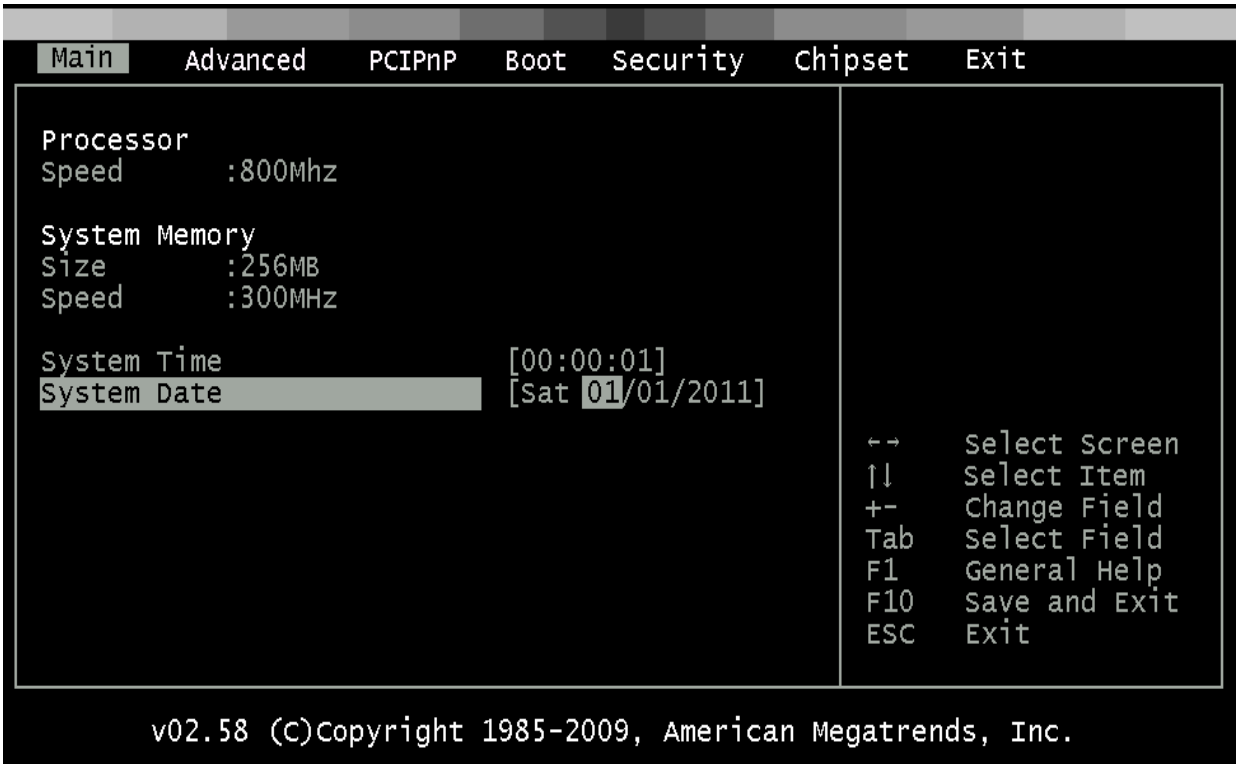
This chapter describes the BIOS menu displays and explains how to perform common tasks needed to get the system up. It also gives detailed explanation of the elements found in each of the BIOS menus. The following topics are covered:

- Main Setup
- Advanced Setup
- PCIPnP Setup
- Boot Setup
- Security Setup
- Chipset Setup
- Exit Setup

3.1 Main Setup

3.1.1 AMI BIOS

This is the interface of AMI BIOS:



3.1.2 Processor

This part shows the auto-detected CPU specification. DM&P Semiconductor is the ***Vortex86SX*** 32-Bit Microprocessor, DDR2 128MB onboard, which is based on x86 structure. It is the x86 SoC (System on Chip) with 0.13 micron process and ultra low power consumption design (less than 1 watt)The CPU on the Vortex86SX is a high performance and fully static 32-bit X86 processor with the compatibility of Windows based, Linux and most popular 32-bit RTOS.

3.1.3 System Memory

This part shows the auto-detected system memory. The ***Vortex86DX*** is a high performance with 256MB RAM and speed 133MHz onboard and fully static 32-bit x86 processor, which is compatible with DOS and Linux. It integrates 32KB write through direct map L1 cache, PCI Rev. 2.1 32-bit bus interface at 33 MHz, SDRAM, DDR2, ROM controller, IPC (Internal Peripheral Controllers with DMA and interrupt timer/counter included), Fast Ethernet MAC, FIFO UART, USB2.0 Host and IDE controller into a System-on-Chip (SoC) design. The Vortex86DX are all 256MB onboard and the speed is 133MHz.

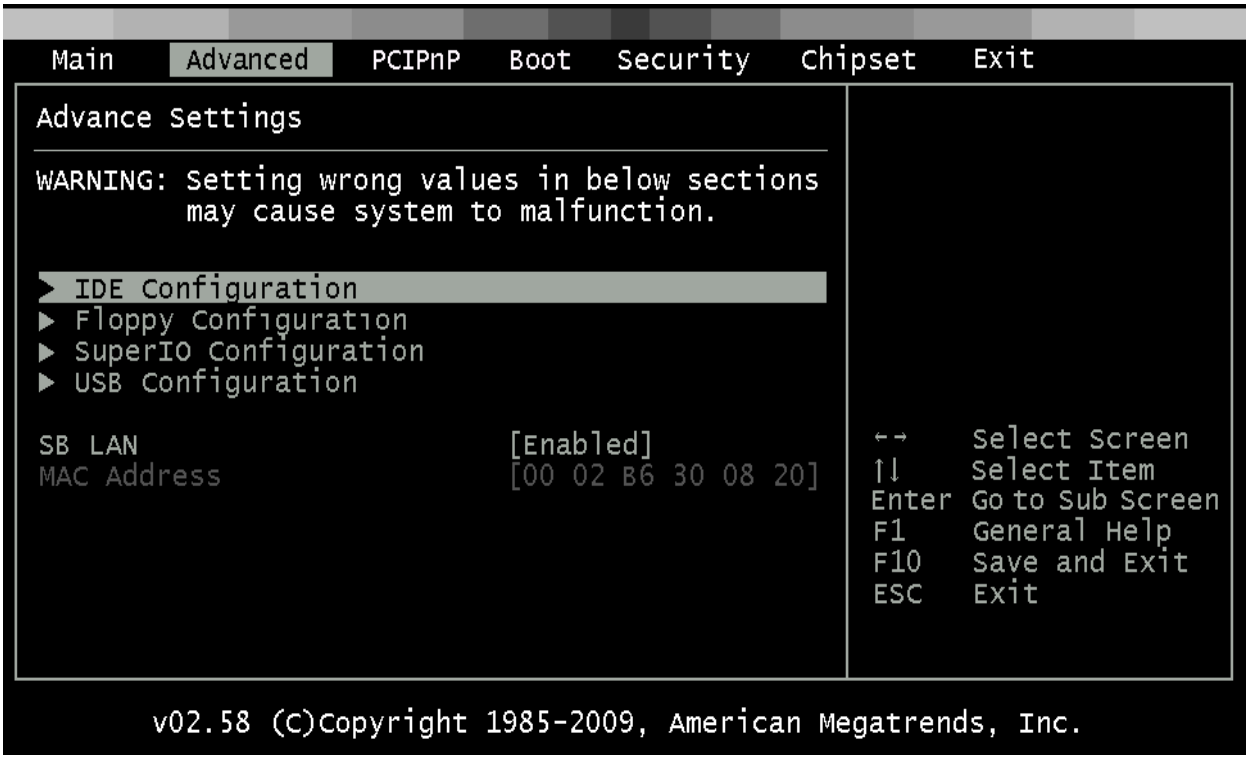
3.1.4 System Time:

The time format is based on the 24-hour military time clock. Press the “+” or “–“ key to increment the setting or type the desired value into the field.

3.1.5 System Date:

Press the “+” or ” –“ to set the date you wanted. The BIOS determines the day of the week from the other date information; this field is for information only.

3.2 Advanced Setup



3.2.1 IDE Configuration

OnBoard PCI IDE Controller
This can select the specification you wanted for the IDE device. This option specifies the channel used by IDE controller on the motherboard,

Option	Description
Disabled	Set this value to prevent the computer system from using the onboard IDE controller.
Primary	Set this value to allow the computer system to detect only the Primary IDE channel. This includes both the Primary Master and the Primary Slave.
Secondary	Set this value to allow the computer system to detect only the Secondary IDE channel. This includes both the Secondary Master and the Secondary Slave.
Both	Set this value to allow the computer system to detect the Primary and Secondary IDE channels. This includes the Primary Master, Primary Slave, Secondary Master, and Secondary Slave. This is the default setting.

Primary IDE Master/Slave

When you entered the IDE devices, the bios will auto-detected and show the detail information of IDE devices. If you want to change IDE configuration, select the item and press the “Enter” to configure the item you wanted.

3.2.3 SuperIO Configuration

You can use this screen to select options for the Super I/O settings. Use the up and down <Arrow> keys to select an item. Use the <Plus> and <Minus> keys to change the value of the selected option. The settings are described on the following pages. The screen is shown below.

Onboard Floppy Controller
This item specifies the Floppy used by the onboard Floppy controller. The settings are Disabled or Enabled.

Floppy Drive Swap
This option allows you to *Enabled* or *Disabled* the Floppy Drive Swap.

3.2.4 USB Configuration

USB Functions
Set this value to allow the system to enable or disable the onboard USB ports. The Optimal and Fail-Safe default setting is Enabled.

Option	Description
Disabled	This setting makes the onboard USB ports unavailable.
Enabled	This setting allows the use of the USB PORTS. This is the default setting.

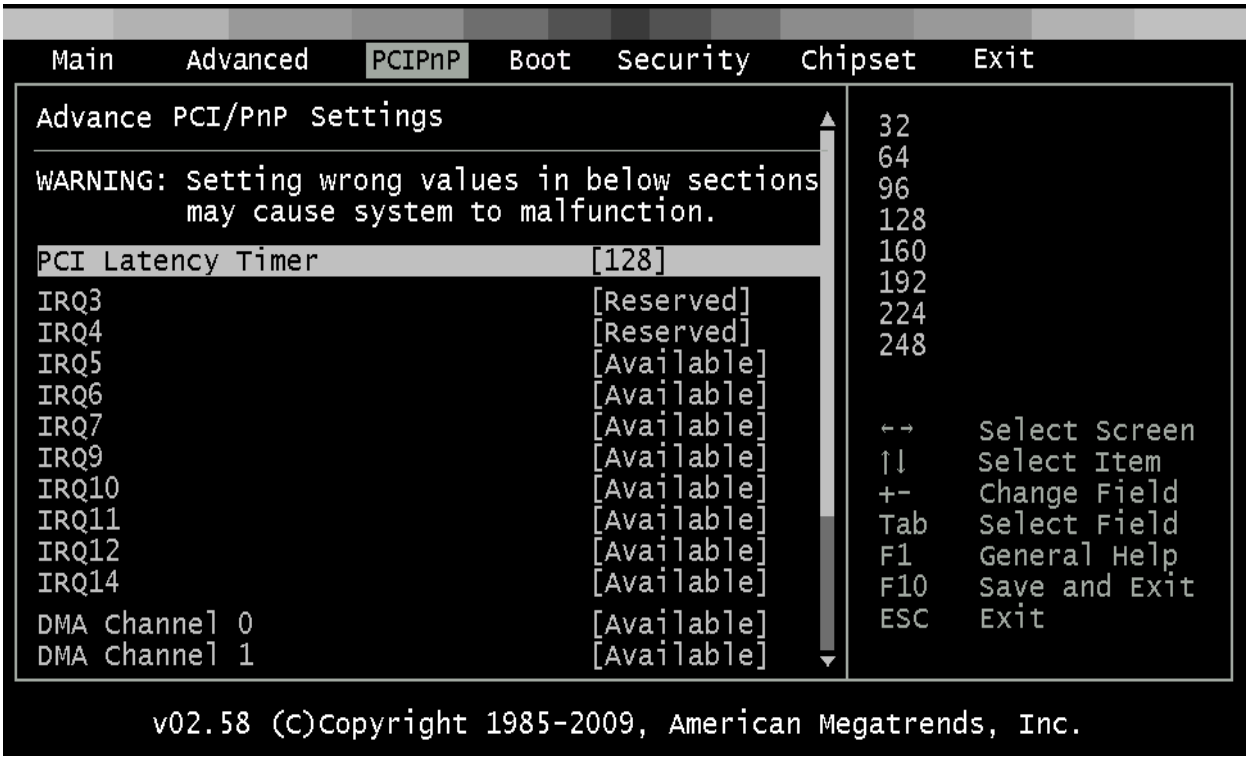
Legacy USB Support
Legacy USB Support refers to the USB mouse and USB keyboard support. Normally if this option is not enabled, any attached USB mouse or USB keyboard will not become available until a USB compatible operating system is fully booted with all USB drivers loaded. When this option is enabled, any attached USB mouse or USB keyboard can control the system even when there is no USB drivers loaded on the system. Set this value to enable or disable the Legacy USB Support. The Optimal and Fail-Safe default setting is Disabled.

Option	Description
Disabled	Set this value to prevent the use of any USB device in DOS or during system boot. This is the default setting.
Enabled	Set this value to allow the use of USB devices during boot and while using DOS.
Auto	This option auto detects USB Keyboards or Mice and if found, allows them to be utilized during boot and while using DOS.

3.3 PCIPnP

3.3.1 Clear NVRAM

Clear NVRAM during system boot.



3.3.2 PCI Latency Timer

Allow you to select the value in units of PCI clocks for all of the PCI device latency timer register. Configuration option: 32, 64, 96, 128, 160, 192, 224, 248.

Option	Description
32	This option sets the PCI latency to 32 PCI clock cycles.
64	This option sets the PCI latency to 64 PCI clock cycles. This is the default setting.
96	This option sets the PCI latency to 96 PCI clock cycles.
128	This option sets the PCI latency to 128 PCI clock cycles.
160	This option sets the PCI latency to 160 PCI clock cycles.
192	This option sets the PCI latency to 192 PCI clock cycles.
224	This option sets the PCI latency to 224 PCI clock cycles.
248	This option sets the PCI latency to 248 PCI clock cycles.

Set this value to allow the PCI Latency Timer to be adjusted. This option sets the latency of all PCI devices on the PCI bus This decides how long a PCI device can hog the PCI bus for , higher setting , hogs the bus a little longer , lower setting lets go quicker but stuff like some sound card (PCI of course) will start to crackle , default on this board was default at 64.

IRQ

This item can select the IRQ with Available or Reserved. And the default of IRQ3, 4 are Reserved and others are Available. When you set available, the specified IRQ is to be used by a PCI/PnP device; as you set reserved, the IRQ will reserved for legacy ISA devices.

Interrupt	Option	Description
IRQ3 IRQ4 IRQ5 IRQ6 IRQ7 IRQ8	Available	This setting allows the specified IRQ to be used by a PCI/PnP device. This is the default setting.
IRQ9 IRQ10 IRQ11 IRQ12 IRQ13 IRQ14 IRQ15	Reserved	This setting allows the specified IRQ to be used by a legacy ISA device.

DMA Channel

This item can select the DMA Channel for Available or Reserved. When set to Available the specified DMA is available for used by PCI/PnP devices; when set to reserved, the specified DMA to be used by a legacy ISA device.

DMA Channel	Option	Description
DMA Channel 0 DMA Channel 1 DMA Channel 3	Available	This setting allows the specified DMA to be used by PCI/PnP device. It is default setting.
DMA Channel 5 DMA Channel 6 DMA Channel 7	Reserved	This setting allows the specified DMA to be used by a legacy ISA device.

Reserved Memory Size

Set this value to allow the system to reserve memory that is used by ISA devices. The optimal and Fail-Safe default setting is *Disabled.*

Option	Description
Disabled	Set this value to prevent BIOS from reserving memory to ISA devices.
16K	Set this value to allow the system to reserve 16K of the system memory to the ISA devices.
32K	Set this value to allow the system to reserve 32K of the system memory to the ISA devices.
64K	Set this value to allow the system to reserve 64K of the system memory to the ISA devices.

3.4 Boot

The Boot menu items allow you to change the system boot options. Select an item then press Enter to display the sub-menu.

3.4.1 Boot Settings Configuration

Allow you to configure the system boot setting with bellow submenus.

Quick Boot
Set the value to *Enable* to allow the BIOS to skip some Power On Self Tests (POST) while booting to decrease the time needed to boot the system. When you set the value to Disable the BIOS will performs all the POST items.

Option	Description
Disabled	Set this value to allow BIOS to perform all POST tests.
Enabled	Set this value to allow BIOS to skip certain POST tests to boot faster.

PS/2 Mouse Support
Set this value to allow the PS/2 mouse support to be adjusted. The Optimal and Fail-Safe default setting is Enabled.

Option	Description
Disabled	This option will prevent the PS/2 mouse port from using system resources and will prevent the port from being active. Use this setting if installing a serial mouse.
Enabled	Set this value to allow the system to use a PS/2 mouse. This is the default setting.

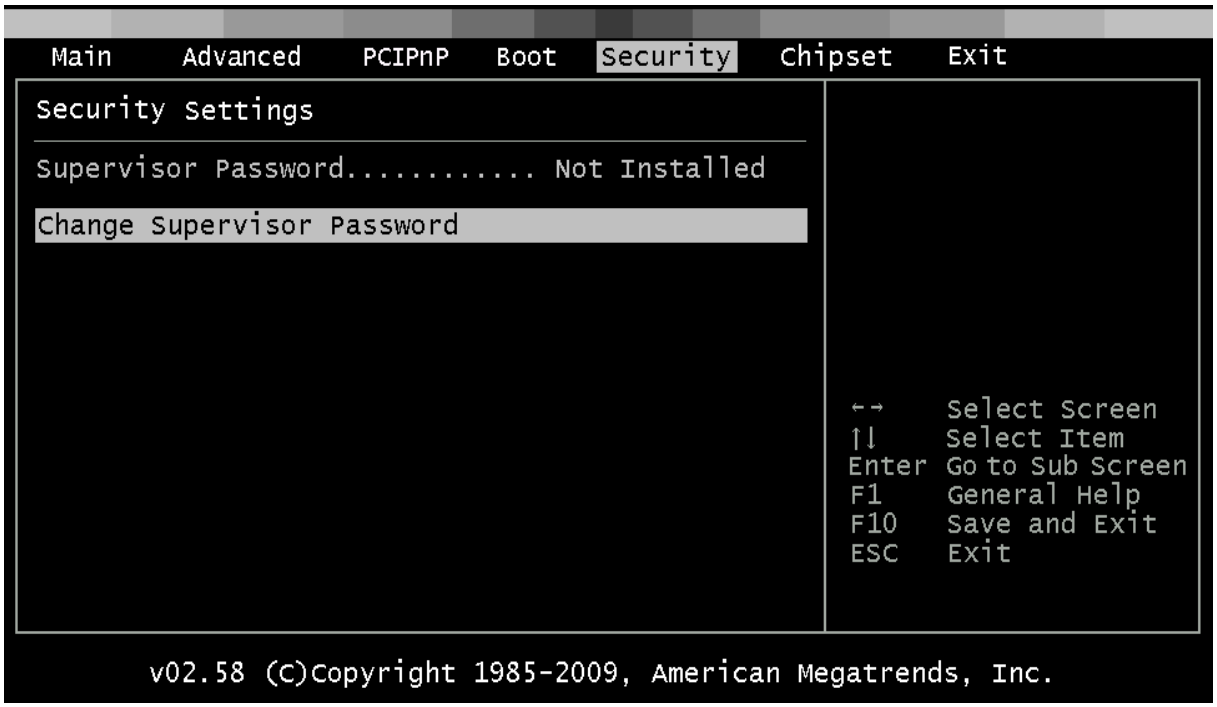
Hit “DEL” Massage Display

Set this value to allow the *Hit “DEL” to enter Setup* Message Display to be modified. The Optimal and Fail-Safe default setting is *Enabled*.

Option	Description
Disabled	This prevents the export to display Hit Del to enter Setup during memory initialization. If Quiet Boot is enabled, the Hit 'DEL' message will not display.
Enabled	This allows the export to display Hit Del to enter Setup during memory initialization. This is the default setting.

3.5 Security

The Security menu items allow you to change the system security settings. Select an item then press Enter to display the configuration options.



Supervisor Password
Indicate whether a supervisor password has been set. If the password has been installed, *Installed* displays. If not, *Not Installed* displays.

Change Supervisor Password
Select this option and press <Enter> to access the sub menu. You can use the sub menu to change the supervisor password. Select Change Supervisor Password from the Security Setup menu and press <Enter>. Enter New Password: appears. Type the password and press <Enter>. The screen does not display the characters entered. Retype the password as prompted and press <Enter>. If the password confirmation is incorrect, an error message appears. The password is stored in NVRAM.

Change User Password
Select this option and press <Enter> to access the sub menu. You can use the sub menu to change the user password.

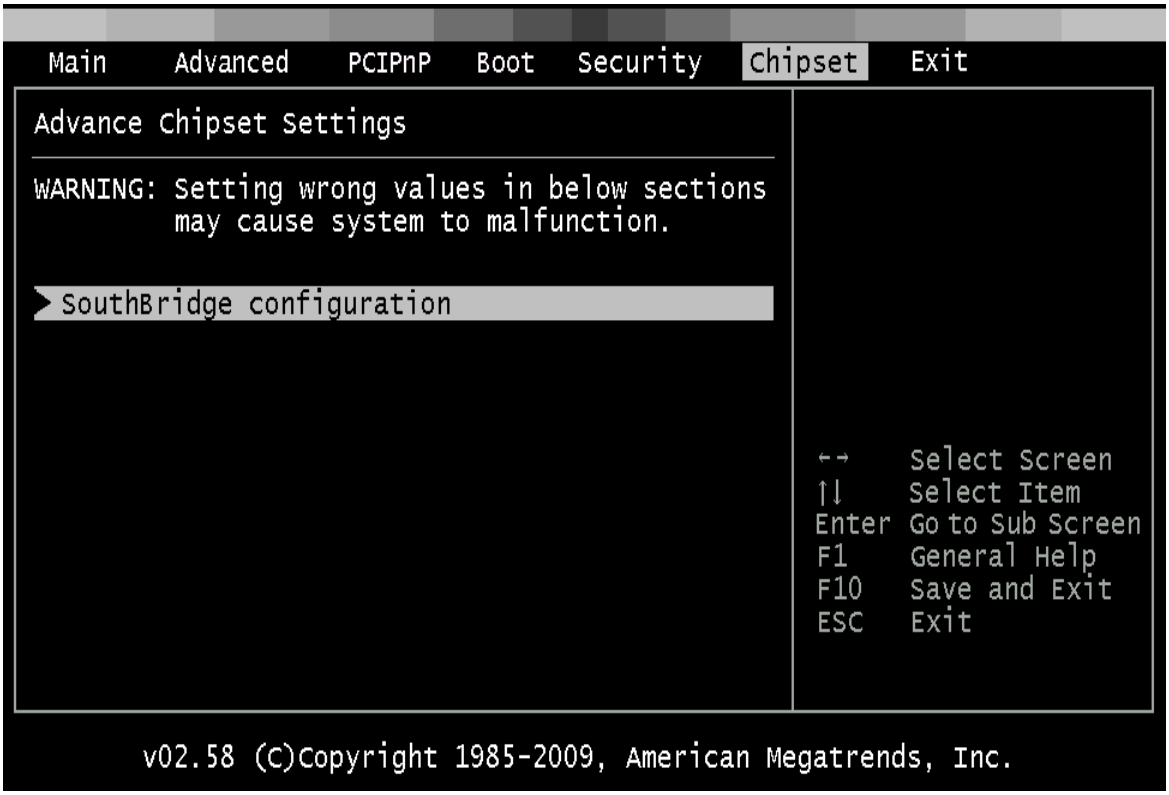
Clear User Password
Select this option and press <Enter> to access the sub menu. You can use the sub menu to clear the user password. Select Change User Password from the Security Setup menu and press <Enter>. Enter New Password: appears. Type the password and press <Enter>. The screen does not display the characters entered. Retype the password as prompted and press <Enter>. If the password confirmation is incorrect, an error message appears. The password is stored in NVRAM

Clear User Password
Select Clear User Password from the Security Setup menu and press <Enter>. Clear New Password [Ok] [Cancel] appears. Type the password and press <Enter>. The screen does not display the characters entered. Retype the password as prompted and press <Enter>. If the password confirmation is incorrect, an error message appears. The password is stored in NVRAM.

3.6 Chipset Setup

3.6.1 SouthBridge Configuration

You can use this screen to select options for the South Bridge Configuration. South Bridge is a chipset on the motherboard that controls the basic I/O functions. Use the up and down. <Arrow> keys to select an item. Use the <Plus> and <Minus> keys to change the value of the selected option.



ISA Configuration

This allows you to set the ISA bus frequency and to select the clock value of I/O and Memory.

Serial/Parallel Port Configuration

These options specify the serial port address and the parallel port mode and select the IRQ of Serial/Parallel Port.

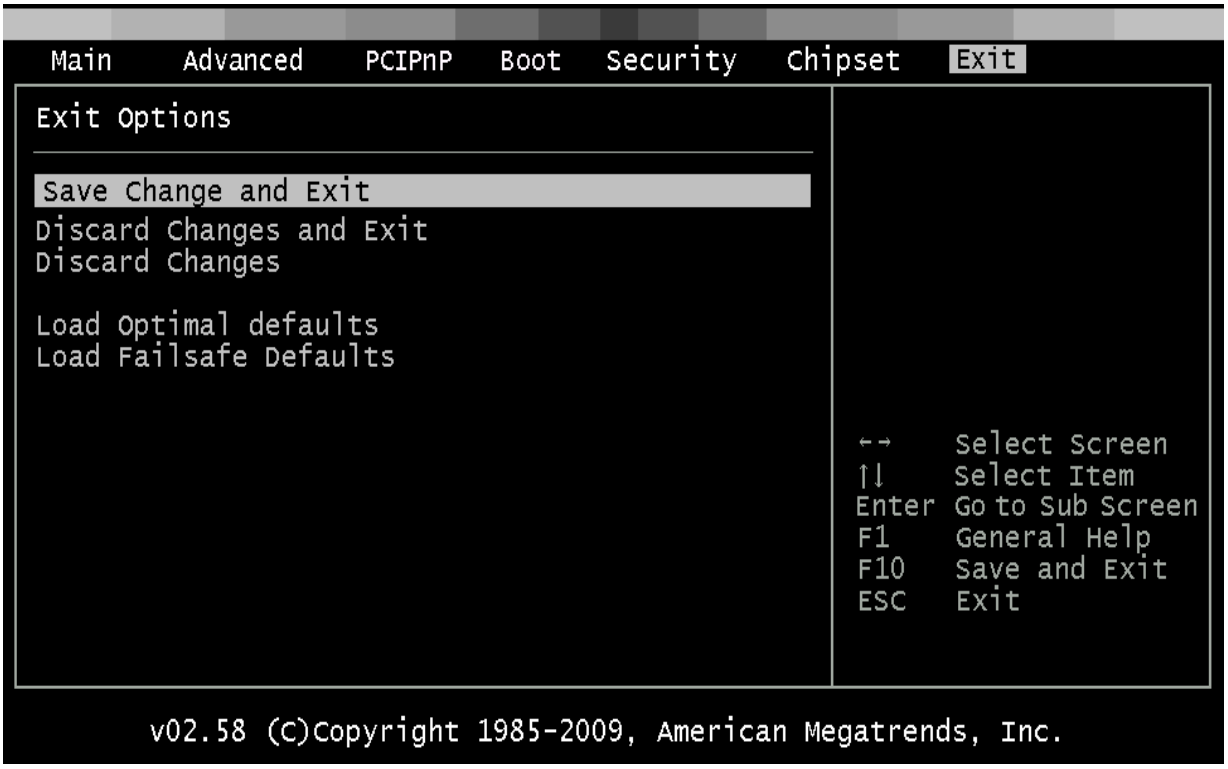
Option	Description
Disabled	Set this value to prevent the serial port from accessing any system resources. When this option is set to disabled, the serial port physically becomes unavailable.
3F8/IRQ4	Set this value to allow the serial port to use 3F8 as its I/O port address and IRQ 4 for the interrupt address. This is the default setting. The majority of serial port 1 or COM1 ports on computer systems use IRQ4 and I/O Port 3F8 as the standard setting. The most common serial device connected to this port is a mouse. If the system will not use a serial device, it is best to set this port to disabled.
2F8/IRQ3	Set this value to allow the serial port to use 2F8 as its I/O port address and IRQ 3 for the interrupt address. If the system will not use a serial device, it is best to set this port

	to disabled.
3E8/IRQ4	Set this value to allow the serial port to use 3E8 as its I/O port address and IRQ 4 for the interrupt address. If the system will not use a serial device, it is best to set this port to disabled.
2E8/IRQ3	Set this value to allow the serial port to use 2E8 as its I/O port address and IRQ 3 for the interrupt address. If the system will not use a serial device, it is best to set this port to disabled.

Option	Description
Normal	Set this value to allow the standard parallel port mode to be used. This is the default setting.
Bi-Directional	lSet this value to allow data to be sent to and received from the parallel port.
EPP	The parallel port can be used with devices that adhere to the Enhanced Parallel Port (EPP) specification. EPP uses the existing parallel port signals to provide asymmetric bi-directional data transfer driven by the host device.
EPP	The parallel port can be used with devices that adhere to the Extended Capabilities Port (ECP) specification. ECP uses the DMA protocol to achieve data transfer rates up to 2.5 Megabits per second. ECP provides symmetric bi-directional communication.

Option	Description
5	Set this value to allow the serial port to use Interrupt 3.
7	Set this value to allow the serial port to use Interrupt 7. This is the default setting. The majority of parallel ports on computer systems use IRQ7 and I/O Port 378H as the standard setting.

3.7 Exit



3.7.1 Save Changes and Exit

Once you finished the selections, this option will allow you to determine whether to accept the modifications or not. Select the “OK” to save the change and exit, if you select “NO”, you will return to Setup utility.

3.7.2 Discard Change and Exit

Select this option to exit the Setup without saving any change you have made in this session. Press “OK” will quit the Setup utility without saving any modifications. Press “NO” will return to Setup utility.

3.7.3 Discard Change

This option allows you to load the default values to your system configuration. These default settings will save the setup without making any permanent changes to the system configuration. Discard Changes This option allows you to discard the selections you made and restore the previously saved value.

3.7.4 Load Optimal Defaults

This option allows you to load the default values to your system configuration. These default settings are optimal and enable all high performance features.

3.7.5 Load Failsafe Defaults

This option allows you to load the failsafe default values for each of the parameters on the Setup menus, this will provide the most stable performance setting.

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 AR-B8172 provides the FLASH BIOS update function for you to easily to update BIOS. Please follow these operating steps to update BIOS:

STEP1	You must boot up system into MS-DOS first and please don't detect files CONFIG.SYS
	and AUTOEXEC.BAT.
STEP2	In the MS-DOS mode, you should execute the AMIFLASH program to update BIOS.
STEP3	Follow all messages then you could update BIOS smoothly.

4.2 WATCHDOG Programming

This section describes the usage of WatchDog. AR-B8172 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

WatchDog demo program in Turbo C++ as following:

```
//=========================================================================
// Turbo C++ Version 3.0 Copyright(c) 1990, 1992 by Borland International,Inc.
//=========================================================================
// Describe : Vortex86DX WatchDog timer test
// Date        : 09/16/2009
// Author     : Willy
//=========================================================================

//=========================================================================
//    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=0x22;  // Index Port 22h, Date Port 23h
    unsigned char Signal;
    unsigned char Time;
    unsigned long Timer;
    unsigned char Counter0;
    unsigned char Counter1;
    unsigned char Counter2;

    int Temp;

    if ( argc != 3 )
      {   Show_Help();     return 1;        }

    clrscr();

    Signal=atoi(argv[1]);      // Signal Set Bits
    Signal=Signal<<4;

    Time=atoi(argv[2]);                // Watchdog counter
    Timer=Time*32768;
    Counter0=(unsigned char)Timer;
    Counter1=(unsigned char)(Timer>>8);
    Counter2=(unsigned char)(Timer>>16);

    // Select Watchdog Signal Source
    outportb(IO_Port_Address,0x38);  // WDT0 signal select
    outportb(IO_Port_Address+1,Signal);

    // Set Watchdog timer
    outportb(IO_Port_Address,0x39);  // WDT0 Counter0
    outportb(IO_Port_Address+1,Counter0);

    outportb(IO_Port_Address,0x3A);  // WDT0 Counter1
    outportb(IO_Port_Address+1,Counter1);

    outportb(IO_Port_Address,0x3B);  // WDT0 Counter2
    outportb(IO_Port_Address+1,Counter2);

    // Set Watchdog Enabled.
    outportb(IO_Port_Address,0x37);  // WDT0 Enabled Control Reg.
    outportb(IO_Port_Address+1,0x40);

    textcolor(YELLOW);
    for(Temp=Time;Temp>0;Temp--)
     {
        gotoxy(20,10);

        if(Signal==0xD0)
            cprintf(">>> After %3d Second will reset the system. <<<",Temp);
        else
            cprintf(">>> After %3d Second Watchdog Signal will occur. <<<",Temp);
```

```
        delay(1000);
      }

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

    if(Signal==0xD0)
        cprintf("If you can see this message, Reset system is Fail");
    else
        cprintf("If you can see this message, Watchdog Signal is occur.");

    return 1;
}

//===========================================================================
// Function : Show_Help()
// Input      : -
// Change  : -
// Return    : -
// Description    : Show Title string.
//===========================================================================
void Show_Help()
{
    clrscr();
    printf("WatchDog Test for Vortex86DX     \n\n");
    printf("Signal Select                    \n");
    printf("1 : IRQ3     2 : IRQ4     4 : IRQ5 \n");
    printf("4 : IRQ6     5 : IRQ7     6 : IRQ9 \n");
    printf("7 : IRQ10    8 : IRQ11    9 : IRQ12\n");
    printf("10: IRQ14    11: IRQ15    12: NMI    \n");
    printf("13: System Reset                 \n\n");
    printf("Sample:                          \n");
    printf("            WDT.EXE 1 10           \n");
    printf("For 10 seconds to IRQ3.          \n\n");
    printf("            WDT.EXE 13 10          \n");
    printf("For 10 seconds to system reset.\n");
}
```

4.3 GPIO Programming

Data Port
(GPIO0 Base Address 0 Refers to the Register of index 61h-60h, IDSEL = AD18/SB of PCI Configuration Register)
(GPIO1 Base Address 1 Refers to the Register of index 63h-62h, IDSEL = AD18/SB of PCI Configuration Register)
(GPIO2 Base Address 2 Refers to the Register of index 65h-64h, IDSEL = AD18/SB of PCI Configuration Register)
(GPIO3 Base Address 3 Refers to the Register of index 67h-66h, IDSEL = AD18/SB of PCI Configuration Register)
(GPIO4 Base Address 4 Refers to the Register of index 69h-68h, IDSEL = AD18/SB of PCI Configuration Register)

IO Address	Register Name
BA[0] + 00h	GPIO PORT0 Data Register
BA[1] + 00h	GPIO PORT1 Data Register
BA[2] + 00h	GPIO PORT2 Data Register
BA3 + 00h	GPIO PORT3 Data Register
BA4 + 00h	GPIO PORT4 Data Register

Direction Port
(Base Address Refers to the Register of index 6Bh-6Ah, IDSEL = AD18/SB of PCI Configuration Register)

IO Address	Register Name
BA + 00h	GPIO PORT0 Data Register
BA + 01h	GPIO PORT1 Data Register
BA + 02h	GPIO PORT2 Data Register
BA + 03h	GPIO PORT3 Data Register
BA + 04h	GPIO PORT4 Data Register
BA + 06h	GPIO PORT1 Interrupt Status Register
BA + 07h	GPIO PORT0 Interrupt Status Register

GPIO demo program in Turbo C++ as following:

```
//==========================================================================
// Turbo C++ Version 3.0 Copyright(c) 1990, 1992 by Borland International,Inc.
//==========================================================================
// Describe : GPIO00~GPIO07 GPIO10~GPIO17 Test utility for Vortex86DX.
// Date         : 09/17/2009
// Author     : Willy
//==========================================================================

//==========================================================================
//   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-B8172";
   unsigned char       IO_PORT_BASE=0x22;   // 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("|| Vortex86DX 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);

   // Set GPIO00~07 to Output
   outportb(IO_PORT_BASE,0x4E);
   outportb(IO_PORT_BASE+1,0xFF);        // bit=1 , output

   // Set GPIO10~GPIO17 to Input
   outportb(IO_PORT_BASE,0x4F);
   outportb(IO_PORT_BASE+1,0x00);// bit=0 , input

   // Set GPIO00~07 to AA
```

```
    outportb(IO_PORT_BASE,0x47);
    outportb(IO_PORT_BASE+1,0xAA);

    // Read GPIO10~17 Status, if not AA error.
    outportb(IO_PORT_BASE,0x4C);
    if(inportb(IO_PORT_BASE+1)!=0xAA)
        result=1;

    // Set GPIO00~07 to 55
    outportb(IO_PORT_BASE,0x47);
    outportb(IO_PORT_BASE+1,0x55);

    // Read GPIO10~17 Status, if not 55 error.
    outportb(IO_PORT_BASE,0x4C);
    if(inportb(IO_PORT_BASE+1)!=0x55)
        result=2;

    // Set GPIO10~GPIO17 to Output
    outportb(IO_PORT_BASE,0x4F);
    outportb(IO_PORT_BASE+1,0xFF);        // bit=1 , output

    // Set GPIO00~07 to Input
    outportb(IO_PORT_BASE,0x4E);
    outportb(IO_PORT_BASE+1,0x00);// bit=0 , input

    // Set GPIO10~17 to AA
    outportb(IO_PORT_BASE,0x4D);
    outportb(IO_PORT_BASE+1,0xAA);

    // Read GPIO00~07 Status, if not AA error.
    outportb(IO_PORT_BASE,0x46);
    if(inportb(IO_PORT_BASE+1)!=0xAA)
        result=3;

    // Set GPIO10~17 to 55
    outportb(IO_PORT_BASE,0x4D);
    outportb(IO_PORT_BASE+1,0x55);

    // Read GPIO00~07 Status, if not 55 error.
    outportb(IO_PORT_BASE,0x46);
    if(inportb(IO_PORT_BASE+1)!=0x55)
        result=4;

    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 Vortex86DX\n\n");
    printf("GPIO00 ▲迁?                    ▲ VCC     \n");
```

```
    printf("GPIO01 ▯◀迂迂?          奼迂迂迂? ▯ GPIO07\n");
    printf("GPIO02 ▯◀迂迂迂?       ?奼迂迂? ▯ GPIO06\n");
    printf("GPIO03 ▯◀迂迂迂迂?    ??奼迂? ▯ GPIO05\n");
    printf("GND    ▯       ????   ???奼? ▯ GPIO04\n");
    printf("GPIO10 ▯◀迂????   ????    ▯ VCC    \n");
    printf("GPIO11 ▯◀迂迂???       迂迂迂? ▯ GPIO17\n");
    printf("GPIO12 ▯◀迂迂迂??          迂迂? ▯ GPIO16\n");
    printf("GPIO13 ▯◀迂迂迂迂?             迂? ▯ GPIO15\n");
    printf("GND    ▯                          ? ▯ GPIO14\n");
}

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

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

5 ELECTRICAL CHARACTERISTICS

5.1 Basic Electrical Characteristics Table

Electrical Characteristics					
Symbol	Parameter / Condition	Value			Unit
		Min.	Typ.	Max.	
TA	Ambient Temperature	0	-	60	℃
Tstg	Storage Temperature	-20	-	80	℃
+12V	External power input for system or +12Vdc power output	11.4	12.0	12.6	V
+5V	+5Vdc power input	4.75	5.0	5.25	V
GPIO VIL	GPIO's maximum Input LOW voltage	-	0	0.8	V
GPIO VIH	GPIO's minimum input HIGH voltage	2.5	3.3	-	V
GPIO VOL	GPIO's typical output LOW voltage	-	-	0.4	V
GPIO VOH	GPIO's typical output HIGH voltage	2.4	-	-	V