CSB100-895

Slim & Compact SBC System

User's Manual

Version 1.0
(Oct. 2016)

iBASE

Compliance

CE

This is a class B product. This product has passed CE tests for environmental specifications and limits. This product is in accordance with the directives of the Union European (EU). If users modify and/or install other devices in this equipment, the CE conformity declaration may no longer apply.

FC

This product has been tested and found to comply with the limits for a Class B device, pursuant to Part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference in a residential installation. This equipment generates, uses and can radiate radio frequency energy and, if not installed and used in accordance with manufacturer's instructions, may cause harmful interference to radio communications.

WEEE

This product must not be disposed of as normal household waste, in accordance with the EU directive of for waste electrical and electronic equipment (WEEE - 2012/19/EU). Instead, it should be disposed of by returning it to a municipal recycling collection point. Check local regulations for disposal of electronic products.

Green IBASE

This product is compliant with the current RoHS restrictions and prohibits use of the following substances in concentrations exceeding 0.1% by weight (1000 ppm) except for cadmium, limited to 0.01% by weight (100 ppm).

- Lead (Pb)
- Mercury (Hg)
- Cadmium (Cd)
- Hexavalent chromium (Cr6+)
- Polybrominated biphenyls (PBB)
- Polybrominated diphenyl ether (PBDE)

Important Safety Information

Carefully read the precautions before using the device.

Environmental conditions:

- Lay the device horizontally on a stable and solid surface in case the device may fall, causing serious damage.
- Leave plenty of space around the device and do not block the openings for ventilation. NEVER DROP OR INSERT ANY OBJECTS OF ANY KIND INTO THE VENTIILATION OPENINGS.
- Slots and openings on the chassis are for ventilation. Do not block or cover these openings. Make sure you leave plenty of space around the device for ventilation. NEVER INSERT OBJECTS OF ANY KIND INTO THE VENTILATION OPENINGS.
- Use this product in environments with ambient temperatures between 0˚C and 45˚C.
- DO NOT LEAVE THIS DEVICE IN AN ENVIRONMENT WHERE THE STORAGE TEMPERATURE MAY GO BELOW -20˚C OR ABOVE 80˚C. This could damage the device. The device must be used in a controlled environment.

Care for your IBASE products:

- Before cleaning the device, turn it off and unplug all cables such as power in case a small amount of electrical current may still flow.
- Use neutral cleaning agents or diluted alcohol to clean the device chassis with a cloth. Then wipe the chassis with a dry cloth.
- Vacuum the dust with a computer vacuum cleaner to prevent the air vent or slots from being clogged.

WARNING

Attention during use:

- Do not use this product near water.
- Do not spill water or any other liquids on your device.
- Do not place heavy objects on the top of the device.
- Operate this device from the type of power indicated on the marking label. If you are not sure of the type of power available, consult your distributor or local power company.
- Do not walk on the power cord or allow anything to rest on it.
- If you use an extension cord, make sure that the total ampere rating of the product plugged into the extension cord does not exceed its limits.

Avoid Disassembly

You are not suggested to disassemble, repair or make any modification to the device. Disassembly, modification, or any attempt at repair could generate hazards and cause damage to the device, even bodily injury or property damage, and will void any warranty.

CAUTION

Danger of explosion if internal lithium-ion battery is replaced by an incorrect type. Replace only with the same or equivalent type recommended by the manufacturer. Dispose of used batteries according to the manufacturer's instructions.

Warranty Policy

- **IBASE standard products:**

  24-month (2-year) warranty from the date of shipment. If the date of shipment cannot be ascertained, the product serial numbers can be used to determine the approximate shipping date.

- **3rd-party parts:**

  12-month (1-year) warranty from delivery for the 3rd-party parts that are not manufactured by IBASE, such as CPU, memory, HDD, power adapter, panel and touchscreen.

* PRODUCTS, HOWEVER, THAT FAILS DUE TO MISUSE, ACCIDENT, IMPROPER INSTALLATION OR UNAUTHORIZED REPAIR SHALL BE TREATED AS OUT OF WARRANTY AND CUSTOMERS SHALL BE BILLED FOR REPAIR AND SHIPPING CHARGES.

Technical Support & Services

1. Visit the IBASE website at www.ibase.com.tw to find the latest information about the product.
2. If you need any further assistance from your distributor or sales representative, prepare the following information of your product and elaborate upon the problem.
   - Product model name
   - Product serial number
   - Detailed description of the problem
   - The error messages in text or in screenshots if there is any
   - The arrangement of the peripherals
   - Software in use (such as OS and application software, including the version numbers)
3. If repair service is required, you can download the RMA form at http://www.ibase.com.tw/english/Supports/RMAService/. Fill out the form and contact your distributor or sales representative.

iBASE

Table of Contents

Chapter 1
General Information

The information provided in this chapter includes:

- Features
- Packing List
- Specifications
- Overview
- Dimensions

1.1 Introduction

The CSB100-895 is a product series of IBASE embedded computing system, applicable to thin clients, smart industrial automation or controller, and retail equipment. It is a compact and fanless design with an Intel®-Atom™ D2550 processor and an Intel® 82583V Ethernet controller. This product features iSMART that allows the device capable of auto-scheduling for general applications and gives energy savings on power. It is able to be operated at the ambient operating temperature ranging from 0 ~ 45 °C, and even from -20 ~ 80 °C for storage.

1.2 Features

- Slim and compact fanless design with IBASE 3.5” disk-sized SBC
- On board Intel® Atom™ D2550 at 1.86GHz
- iSMART for auto-scheduler and power resume
- 2.5” drive tray for SATA HDD
- 6 x USB 2.0, and 2 x mini PCIe
- Wall mount kit included
- 12 ~ 24V wide-range DC power input

1.3 Packing List

Your CSB100-895 package should include the items listed below. If any of the items below is missing, contact the distributor or the dealer from whom you purchased the product.

- CSB100-895 x 1
- Power Adapter x 1
- Power Cord x 1
- Wall Mount Kit x 1
- Screws for Wall Mount Kit x 4
- Disk (including drivers and this user manual) x 1

1.4 Specifications

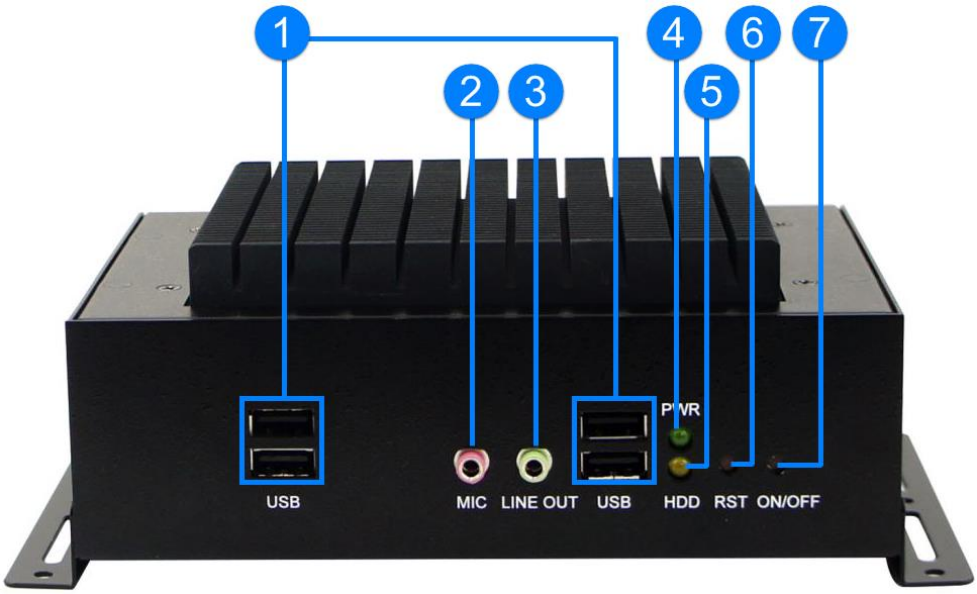
Product Name	CSB100-895
System	
Motherboard	IB895
Operating System	• Windows 7 (32-bit) • Linux
CPU	Intel® Atom™ D2550 at 1.86 GHz
Chipset	Intel® NM10 PCH
Memory	1 x DDR3L-1333 SO-DIMM 2GB, expandable to 4 GB
Graphics	Intel® GMA3650 integrated graphics
Super I/O	Nuvoton NCT6627UD
Audio Codec	Realtek ALC269QHD
Network	Intel® 82583V GbE LAN
Power Supply	60W power adaptor
BIOS	AMI BIOS
Watchdog	Watchdog Timer 256 segments, 0, 1, 2…255 sec/min
Chassis	Aluminum & steel, black
Mounting	• Desktop mount • Wall mount
Dimensions (W x H x D)	113.6 x 52.4 x 172.4 mm (4.47” x 2.06” x 6.79”)
Weight	1.3 kg (2.87 lb)
Certificate	CE / LVD / FCC Class B / CCC
I/O Ports	
DC Input	DC Jack
Storage	1 x 2.5” drive bay for SSD / HDD
LAN	2 x RJ45 GbE LAN
USB	6 x USB 2.0
Serial	• 1 x COM1 (RS-232/422/485, selectable from BIOS) • 1 x COM2 (RS-232 only)
Display	1 x DVI-I Port
Audio Jack	• 1 x Microphone input • 1 x Line-out
SATA	2 x SATA II connector

Expansion	• 2 x Mini PCIe slot (full-sized) • 1 x Mini PCIe slot (half-sized)
Environment	
Temperature	• **Operating:** 0 ~ 45 °C (32 ~ 113 °F) • **Storage:** -20~ 80 °C (-4 ~ 176 °F)
Relative Humidity	5 ~ 90% at 45 °C (non-condensing)
Vibration Protection	• **Operating:** 0.25 Grms / 5 ~ 500 Hz • **Non-operating:** 1 Grms / 5 ~ 500Hz
Shock Protection	• **Operating:** 20 g / 11 ms • **Non-operating:** 40 g / 11 ms

All specifications are subject to change without prior notice.

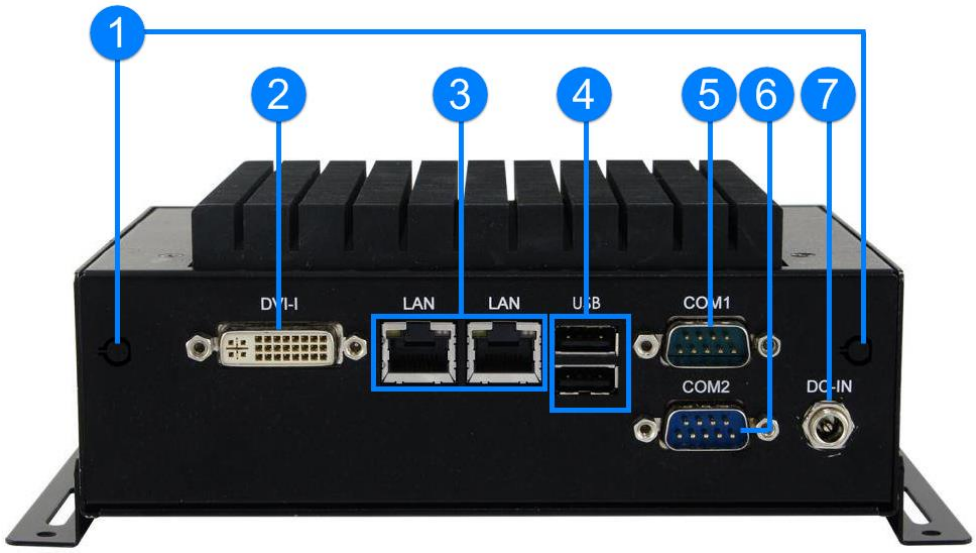
1.5 Overview

Front View



No.	Name	No.	Name
1	USB 2.0 Ports	5	HDD LED Indicator
2	Audio Jack – Microphone Input	6	Reset Button
3	Audio Jack – Line-Out	7	Power Button
4	Power LED Indicator		

Rear View

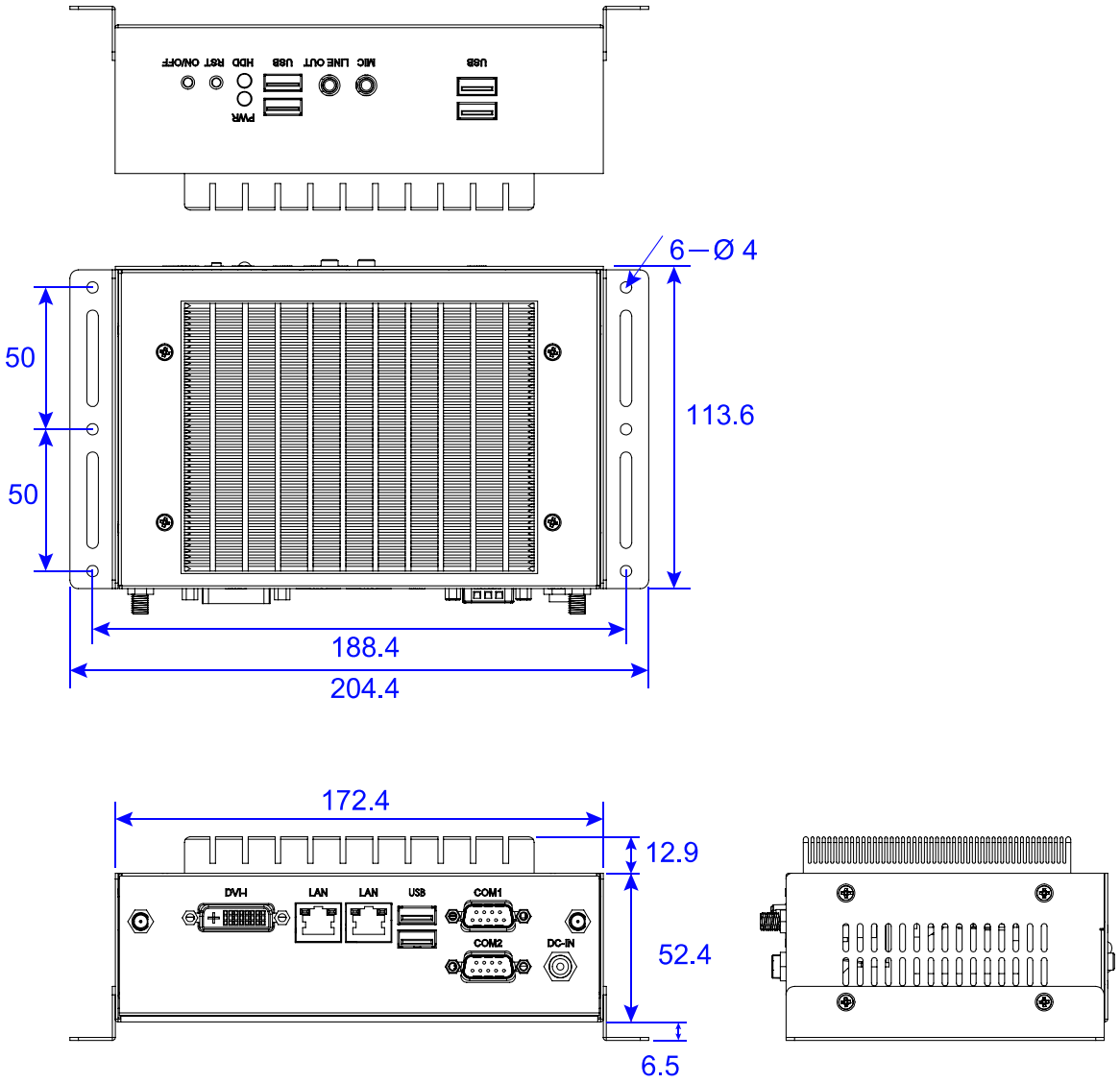


No.	Name	No.	Name
1	Antenna Holes	5	COM1 Port - RS-232 / 422 / 485
2	DVI-I Port	6	COM2 Ports – RS-232 Only
3	LAN Ports	7	DC-In Power Connector
4	USB 2.0 Ports		

Oblique View

1.6 Dimensions

Unit: mm



Chapter 2
Hardware Configuration

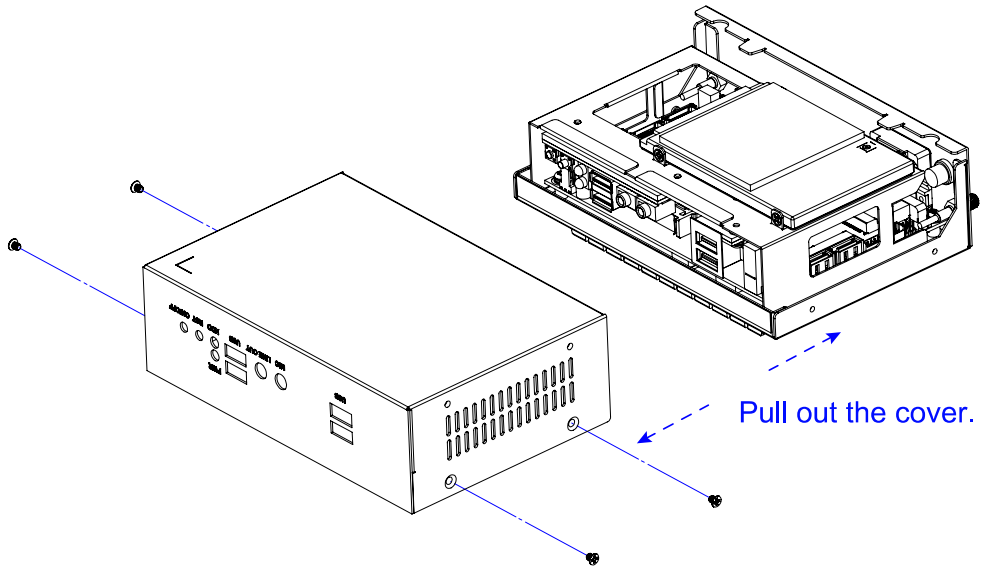
The information provided in this chapter includes:

- Essential installations before you begin
- Information and locations of connectors

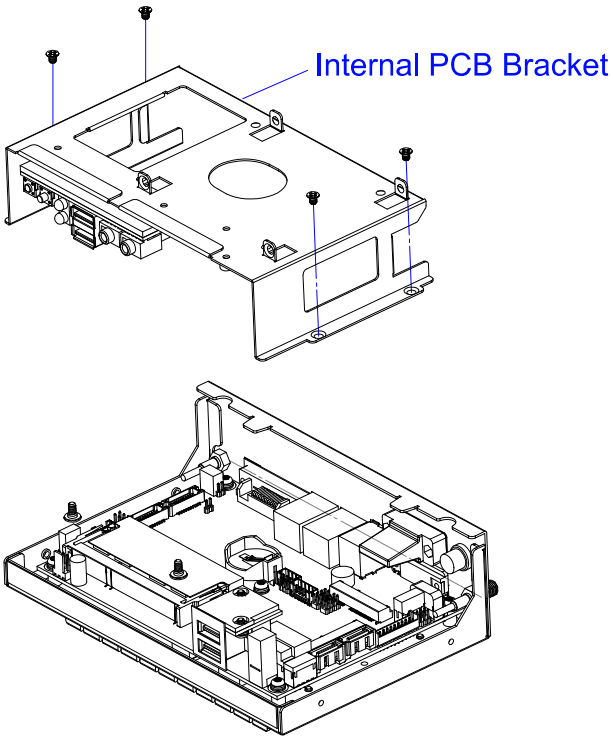
2.1 Essential Installations Before You Begin

Before installations, you need to:

1. Disassemble the device cover by loosen 4 screws from the device and pull out the cover.



2. Loosen 4 screws to free up the internal PCB bracket for removal.

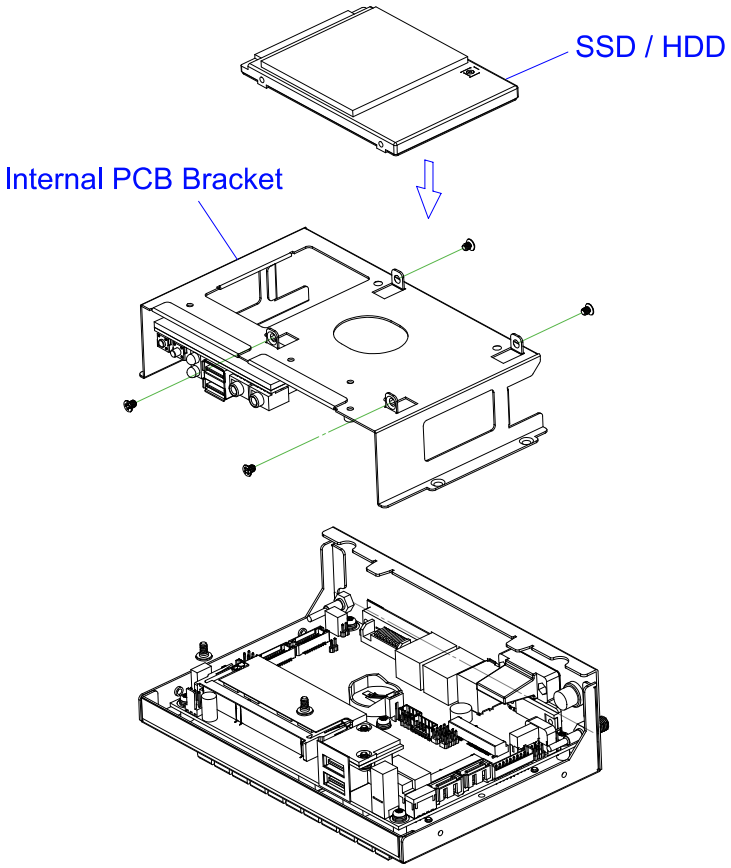


2.1.1 HDD Installation

If you need to install or replace a SSD / HDD, follow the instructions below for installation after you disassemble the device cover and the internal PCB bracket.

1. Loosen 4 screws below.

2. Attach your SSD / HDD and tighten these screws.



2.1.2 Memory Installation

There is one SO-DIMM DDR3L memory slots inside CSB100-895 and the maximum memory is expandable up to 4 GB.

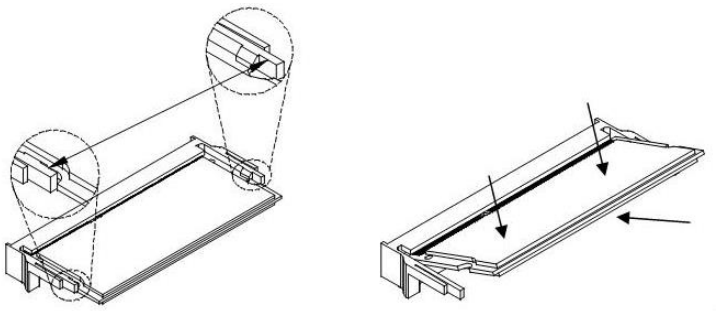
If you need to install or replace a memory module, follow the instructions below.

1. Take out the PCB bracket along with the I/O module locating above the memory slot.

2. Locate the memory slot on the board.

3. Align the key of the memory module with that on the memory slot and insert the module slantwise.

4. Gently push the module down in an upright position until the clips of the slot close to hold the module in place when the module touches the bottom of the slot.



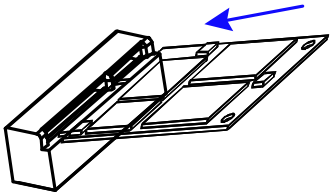
To remove the module, press the clips outwards with both hands.

2.1.3 Mini PCIe Card Installation

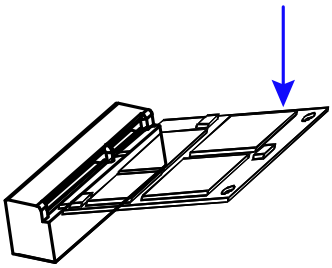
If you need to use a mini PCIe card for expansion slots, follow the instructions below for installation after you disassemble the device cover and the internal PCB bracket.

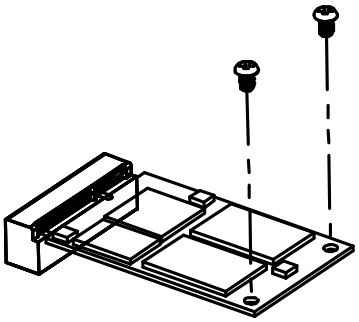
1. Take out the PCB bracket with the I/O module carefully.

2. Align the key of the mini PCIe card to the Mini PCIe interface, and insert the card slantwise.



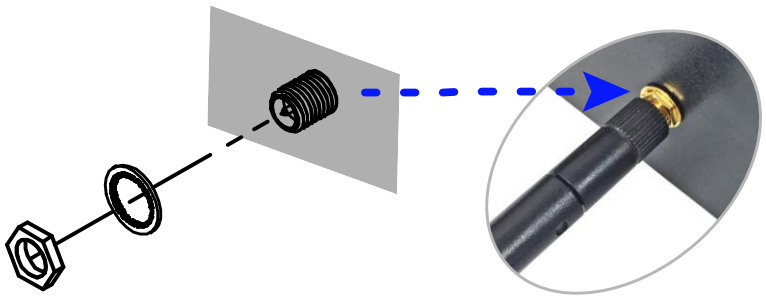
3. Push the mini PCIe card down, fix it onto the standoff(s) with one or two screws.





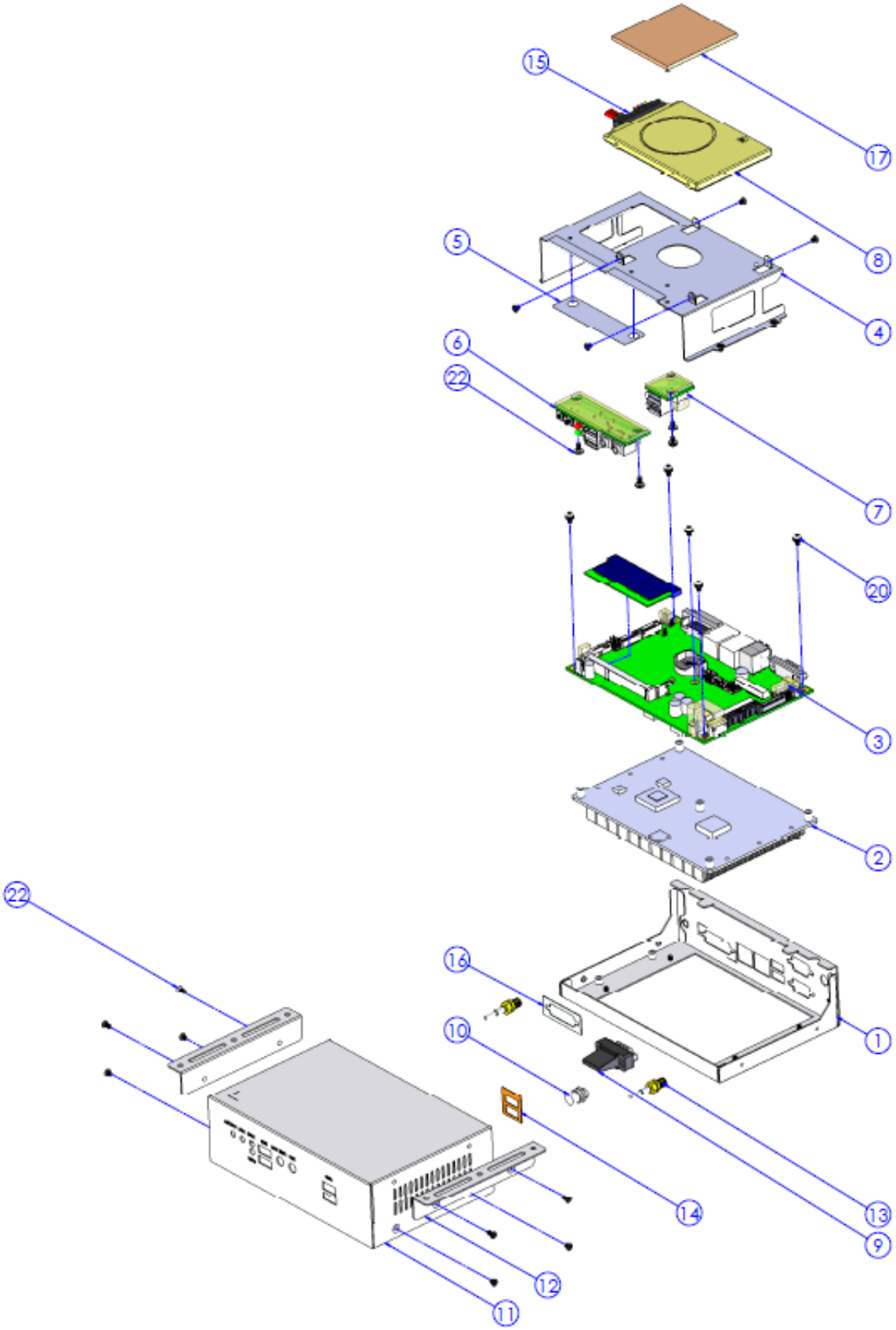
2.1.4 WiFi / 3G / 4G Antenna Installation

Thread the WiFi / 3G / 4G antenna cable through an antenna hole. Then fasten the antenna as shown below.



Info: The diameter of the nut is around 6.35 mm (0.25”-36UNC).

2.1.5 Device Exploded Diagram



Item	Name	Q'ty
1	CSB100-895 base	1
2	CSB100-895 heatsink	1
3	IB895-1.0	1
4	CSB100-895 hard disk bracket	1
5	CSB100-895 mylar	1
6	ID-board	1
7	ID112	1
8	SATA 2.5" HDD	1
9	D-Sub 9 pin cable	1
10	Power Switch connector	1
11	CSB100-895 cover	1
12	CSB100-895 bracket	2
13	Antenna	2
14	CSB110-905 gasket	1
15	Cable	1
16	DVI-0.5T gasket	1
17	Pad	1
18	CSB100-895 base gasket	1
19	Screw (M3*4L)	16
20	Screw (M3*6L)	11
21	Screw (M3*6L)	4
22	Screw	2

2.1.6 Mounting Brackets Installation

Note: Before mounting the system on wall, ensure that you are following all applicable building and electric codes.

Requirements

When mounting, ensure that you have enough room for power and signal cable routing. And have good ventilation for power adapter. The method of mounting must be able to support weight of the CSB100-895 plus the suspend weight of all the cables to be attached to the system. Use the following methods for mounting your system:

Selecting the Location

Plan the mounting location thoroughly. Locations such as walkway areas, hallways, and crowded areas are not recommended. Mount the product to a flat, sturdy, structurally sound column or wall surface.

The best mounting surface is a standard countertop, cabinet, table, or other structure that is minimally the width and length of the product. This will reduce the risk that someone may accidentally wall into and damage the product. Local laws governing the safety of individuals might require this type of consideration.

Selecting the type of wall construction

1. Mounting on a hollow wall

   - **Wood surface**

     Use construction-grade wood and the recommended minimum thickness is 38 x 25.4 mm (1.5” x 10”).

     Note: This method provides the most reliable attachment for the product with little risk that the product may come loose or require ongoing maintenance.

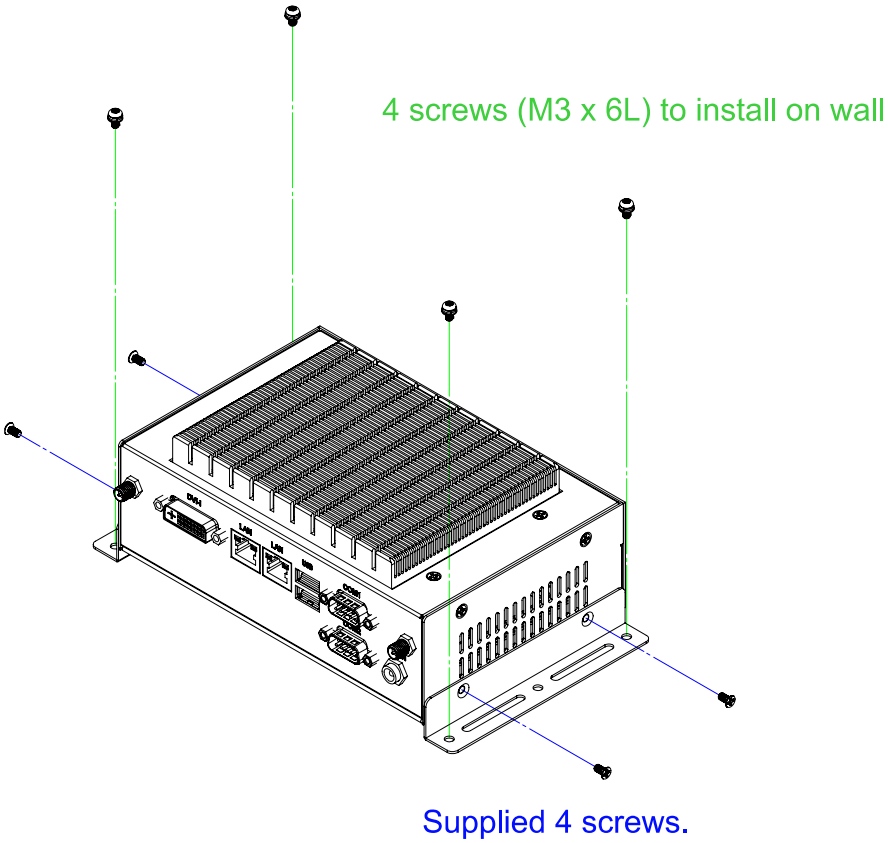
   - **Drywall**

     Drywall over wood studs is acceptable.

2. Mounting on a solid concrete or brick wall with flat and smooth surface

Wall mount installation instructions:

1. Attach the mounting brackets to your CSB100-895, and secure with the supplied 4 screws.
2. Prepare at least 4 screws (M3, 6 mm) to install the device on wall .



You can install CSB100-895 on plastic (LCD monitor), wood, drywall surface over studs, or a solid concrete or metal plane directly. The types of fasteners required are dependent on the type of wall construction.

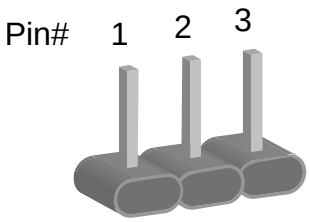
Fasteners are not supplied in the product package. You will need to prepare the fasteners. Choose fasteners that are rated either **Medium Duty** or **Heavy Duty**. To assure proper fastener selection and installation, follow the fastener manufacturer's recommendations.

2.2 Setting the Jumpers

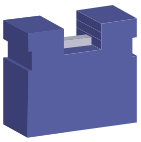
Set up and configure your CSB100-895 by using jumpers for various settings and features according to your needs and applications. Contact your supplier if you have doubts about the best configuration for your use.

2.2.1 How to Set Jumpers

Jumpers are short-length conductors consisting of several metal pins with a non-conductive base mounted on the circuit board. Jumper caps are used to have the functions and features enabled or disabled. If a jumper has 3 pins, you can connect either PIN1 to PIN2 or PIN2 to PIN3 by shorting.

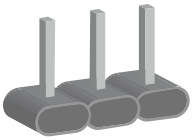
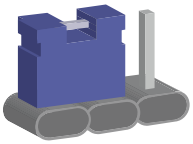
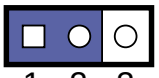
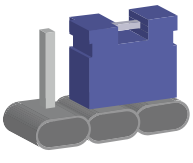
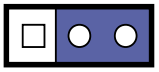


A 3-pin jumper



A jumper cap

Refer to the illustration below to set jumpers.

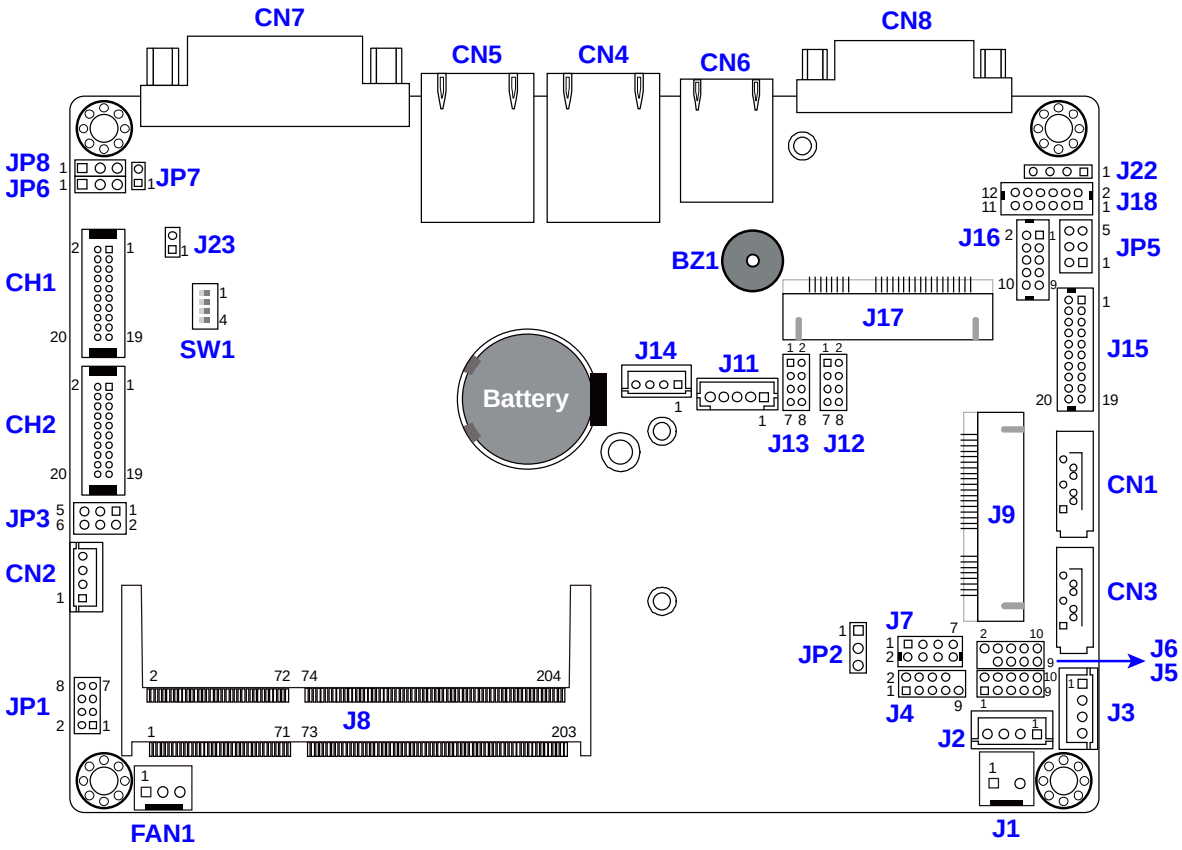
Pin closed	Oblique view	Schematic illustration in the manual
Open		
1-2		
2-3		

When two pins of a jumper are encased in a jumper cap, this jumper is **closed**, i.e. turned **On**.

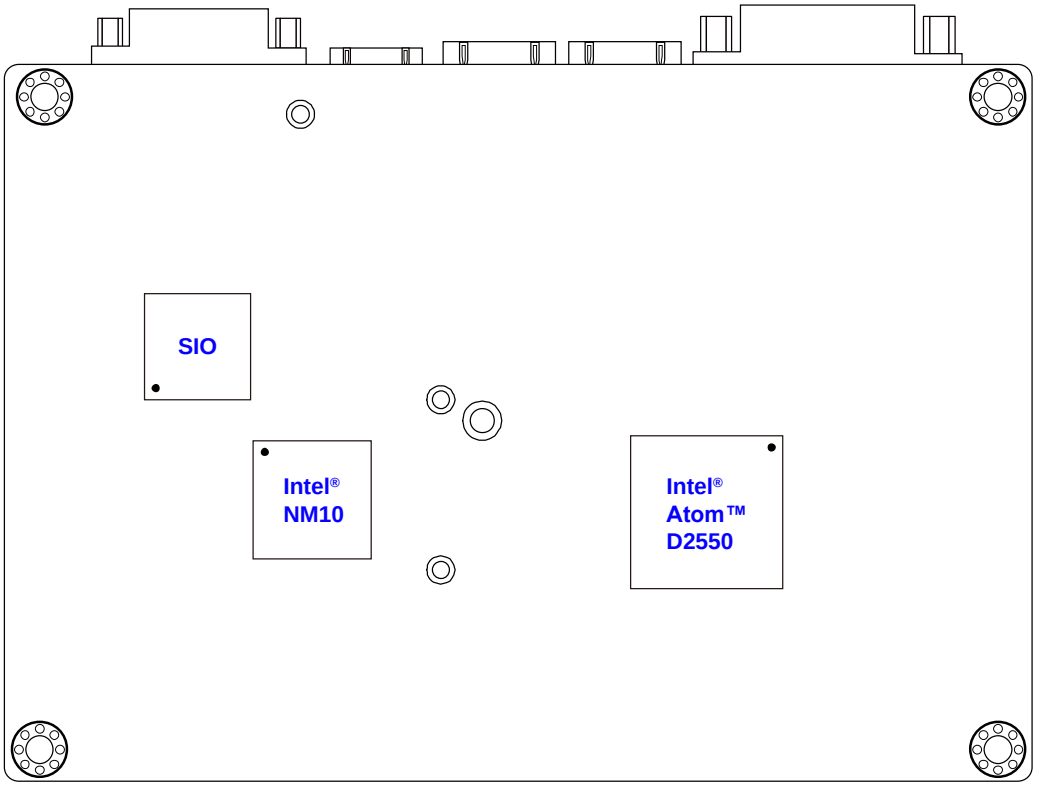
When a jumper cap is removed from two jumper pins, this jumper is **open**, i.e. turned **Off**.

2.3 Jumper & Connector Locations on Motherboard

Motherboard: IB895



IB895 - top

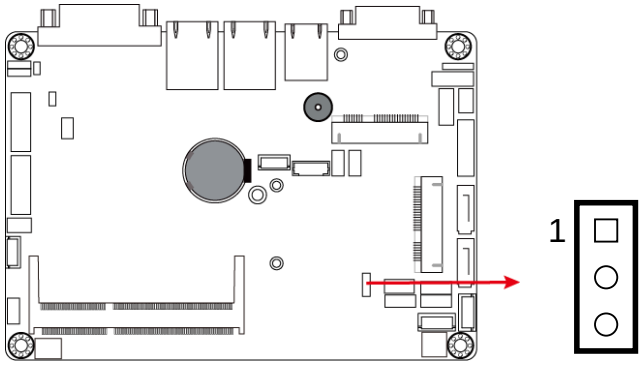


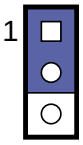
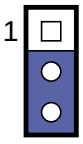
IB895 - bottom

2.4 Jumpers Quick Reference

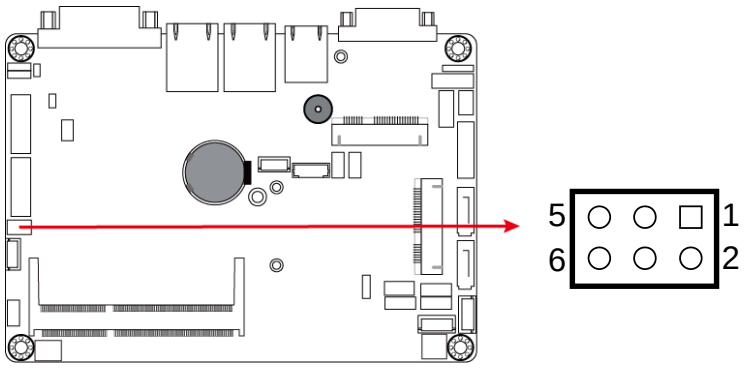
Function	Connector Name	Page
CMOS Data Clearance	JP2	23
Backlight Power Selection	JP3	24
COM1 RS-232 Port Power Selection	JP5	25
LVDS Panel Power Selection	JP6	26
Backlight Adjustment Level Selection (PWM Mode)	JP7	26
Factory Use Only	JP8	--

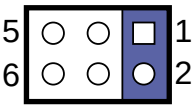
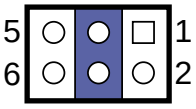
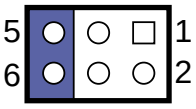
2.4.1 CMOS Data Clearance (JP2)



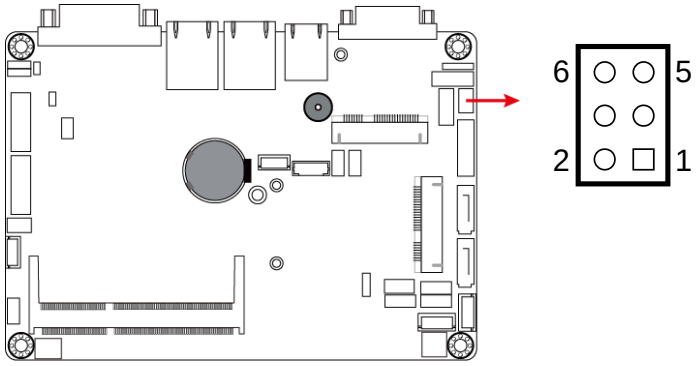
Function	Pin closed	Illustration
Normal (default)	1-2	
Clear CMOS	2-3	

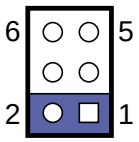
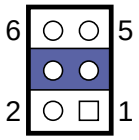
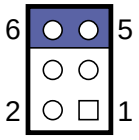
2.4.2 Backlight Power Selection (JP3)



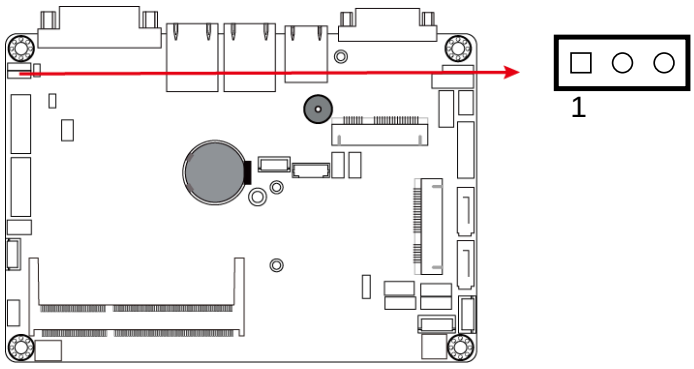
Function	Pin closed	Illustration
3.3V (default)	1-2	
5V	3-4	
12V	5-6	

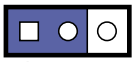
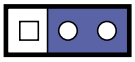
2.4.3 COM1 RS-232 Power Selection (JP5)



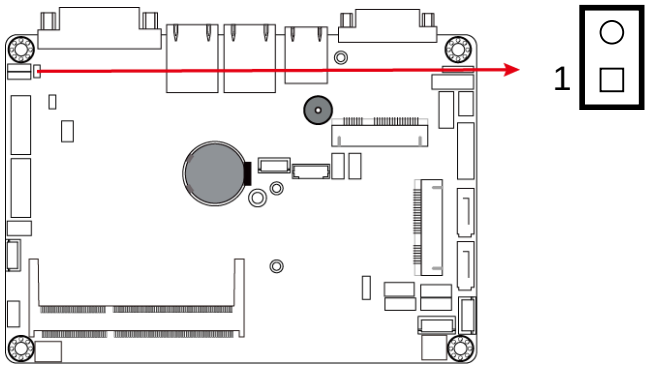
Function	Pin closed	Illustration
12V	1-2	
RI (default)	3-4	
5V	5-6	

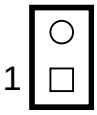
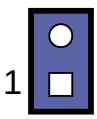
2.4.4 LVDS Panel Power Selection (JP6)



Function	Pin closed	Illustration
3.3V (default)	1-2	
5V	2-3	

2.4.5 Backlight Adjustment Level Selection (PWM Mode) (JP7)

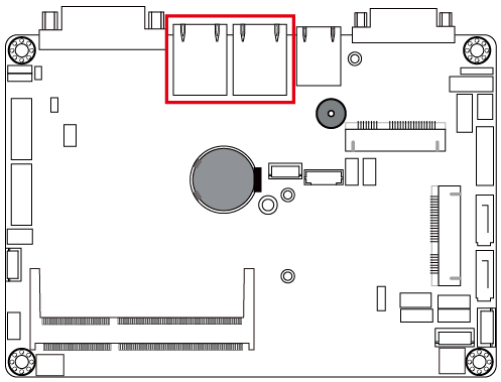


Function	Pin closed	Illustration
3.3V (default)	Open	
5V	Close	

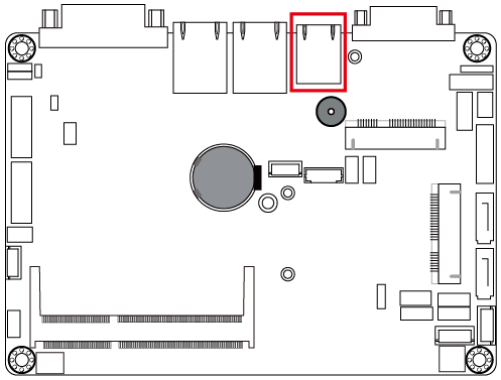
2.5 Connectors Quick Reference

Function	Connector Name	Page
LAN Port (GbE)	CN4, CN5	28
USB 2.0 Port	CN6	28
DVI-I Port	CN7	28
COM1 RS-232/422/485 Port	CN8	29
SATA II Port	CN1, CN3	30
LCD Backlight Connector	CN2	30
Motherboard Power Input Connector	J1	31
SATA HDD Power Connector	J2, J3	31
Digital I/O Connector	J5	32
Keyboard & Mouse Connector	J7	32
DDR3L SO-DIMM Socket	J8	33
Full-Size Mini-PCIe / mSATA Connector	J9	33
Half-Size Mini-PCIe Connector	J17	33
Smart Battery Interface Connector	J11	34
USB 2.0 Connector	J12, J13	34
COM3 & COM4 Port	J15	35
COM2 RS-232 Connector	J16	36
Audio Connector	J18	36
Amplify Connector	J22	37
Front Panel Setting Connector for LED Indicators	JP1	37
LVDS Connector	CH1, CH2	38
CPU Fan Power Connector	FAN1	39
Factory Use Only	J4, J6, J14, J23	--

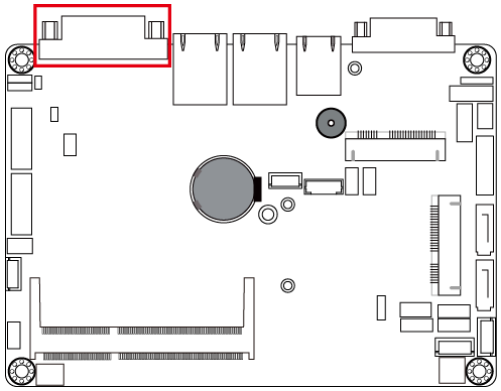
2.5.1 LAN Port (GbE) (CN4, CN5)



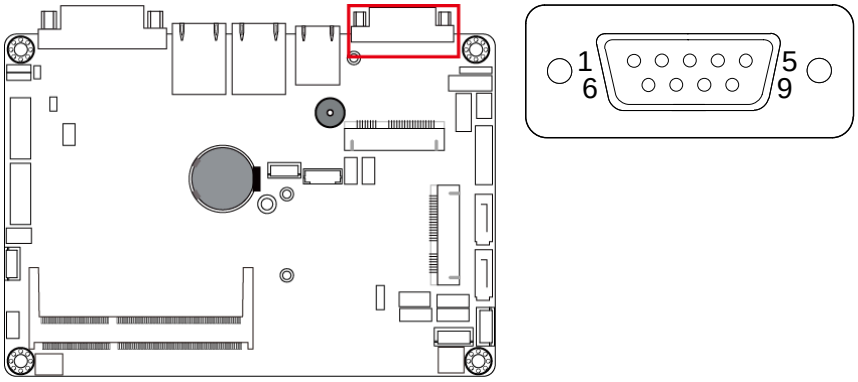
2.5.2 USB 2.0 Port (CN6)



2.5.3 DVI-I Port (CN7)



2.5.4 COM1 RS-232/422/485 Port (CN8)

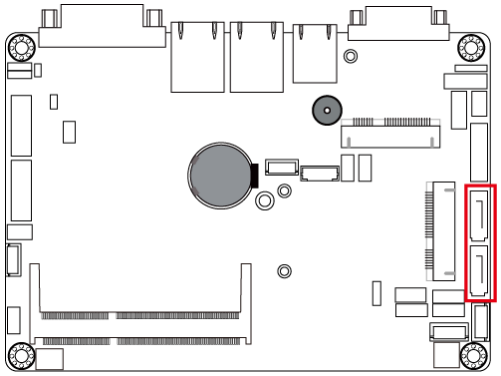


COM1 port is jumper-less and configurable in BIOS.

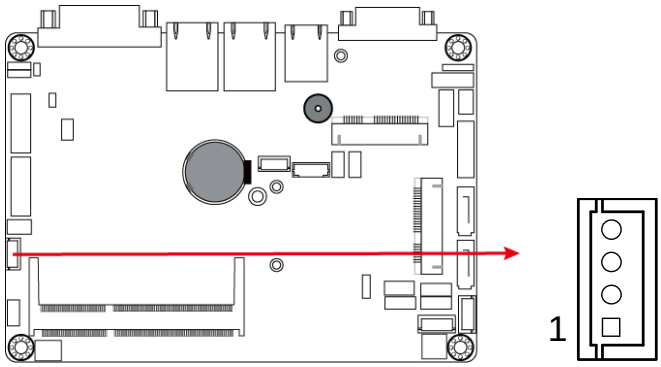
Pin	Assigment	Pin	Assigment
1	DCD, Data carrier detect	6	DSR, Data set ready
2	RXD, Receive data	7	RTS, Request to send
3	TXD, Transmit data	8	CTS, Clear to send
4	DTR, Data terminal ready	9	RI, Ring indicator
5	GND, ground		

Pin	Assignment		
	RS-232	RS-422	RS-485
1	DCD	TX-	DATA-
2	RX	TX+	DATA+
3	TX	RX+	NC
4	DTR	RX-	NC
5	Ground	Ground	Ground
6	DSR	NC	NC
7	RTS	NC	NC
8	CTS	NC	NC
9	RI	NC	NC

2.5.5 SATA II Port (CN1, CN3)



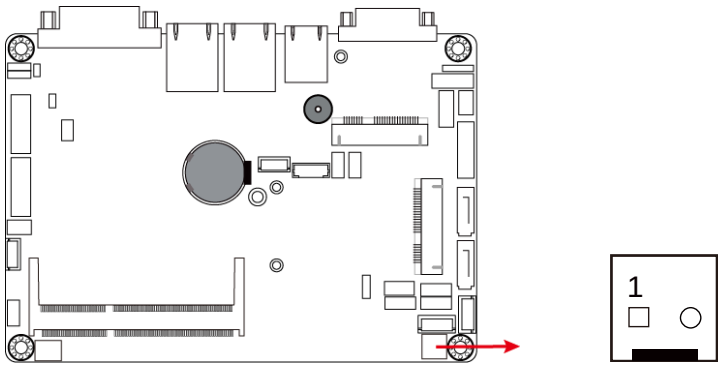
2.5.6 LCD Backlight Connector (CN2)



Brightness Control is only supported on panels with PWM Mode.

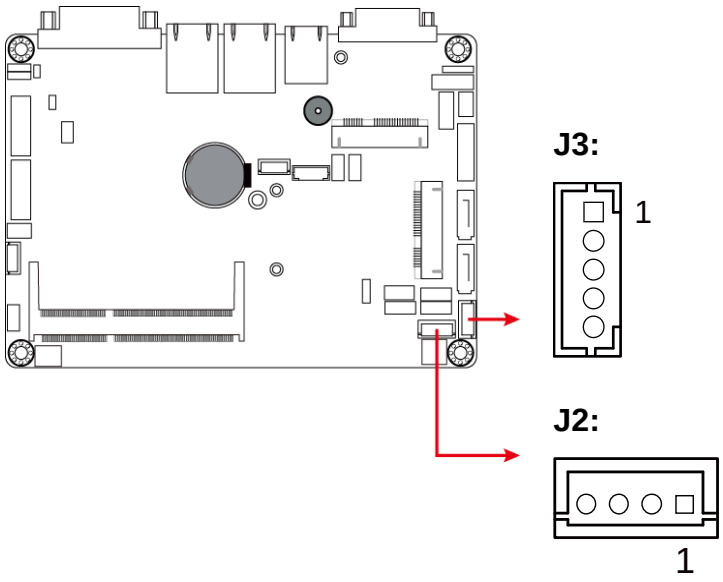
Pin	Assigment	Pin	Assigment
1	Backlight Power	3	Brightness Control
2	Backlight Enable	4	Ground

2.5.7 Motherboard Power Input Connector (J1)



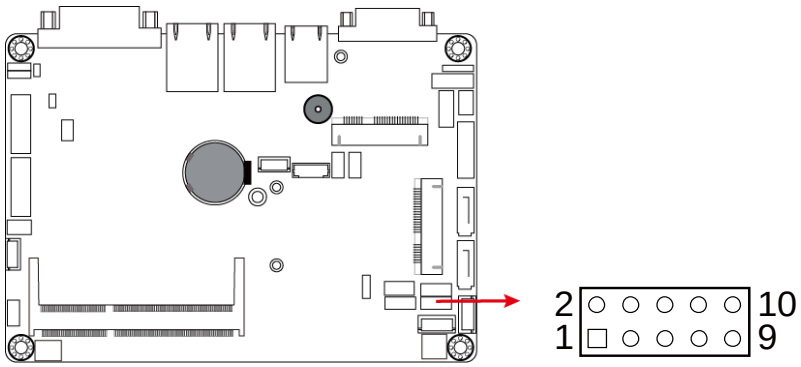
Pin	Assigment
1	12V ~ 24V
2	Ground

2.5.8 SATA HDD Power Connector (J2, J3)



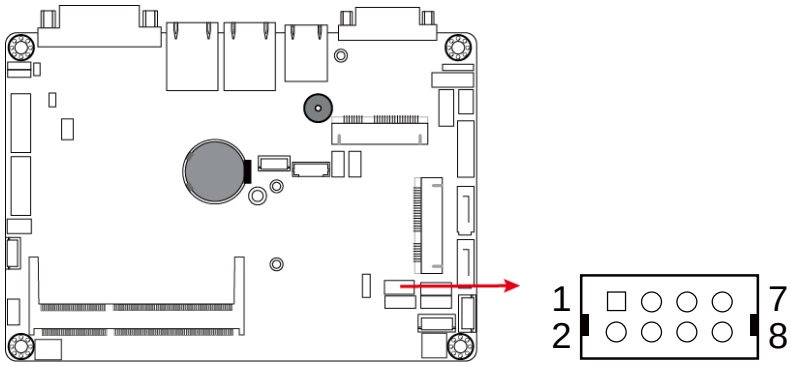
Pin	Assigment	Pin	Assigment
1	+5V	3	Ground
2	Ground	4	+12V

2.5.9 Digital I/O Connector (J5)



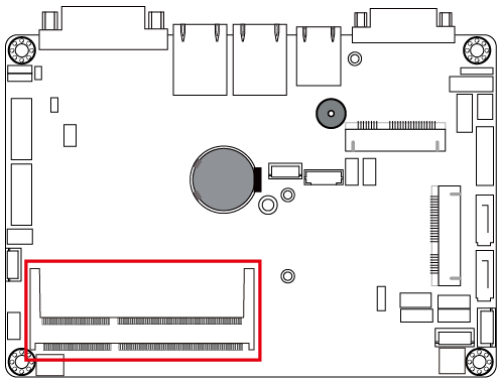
Pin	Assigment	Pin	Assigment
1	Ground	6	OUT0
2	VCC	7	IN3
3	OUT3	8	IN1
4	OUT1	9	IN2
5	OUT2	10	IN0

2.5.10 Keyboard & Mouse Connector (J7)

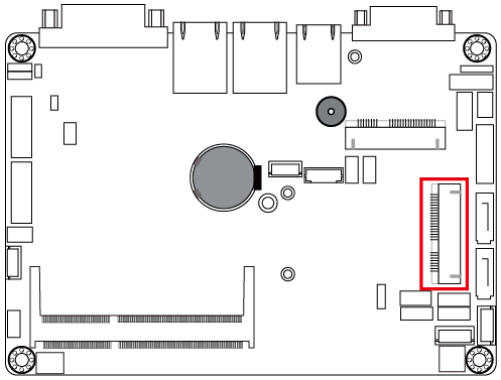


Pin	Assigment	Pin	Assigment
1	VCC	5	MCL
2	VCC	6	KBCL
3	MDA	7	Ground
4	KBDA	8	Ground

2.5.11 DDR3L SO-DIMM Socket (J8)

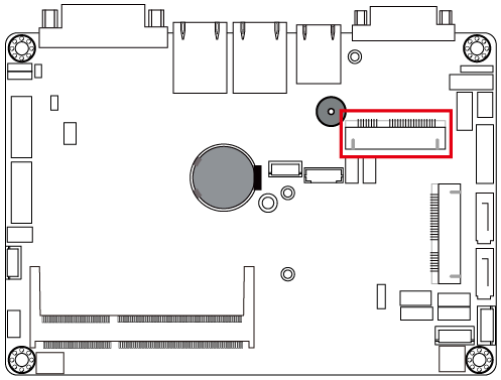


2.5.12 Full-Size Mini-PCIe / mSATA Connector (J9)

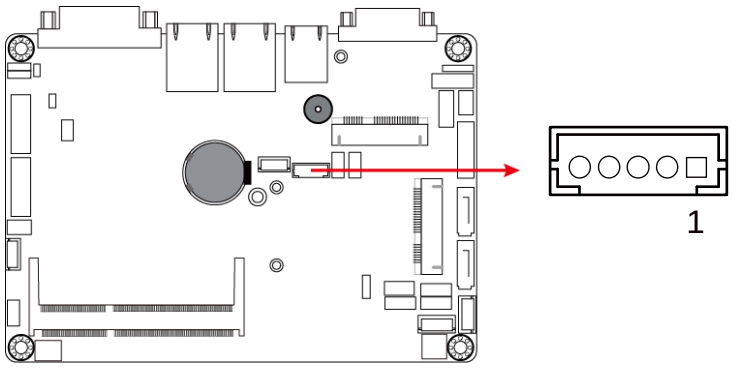


J9 is a full-size mini PCIe connector, also supporting mSATA, but cannot be used with CN3 (SATA II port) at the time. Use either J9 or CN3 at a time.

2.5.13 Half-Size Mini-PCIe Connector (J17)

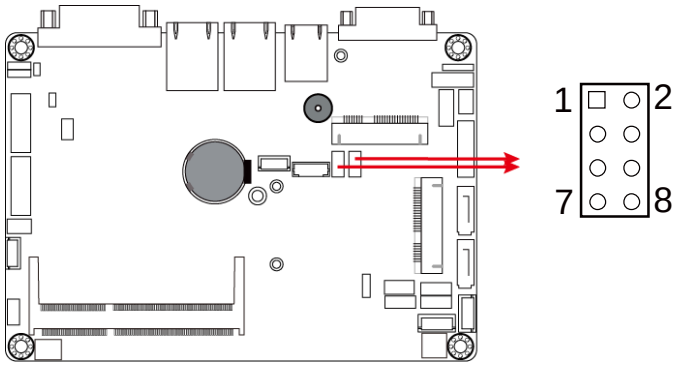


2.5.14 Smart Battery Interface Connector (J11)



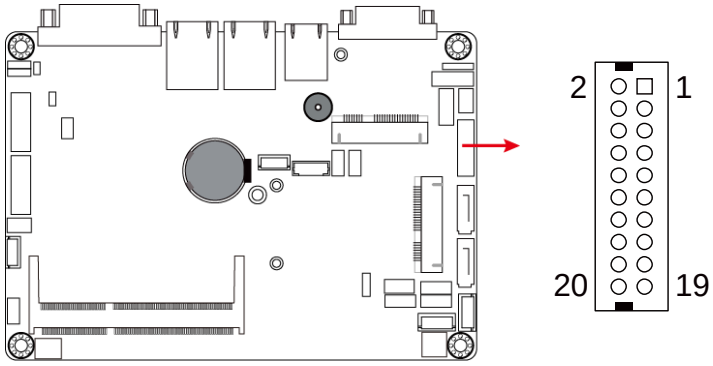
Pin	Assigment	Pin	Assigment
1	RST	4	DATA
2	EXTSMI	5	CLK
3	Ground		

2.5.15 USB 2.0 Connector (J12, J13)



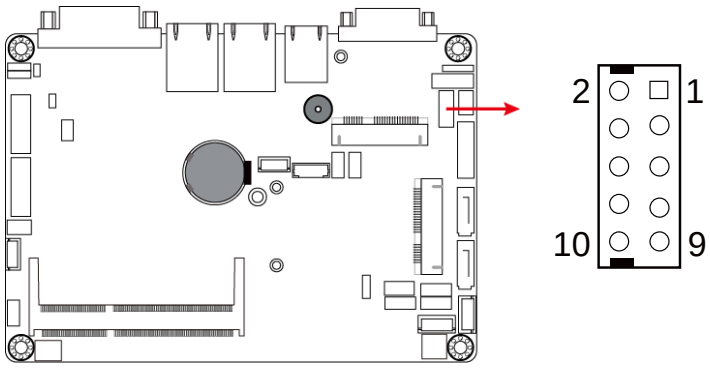
Pin	Assigment	Pin	Assigment
1	VCC	5	D+
2	Ground	6	D-
3	D-	7	Ground
4	D+	8	VCC

2.5.16 COM3 & COM4 Port (J15)



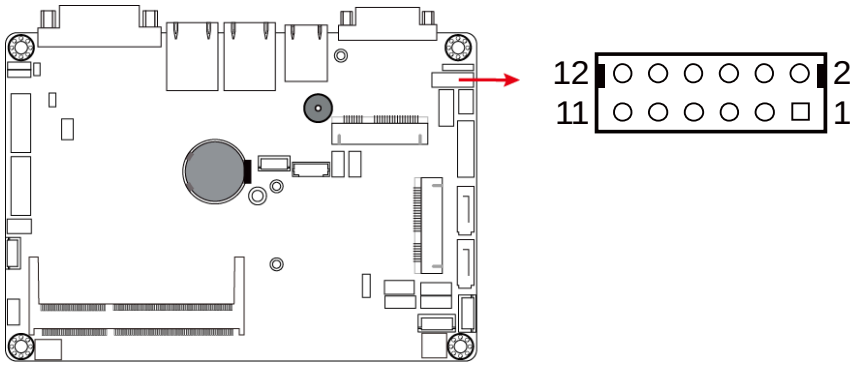
Pin	Assigment	Pin	Assigment
1	DCD3	11	DCD4
2	DSR3	12	DSR4
3	RXD3	13	RXD4
4	RTS3	14	RTS4
5	TXD3	15	TXD4
6	CTS3	16	CTS4
7	DTR3	17	DTR4
8	RI3	18	RI4
9	Ground	19	Ground
10	NC	20	NC

2.5.17 COM2 RS-232 Port (J16)



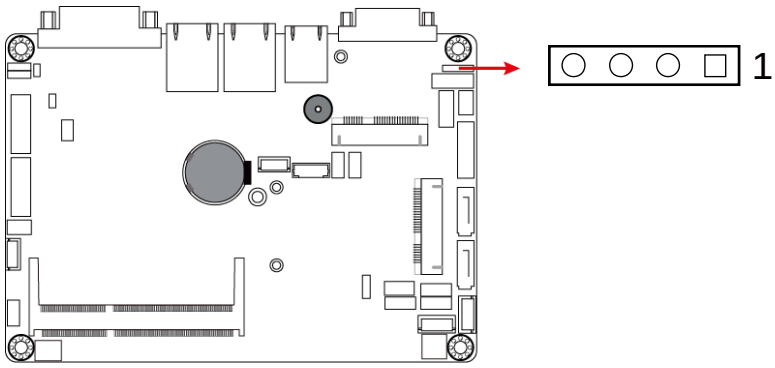
Pin	Assigment	Pin	Assigment
1	DCD, Data carrier detect	6	DSR, Data set ready
2	RXD, Receive data	7	RTS, Request to send
3	TXD, Transmit data	8	CTS, Clear to send
4	DTR, Data terminal ready	9	RI, Ring indicator
5	Ground	10	Not Used

2.5.18 Audio Connector (J18)



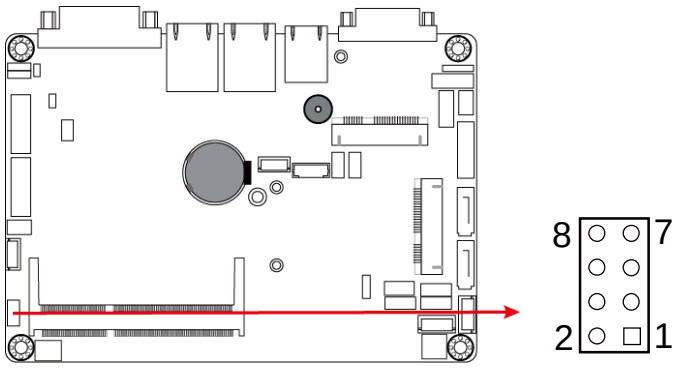
Pin	Assigment	Pin	Assigment
1	LINEOUT_L	7	JD_LINEIN
2	LINEOUT_R	8	Ground
3	JD_FRONT	9	MIC_L
4	Ground	10	MIC-R
5	LINEIN_L	11	JD_MIC1
6	LINEIN_R	12	Ground

2.5.19 Amplifier Connector (J22)



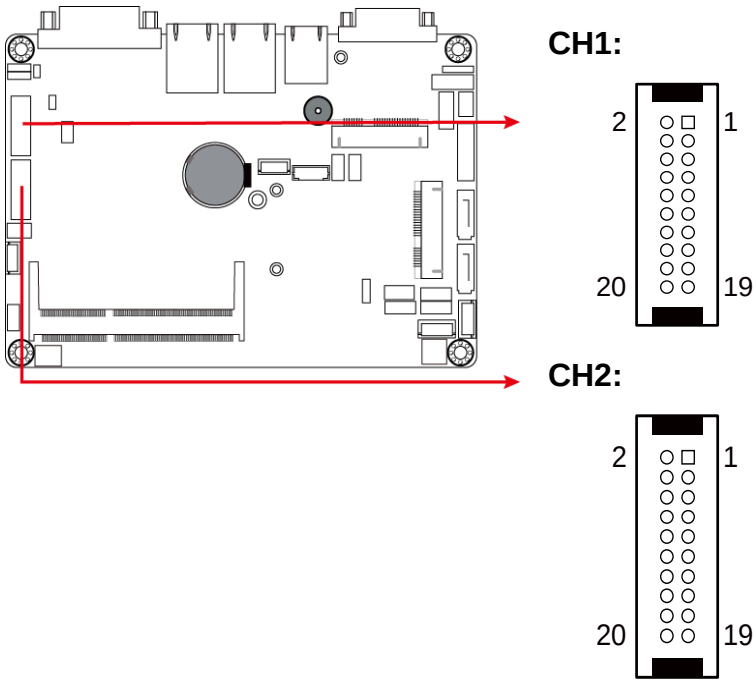
Pin	Assigment	Pin	Assigment
1	OUTL+	3	OUTR-
2	OUTL-	4	OUTR+

2.5.20 Front Panel Setting Connector for LED Indicators (JP1)



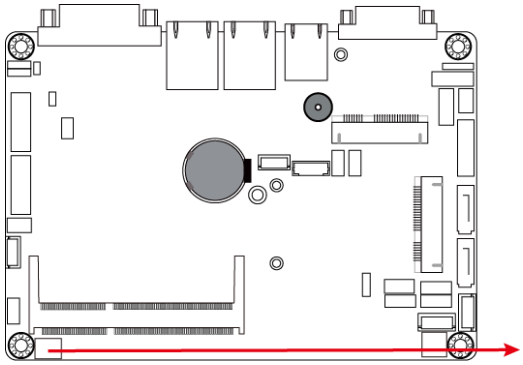
Pin	Assigment	Pin	Assigment
1	Ground	5	HDD_LED+
2	PWR_SW	6	HDD_LED-
3	PWR_LED+	7	Ground
4	PWR_LED- (Ground)	8	Reset-

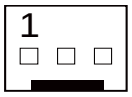
2.5.21 LVDS Connector (CH1, CH2)



Pin	Assigment	Pin	Assigment
1	LD0+	11	LD2+
2	LD0-	12	LD2-
3	GND	13	GND
4	GND	14	GND
5	LD1+	15	CLK+
6	LD1-	16	CLK-
7	GND	17	ENABLE
8	LCD_PWR	18	LCD_PWR
9	LD3+	19	NC
10	LD3-	20	NC

2.5.22 CPU Fan Power Connector (FAN1)





Pin	Assigment	Pin	Assigment
1	GND	3	Rotation detection
2	12V		

Chapter 3
Driver Installation

The information provided in this chapter includes:

- Intel® Chipset Software Installation Utility
- VGA Driver Installation
- HD Audio Driver Installation
- LAN Driver Installation

3.1 Introduction

This section describes the installation procedures for software drivers. The software drivers are in a disk enclosed with the product package. If you find anything missing, please contact the distributor where you made the purchase.

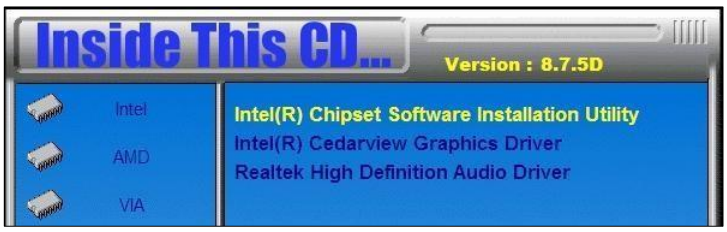
Note: After installing your Windows operating system, you must install the Intel® Chipset Software Installation Utility first before proceeding with the drivers installation.

3.2 Intel® Chipset Software Installation Utility

The Intel® Chipset drivers should be installed first before the software drivers to install INF files for Plug & Play function for the chipset components. Follow the instructions below to complete the installation.

1. Insert the disk enclosed in the package. Click **Intel** and then **Intel(R) Cedarview Chipset Drivers**.

2. Click **Intel(R) Chipset Software Installation Utility**.



3. When the *Welcome* screen to the Intel® Chipset Device Software appears, click **Next** to continue.

4. Click **Yes** to accept the software license agreement and proceed with the installation process.

5. On the *Readme File Information* screen, click **Next** for installation.

6. The driver has been completely installed. Click **Finish** to restart the computer and for changes to take effect.

3.3 VGA Driver Installation

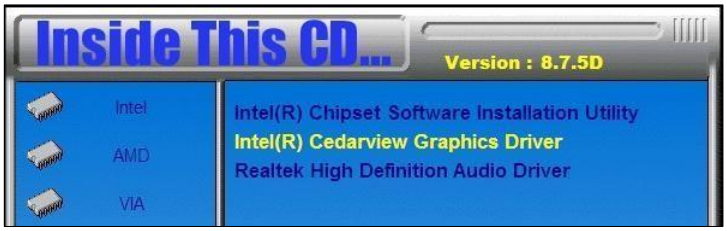
Note: LVDS is the primary default display. If you need to switch to any extended display of VGA / CRT / DVI-I, use the hot keys below.

For VGA / CRT: Ctrl + Alt + F1

For DVI-I: Ctrl + Alt + F3

1. Insert the disk enclosed in the package. Click **Intel** and then **Intel(R) Cedarview Chipset Drivers**.

2. Click **Intel(R) Cedarview Graphics Driver**.



3. When the *Welcome* screen appears, click **Next** to continue.

4. Click **Yes** to agree with the license agreement and continue the installation.

5. On the *Readme File Information* and *Setup Progress* screens, click **Next** for installation.

6. The driver has been completely installed. Click **Finish** to restart the computer and for changes to take effect.

3.4 HD Audio Driver Installation

1. Insert the disk enclosed in the package. Click **Intel** and then **Intel(R) Cedarview Chipset Drivers**.

2. Click **Realtek High Definition Audio Driver**.

3. On the *Welcome* screen of the InstallShield Wizard, click **Next** for installation.

4. The driver has been completely installed. Click **Finish** to restart the computer and for changes to take effect.

3.5 LAN Driver Installation

1. Insert the disk enclosed in the package with the product. Click **LAN Card** and then **Intel(R) LAN Controller Drivers.**

2. When the *Welcome* screen appears, click **Next** to continue.

3. Accept the license agreement and click **Next** to continue.

4. Tick the checkbox for **Drivers** to select the related drivers and click **Next**.

5. When the wizard is ready for installation, click **Install**.

6. The driver has been completely installed. Click **Finish** to restart the computer and for changes to take effect.

Chapter 4
BIOS Setup

This chapter describes the different settings available in the AMI BIOS that comes with the board. The topics covered in this chapter are as follows:

- Main Settings
- Advanced Settings
- Chipset Settings
- Security Settings
- Book Settings
- Save & Exit

4.1 Introduction

The BIOS (Basic Input/Output System) installed in the ROM of your computer system supports Intel® processors. The BIOS provides critical low-level support for standard devices such as disk drives, serial ports and parallel ports. It also provides password protection as well as special support for detailed fine-tuning of the chipset controlling the entire system.

4.2 BIOS Setup

The BIOS provides a Setup utility program for specifying the system configurations and settings. The BIOS ROM of the system stores the Setup utility. When you turn on the computer, the BIOS is immediately activated. Press the <Del> key immediately allows you to enter the Setup utility. If you are a little bit late pressing the <Del> key, POST (Power On Self Test) will continue with its test routines, thus preventing you from invoking the Setup.

If you still need to enter Setup, restart the system by pressing the ”Reset” button or simultaneously pressing the <Ctrl>, <Alt> and <Delete> keys. You can also restart by turning the system Off and back On again.

The following message will appear on the screen:

```
Press <DEL> to Enter Setup
```

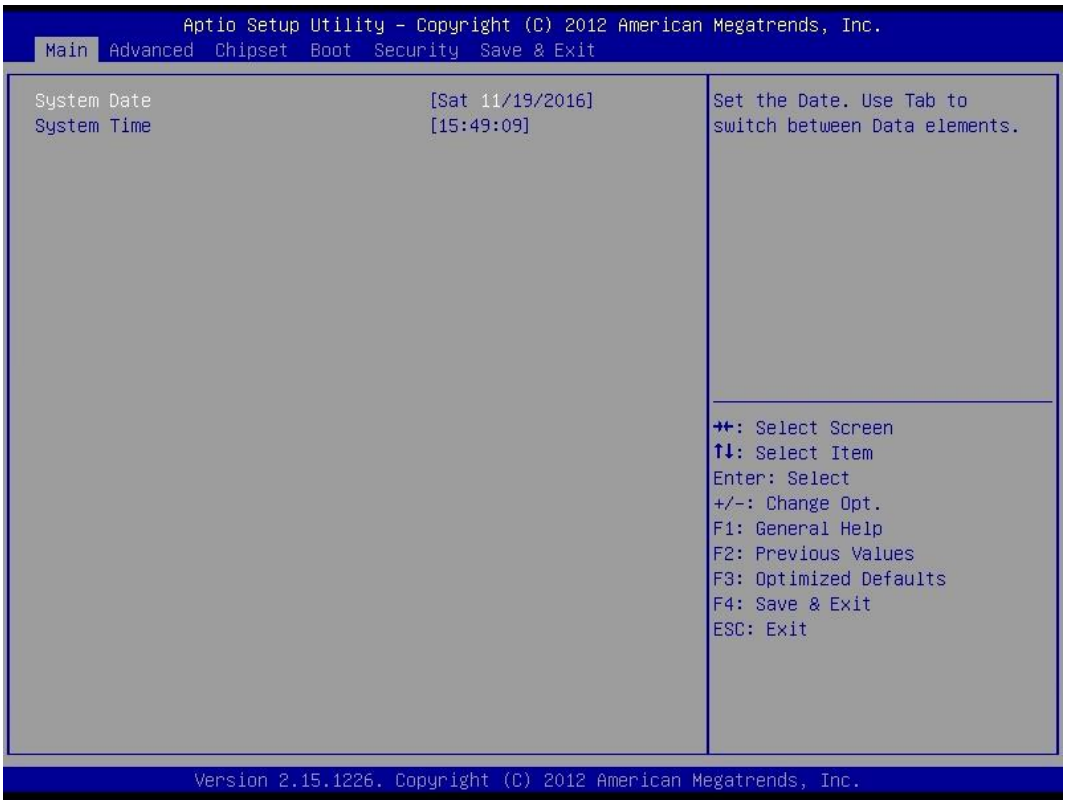
In general, press the arrow keys to highlight items, <Enter> to select, the <PgUp> and <PgDn> keys to change entries, <F1> for help, and <Esc> to quit.

When you enter the BIOS Setup utility, the *Main Menu* screen will appear on the screen. The Main Menu allows you to select from various setup functions and exit choices.

Warning: It is strongly recommended that you avoid making any changes to the chipset defaults.

These defaults have been carefully chosen by both AMI and your system manufacturer to provide the absolute maximum performance and reliability. Changing the defaults could make the system unstable and crash in some cases.

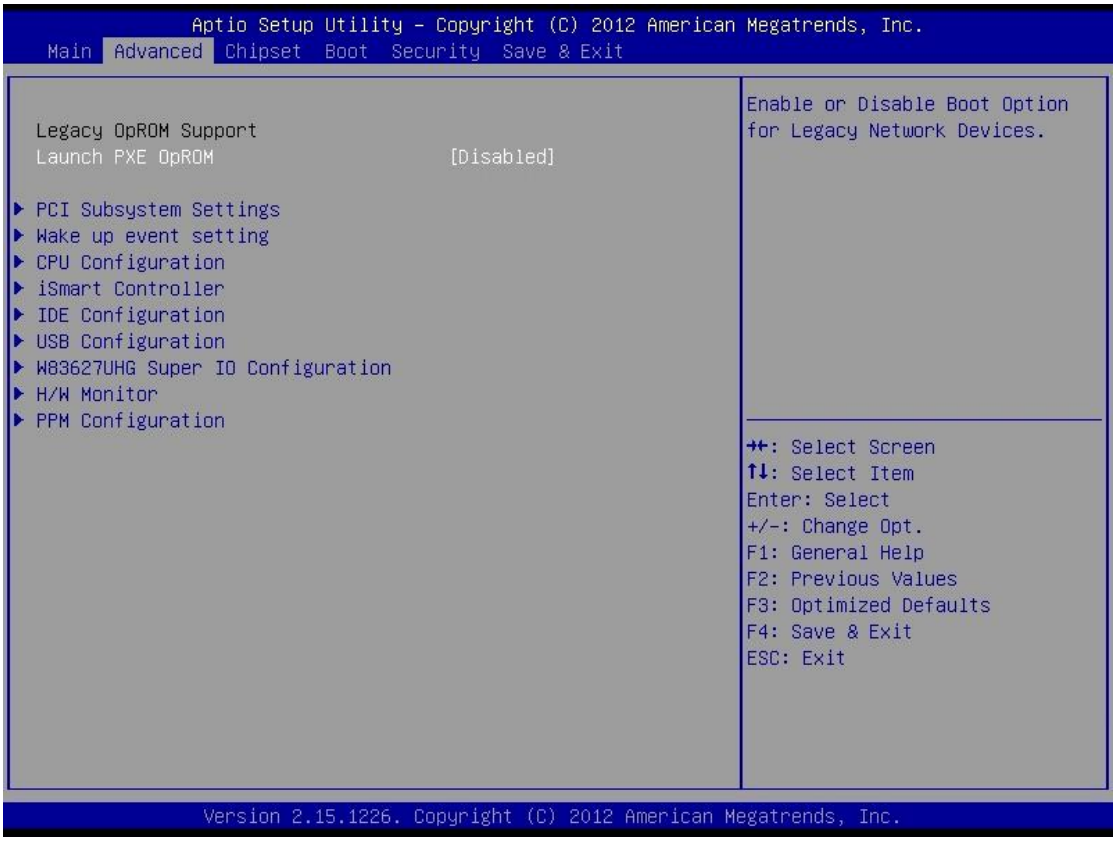
4.3 Main Settings



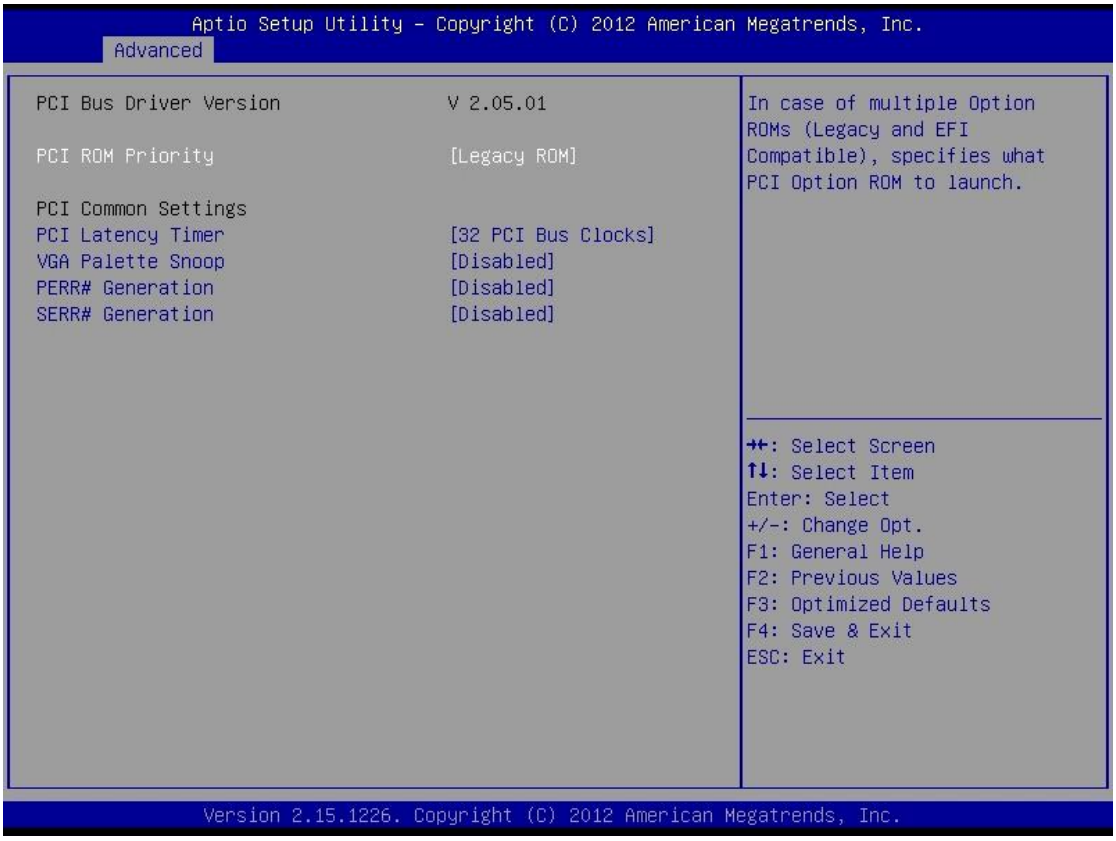
BIOS Setting	Description
System Date	Sets the date. Use the <Tab> key to switch between the data elements.
System Time	Set the time. Use the <Tab> key to switch between the data elements.

4.4 Advanced Settings

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

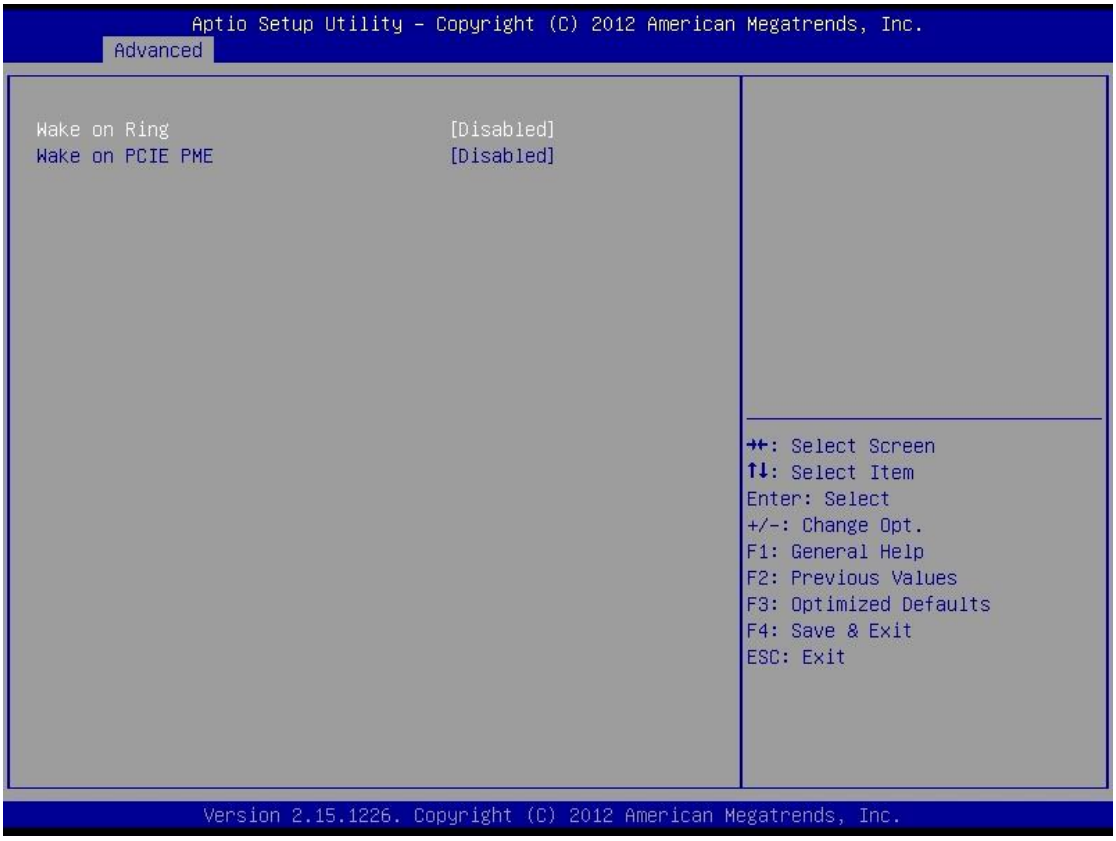


4.4.1 PCI Subsystem Settings



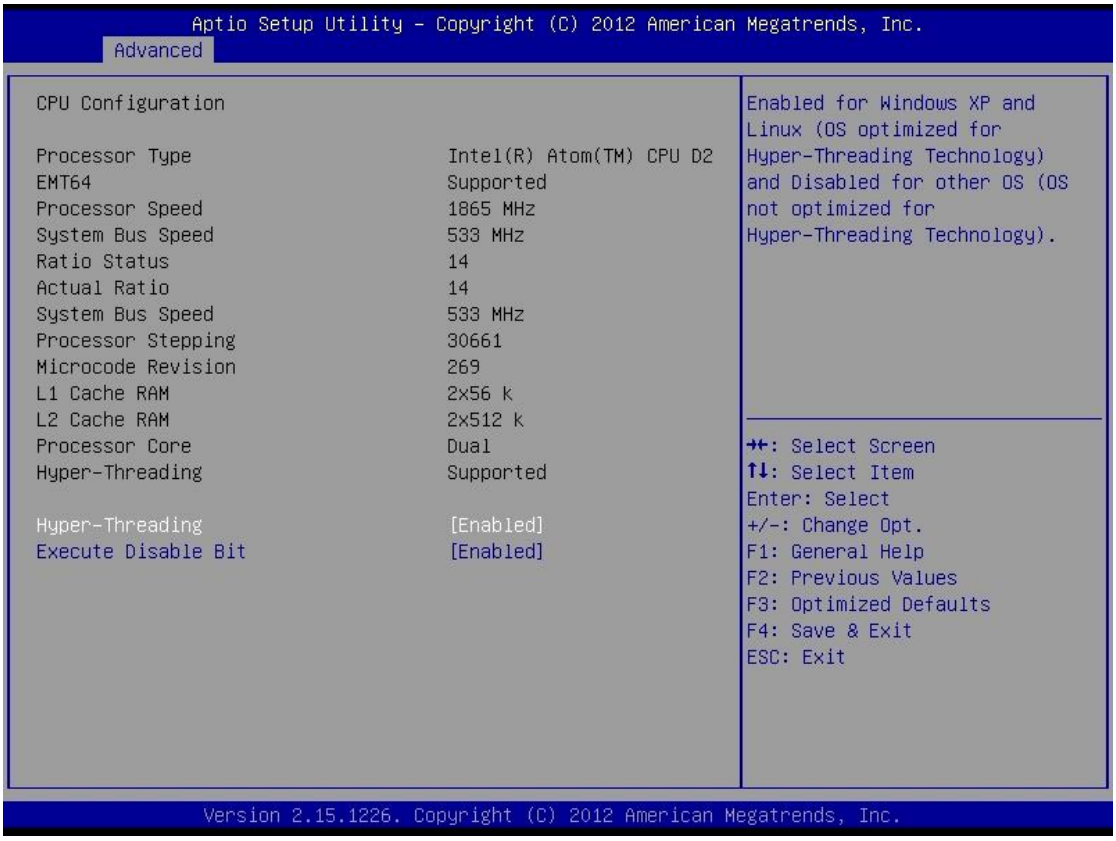
BIOS Setting	Description
PCI ROM Priority	Specifies a PCI Option ROM to launch in case of multiple Option ROMs (Legacy and EFI compatible).
PCI Latency Timer	Sets a value to program into PCI Latency Timer Register. Options: 32, 64, 96, 128, 160, 192, 224, 248 PCI Bus Clocks
VGA Palette Snoop	Enables / Disables VGA Palette Registers Snooping.
PERR# Generation	Enables / Disables PCI devices to generate PERR# error codes.
SERR# Generation	Enables / Disables PCI devices to generate SERR# error codes.

4.4.2 Wake up Event Setting



BIOS Setting	Description
Wake on Ring	Enables / Disables the device to wake up and resume from a suspend state on alarm events.
Wake on PCIe PME	Enables / Disables the device to wake up and resume from a suspend state.

4.4.3 CPU Configuration

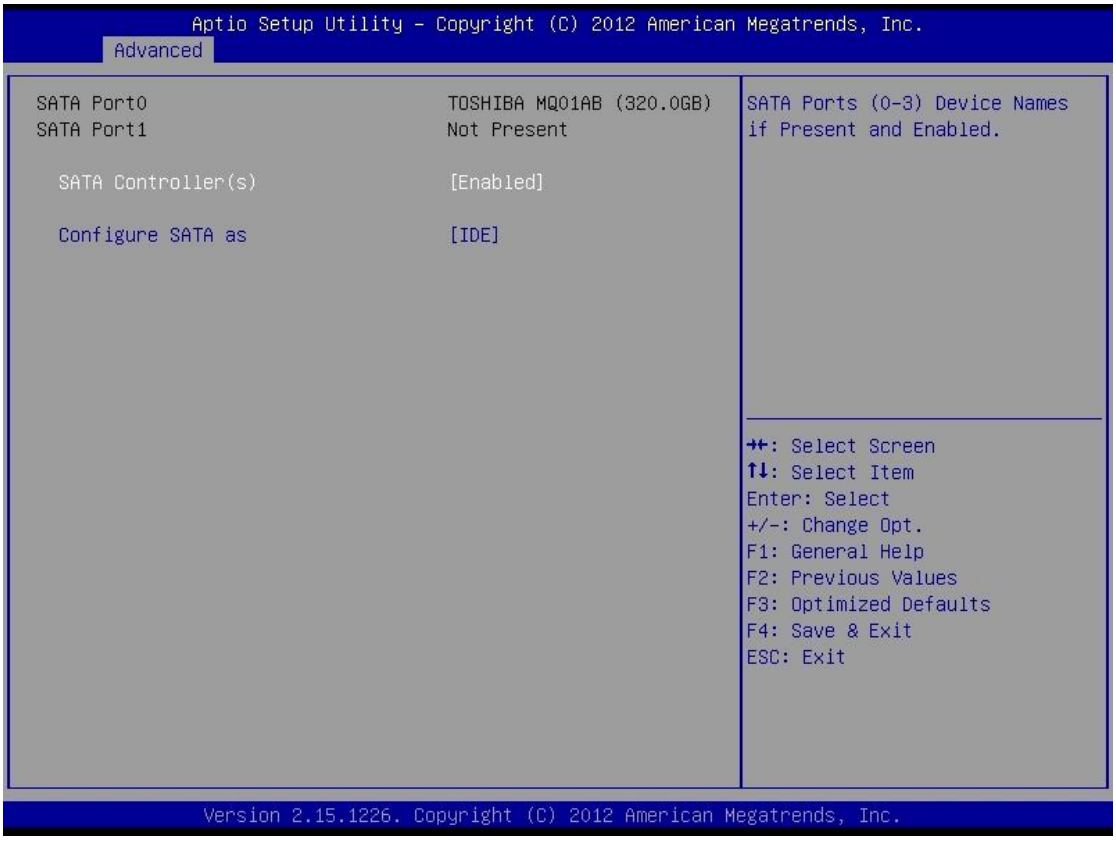


BIOS Setting	Description
Hyper-threading	Enables for Windows XP and Linux (OS optimized for Hyper-Threading Technology). Disables for other OS (OS not optimized for Hyper-Threading Technology). When disabled, only one thread per enabled core is enabled.
Execute Disable Bit	Prevents certain classes of malicious buffer oerflow attacks when combined with a supporting OS (Windows Server 2003 SP1, Windows XP SP2, SuSE Linux 9.2, RedHat Enterprise 3 Update 3.)

4.4.4 iSmart Controller

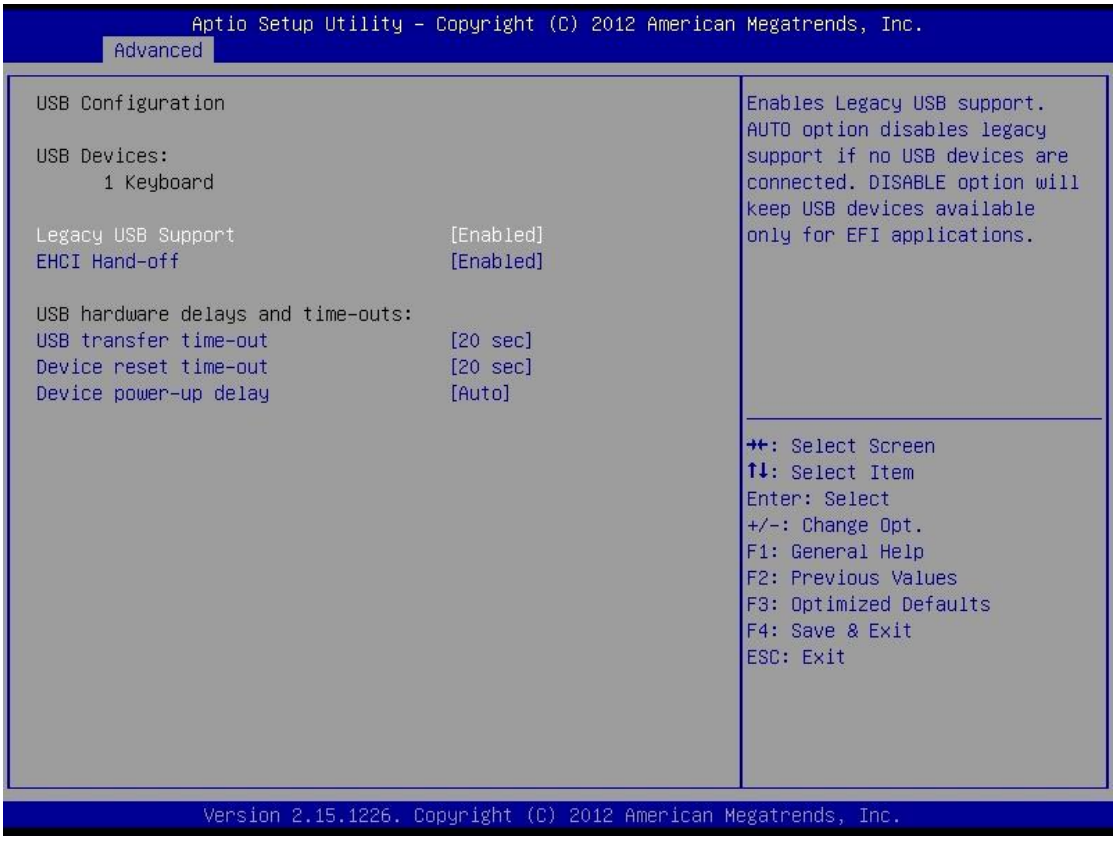
BIOS Setting	Description
Power-On after Power failure	Enables / Disables the system to be turned on automatically after a power failure.
Schedule Slot 1 / 2	Sets up the hour / minute / day for the power-on schedule for the system. Options: • None • Power On • Power On / Off

4.4.5 IDE Configuration



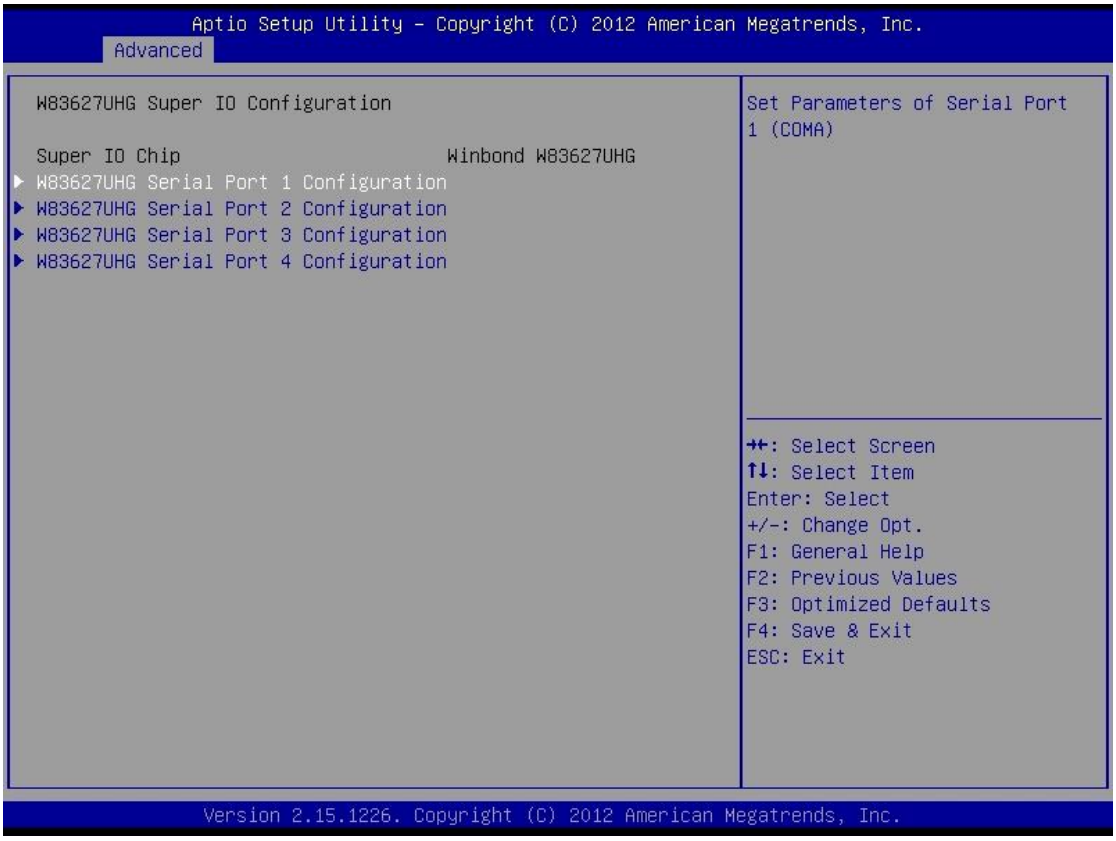
BIOS Setting	Description
SATA Controller(s)	Enables / Disables the Serial ATA.
Configure SATA as	Selects IDE or AHCI Mode.
SATA Port 0 HotPlug	Enables / Disables SATA Port 0 HotPlug.
SATA Port 1 HotPlug	Enables / Disables SATA Port 1 HotPlug.

4.4.6 USB Configuration



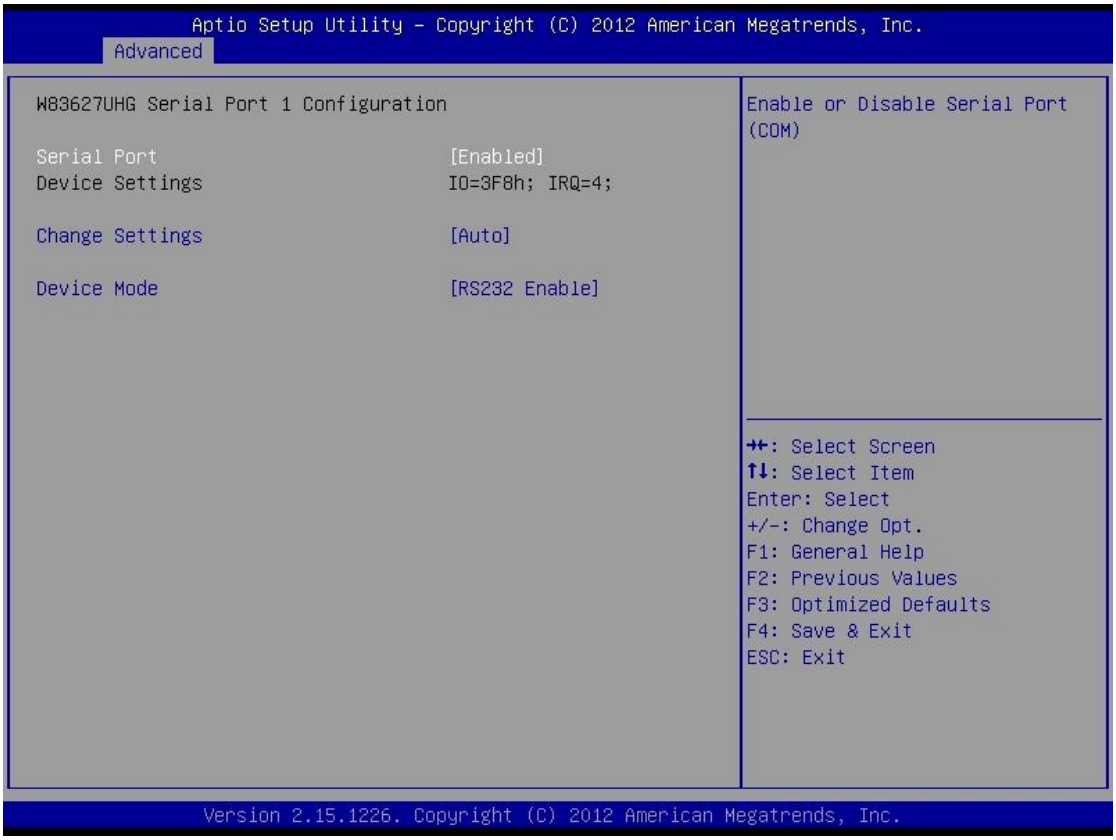
BIOS Setting	Description
Legacy USB Support	Enables / Disables Legacy USB support. • "Auto" disables legacy support if there is no USB device connected. • "Disable" keeps USB devices available only for EFI applications.
EHCI Hand-pff	This is a workaround for OSes without EHCI hand-off support. The EHCI ownership change should be claimed by EHCI driver.
USB Transfer time-out	Sets the time-out value 1, 5, 10 or 20 sec(s) for Control, Bulk, and Interrupt transfers.
Device reset time-out	Sets the seconds (10, 20, 30, 40 secs) of delaying execution of start unit command to USB mass storage device.
Device power-up delay	The maximum time the device will take before it properly reports itself to the Host Controller. "Auto" uses default value. For a Root port, it is 100 ms. For a Hub port, the delay is taken from Hub descriptor.

4.4.7 Super IO Configuration



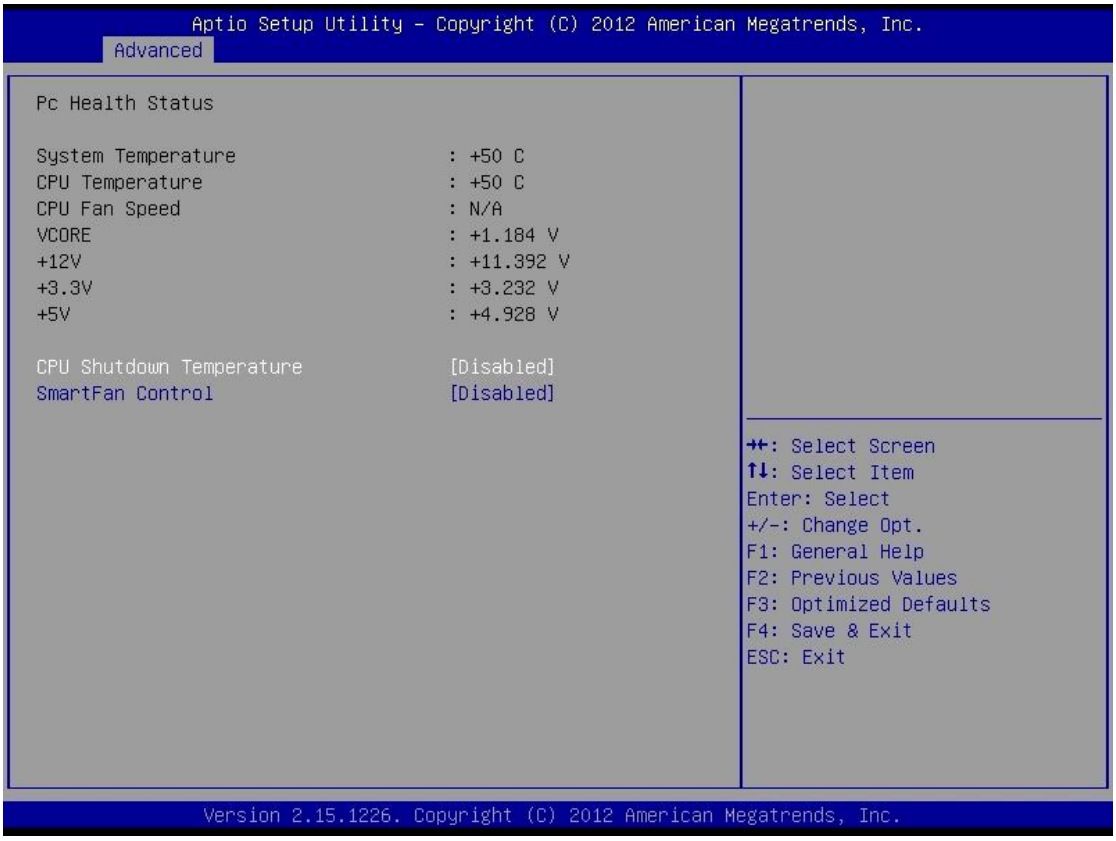
BIOS Setting	Description
Serial Port Configuration	Sets Parameters of Serial Ports. You can enable / disable the serial port and select an optimal settings for the Super IO device.

4.4.7.1. Serial Port 1 Configuration



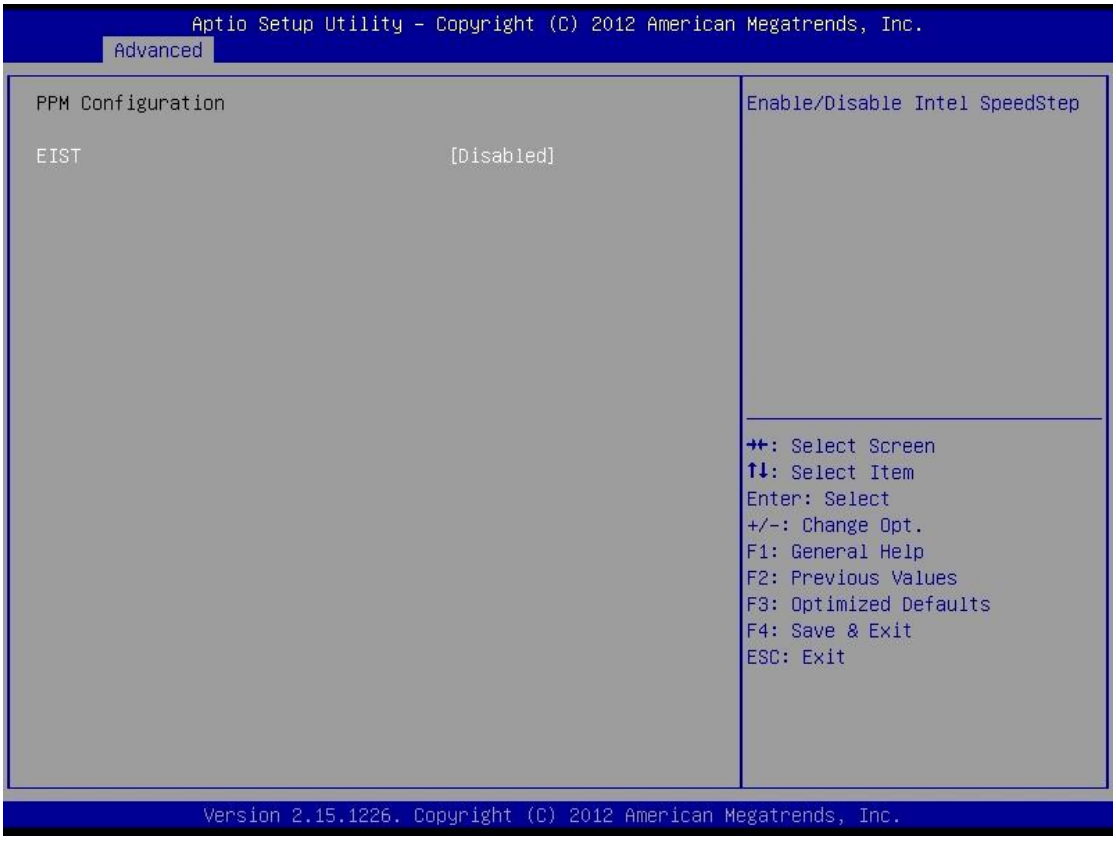
BIOS Setting	Description
Change Settings	Selects an optimal settings for the Super I/O device. Options: • Auto • IO=3F8h ; IRQ=4 • IO=3F8h ; IRQ=3, 4, 5, 6, 7, 9. 10, 11, 12 • IO=2F8h ; IRQ=3, 4, 5, 6, 7, 9. 10, 11, 12 • IO=3E8h ; IRQ=3, 4, 5, 6, 7, 9. 10, 11, 12 • IO=2E8h ; IRQ=3, 4, 5, 6, 7, 9. 10, 11, 12
Device Mode	Changes the mode of serial port. Options: • RS232 Enable • RS485 Enable • RS422 Enable

4.4.8 Hardware Monitor



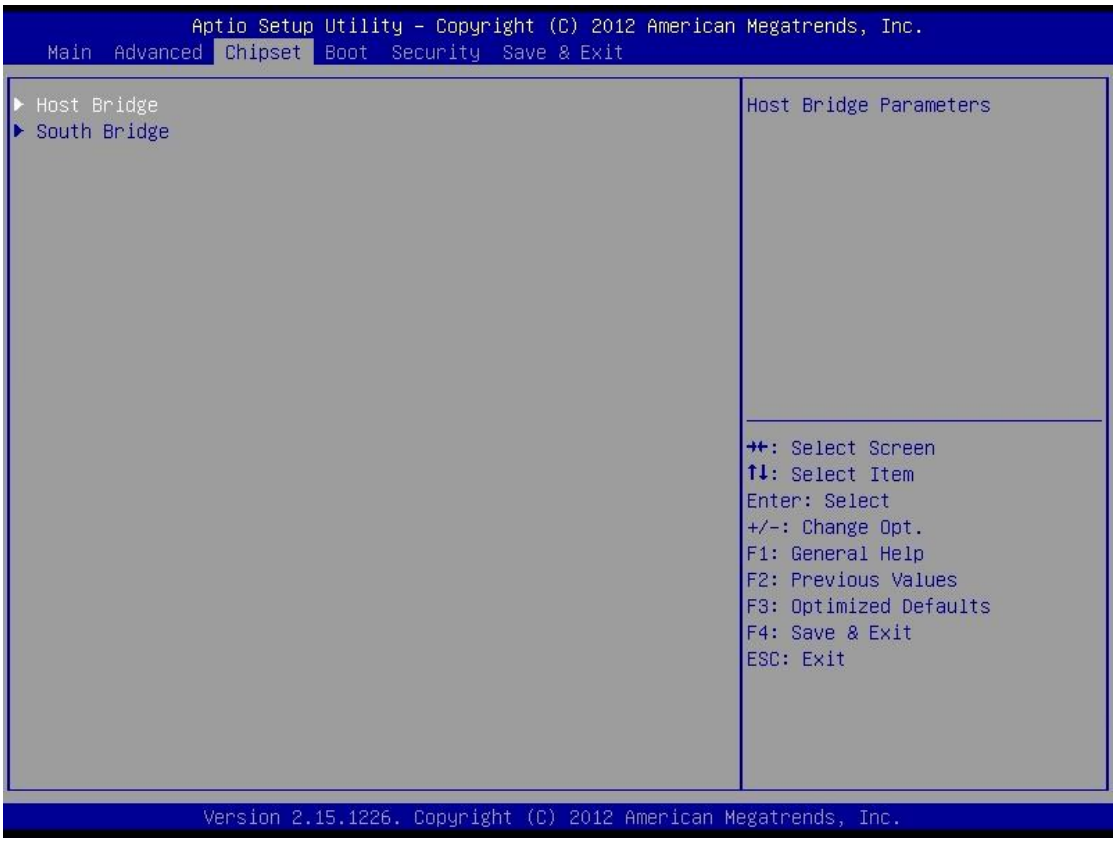
BIOS Setting	Description
Shutdown Temperature	This field enables or disables the Shutdown Temperature Options: Disabled (default),. 70 ℃, 75 ℃, 80 ℃, 85 ℃, 90 ℃, 95 ℃
Temperatures / Voltages	These fields are the parameters of the hardware monitoring function feature of the motherboard. The values are read-only values as monitored by the system and show the PC health status
Smart Fan Function	This field enables or disables the smart fan feature. Options: Disabled (default), 60 ℃, 70 ℃, 80 ℃, 90 ℃

4.4.9 CPU PPM Configuration

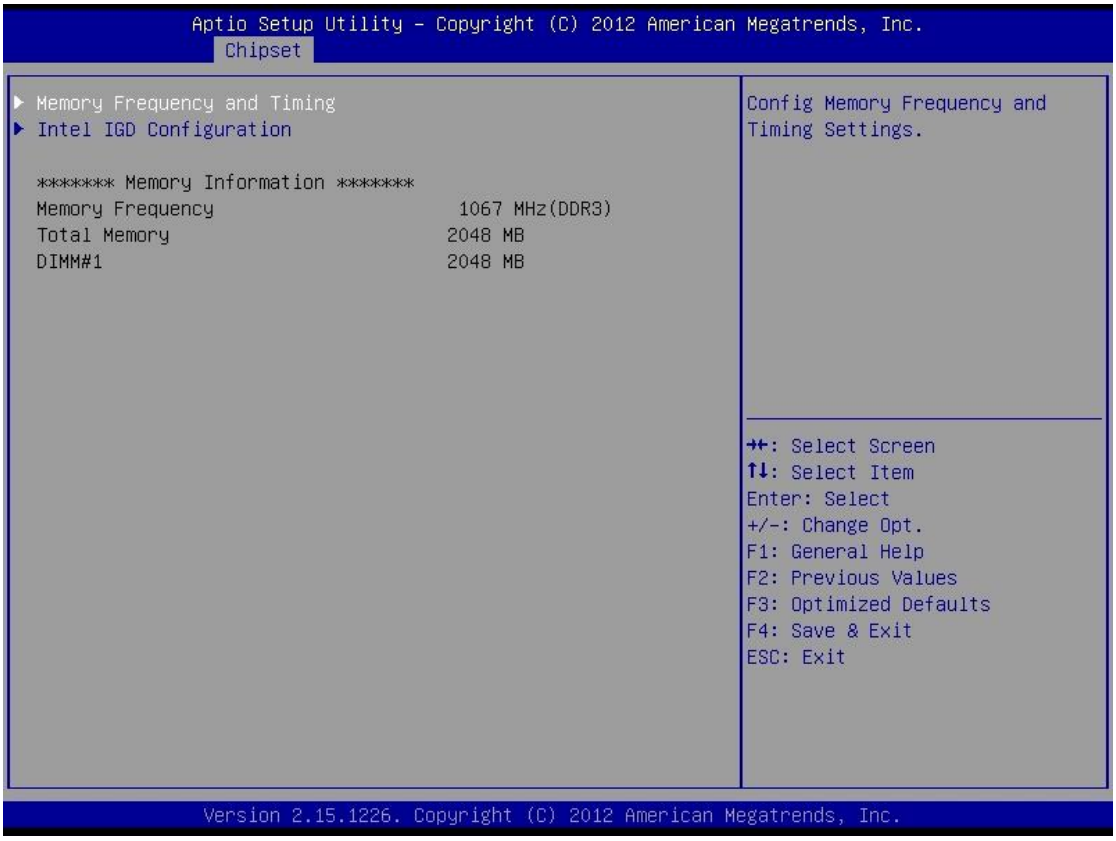


BIOS Setting	Description
EIST	Enables / Disables Intel SpeedStep.

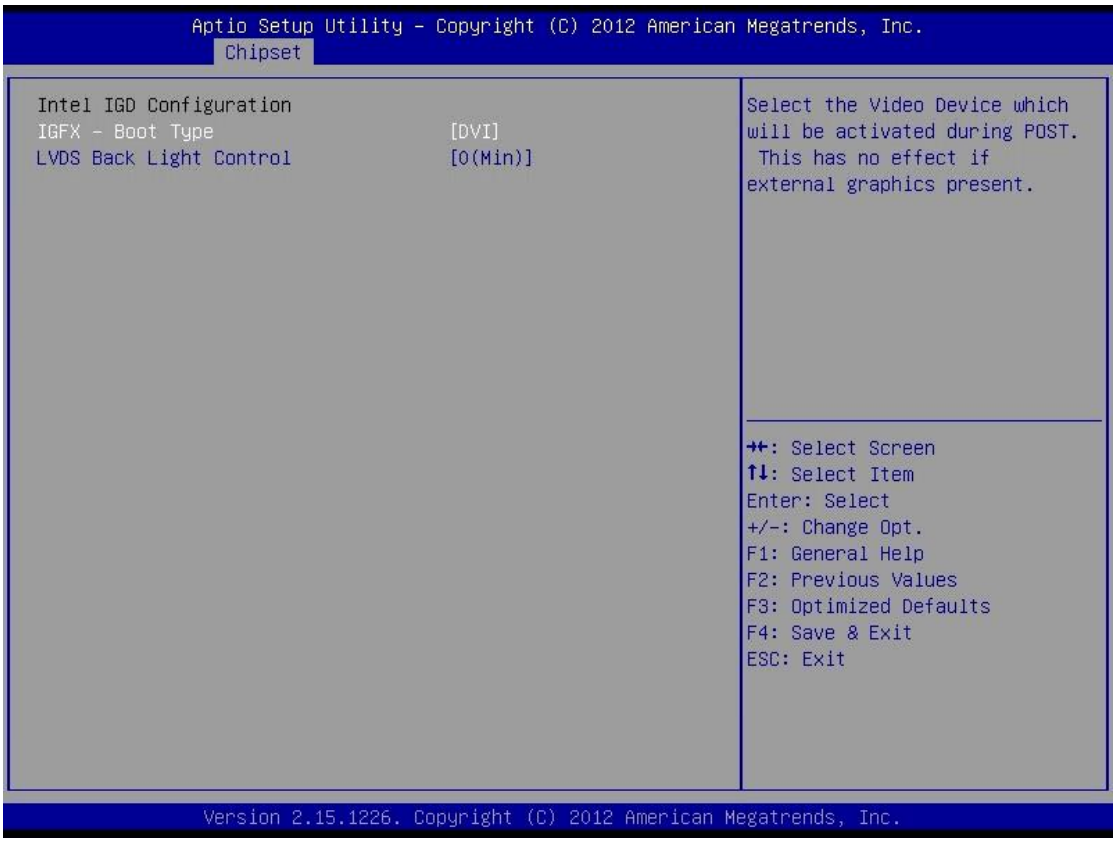
4.5 Chipset Settings



4.5.1 Host Bridge

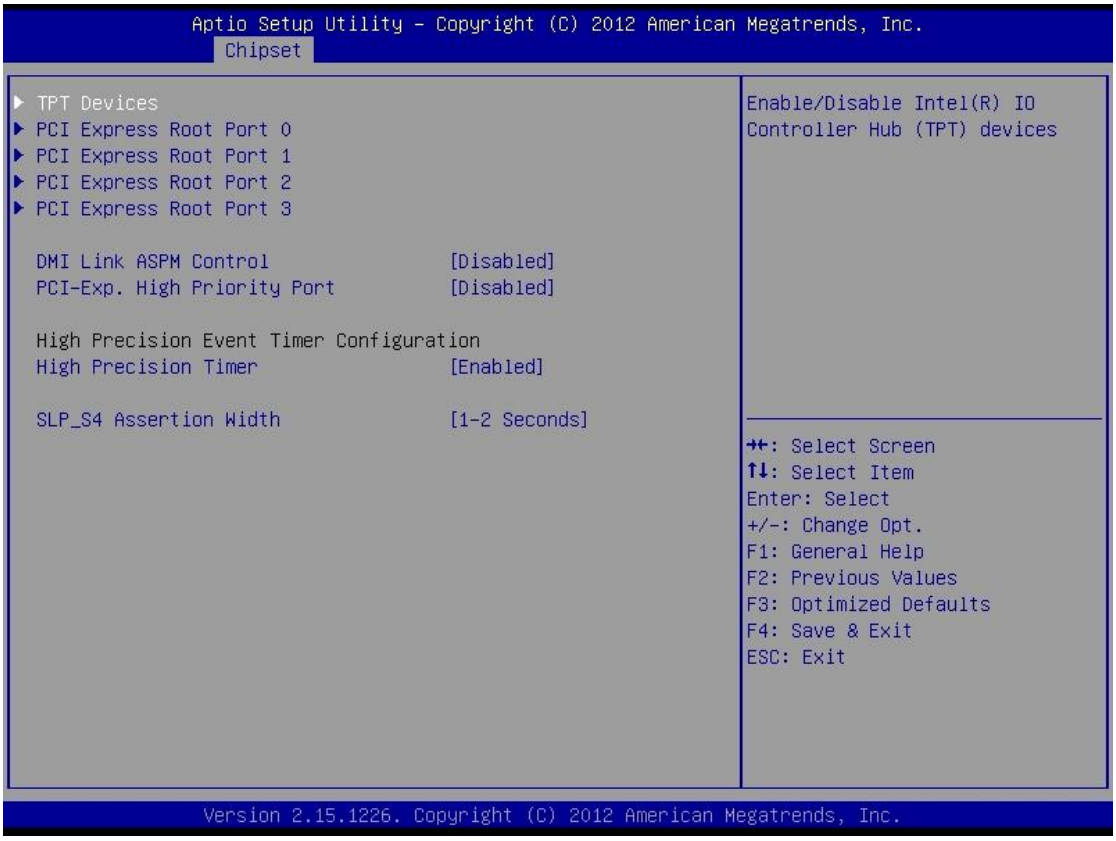


BIOS Setting	Description
Memory Frequency and Timing	Configs Memory Frequency and timing Settings.
Intel IGD Configuration	Configs Intel IGD Settings.

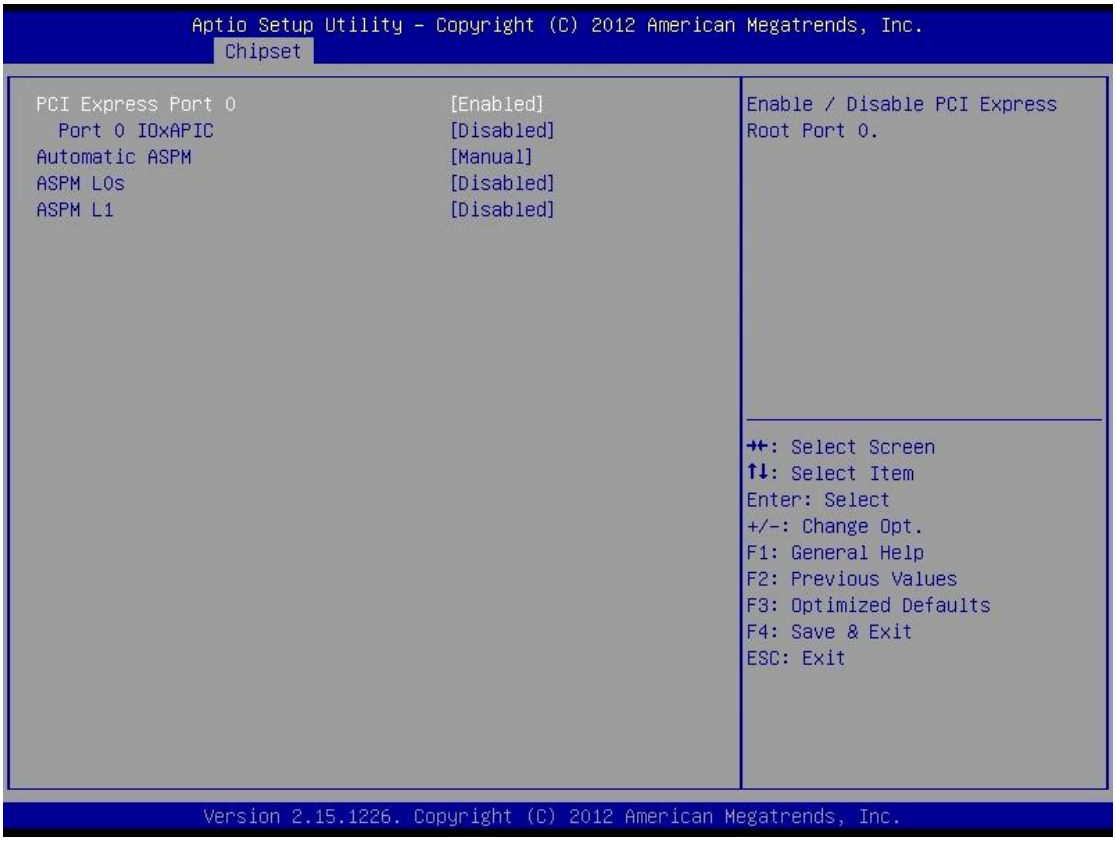


BIOS Setting	Description
IGFX – Boot Type	Selects the Video Device which will be activated during POST. This has no effect if external graphics present. Options: DVI, CRT+DVI
LVDS Back Light Control	Adjusts the LVDS backlight Opions: 7 (Max), 6, 5, 4, 3, 2, 1, 0 (Min)

4.5.2 South Bridge

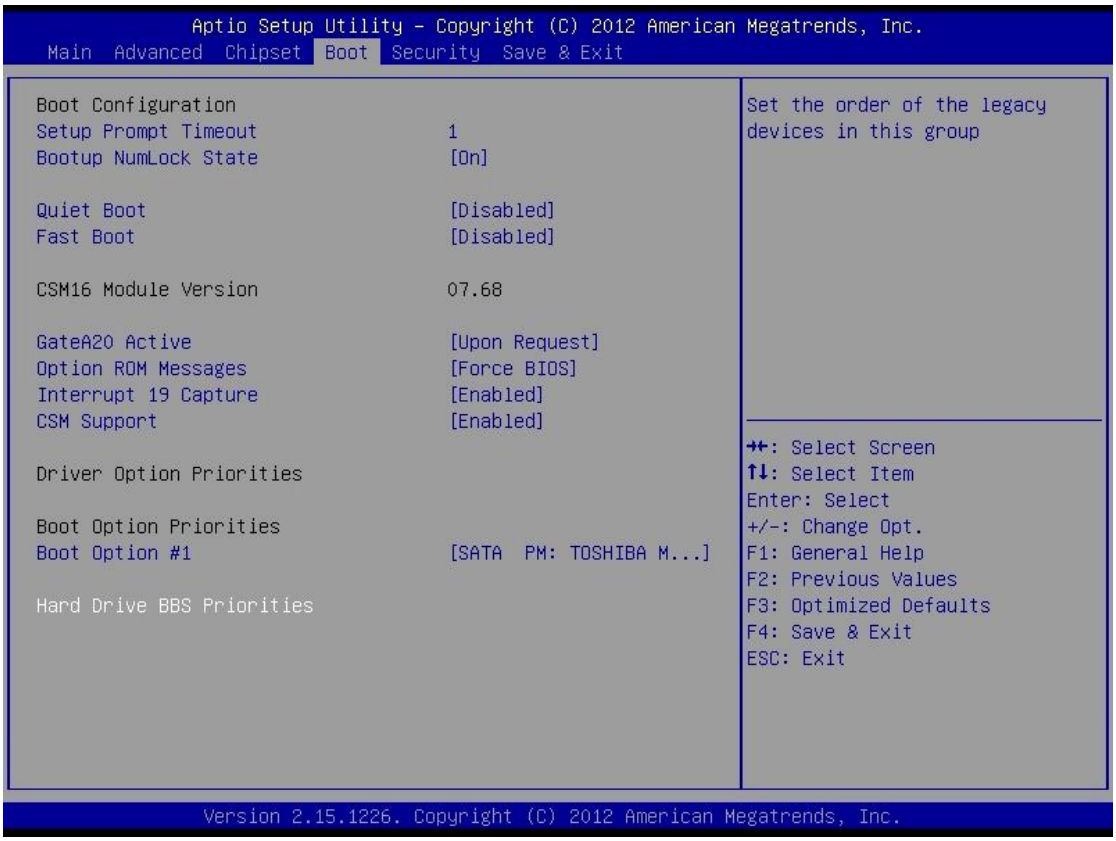


BIOS Setting	Description
TPT Devices	Enables / Disables Intel(R) IO Controller Hub (TPT) devices.
PCI Express Root Ports	Enables / Disables PCI Express Root Ports.
DMI Link ASPM Control	Enables / Disables the control of Active State Power Management on both NB side and SB side of the DMI Link.
PCI-Exp. High Proority Port	Selects a PCI Express High Priority Port. Options: Disabled, Port 0, Port1, Port 2, Port 3
High Precision Timer	Enables / Disables the High Precision Timer
SLP_S4 Assertion Width	Selects a minimum assertion width of the SLP_S4# signal. Options: 1~2 / 2~3 / 3~4 / 4~5 seconds



BIOS Setting	Description
PCI Express Port 0	Enables / Disables Intel(R) IO Controller Hub (TPT) devices.
Automatic ASPM	Automatically enable ASPM based on reported capabilities and known issues. Options: Manual, Auto
ASPM L0s	Enables PCIe ASPM L0s. Options: • Disabled • Root Port Only • Endpoint Port Only • Both Root and Endpoint Ports
ASPM L1	Enables PCIe ASPM L1.

4.6 Boot Settings

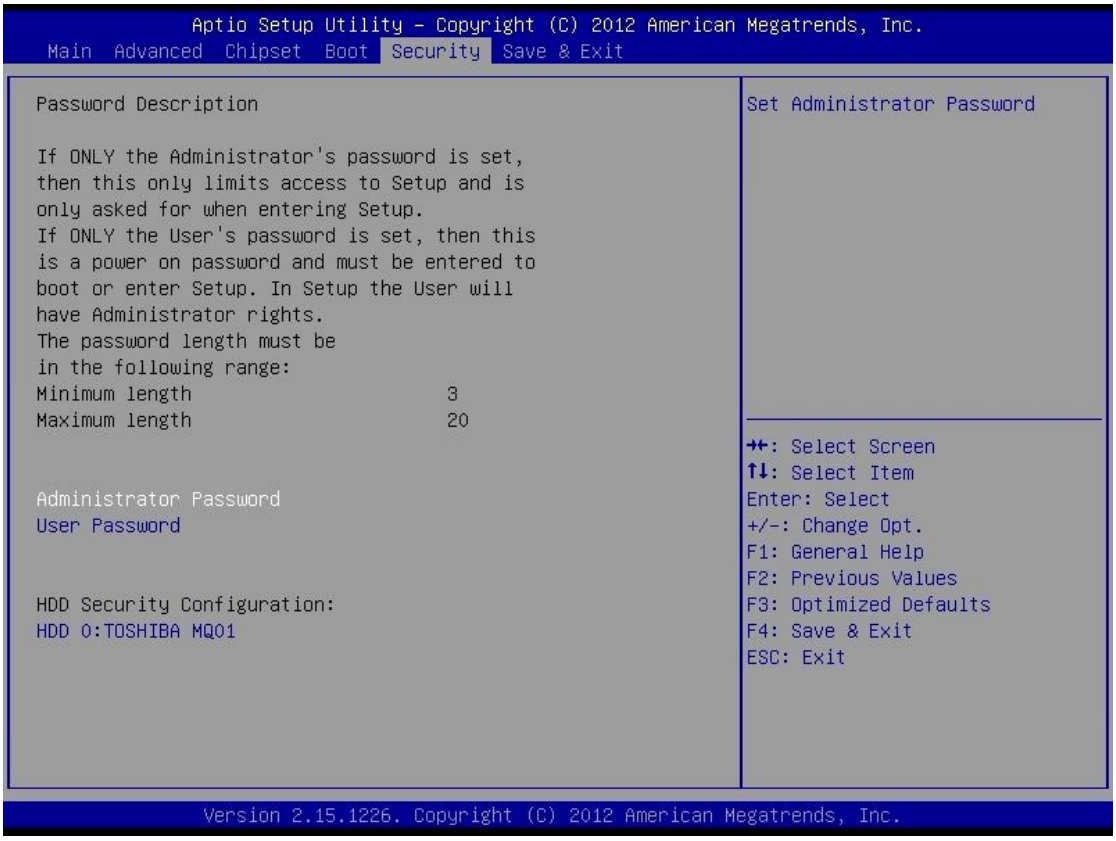


BIOS Setting	Description
Setup Prompt Timeout	Number of seconds to wait for setup activation key. 65535 (0xFFFF) means indefinite waiting.
Bootup NumLock State	Selects the keyboard NumLock state.
Quiet Boot	Enables / Disables Quiet Boot option.
Fast Boot	Enables / Disables boot with initialization of a minimal set of devices required to launch the active boot option. Has no effect for BBS boot options.
GateA20 Active	The option Upon Request disables GA20 when using BIOS services. The option Always cannot disable GA20, but is useful when any RT code is executed above 1 MB.
Option ROM Messages	Sets a display mode, Force BIOS or Keep Current, for Option ROM.
Interrupt 19 Capture	Allows Option ROMs to trap Interrupt 19.

BIOS Setting	Description
CSM Support	Enables / Disables CSM support. “Auto” makes CSM support enabled or disabled automatically according to your OS.
Boot Option Priorities	Sets the system boot order.

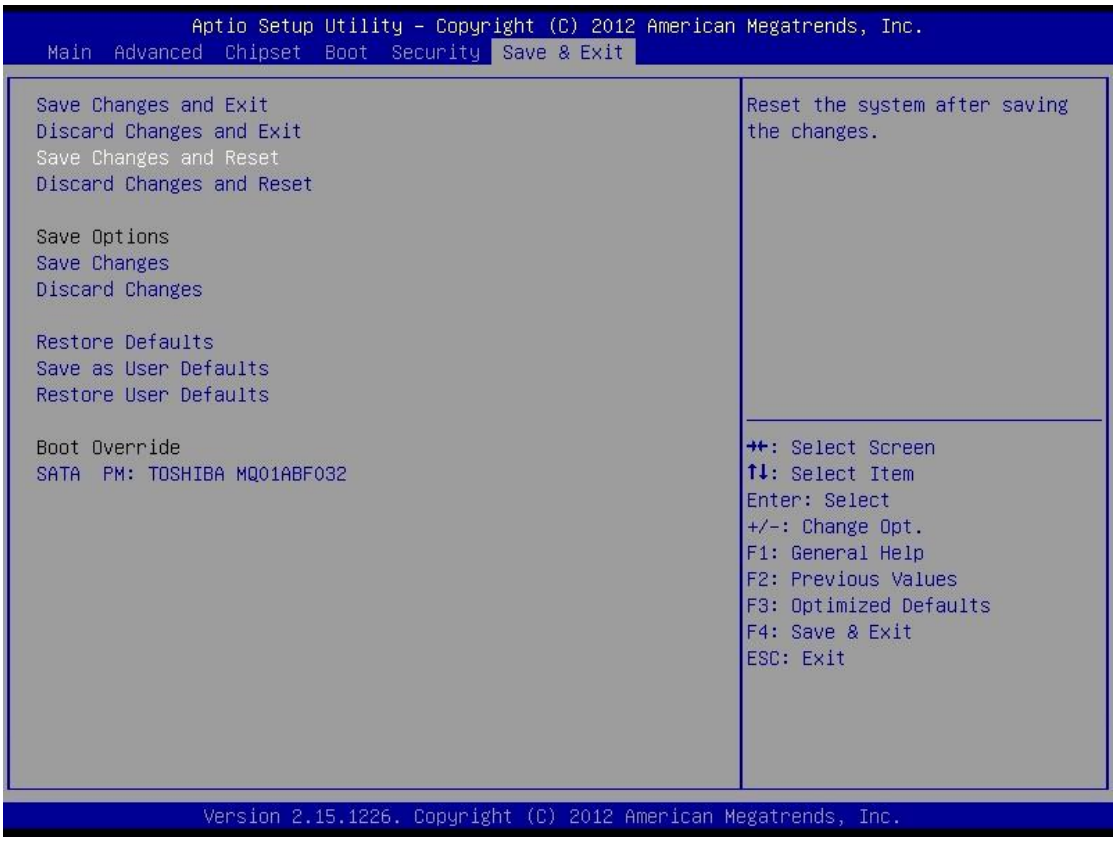
4.7 Security Settings

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



BIOS Setting	Description
Administrator Password	Sets an administrator password for the setup utility.
User Password	Sets a user password.

4.8 Save & Exit Settings



BIOS Setting	Description
Save Changes and Exit	Exits system setup after saving the changes.
Discard Changes and Exit	Exits system setup without saving any changes.
Save Changes and Reset	Resets the system after saving the changes.
Discard Changes and Reset	Resets system setup without saving any changes.
Save Changes	Saves changes done so far to any of the setup options.
Discard Changes	Discards changes done so far to any of the setup options.
Restore Defaults	Restores / Loads defaults values for all the setup options.
Save as User Defaults	Saves the changes done so far as user defaults.
Restore User Defaults	Restores the user defaults to all the setup options.

Appendix

This section provides the mapping addresses of peripheral devices and the sample code of watchdog timer configuration.

- I/O Port Address Map
- Interrupt Request Lines (IRQ)
- Digital I/O Sample Code
- Watchdog Timer Configuration

A. I/O Port Address Map

Each peripheral device in the system is assigned a set of I/O port addresses which also becomes the identity of the device. The following table lists the I/O port addresses used.

Address	Device Description
0x000000F0-0x000000F0	Numeric data processor
0x0000D000-0x0000DFFF	Intel(R) N10/ICH7 Family PCI Express Root Port - 27D2
0x0000F0F0-0x0000F0F7	Intel(R) Graphics Media Accelerator 3600 Series
0x000003B0-0x000003BB	Intel(R) Graphics Media Accelerator 3600 Series
0x000003C0-0x000003DF	Intel(R) Graphics Media Accelerator 3600 Series
0x0000F0E0-0x0000F0E7	Intel(R) N10/ICH7 Family Serial ATA Storage Controller - 27C0
0x0000F0D0-0x0000F0D3	Intel(R) N10/ICH7 Family Serial ATA Storage Controller - 27C0
0x0000F0C0-0x0000F0C7	Intel(R) N10/ICH7 Family Serial ATA Storage Controller - 27C0
0x0000F0B0-0x0000F0B3	Intel(R) N10/ICH7 Family Serial ATA Storage Controller - 27C0
0x0000F0A0-0x0000F0AF	Intel(R) N10/ICH7 Family Serial ATA Storage Controller - 27C0
0x00000020-0x00000021	Programmable interrupt controller
0x00000024-0x00000025	Programmable interrupt controller
0x00000028-0x00000029	Programmable interrupt controller
0x0000002C-0x0000002D	Programmable interrupt controller
0x00000030-0x00000031	Programmable interrupt controller
0x00000034-0x00000035	Programmable interrupt controller
0x00000038-0x00000039	Programmable interrupt controller
0x0000003C-0x0000003D	Programmable interrupt controller
0x000000A0-0x000000A1	Programmable interrupt controller
0x000000A4-0x000000A5	Programmable interrupt controller
0x000000A8-0x000000A9	Programmable interrupt controller

Address	Device Description
0x000000AC-0x000000AD	Programmable interrupt controller
0x000000B0-0x000000B1	Programmable interrupt controller
0x000000B4-0x000000B5	Programmable interrupt controller
0x000000B8-0x000000B9	Programmable interrupt controller
0x000000BC-0x000000BD	Programmable interrupt controller
0x000004D0-0x000004D1	Programmable interrupt controller
0x000004D0-0x000004D1	Motherboard resources
0x00000040-0x00000043	System timer
0x00000050-0x00000053	System timer
0x0000F080-0x0000F09F	Intel(R) N10/ICH7 Family USB Universal Host Controller - 27C8
0x0000F000-0x0000F01F	Intel(R) N10/ICH7 Family SMBus Controller - 27DA
0x00000000-0x0000001F	Direct memory access controller
0x00000000-0x0000001F	PCI bus
0x00000081-0x00000091	Direct memory access controller
0x00000093-0x0000009F	Direct memory access controller
0x000000C0-0x000000DF	Direct memory access controller
0x00000060-0x00000060	Standard PS/2 Keyboard
0x00000064-0x00000064	Standard PS/2 Keyboard
0x0000F060-0x0000F07F	Intel(R) N10/ICH7 Family USB Universal Host Controller - 27C9
0x000003F8-0x000003FF	Serial Port (COM1)
0x000002F8-0x000002FF	Serial Port (COM2)
0x000003E8-0x000003EF	Serial Port (COM3)
0x0000F040-0x0000F05F	Intel(R) N10/ICH7 Family USB Universal Host Controller - 27CA
0x000002E8-0x000002EF	Serial Port (COM4)
0x00000D00-0x0000FFFF	PCI bus
0x00000070-0x00000077	System CMOS/real time clock
0x00000070-0x00000077	Motherboard resources
0x0000F020-0x0000F03F	Intel(R) N10/ICH7 Family USB Universal Host Controller - 27CB

Address	Device Description
0x00000010-0x0000001F	Motherboard resources
0x00000022-0x0000003F	Motherboard resources
0x00000044-0x0000005F	Motherboard resources
0x00000062-0x00000063	Motherboard resources
0x00000065-0x0000006F	Motherboard resources
0x00000065-0x0000006F	Motherboard resources
0x00000072-0x0000007F	Motherboard resources
0x00000080-0x00000080	Motherboard resources
0x00000080-0x00000080	Motherboard resources
0x00000084-0x00000086	Motherboard resources
0x00000088-0x00000088	Motherboard resources
0x0000008C-0x0000008E	Motherboard resources
0x00000090-0x0000009F	Motherboard resources
0x000000A2-0x000000BF	Motherboard resources
0x000000E0-0x000000EF	Motherboard resources
0x0000002E-0x0000002F	Motherboard resources
0x0000004E-0x0000004F	Motherboard resources
0x00000061-0x00000061	Motherboard resources
0x00000063-0x00000063	Motherboard resources
0x00000067-0x00000067	Motherboard resources
0x00000092-0x00000092	Motherboard resources
0x000000B2-0x000000B3	Motherboard resources
0x00000680-0x0000069F	Motherboard resources
0x00001000-0x0000100F	Motherboard resources
0x0000FFFF-0x0000FFFF	Motherboard resources
0x0000FFFF-0x0000FFFF	Motherboard resources
0x00000400-0x0000047F	Motherboard resources
0x00000400-0x0000047F	Motherboard resources
0x00000500-0x0000057F	Motherboard resources
0x00000500-0x0000057F	Motherboard resources
0x00000600-0x0000061F	Motherboard resources
0x000006A0-0x000006AF	Motherboard resources

Address	Device Description
0x000006B0-0x000006EF	Motherboard resources
0x00000295-0x00000296	Motherboard resources
0x0000E000-0x0000EFFF	Intel(R) N10/ICH7 Family PCI Express Root Port - 27D0

B. Interrupt Request Lines (IRQ)

Peripheral devices use interrupt request lines to notify CPU for the service required. The following table shows the IRQ used by the devices on board.

Level	Function
IRQ 0	System timer
IRQ 1	Standard PS/2 Keyboard
IRQ 4	Serial Port (COM1)
IRQ 3	Serial Port (COM2)
IRQ 5	Serial Port (COM3)
IRQ 7	Serial Port (COM4)
IRQ 8	System CMOS/real time clock
IRQ 11	Intel(R) N10/ICH7 Family SMBus Controller - 27DA
IRQ 12	Microsoft PS/2 Mouse
IRQ 13	Numeric data processor
IRQ 16	Intel(R) N10/ICH7 Family USB Universal Host Controller - 27CB
IRQ 18	Intel(R) N10/ICH7 Family USB Universal Host Controller - 27CA
IRQ 19	Intel(R) N10/ICH7 Family Serial ATA Storage Controller - 27C0
IRQ 19	Intel(R) N10/ICH7 Family USB Universal Host Controller - 27C9
IRQ 22	High Definition Audio Controller
IRQ 23	Intel(R) N10/ICH7 Family USB Universal Host Controller - 27C8
IRQ 23	Intel(R) N10/ICH7 Family USB2 Enhanced Host Controller - 27CC
IRQ 81 ~ 190	Microsoft ACPI-Compliant System
IRQ 4294967290	Intel(R) 82583V Gigabit Network Connection #2
IRQ 4294967291	Intel(R) 82583V Gigabit Network Connection
IRQ 4294967292	Intel(R) Graphics Media Accelerator 3600 Series
IRQ 4294967293	Intel(R) N10/ICH7 Family PCI Express Root Port - 27D2
IRQ 4294967294	Intel(R) N10/ICH7 Family PCI Express Root Port - 27D0

C. Digital I/O Sample Code

1. The file W627UHG.H

```
//---------------------------------------------------------------------------
//
// THIS CODE AND INFORMATION IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY
// KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE
// IMPLIED WARRANTIES OF MERCHANTABILITY AND/OR FITNESS FOR A
PARTICULAR
// PURPOSE.
//
//---------------------------------------------------------------------------
#ifndef      W627UHG_H
#define      W627UHG_H  1
//---------------------------------------------------------------------------
#define     W627UHG_INDEX_PORT        (W627UHG_BASE)
#define     W627UHG_DATA_PORT         (W627UHG_BASE+1)
//---------------------------------------------------------------------------
#define     W627UHG_REG_LD            0x07
//---------------------------------------------------------------------------
#define W627UHG_UNLOCK                0x87
#define     W627UHG_LOCK              0xAA
//---------------------------------------------------------------------------
unsigned int Init_W627UHG(void);
void Set_W627UHG_LD( unsigned char);
void Set_W627UHG_Reg( unsigned char, unsigned char); unsigned char
Get_W627UHG_Reg( unsigned char);
//---------------------------------------------------------------------------
#endif//     W627UHG_H
```

The file W627UHG.CPP

```
//---------------------------------------------------------------------------
//
// THIS CODE AND INFORMATION IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY
// KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE
// IMPLIED WARRANTIES OF MERCHANTABILITY AND/OR FITNESS FOR A
PARTICULAR
// PURPOSE.
//
//---------------------------------------------------------------------------
#include "W627UHG.H"
#include <dos.h>
//---------------------------------------------------------------------------
unsigned int W627UHG_BASE; void Unlock_W627UHG (void); void Lock_W627UHG (void);
//---------------------------------------------------------------------------
unsigned int Init_W627UHG(void)
{
	unsigned int result; unsigned char ucDid;

	W627UHG_BASE = 0x4E;
	result = W627UHG_BASE;

	ucDid = Get_W627UHG_Reg(0x20);
	if (ucDid == 0xA2)				//W83627UHG??
	{	goto Init_Finish;	}

	W627UHG_BASE = 0x2E;
	result = W627UHG_BASE;

	ucDid = Get_W627UHG_Reg(0x20);
	if (ucDid == 0xA2)				//W83627UHG??
	{	goto Init_Finish;	}

	W627UHG_BASE = 0x00;
	result = W627UHG_BASE;

Init_Finish:
	return (result);
}
//---------------------------------------------------------------------------
void Unlock_W627UHG (void)
{
	outportb(W627UHG_INDEX_PORT,	W627UHG_UNLOCK);
	outportb(W627UHG_INDEX_PORT,	W627UHG_UNLOCK);
}
//---------------------------------------------------------------------------
void Lock_W627UHG (void)
{
	outportb(W627UHG_INDEX_PORT,	W627UHG_LOCK);
}
//---------------------------------------------------------------------------
void Set_W627UHG_LD( unsigned char LD)
```

```
{
	Unlock_W627UHG();
	outportb(W627UHG_INDEX_PORT,	W627UHG_REG_LD);
	outportb(W627UHG_DATA_PORT, LD);
	Lock_W627UHG();
}
//----------------------------------------------------------------------------
void Set_W627UHG_Reg( unsigned char REG, unsigned char DATA)
{
	Unlock_W627UHG();
	outportb(W627UHG_INDEX_PORT,	REG);
	outportb(W627UHG_DATA_PORT,	DATA);
	Lock_W627UHG();
}
//----------------------------------------------------------------------------
unsigned char Get_W627UHG_Reg(unsigned char REG)
{
	unsigned char Result;
	Unlock_W627UHG();
	outportb(W627UHG_INDEX_PORT, REG);
	Result = inportb(W627UHG_DATA_PORT);
	Lock_W627UHG();
	return Result;
}
//----------------------------------------------------------------------------
```

2. The file MAIN.CPP

```
//---------------------------------------------------------------------------
//
// THIS CODE AND INFORMATION IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY
// KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE
// IMPLIED WARRANTIES OF MERCHANTABILITY AND/OR FITNESS FOR A
PARTICULAR
// PURPOSE.
//
//---------------------------------------------------------------------------
#include <dos.h>
#include <conio.h>
#include <stdio.h>
#include <stdlib.h>
#include "W627UHG.H"
//---------------------------------------------------------------------------
int main (void);

void Dio5Initial(void);
void Dio5SetOutput(unsigned char); unsigned char Dio5GetInput(void); void
Dio5SetDirection(unsigned char); unsigned char Dio5GetDirection(void);
//---------------------------------------------------------------------------
int main (void)
{
	char SIO;

	SIO = Init_W627UHG();
	if (SIO == 0)
	{
		printf("Can not detect Winbond 83627UHG, program abort.\n");
		return(1);
	}

	Dio5Initial();

	//for GPIO50..57
	Dio5SetDirection(0x0F); //GP50..53 = input, GP54..57=output
	printf("Current DIO direction = 0x%X\n", Dio5GetDirection());

	printf("Current DIO status = 0x%X\n", Dio5GetInput());

	printf("Set DIO output to high\n");
	Dio5SetOutput(0x0F);

	printf("Set DIO output to low\n");
	Dio5SetOutput(0x00);

	return 0;
}
//---------------------------------------------------------------------------
void Dio5Initial(void)
{
	unsigned char ucBuf;
```

```
        Set_W627UHG_LD(0x08);                           //switch to logic device 8
        //enable the GP5 group
        ucBuf = Get_W627UHG_Reg(0x30);
        ucBuf |= 0x02;
        Set_W627UHG_Reg(0x30, ucBuf);
}
//----------------------------------------------------------------------------
void Dio5SetOutput(unsigned char NewData)
{
        Set_W627UHG_LD(0x08);                           //switch to logic device 8
        Set_W627UHG_Reg(0xE1, NewData);
}
//----------------------------------------------------------------------------
unsigned char Dio5GetInput(void)
{
        unsigned char result;

        Set_W627UHG_LD(0x08);                           //switch to logic device 8
        result = Get_W627UHG_Reg(0xE1);
        return (result);
}
//----------------------------------------------------------------------------
void Dio5SetDirection(unsigned char NewData)
{
        //NewData : 1 for input, 0 for output
        Set_W627UHG_LD(0x08);                           //switch to logic device 8
        Set_W627UHG_Reg(0xE0, NewData);
}
//----------------------------------------------------------------------------
unsigned char Dio5GetDirection(void)
{
        unsigned char result;

        Set_W627UHG_LD(0x08);                           //switch to logic device 8
        result = Get_W627UHG_Reg(0xE0);
        return (result);
}
//----------------------------------------------------------------------------
```

D. Watchdog Timer Configuration

The Watchdog Timer (WDT) is used to generate a variety of output signals after a user programmable count. The WDT is suitable for the use in the prevention of system lock-up, such as when software becomes trapped in a deadlock. Under these sorts of circumstances, the timer will count to zero and the selected outputs will be driven.

Under normal circumstance, you will need to restart the WDT at regular intervals before the timer counts to zero.

1. Sample Code: The file W627UHG.CPP

```
//---------------------------------------------------------------------------
//
// THIS CODE AND INFORMATION IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY
// KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE
// IMPLIED WARRANTIES OF MERCHANTABILITY AND/OR FITNESS FOR A
PARTICULAR
// PURPOSE.
//
//---------------------------------------------------------------------------
#include "W627UHG.H"
#include <dos.h>
//---------------------------------------------------------------------------
unsigned int W627UHG_BASE;
void Unlock_W627UHG (void);
void Lock_W627UHG (void);
//---------------------------------------------------------------------------
unsigned int Init_W627UHG(void)
{
        unsigned int result;
        unsigned char ucDid;

        W627UHG_BASE = 0x4E;
        result = W627UHG_BASE;

        ucDid = Get_W627UHG_Reg(0x20);
        if (ucDid == 0xA2)      //W83627UHG??
        {       goto Init_Finish;       }

        W627UHG_BASE = 0x2E;
        result = W627UHG_BASE;

        ucDid = Get_W627UHG_Reg(0x20);
        if (ucDid == 0xA2)      //W83627UHG??
        {       goto Init_Finish;       }

        W627UHG_BASE = 0x00;
        result = W627UHG_BASE;
```

```
Init_Finish:
        return (result);
}
//-----------------------------------------------------------------------
void Unlock_W627UHG (void)
{
        outportb(W627UHG_INDEX_PORT,        W627UHG_UNLOCK);
        outportb(W627UHG_INDEX_PORT,        W627UHG_UNLOCK);
}
//-----------------------------------------------------------------------
void Lock_W627UHG (void)
{
        outportb(W627UHG_INDEX_PORT,        W627UHG_LOCK);
}
//-----------------------------------------------------------------------
void Set_W627UHG_LD( unsigned char LD)
{
        Unlock_W627UHG();
        outportb(W627UHG_INDEX_PORT,        W627UHG_REG_LD);
        outportb(W627UHG_DATA_PORT, LD); Lock_W627UHG();
}
//-----------------------------------------------------------------------
void Set_W627UHG_Reg( unsigned char REG, unsigned char DATA)
{
        Unlock_W627UHG(); outportb(W627UHG_INDEX_PORT,      REG);
        outportb(W627UHG_DATA_PORT,       DATA); Lock_W627UHG();
}
//-----------------------------------------------------------------------
unsigned char Get_W627UHG_Reg(unsigned char REG)
{
        unsigned char Result;
        Unlock_W627UHG();
        outportb(W627UHG_INDEX_PORT, REG);
        Result = inportb(W627UHG_DATA_PORT);
        Lock_W627UHG();
        return Result;
}
//-----------------------------------------------------------------------
```

2. Sample Code: The file W627UHG.H

```
//---------------------------------------------------------------------------
//
// THIS CODE AND INFORMATION IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY
// KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE
// IMPLIED WARRANTIES OF MERCHANTABILITY AND/OR FITNESS FOR A
PARTICULAR
// PURPOSE.
//
//---------------------------------------------------------------------------
#ifndef      W627UHG_H
#define      W627UHG_H                    1
//---------------------------------------------------------------------------
#define     W627UHG_INDEX_PORT        (W627UHG_BASE)
#define     W627UHG_DATA_PORT         (W627UHG_BASE+1)
//---------------------------------------------------------------------------
#define     W627UHG_REG_LD            0x07
//---------------------------------------------------------------------------
#define W627UHG_UNLOCK                0x87
#define     W627UHG_LOCK              0xAA
//---------------------------------------------------------------------------
unsigned int Init_W627UHG(void);
void Set_W627UHG_LD( unsigned char);
void Set_W627UHG_Reg( unsigned char, unsigned char); unsigned char
Get_W627UHG_Reg( unsigned char);
//---------------------------------------------------------------------------
#endif//     W627UHG_H
```

3. Sample Code: The file MAIN.CPP

```
//---------------------------------------------------------------------------
//
// THIS CODE AND INFORMATION IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY
// KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE
// IMPLIED WARRANTIES OF MERCHANTABILITY AND/OR FITNESS FOR A
PARTICULAR
// PURPOSE.
//
//---------------------------------------------------------------------------
#include <dos.h>
#include <conio.h>
#include <stdio.h>
#include <stdlib.h>
#include "W627UHG.H"
//---------------------------------------------------------------------------
int main (void);

void WDTInitial(void);
void WDTEnable(unsigned char); void WDTDisable(void);

//---------------------------------------------------------------------------
int main (void)
{
        char SIO;

        SIO = Init_W627UHG(); if (SIO == 0)
        {
        ..... printf("Can not detect Winbond 83627UHG, program abort.\n");
        ..... return(1);
        }

        WDTInitial();
        WDTEnable(10);
        WDTDisable();
        return 0;
}
//---------------------------------------------------------------------------
void WDTInitial(void)
{
        unsigned char bBuf;
        Set_W627UHG_LD(0x08); ..... //switch to logic device 8
        bBuf = Get_W627UHG_Reg(0x30);
        bBuf &= (~0x01);
        Set_W627UHG_Reg(0x30, bBuf); ..... //Enable WDTO
}
```

```
//---------------------------------------------------------------------------
void WDTEnable(unsigned char NewInterval)
{
	unsigned char bBuf;

	Set_W627UHG_LD(0x08); ..... //switch to logic device 8
	Set_W627UHG_Reg(0x30, 0x01); ..... //enable timer
	bBuf = Get_W627UHG_Reg(0xF5);
	bBuf &= (~0x08);
	Set_W627UHG_Reg(0xF5, bBuf); ..... //count mode is second

	Set_W627UHG_Reg(0xF6, NewInterval);..... //set timer
}
//---------------------------------------------------------------------------
void WDTDisable(void)
{
	Set_W627UHG_LD(0x08);
	Set_W627UHG_Reg(0xF6, 0x00); ..  ..... //clear watchdog timer
	Set_W627UHG_Reg(0x30, 0x00); ...../watchdog disabled
}
//---------------------------------------------------------------------------
```