# UPC-6210

## 21.5” Ultra-Slim Retail Panel PC

# User’s Manual

Version 1.0
(Feb. 2019)

iBASE

## Copyright



## Disclaimer

IBASE reserves the right to make changes and improvements to the products described in this document without prior notice. Every effort has been made to ensure the information in the document is correct; however, IBASE does not guarantee this document is error-free. IBASE assumes no liability for incidental or consequential damages arising from misapplication or inability to use the product or the information contained herein, nor for any infringements of rights of third parties, which may result from its use.

## Trademarks

All the trademarks, registrations and brands mentioned herein are used for identification purposes only and may be trademarks and/or registered trademarks of their respective owners.

# Compliance

CE

This product may cause radio interference in which case users may be required to take adequate measures.

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.

Operation is subject to the following two conditions:

- This product may not cause harmful interference
- This product must accept any interference received including interference that may cause undesired operation.

However, there is no guarantee that interference will not occur in a particular installation. If this equipment causes harmful interference to radio or television reception which can be determined by turning the equipment off and on, you may correct the interference by one or more of the following measures:

- Reorient or relocate the receiving antenna.
- Increase the separation between the equipment and the receiver.
- Connect the equipment to an outlet on a circuit different from that to which the receiver is connected.
- Consult the distributor or an experienced radio/TV technician for help.

## 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.

iBASE

**Green IBASE**


This product complies with the current RoHS restrictions that prohibit the 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:

- Put the device horizontally on a stable and solid surface during installation in case the device may fall, causing serious damage.
- Leave plenty of space around the device for ventilation.
- Use this product in environments with ambient temperatures between 0˚C and 40˚C.
- DO NOT LEAVE THIS DEVICE IN AN ENVIRONMENT WHERE THE STORAGE TEMPERATURE MAY BE BELOW -20° C OR ABOVE 60° C. To prevent from damages, the device must be used in a controlled environment.
- Keep the device away from humidity to avoid fog or condensation from accumulating on the inner surface of the panel.

### Care for your IBASE products:

- Before cleaning the device, turn it off and unplug all cables 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.
- Use a computer vacuum cleaner to remove dust to prevent the air vent or slots from getting 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.
- Ensure to use the correct power supply voltage.
- 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**

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 on the product.

# 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, CPU cooler, memory, storage devices, power adapter, panel and touch screen.

* Products, however, that fail 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.

iBASE

# 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 assistance from your distributor or sales representative concerning problems that you may have encountered, please prepare the following information:

   - Product model name
   - Product serial number
   - Detailed description of the problem
   - Error messages in text or in screenshots if there is any
   - The arrangement of the peripherals
   - Software used (such as OS and application software, including the version numbers)

3. For repair service, please download the RMA form from http://www.ibase.com.tw/english/Supports/RMAService/.
   Fill out the form and contact your distributor or sales representative.

# Table of Contents

# Chapter 1
# General Information

The information provided in this chapter includes:

- Features
- Packing List
- Specifications
- Overview
- Dimensions

## 1.1 Introduction

UPC-6210 is a 21.5” retail Panel PC with IPS LCD and projected capacitive touchscreen and supports multiple touch and 1920 x 1080 full HD for display. With a fanless design, the ultra slim system operates in 0°C ~ 50°C wide-range temperature. The system is empowered by Intel® Celeron® J1900 or Atom™ E3825 processor and has an IP65 ingress protection rating for flat front bezel. Wireless solution is available for option.

## 1.2 Features

- 21.5” 1920 x 1080 IPS LCD panel
- Projected capacitive touch screen
- IP65 rated front panel
- Built-in Intel® Celeron® J1900 / Atom™ E3825 processor
- Optional wireless solution
- Three Mini-PCIe expansion slots
- Easy replacement for SSD from the rear side

## 1.3 Packing List

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

- UPC-6210
- Power Cord

## 1.4 Optional Accessories

- Wi-Fi Cable Kit

## 1.5 Specifications

| | |
|---|---|
| **Product Name** | **UPC-6210-J19**<br>**UPC-6210-I25** |
| **Motherboard** | IB898 |
| **Operating System** | • Windows 10 & 7 (32 / 64-bit)<br>• Linux Kernel 3+ |
| **System** | |
| **CPU** | • Intel® Celeron® J1900, 2.0 GHz ~ 2.4 GHz, 2 MB cache (for **UPC-6210-J19**)<br>• Intel® Atom™ E3825, 1.33 GHz, 1 MB cache (for **UPC-6210-I25**) |
| **Memory** | 1 x DDR3L SO-DIMM 4GB (max. and default) |
| **Membrane Control** | N/A |
| **Built-in speaker / Microphone** | 2 x 2W 4Ω speakers |
| **Power** | DC 12V |
| **Super I/O** | Nuvoton NCT5523D |
| **BIOS** | AMI BIOS |
| **Watchdog** | Watchdog Timer 256 segments, 0, 1, 2…255 sec/min |
| **iSmart** | Yes (auto-scheduler / power resume) |
| **Chassis** | AL 5052, all black |
| **Mounting** | VESA 100 x 100 mm |
| **Dimensions (W x H x D)** | 529.50 x 321 x 59 mm (20.85” x 12.28” x 2.32") |
| **Net Weight** | 9 kg (19.84 lb) |
| **Ingress Protection** | IP65 rated front side |
| **RoHS** | Yes |
| **Certificate** | CE, FCC Class B, LVD |
| **Display & Touchscreen** | |
| **Display Type** | 21.5” IPS LCD panel |
| **Touch Type** | Projected capacitive touch |
| **Resolution** | Max. 1920 x 1080 |
| **Luminance** | 250 cd/m2 |
| **Contrast** | 1:1000 |
| **Colors** | Max. 16.2M |

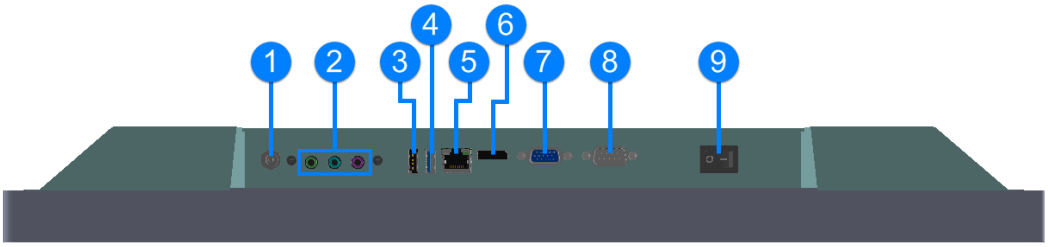
| | |
|---|---|
| **Viewing Angle** | H/V: 178° / 178° |
| **Backlight Lifetime** | 30,000 hrs |
| **Touch Interface** | USB |
| **Light Transmission** | 89% |
| **Point of Touch** | 10 points multi-touch |
| **I/O Ports** | |
| **Display** | • 1 x VGA<br>• 1 x DisplayPort |
| **LAN** | 1 x GbE LAN port |
| **Serial** | 1 x RS-232/422/485 (selectable in BIOS) |
| **SATA** | 1 x SATA III connector on board |
| **USB** | • 1 x USB 3.0<br>• 1 x USB 2.0 |
| **Audio** | 1 x 3.5 mm Line-Out jack |
| **Storage** | 1 x 2.5” SATA HDD (64G SSD by default) |
| **Internal Expansion** | • 2 x full-sized Mini-PCIe, one with mSATA and the other with USB2.0<br>• 1 x half-sized Mini-PCIe with USB |
| **Environment** | |
| **Temperature** | • **Operating:** 0°C ~ 50°C (32 ~ 140 °F)<br>• **Storage:** --20°C ~ 70°C (-4 ~ 158 °F) |
| **Relative Humidity** | 10 ~ 90% at 40°C (non-condensing) |

All specifications are subject to change without prior notice.

## 1.6 Overview

### Front View

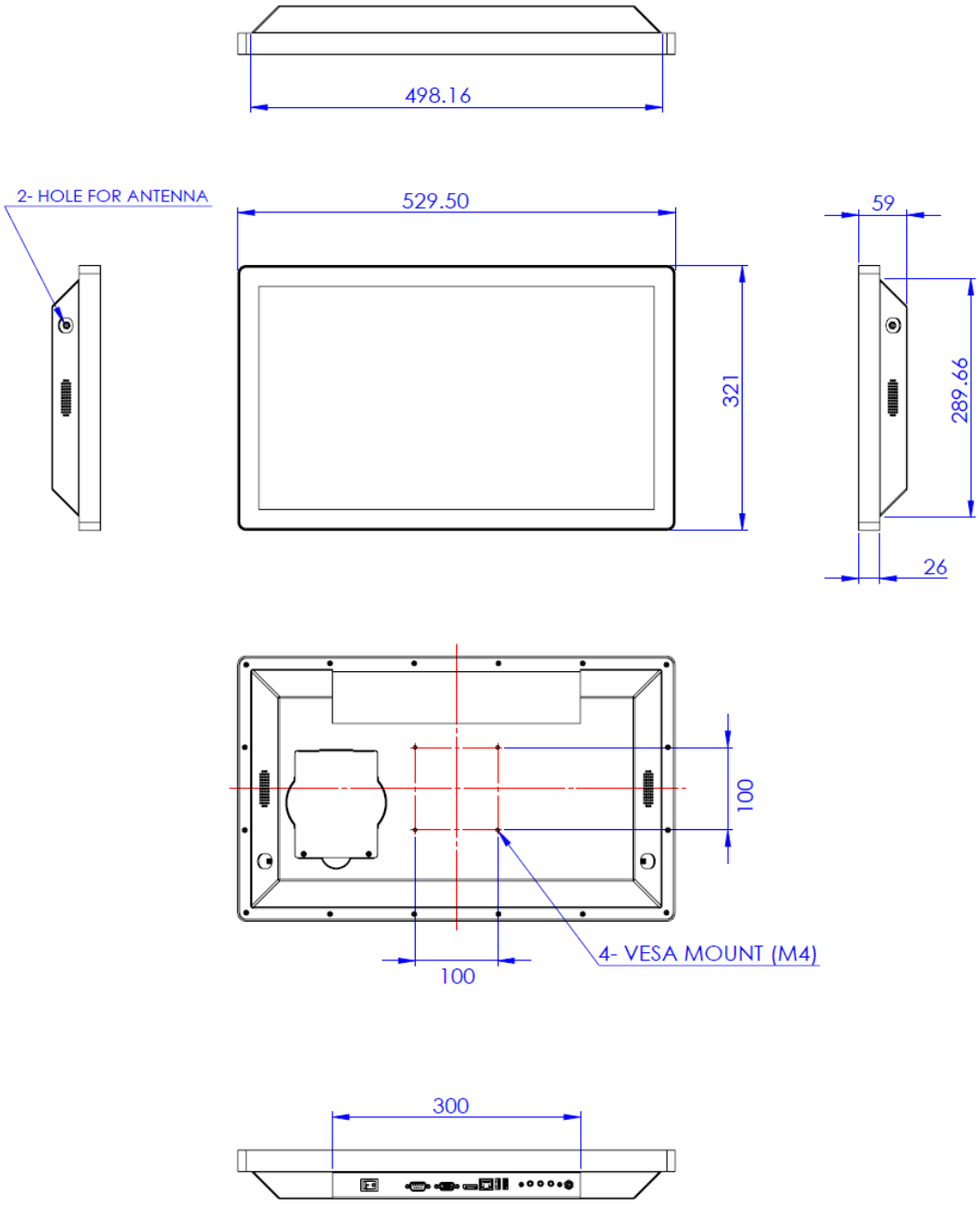
### I/O View




| No. | Name | No. | Name |
|---|---|---|---|
| 1 | Power Jack | 6 | DisplayPort |
| 2 | Audio Jacks<br>(From left to right: Line-Out, Line-In, Mic-In) | 7 | VGA Port |
| 3 | USB 2.0 Port | 8 | COM RS-232/422/485 Port |
| 4 | USB 3.0 Port | 9 | Power Switch |
| 5 | GbE LAN Port | | |

## 1.7 Dimensions

Unit: mm



# Chapter 2
# Hardware Configuration

The information provided in this chapter includes:

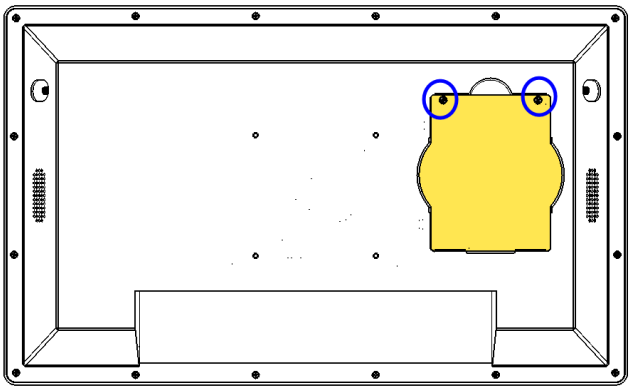
- Memory installation and membrane keypad extension
- Information and locations of connectors

## 2.1 Installations

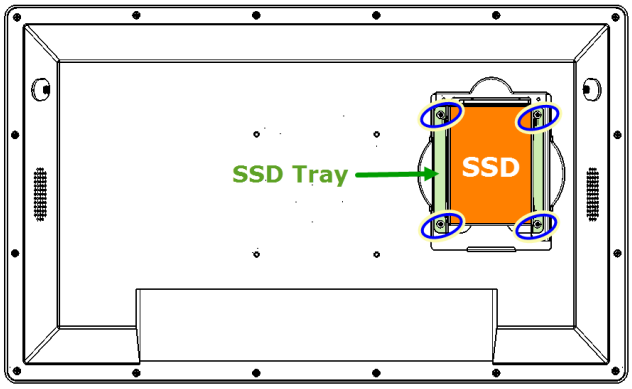
**Avoid system disassembly:** Disassembly, modification, or any attempt at repair could generate hazards and cause damage to the system, injury, or property damage, and will void any warranty. If you need to make any changes to the system, be sure to unplug the power cord of the system and have qualified engineers or technicians do the disassembly or installation.

### 2.1.1 SSD Replacement

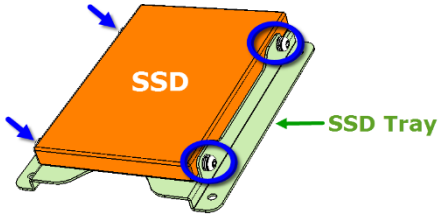
1. Remove the following 2 screws from the rear side.



2. Remove the SSD tray by loosening the 4 screws below.



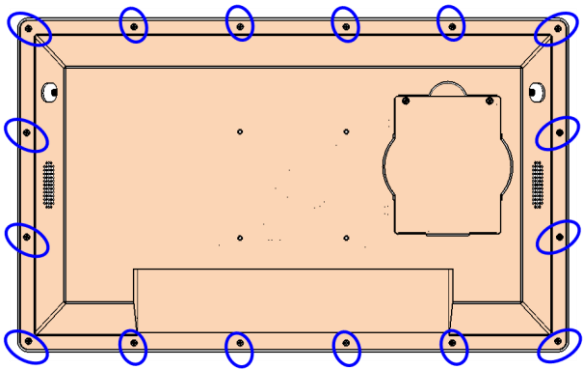

3. Unplug the cables of the SSD. Then unscrew the indicated 4 screws to release the SSD and replace it with a new one.




4. Connect the cables to the new SSD and secure it to the tray. Then fix the SSD along with the tray back to the system.

### 2.1.2 Rear Cover Removal

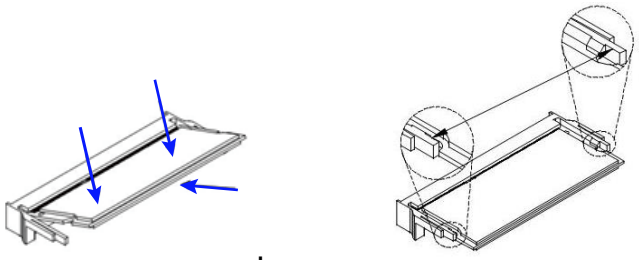
In case you need to remove the rear cover to install or replace a memory module or mini-PCIe card, remove the 16 screws shown below to remove the rear cover.



### 2.1.3 Memory Replacement

To replace or install memory modules, perform the following steps after removing the system rear cover.

1. Locate the memory slot and align the key of the memory module with that on the memory slot.
2. Insert the module slantwise and gently push the module straight down until the clips of the slot close to hold the module in place when the module touches the bottom of the slot.

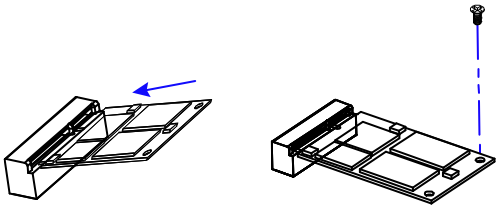


To remove the module, press the clips outwards with both hands.

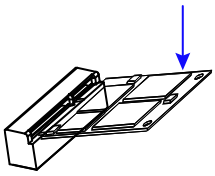
### 2.1.4 Mini-PCIe Card Installation

To replace or install a mini-PCIe card, perform the following steps after removing the rear cover.

1. Locate the mini-PCIe slot, align the key of the card to the interface, and insert the card slantwise.

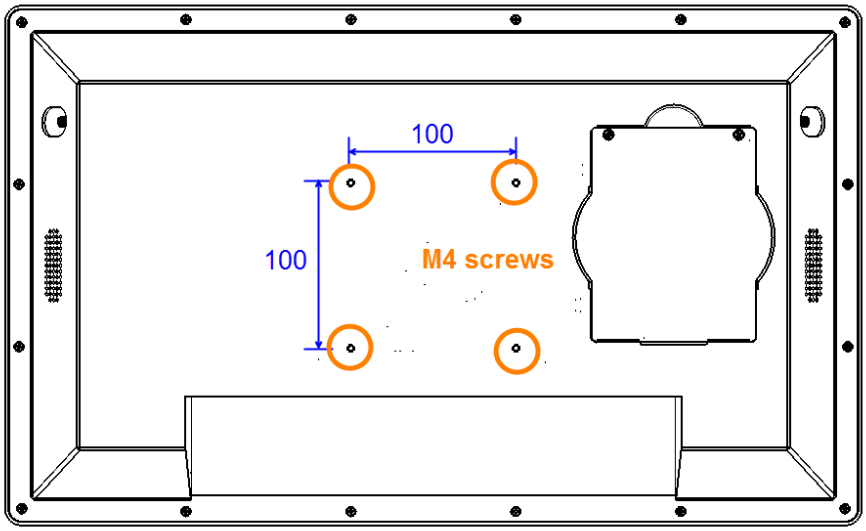


2. Push the card down and fix it with the supplied flat head screw.



### 2.1.5 VESA Mounting Installation

You will need to prepare the VESA mounting bracket in advance. Attach VESA mounting brackets to the rear side of the system and tighten four M4 screws as indicated below.



## 2.2 Pinout for COM RS-232/422/485 Port


This COM port features jumperless selection for RS-232/422/485, which is configurable in the BIOS setting.

| Pin | Signal Name | Pin | Signal Name |
|---|---|---|---|
| 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 | | |

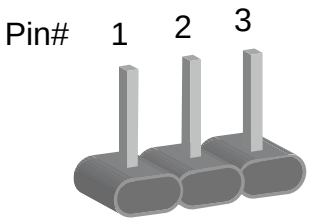
| Pin | Signal Name | | |
|---|---|---|---|
| | RS-232 | RS-422 | RS-485 |
| 1 | DCD | TX- | DATA- |
| 2 | RX | TX+ | DATA+ |
| 3 | TX | RX+ | NC |
| 4 | DTR | RX- | NC |
| 5 | Ground | Ground | Ground |
| 6 | DSR | NC | NC |
| 7 | RTS | NC | NC |
| 8 | CTS | NC | NC |
| 9 | RI | NC | NC |

## 2.3 Setting the Jumpers

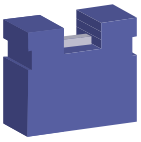
Configure the jumpers with the settings required to be able to use the features needed for your application. Contact your supplier if you have doubts about the best configuration for your use.

### 2.3.1 How to Set the 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 with the jumper cap.

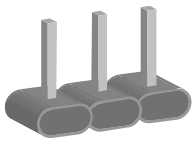
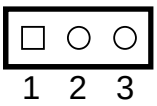
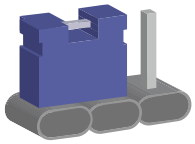
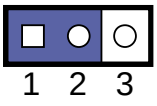
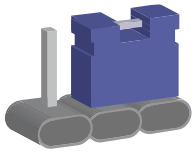
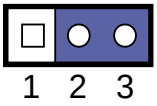



**A 3-pin jumper**



**A jumper cap**

Refer to the illustration below to set the jumpers.

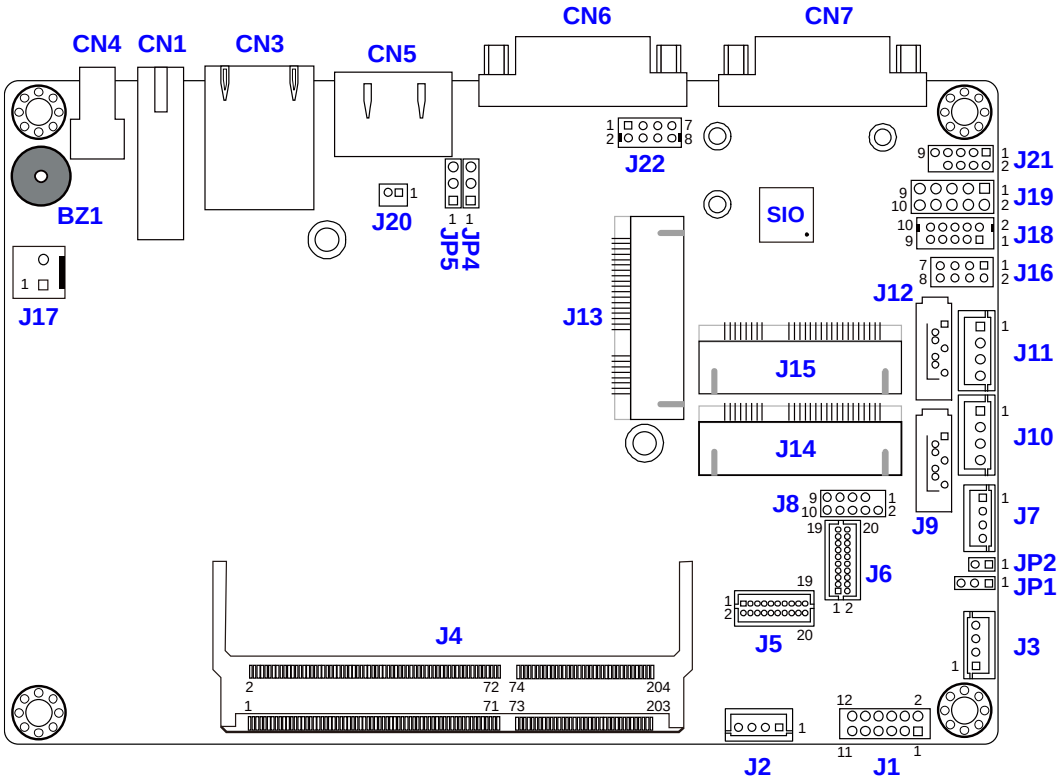
| Pin closed | Oblique view | Illustration |
|---|---|---|
| Open |  |  |
| 1-2 |  |  |
| 2-3 |  |  |

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

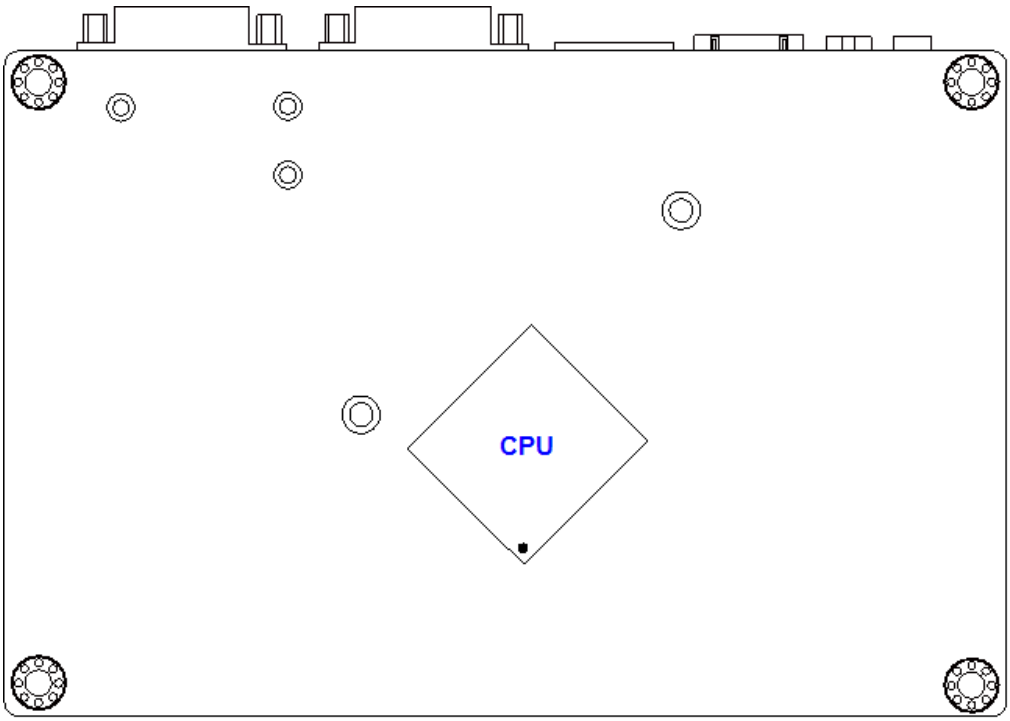
When a jumper cap is removed from two jumper pins, this jumper is **open**, i.e. turned **Off**.

## 2.4 Jumper & Connector Locations

Motherboard: IB898




**IB898 – top**

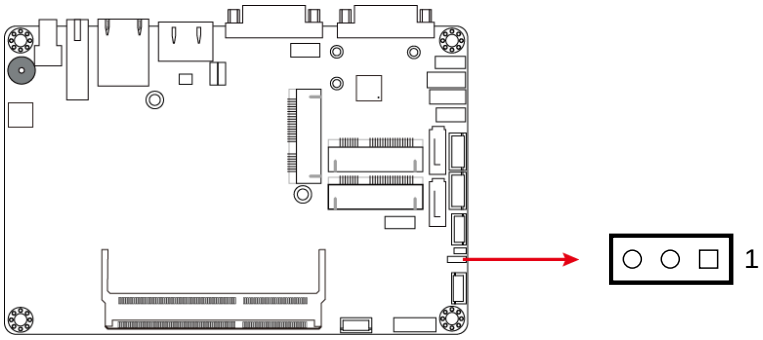


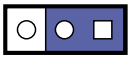
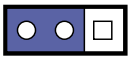

**IB898 - bottom**

## 2.5 Jumpers Quick Reference

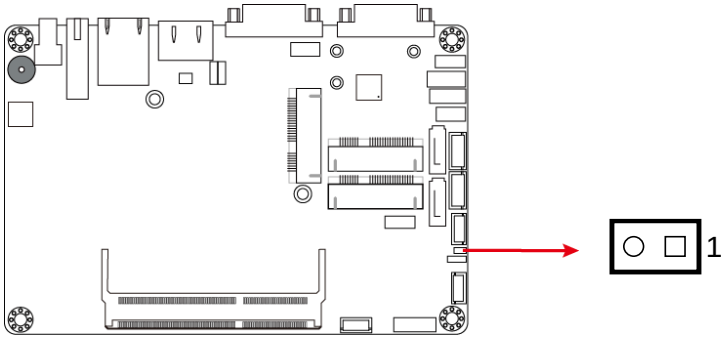
| Function | Connector Name | Page |
|---|---|---|
| LVDS Panel Power Selection | JP1 | 14 |
| LVDS Panel Brightness Control Selection | JP2 | 14 |
| CMOS Data Clearance | JP4 | 15 |
| ME Register Clearance | JP5 | 15 |

### 2.5.1 LVDS Panel Power Selection (JP1)



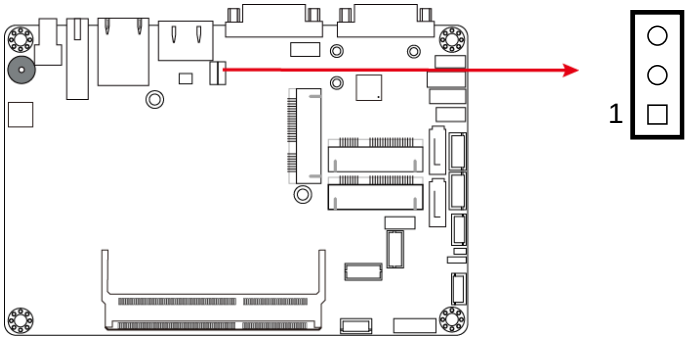

| Function | Pin closed | Illustration |
|---|---|---|
| 3.3V (default) | 1-2 |  1 |
| 5V | 2-3 |  1 |

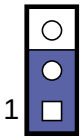
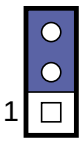
### 2.5.2 LVDS Panel Brightness Control Selection (JP2)



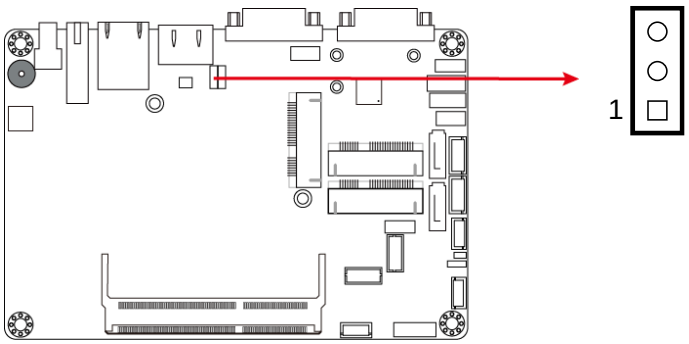

| Function | Pin closed | Illustration |
|---|---|---|
| 3.3V | Open |  1 |
| 5V (default) | Close |  1 |

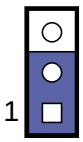
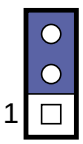
### 2.5.3 Clearing CMOS Data (JP4)




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

### 2.5.4 Clearing ME Register (JP5)



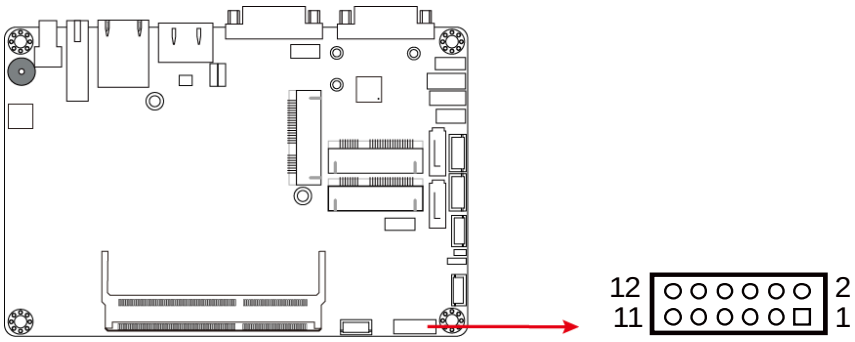

| Function | Pin closed | Illustration |
|---|---|---|
| Normal (default) | 1-2 |   |
| Clear ME Register | 2-3 |   |

## 2.6 Connectors Quick Reference

| Function | Connector Name | Page |
|---|---|---|
| Audio Connector | J1 | 17 |
| Amplifier Connector | J3 | 17 |
| LCD Backlight Connector | J7 | 18 |
| SATA HDD Power Connector | J10, J11 | 18 |
| Front Panel Setting Connector | J16 | 19 |
| Motherboard Power Input Connector | J17 | 19 |
| COM2 (RS-232) Port | J18 | 20 |
| Digital I/O Connector | J19 | 20 |
| USB 2.0 Connector | J22 | 21 |
| DDR3L SO-DIMM Socket | J4 | -- |
| SATA II / mSATA Port | J9 | -- |
| SATA II Port | J12 | -- |
| Full-Size Mini-PCIe Connector | J13 | -- |
| Full-Size Mini-PCIe / mSATA Connector | J14 | -- |
| Half-Size Mini-PCIe Connector | J15 | -- |
| Front Panel Setting Connector | J16 | -- |
| USB 2.0 Port | CN4 | -- |
| USB 3.0 Port | CN1 | -- |
| GbE LAN Port | CN3 | -- |
| Display Port | CN5 | -- |
| VGA Port | CN6 | -- |
| COM RS-232/422/485 Port [1] | CN7 | -- |
| Factory Use Only | J2, J8, J21 | -- |

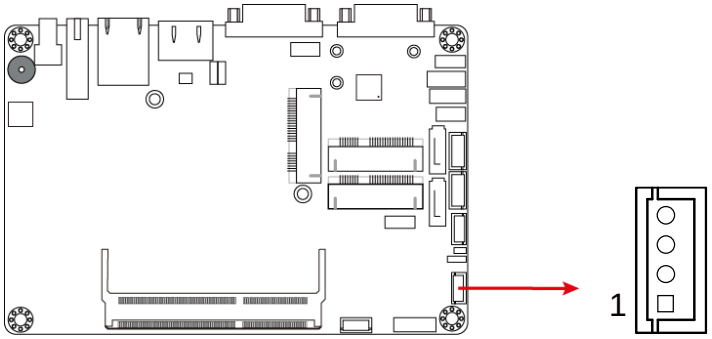
[1]: Refer to *2.2 Pinout for COM RS-232/422/485 Port*.

### 2.6.1 Audio Connector (J1)



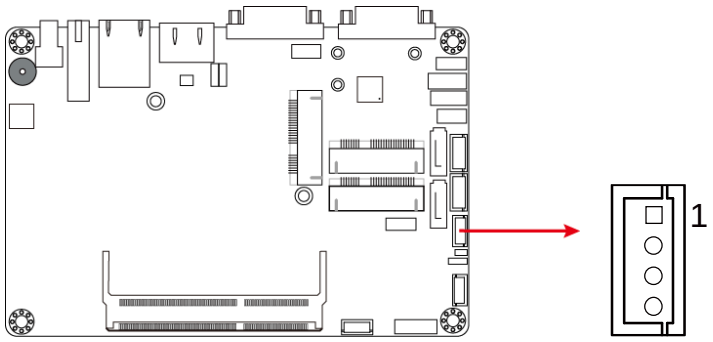

| Pin | Signal Name | Pin | Signal Name |
|---|---|---|---|
| 1 | Lineout_L | 7 | JD_Linein |
| 2 | Lineout_R | 8 | GNd |
| 3 | JD_Front | 9 | MIC_L |
| 4 | GND | 10 | MIC-R |
| 5 | Linein_L | 11 | JD_MIC1 |
| 6 | Linein_R | 12 | GND |

### 2.6.2 Amplifier Connector (J3)



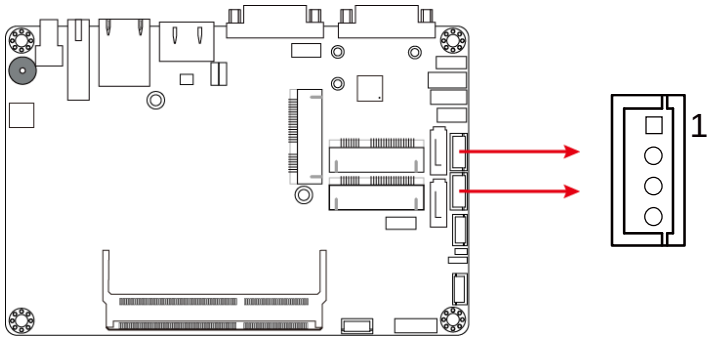

| Pin | Signal Name | Pin | Signal Name |
|---|---|---|---|
| 1 | OUTL+ | 3 | OUTR- |
| 2 | OUTL- | 4 | OUTR+ |

### 2.6.3 LCD Backlight Connector (J7)



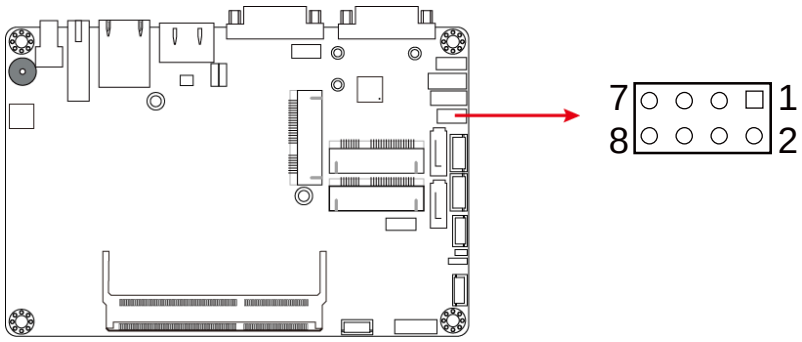

| Pin | Signal Name | Pin | Signal Name |
|---|---|---|---|
| 1 | +12V(1A) | 3 | Brightness Control |
| 2 | Backlight Enable | 4 | Ground |

### 2.6.4 SATA HDD Power Connector (J10, J11)



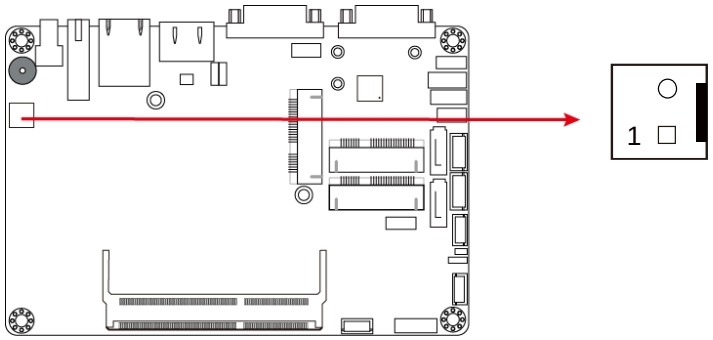

| Pin | Signal Name | Pin | Signal Name |
|---|---|---|---|
| 1 | +5V (1A) | 3 | Ground |
| 2 | Ground | 4 | +12V (1A) |

### 2.6.5 Front Panel Setting Connector (J16)



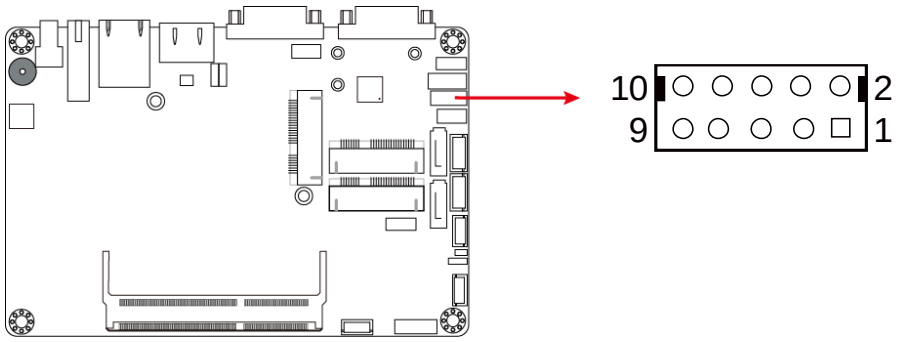

| Pin | Signal Name | Pin | Signal Name |
|---|---|---|---|
| 1 | Power BTN | 5 | Reset BTN |
| 2 | Power BTN | 6 | Reset BTN |
| 3 | HDD LED+ | 7 | Power LED+ |
| 4 | HDD LED- | 8 | Power LED- |

### 2.6.6 Motherboard Power Input Connector (J17)



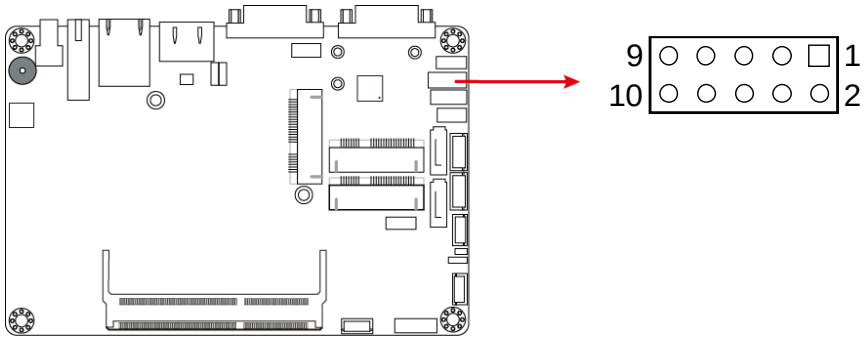

| Pin | Signal Name | Pin | Signal Name |
|---|---|---|---|
| 1 | VCC12 | 2 | GND |

### 2.6.7 COM2 RS-232 Port (J18)



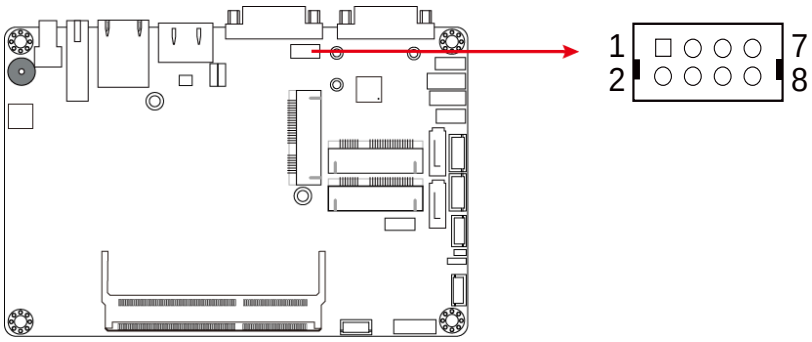

| Pin | Signal Name | Pin | Signal Name |
|---|---|---|---|
| 1 | DCD, Data carrier detect | 6 | DSR, Data set ready |
| 2 | RXD, Receive data | 7 | RTS, Request to send |
| 3 | TXD, Transmit data | 8 | CTS, Clear to send |
| 4 | Data terminal ready | 9 | RI, Ring indicator |
| 5 | GND, ground | 10 |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

### 2.6.8 Digital I/O Connector (J19)




| Pin | Signal Name | Pin | Signal Name |
|---|---|---|---|
| 1 | GND | 6 | OUT0 |
| 2 | VCC (500mA) | 7 | IN3 |
| 3 | OUT3 | 8 | IN1 |
| 4 | OUT1 | 9 | IN2 |
| 5 | OUT2 | 10 | IN0 |

### 2.6.9 USB 2.0 Connector (J22)




| Pin | Signal Name | Pin | Signal Name |
|---|---|---|---|
| 1 | VCC | 5 | D0+ |
| 2 | GND | 6 | D1- |
| 3 | D0- | 7 | GND |
| 4 | D1+ | 8 | VCC |

# Chapter 3
# Driver Installation

The information provided in this chapter includes:

- Intel® Chipset Software Installation Utility
- Intel® Graphics Driver Installation
- HD Audio Driver Installation
- Intel® Trusted Execution Engine Installation
- USB 3.0 Driver Installation
- LAN Driver Installation

## 3.1 Introduction

This section describes the installation procedures for software drivers.

---

**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. Go to the download page of the product. Copy the compressed drivers file to your computer. Double click the file to decompress it. Run “CDGuide” to go to the main drivers page as shown. Click **Intel** and then **Intel(R) Baytrail 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. The driver has been completely installed. Click **Finish** and restart the computer for changes to take effect.

## 3.3 Graphics Driver Installation

1. Click **Intel** and then **Intel(R) Baytrail Chipset Drivers**.


2. Click **Intel(R) Baytrail 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. The driver has been completely installed. Click **Finish** and restart the computer for changes to take effect.

## 3.4 HD Audio Driver Installation

1. Click **Intel** and then **Intel(R) Baytrail 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** and restart the computer for changes to take effect.

## 3.5 Intel® Trusted Execution Engine Installation

**Note:** The driver is for Windows 7 only.

1. Click **Intel** and then **Intel(R) Baytrail Chipset Drivers**.


2. Click **Intel(R) TXE Drivers**.


3. When the *Weocome* screen appears, click **Next**.
4. Click **Next** to agree with the license agreement and continue the installation.
5. The driver has been completely installed. Click **Finish** and restart the computer for changes to take effect.

## 3.6 USB 3.0 Driver Installation

1. Click **Intel** and then **Intel(R) Baytrail Chipset Drivers**.


2. Click **Intel(R) USB 3.0 Drivers**.

3. On the *Welcome* screen of the InstallShield Wizard, click **Next**.
4. Click **Yes** to agree with the license agreement.
5. On the *Readme File Information* screen, click **Next** for installation.
6. The driver has been completely installed. Click **Finish** and restart the computer for changes to take effect.

## 3.7 LAN Driver Installation

1. Click **LAN Card** on the left pane. Then click **Intel(R) LAN Controller Drivers,** and **Intel(R) I21x Gigabit Network Drivers**.


2. On the *Welcome* screen of the InstallShield Wizard, click **Next**.
3. Click **Next** to agree with the license agreement.
4. When the wizard is ready for installation, click **Install**.
5. The driver has been completely installed. Click **Finish** and restart the computer 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
- Boot Settings
- Save & Exit

**Avoid changing the default settings:**

We strongly suggest you contact IBASE for technical support before you make any changes in BIOS. Please be noted that the device may malfunction if you change the BIOS settings improperly.

## 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 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. You can also press <F7> to call the pop-up Boot menu immediately.

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:

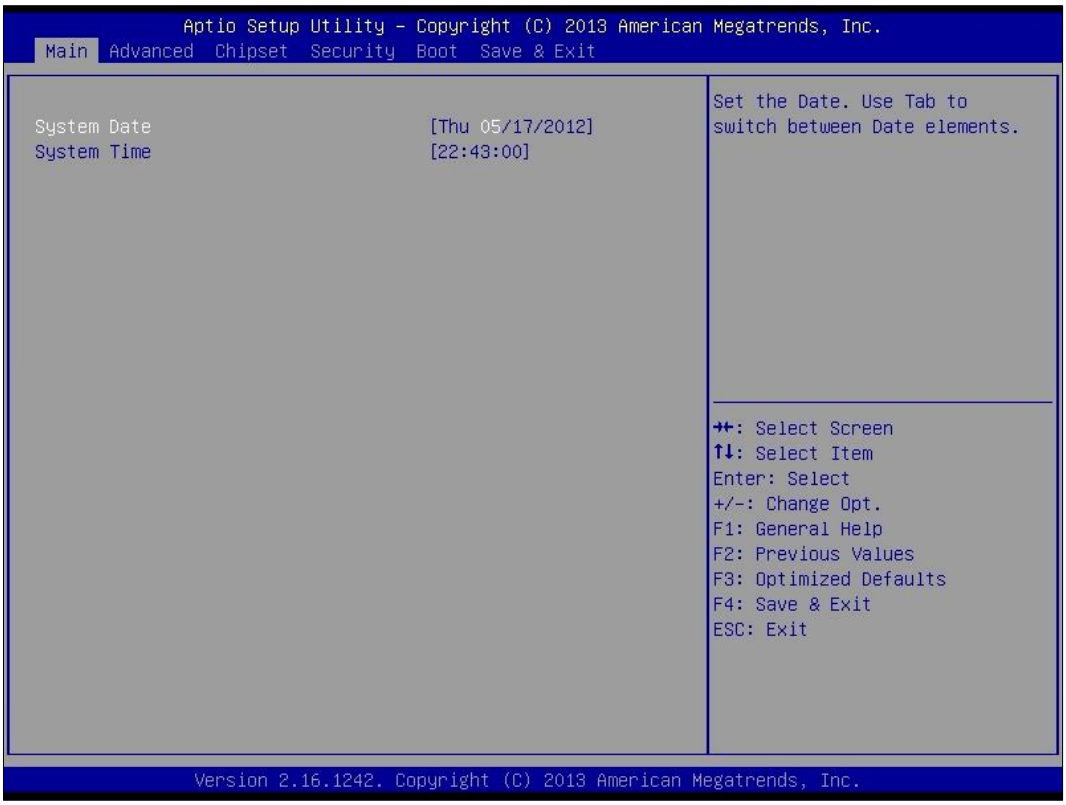
```
Press <DEL> to Enter Setup
```

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.

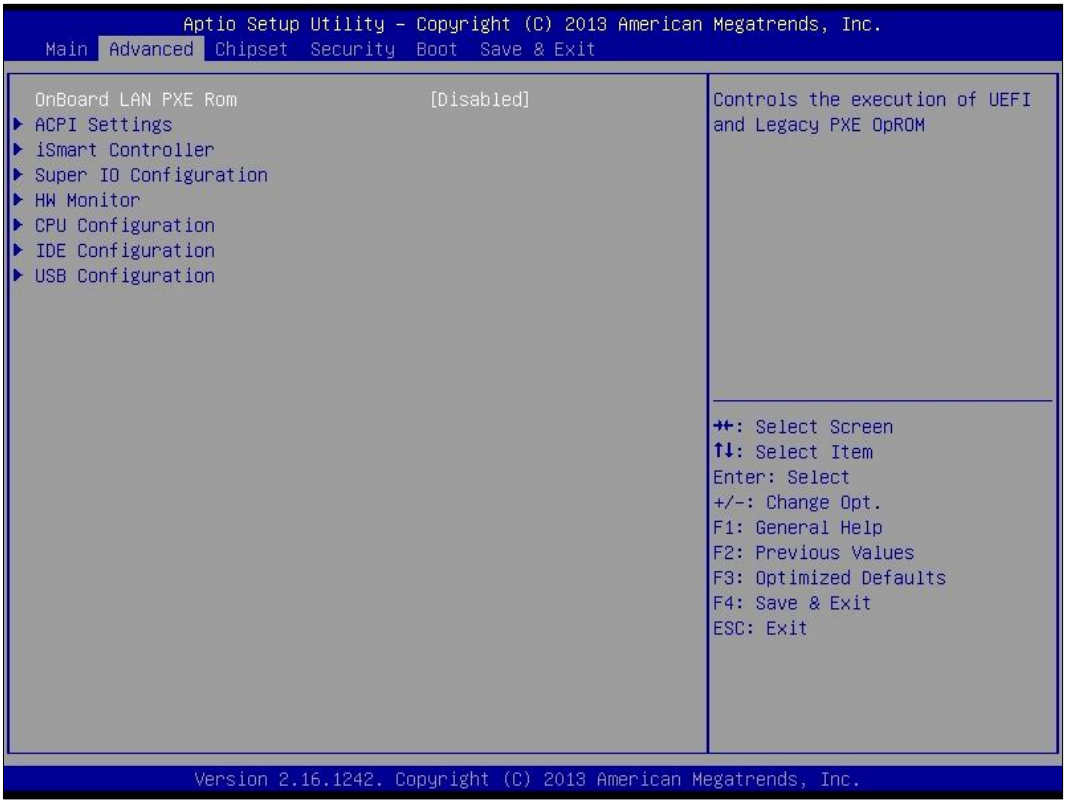
## 4.3 Main Settings




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

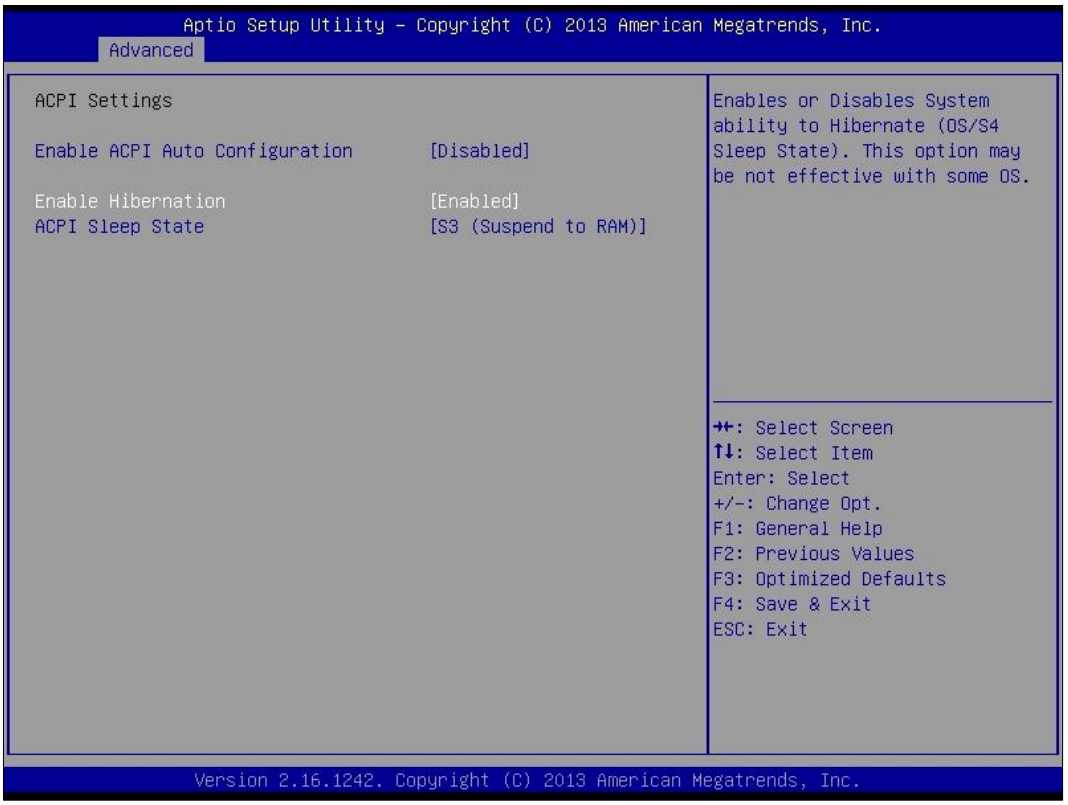
## 4.4 Advanced Settings

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



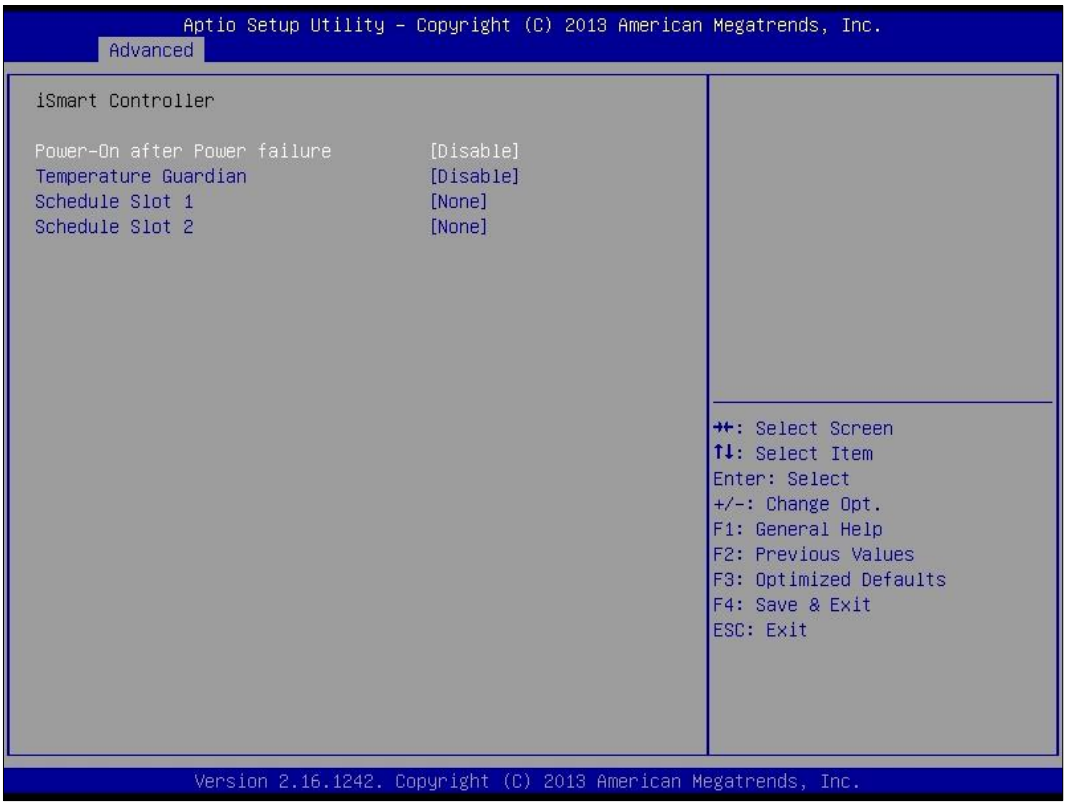

| BIOS Setting | Description |
|---|---|
| OnBoard LAN PXE Rom | Enables or disables the execution of UEFI and Legacy PXE OpROM. |

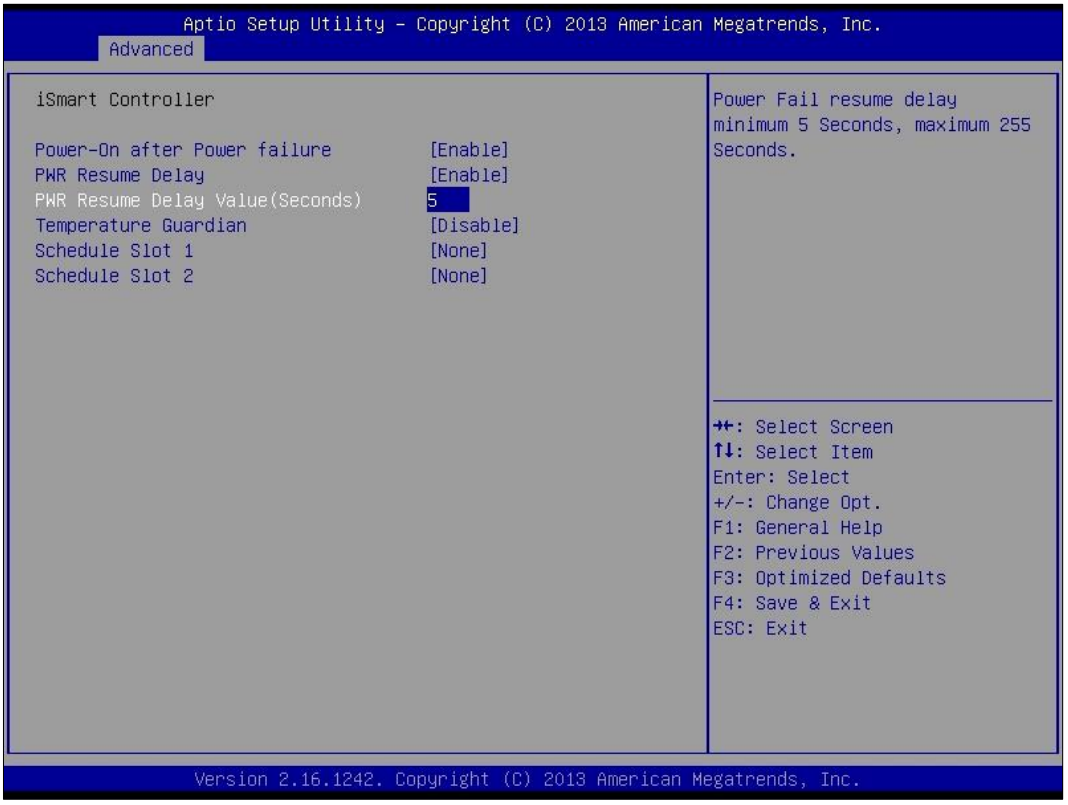
### 4.4.1 ACPI Settings

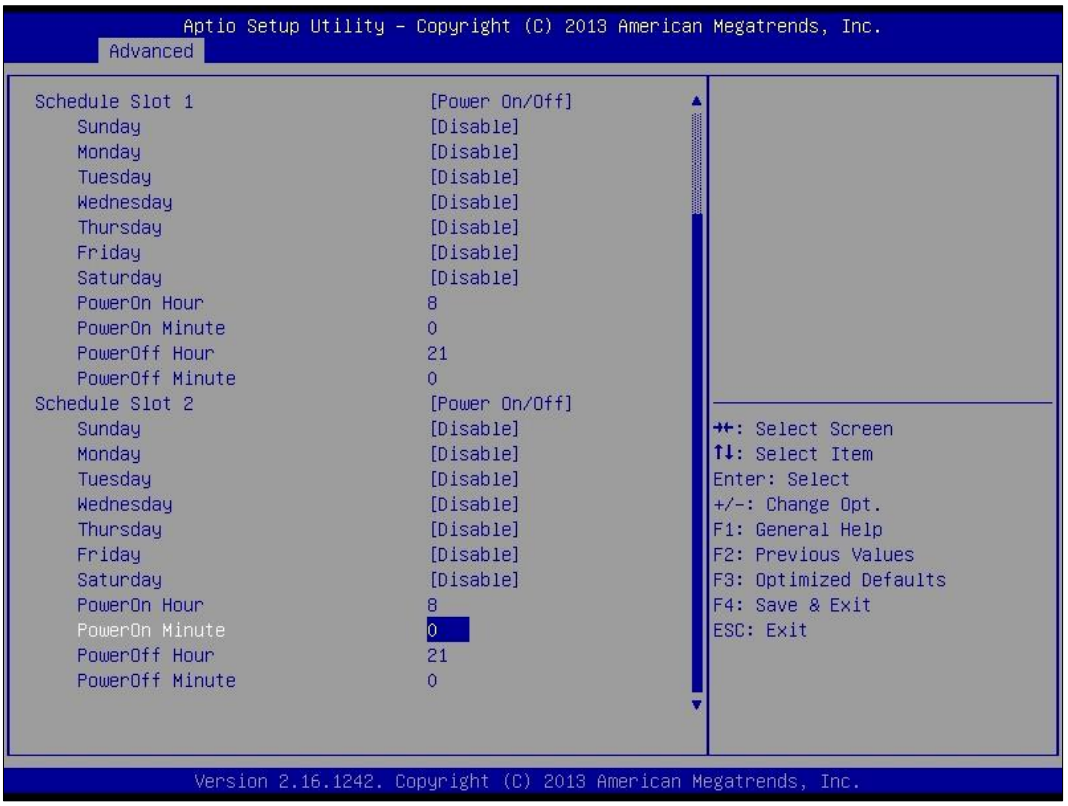



| BIOS Setting | Description |
|---|---|
| Enable ACPI Auto Configuration | Enables / Disables the ACPI Auto configuration. |
| Enable Hibernation | Enables / Disables the system ability to hibernate (OS/S4 Sleep State). This option may not be effective with some OS. |
| ACPI Sleep State | Selects a ACPI sleep state for the system to enter.<br><br>Options:<br>• Suspend Disabled<br>• S3 (Suspend to RAM) |

### 4.4.2 iSmart Controller



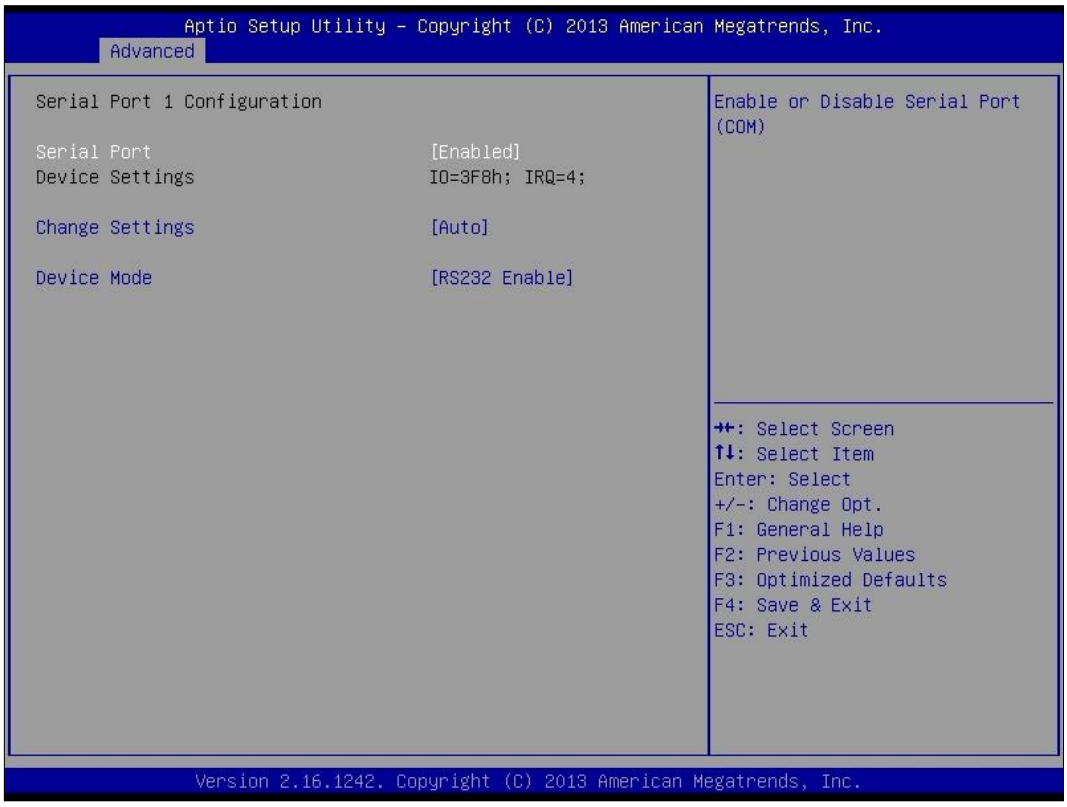



| BIOS Setting | Description |
|---|---|
| Power-On after Power failure | Enables / Disables the system to be turned on automatically after a power failure. |
| Power Resume Delay | Enables / Disables to delay the time for system to turn on. |
| Power Resume Delay Value (Seconds) | Sets the delay timer for the system to resume power if power failure occurs.<br>The minimum delay timer is 5 seconds, and the maximum is 255 seconds. |
| Temperature Guardian | Generate the reset signal when system hands up on POST. |
| Schedule Slot 1 / 2 | Sets up the hour / minute / day for the power-on schedule for the system.<br><br>Options:<br>• None<br>• Power On<br>• Power On / Off |

### 4.4.3 Super IO Configuration

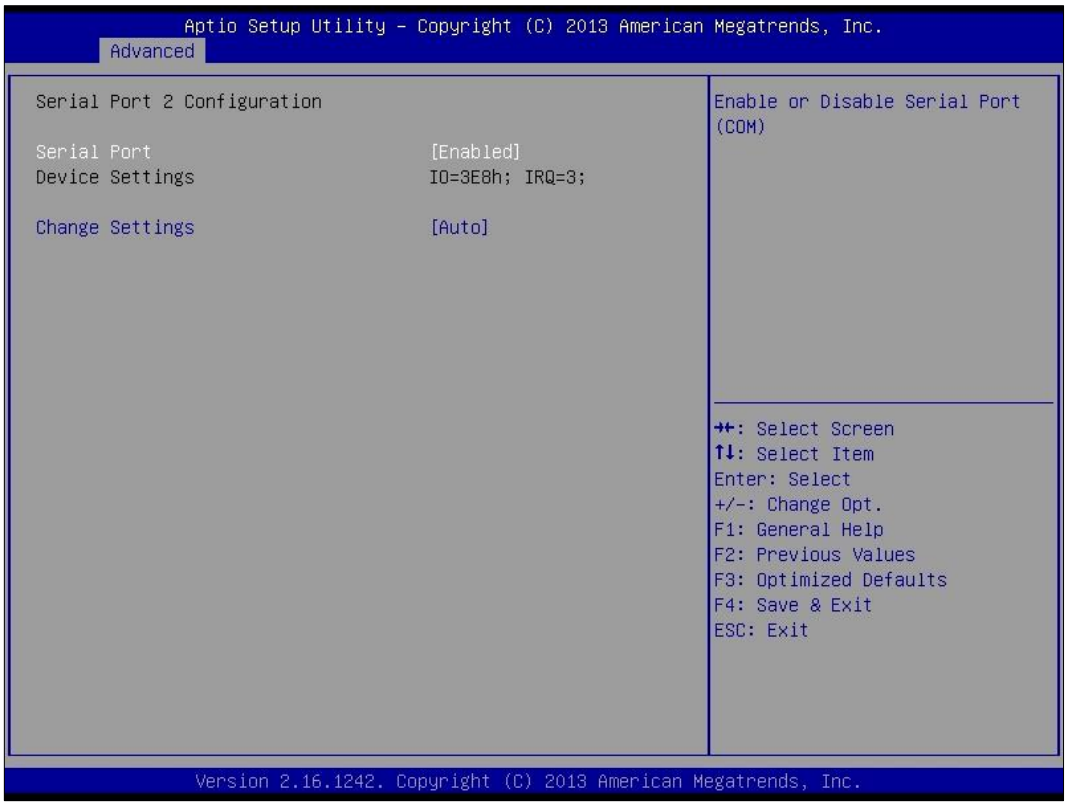

| BIOS Setting | Description |
|---|---|
| Serial Port Configuration | Sets Parameters of Serial Ports.<br>You can enable / disable the serial port and select an optimal settings for the Super IO device. |

### 4.4.4.1. Serial Port 1 Configuration



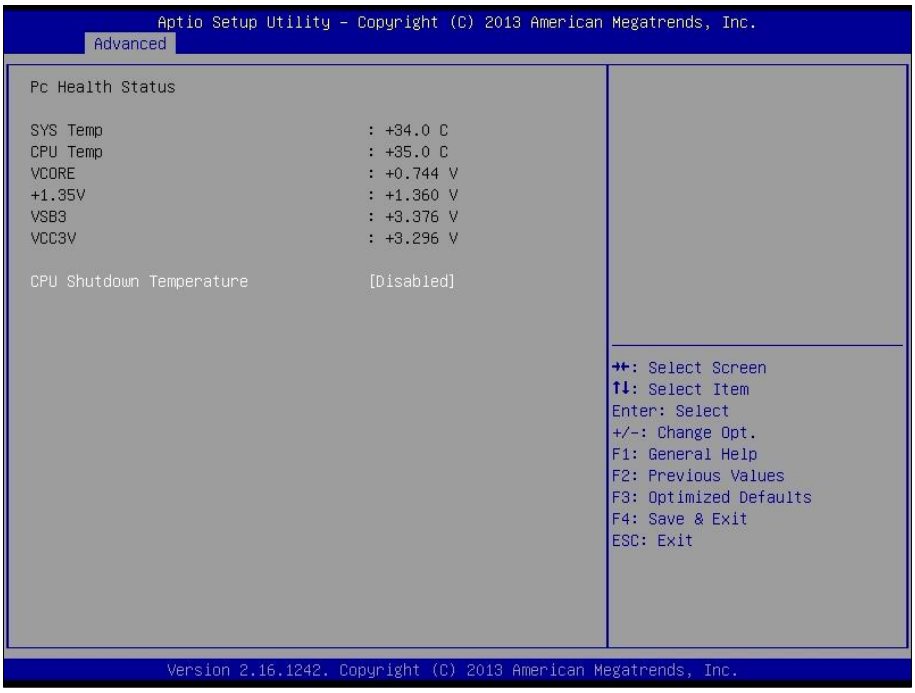

| BIOS Setting | Description |
|---|---|
| Change Settings | Selects an optimal settings for the Super I/O device.<br><br>Options:<br>• Auto<br>• IO=3F8h ; IRQ=4<br>• IO=3F8h ; IRQ=3, 4, 5, 6, 7, 9. 10, 11, 12<br>• IO=2F8h ; IRQ=3, 4, 5, 6, 7, 9. 10, 11, 12<br>• IO=3E8h ; IRQ=3, 4, 5, 6, 7, 9. 10, 11,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

### 4.4.4.2. Serial Port 2 Configuration




| BIOS Setting | Description |
|---|---|
| Change Settings | Selects an optimal settings for the Super I/O device.<br><br>Options:<br>• Auto<br>• IO=3E8h ; IRQ=3<br>• IO=3E8h ; IRQ=3, 4, 5, 6, 7, 9. 10, 11, 12<br>• IO=2E8h ; IRQ=3, 4, 5, 6, 7, 9. 10, 11, 12<br>• IO=2F0h ; IRQ=3, 4, 5, 6, 7, 9. 10, 11, 12<br>• IO=2E0h ; IRQ=3, 4, 5, 6, 7, 9. 10, 11, 12 |

### 4.4.4 Hardware Monitor

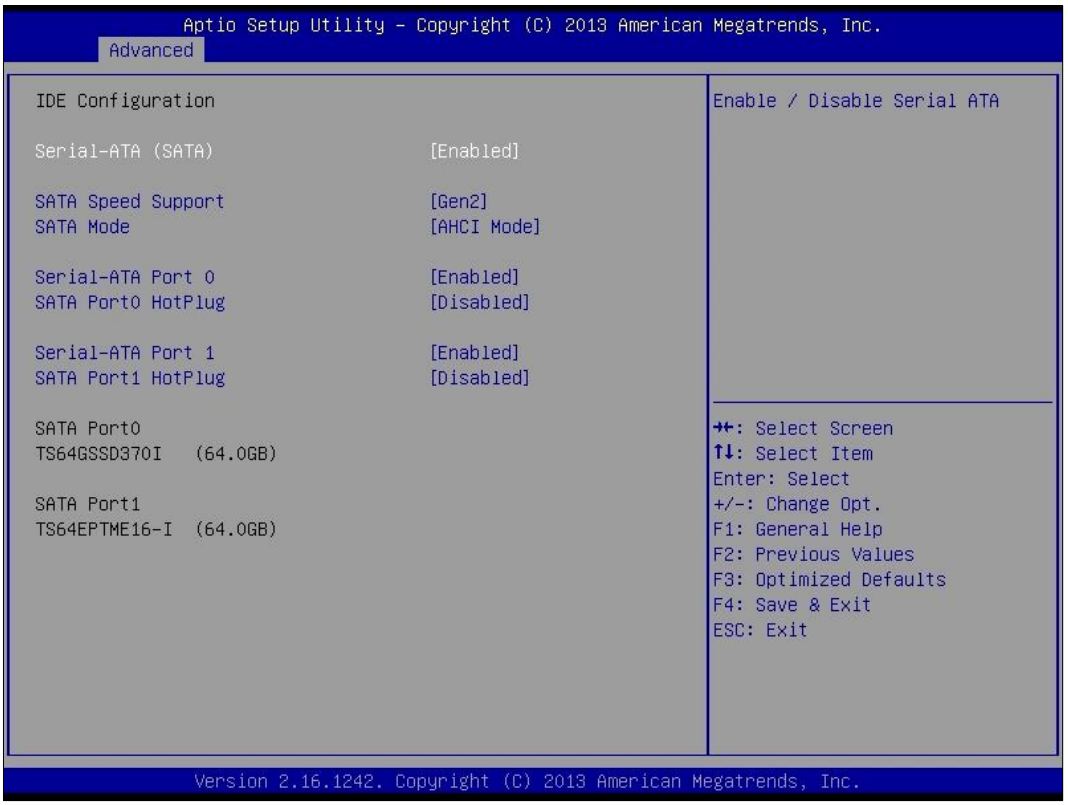



| BIOS Setting | Description |
|---|---|
| CPU Shutdown Temperature | This field enables or disables the Shutdown Temperature<br>Options: Disabled, 70°C, 75°C, 80°C, 85°C, 90°C, 95°C |

### 4.4.5 CPU Configuration

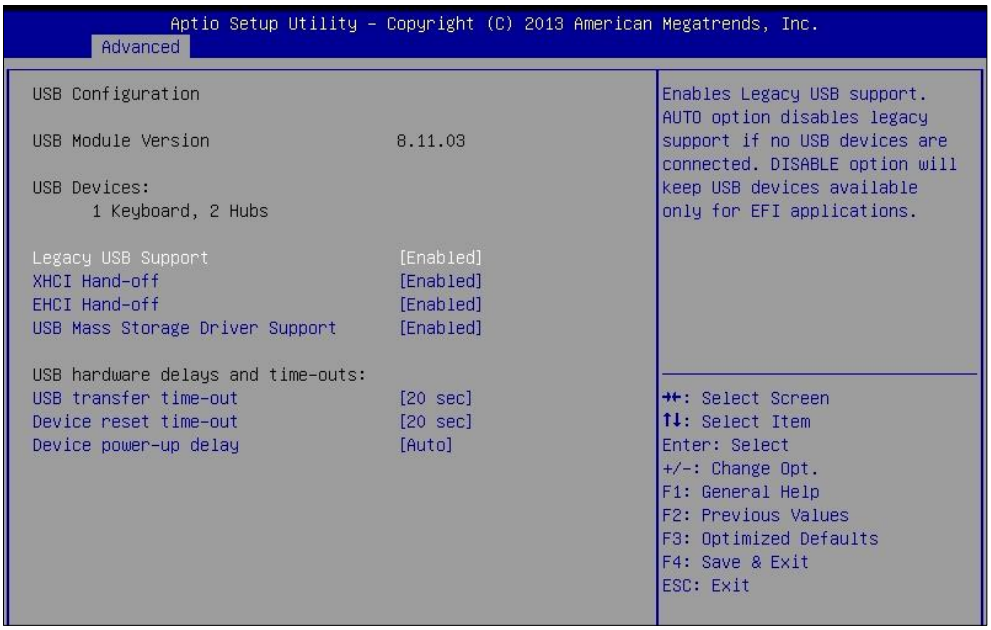

| BIOS Setting | Description |
|---|---|
| Socket 0 CPU Information | Displays the specific socket CPU Information. |

### 4.4.6 IDE Configuration



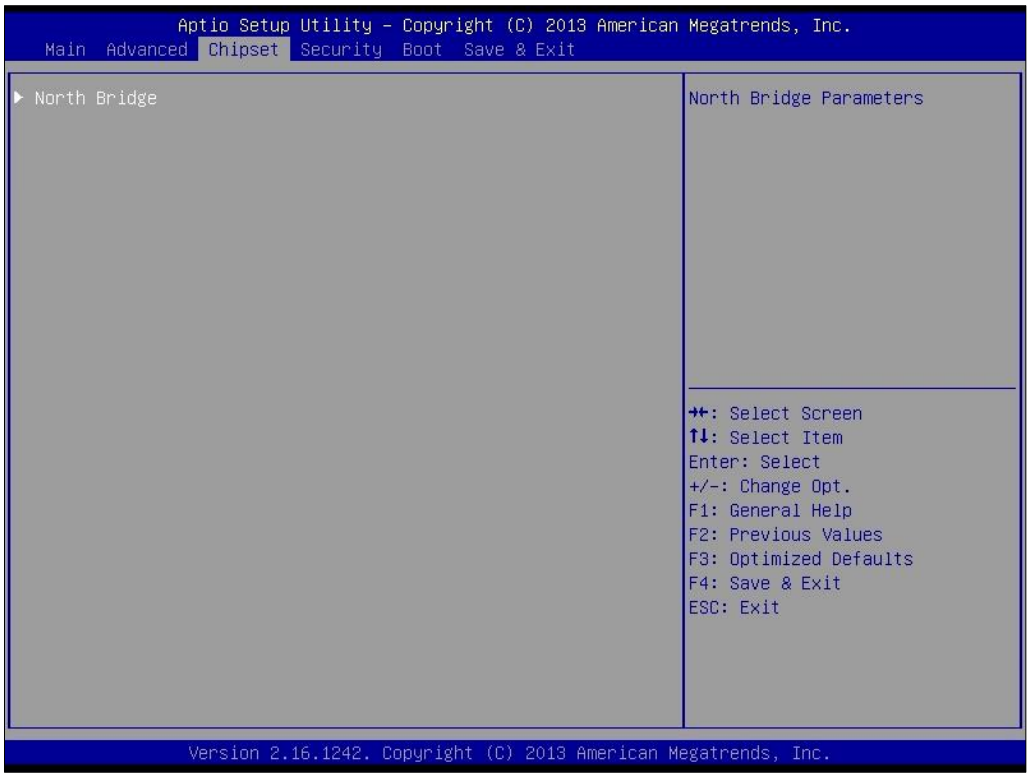

| BIOS Setting | Description |
|---|---|
| Serial-ATA (SATA) | Enables / Disables the Serial ATA. |
| SATA Speed Support | Supports SATA speed Gen.1 or Gen. 2. |
| SATA Mode | Selects IDE / AHCI Mode. |
| Serial –ATA Port 0 | Enables / Disables Serial Port 0. |
| SATA Port0 HotPlug | Enables / Disables SATA Port 0 HotPlug. |
| Serial –ATA Port 1 | Enables / Disables Serial Port 1. |
| SATA Port1 HotPlug | Enables / Disables SATA Port 1 HotPlug. |

### 4.4.7 USB Configuration

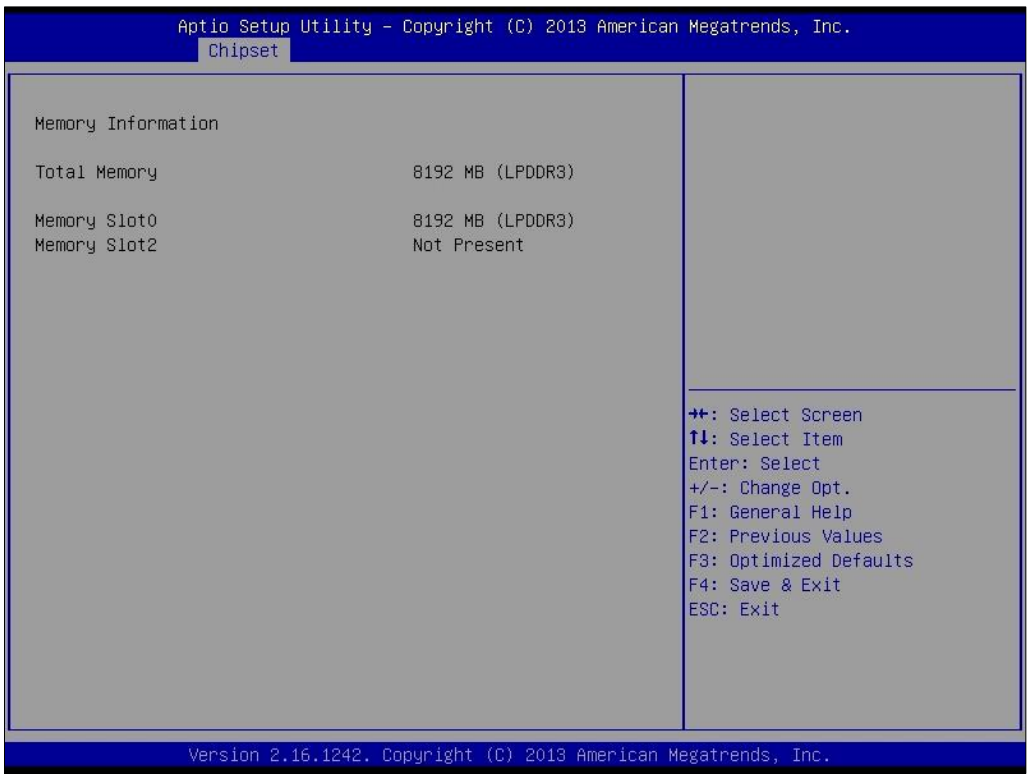



| BIOS Setting | Description |
|---|---|
| Legacy USB Support | Enables / Disables Legacy USB support.<br>• “Auto” disables legacy support if there is no USB device connected.<br>• “Disable” keeps USB devices available only for EFI applications. |
| XHCI Hand-off | This is a workaround for OSes without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

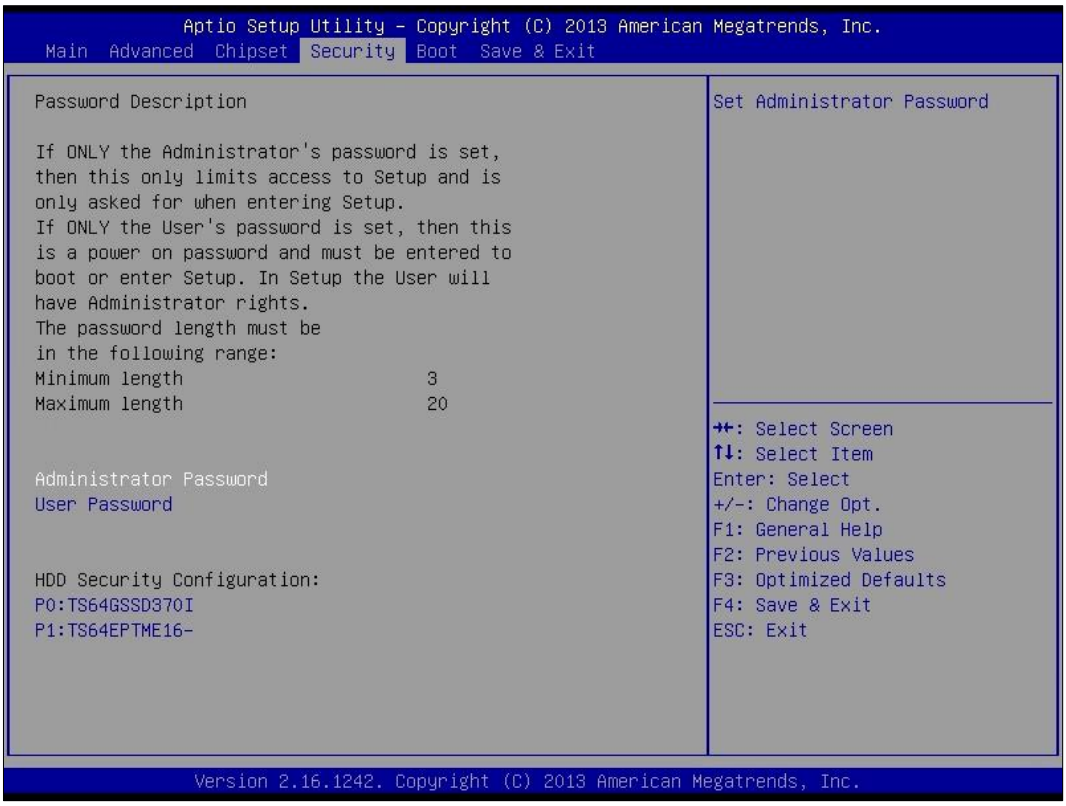
## 4.5 Chipset Settings




### 4.5.1 North Bridge

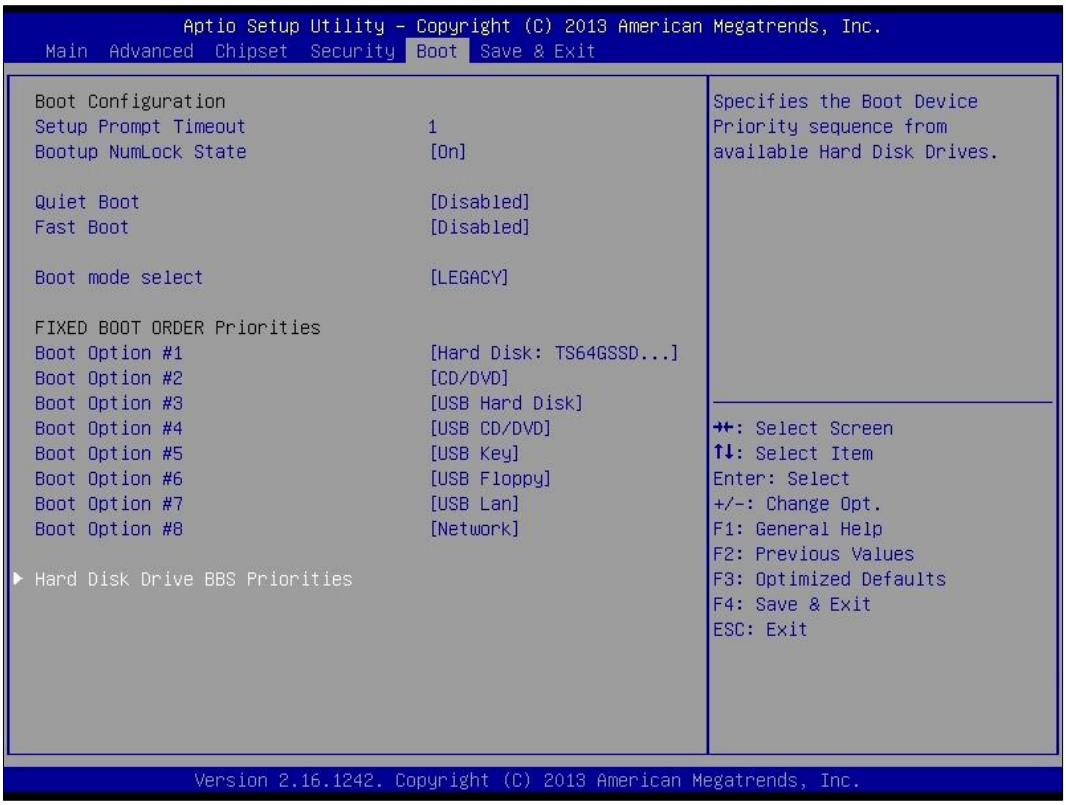


## 4.6 Security Settings



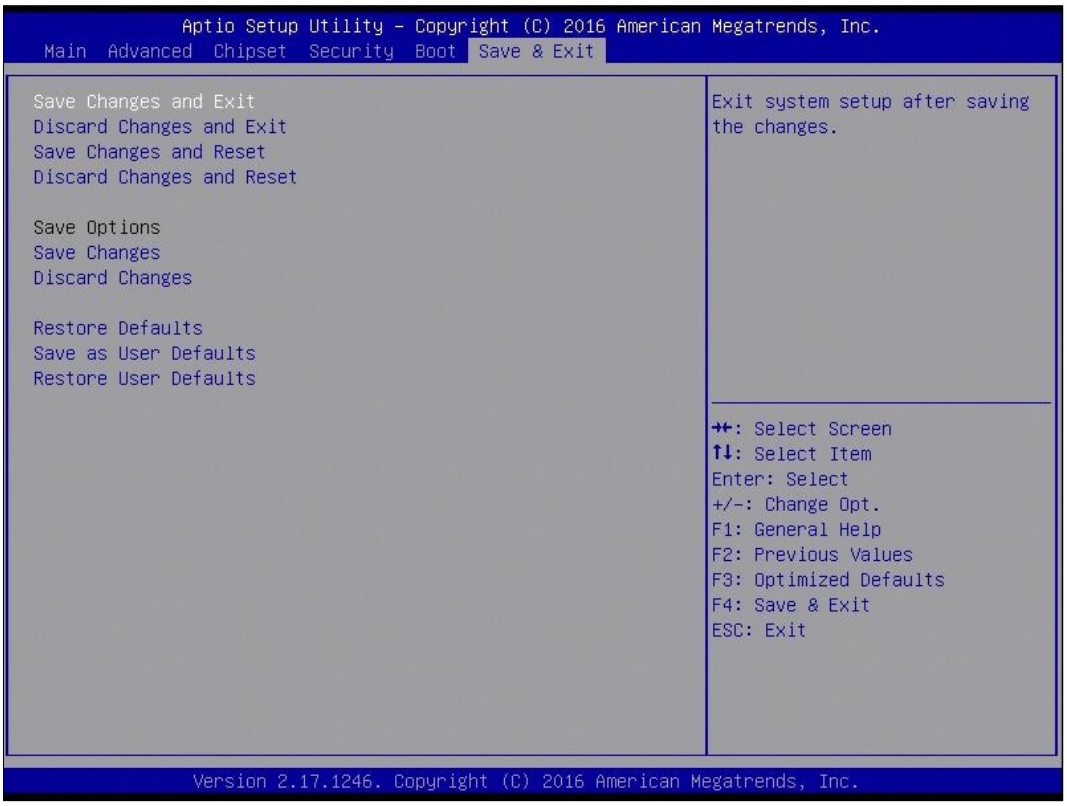

| BIOS Setting | Description |
|---|---|
| Administrator Password | Sets an administrator password for the setup utility. |
| User Password | Sets a user password. |

## 4.7 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 as On or Off. |
| 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. |
| Boot Option Priorities | Sets the system boot order.<br>Options:<br>• Hard Disk<br>• CD/DVD<br>• USB - Hard Disk, CD/DVD, Key, Floppy, Lan<br>• Network<br>• Disabled |
| Hard Disk Drive BBS Priorities | Specificies the priority sequence of the bood devices from available hard disk drives. |

## 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 |
|---|---|
| 0x00000070-0x00000077 | System CMOS/real time clock |
| 0x00000070-0x00000077 | Motherboard resources |
| 0x00000A00-0x00000A0F | Motherboard resources |
| 0x00000290-0x0000029F | Motherboard resources |
| 0x00000A20-0x00000A2F | Motherboard resources |
| 0x00000A30-0x00000A3F | Motherboard resources |
| 0x0000E070-0x0000E077 | Intel(R) Pentium(R) processor N- and J-series / Intel(R) Celeron(R) processor N- and J-series AHCI - 0F23 |
| 0x0000E060-0x0000E063 | Intel(R) Pentium(R) processor N- and J-series / Intel(R) Celeron(R) processor N- and J-series AHCI - 0F23 |
| 0x0000E050-0x0000E057 | Intel(R) Pentium(R) processor N- and J-series / Intel(R) Celeron(R) processor N- and J-series AHCI - 0F23 |
| 0x0000E040-0x0000E043 | Intel(R) Pentium(R) processor N- and J-series / Intel(R) Celeron(R) processor N- and J-series AHCI - 0F23 |
| 0x0000E020-0x0000E03F | Intel(R) Pentium(R) processor N- and J-series / Intel(R) Celeron(R) processor N- and J-series AHCI - 0F23 |
| 0x0000D000-0x0000D01F | Ethernet Controller |
| 0x0000D000-0x0000D01F | Intel(R) Pentium(R) processor N- and J-series / Intel(R) Celeron(R) processor N- and J-series PCI Express - Root Port 1 - 0F48 |
| 0x0000002E-0x0000002F | Motherboard resources |
| 0x0000004E-0x0000004F | Motherboard resources |
| 0x00000061-0x00000061 | Motherboard resources |
| 0x00000063-0x00000063 | Motherboard resources |
| 0x00000065-0x00000065 | Motherboard resources |
| 0x00000067-0x00000067 | Motherboard resources |
| 0x00000080-0x0000008F | Motherboard resources |
| 0x00000092-0x00000092 | Motherboard resources |
| 0x000000B2-0x000000B3 | Motherboard resources |
| 0x00000680-0x0000069F | Motherboard resources |
| 0x00000400-0x0000047F | Motherboard resources |

| Address | Device Description |
|---|---|
| 0x00000500-0x000005FE | Motherboard resources |
| 0x00000600-0x0000061F | Motherboard resources |
| 0x0000E080-0x0000E087 | Intel(R) Atom(TM) Processor E3800 Series/Intel(R) Celeron(R) Processor N2920/J1900 |
| 0x000003B0-0x000003BB | Intel(R) Atom(TM) Processor E3800 Series/Intel(R) Celeron(R) Processor N2920/J1900 |
| 0x000003C0-0x000003DF | Intel(R) Atom(TM) Processor E3800 Series/Intel(R) Celeron(R) Processor N2920/J1900 |
| 0x0000E000-0x0000E01F | Intel(R) Pentium(R) processor N- and J-series / Intel(R) Celeron(R) processor N- and J-series Platform Control Unit - SMBus Port - 0F12 |
| 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 |
| 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 |
| 0x00000040-0x00000043 | System timer |
| 0x00000050-0x00000053 | System timer |
| 0x000003F8-0x000003FF | Serial port #1 (COM1) |
| 0x000003E8-0x000003EF | Serial port #2 (COM2) |
| 0x00000000-0x0000006F | PCI bus |
| 0x00000078-0x00000CF7 | PCI bus |
| 0x00000D00-0x0000FFFF | PCI bus |

## 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 3 | Serial Port #2 (COM2) |
| IRQ 4 | Serial Port #1 (COM1) |
| IRQ 8 | High precision event timer |
| IRQ 11 | Ethernet Controller |
| IRQ 11 | Intel(R) Pentium(R) processor N-series and J-series / Intel(R) Celeron(R) processor N- and J-series Platform Control Unit - SMBus Port - 0F12 |
| IRQ 19 | Intel(R) Pentium(R) processor N-series and J-series / Intel(R) Celeron(R) processor N-series and J-series AHCI - 0F23 |
| IRQ 19 | Intel(R) Pentium(R) processor N-series and J-series / Intel(R) Celeron(R) processor N-series and J-series PCI Express - Root Port 4 - 0F4E |
| IRQ 16 | Intel(R) Pentium(R) processor N-series and J-series / Intel(R) Celeron(R) processor N- and J-series PCI Express - Root Port 1 - 0F48 |
| IRQ 17 | Intel(R) Pentium(R) processor N-series and J-series / Intel(R) Celeron(R) processor N- and J-series PCI Express - Root Port 2 - 0F4A |
| IRQ 18 | Intel(R) Pentium(R) processor N-series and J-series / Intel(R) Celeron(R) processor N-series and J-series PCI Express - Root Port 3 - 0F4C |
| IRQ 22 | High Definition Audio Controller |
| IRQ 81 - IRQ 190 | Microsoft ACPI-Compliant System |
| IRQ 4294967292 | Intel(R) Trusted Execution Engine Interface |
| IRQ 4294967293 | Intel(R) USB 3.0 Extensible Host Controller |
| IRQ 4294967294 | Intel(R) Atom(TM) Processor E3800 Series/Intel(R) Celeron(R) Processor N2920/J1900 |

# C. Digital I/O Sample Code

## 1. The file NCT5523D.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 __NCT5523D_H
#define __NCT5523D_H                    1
//---------------------------------------------------------------------------
#define     NCT5523D_INDEX_PORT     (NCT5523D_BASE)
#define     NCT5523D_DATA_PORT      (NCT5523D_BASE+1)
//---------------------------------------------------------------------------
#define     NCT5523D_REG_LD         0x07
//---------------------------------------------------------------------------
#define NCT5523D_UNLOCK             0x87
#define     NCT5523D_LOCK           0xAA
//---------------------------------------------------------------------------
unsigned int Init_NCT5523D(void);
void Set_NCT5523D_LD( unsigned char);
void Set_NCT5523D_Reg( unsigned char, unsigned char);
unsigned char Get_NCT5523D_Reg( unsigned char);
//---------------------------------------------------------------------------
#endif      //__NCT5523D_H
```

## 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 "NCT5523D.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_NCT5523D();
	if (SIO == 0)
	{
		printf("Can not detect Nuvoton NCT5523D, program abort.\n");
		return(1);
	}

	Dio5Initial();

	//for GPIO20..27
	Dio5SetDirection(0x0F);	//GP20..23 = input, GP24..27=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;

	ucBuf = Get_NCT5523D_Reg(0x1C);
	ucBuf &= ~0x02;
	Set_NCT5523D_Reg(0x1C, ucBuf);

	Set_NCT5523D_LD(0x07);                          //switch to logic device 7
	//enable the GP2 group
	ucBuf = Get_NCT5523D_Reg(0x30);
	ucBuf |= 0x04;
	Set_NCT5523D_Reg(0x30, ucBuf);
}
//---------------------------------------------------------------------------
void Dio5SetOutput(unsigned char NewData)
{
	Set_NCT5523D_LD(0x07);                          //switch to logic device 7
	Set_NCT5523D_Reg(0xE1, NewData);
}
//---------------------------------------------------------------------------
unsigned char Dio5GetInput(void)
{
	unsigned char result;

	Set_NCT5523D_LD(0x07);                          //switch to logic device 7
	result = Get_NCT5523D_Reg(0xE1);
	return (result);
}
//---------------------------------------------------------------------------
void Dio5SetDirection(unsigned char NewData)
{
	//NewData : 1 for input, 0 for output
	Set_NCT5523D_LD(0x07);                          //switch to logic device 7
	Set_NCT5523D_Reg(0xE8, NewData);
}
//---------------------------------------------------------------------------
unsigned char Dio5GetDirection(void)
{
	unsigned char result;

	Set_NCT5523D_LD(0x07);                          //switch to logic device 7
	result = Get_NCT5523D_Reg(0xE8);
	return (result);
}
//---------------------------------------------------------------------------
```

## 3. The file NCT5523D.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 "NCT5523D.H"
#include <dos.h>
//---------------------------------------------------------------------------
unsigned int NCT5523D_BASE;
void Unlock_NCT5523D (void);
void Lock_NCT5523D (void);
//---------------------------------------------------------------------------
unsigned int Init_NCT5523D(void)
{
	unsigned int result;
	unsigned char ucDid;

	NCT5523D_BASE = 0x4E;
	result = NCT5523D_BASE;

	ucDid = Get_NCT5523D_Reg(0x20);
	if (ucDid == 0xC4)                              //NCT5523D??
	{	goto Init_Finish;}

	NCT5523D_BASE = 0x2E;
	result = NCT5523D_BASE;

	ucDid = Get_NCT5523D_Reg(0x20);
	if (ucDid == 0xC4)                              //NCT5523D??
	{	goto Init_Finish;}

	NCT5523D_BASE = 0x00;
	result = NCT5523D_BASE;

Init_Finish:
	return (result);
}
//---------------------------------------------------------------------------
void Unlock_NCT5523D (void)
{
	outportb(NCT5523D_INDEX_PORT, NCT5523D_UNLOCK);
	outportb(NCT5523D_INDEX_PORT, NCT5523D_UNLOCK);
}
//---------------------------------------------------------------------------
void Lock_NCT5523D (void)
{
	outportb(NCT5523D_INDEX_PORT, NCT5523D_LOCK);
}
//---------------------------------------------------------------------------
```

```
void Set_NCT5523D_LD( unsigned char LD)
{
	Unlock_NCT5523D();
	outportb(NCT5523D_INDEX_PORT, NCT5523D_REG_LD);
	outportb(NCT5523D_DATA_PORT, LD);
	Lock_NCT5523D();
}
//---------------------------------------------------------------------------
void Set_NCT5523D_Reg( unsigned char REG, unsigned char DATA)
{
	Unlock_NCT5523D();
	outportb(NCT5523D_INDEX_PORT, REG);
	outportb(NCT5523D_DATA_PORT, DATA);
	Lock_NCT5523D();
}
//---------------------------------------------------------------------------
unsigned char Get_NCT5523D_Reg(unsigned char REG)
{
	unsigned char Result;
	Unlock_NCT5523D();
	outportb(NCT5523D_INDEX_PORT, REG);
	Result = inportb(NCT5523D_DATA_PORT);
	Lock_NCT5523D();
	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 NCT5523D.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 __NCT5523D_H
#define __NCT5523D_H                    1
//-----------------------------------------------------------------------------
#define     NCT5523D_INDEX_PORT     (NCT5523D_BASE)
#define     NCT5523D_DATA_PORT      (NCT5523D_BASE+1)
//-----------------------------------------------------------------------------
#define     NCT5523D_REG_LD         0x07
//-----------------------------------------------------------------------------
#define NCT5523D_UNLOCK             0x87
#define     NCT5523D_LOCK           0xAA
//-----------------------------------------------------------------------------
unsigned int Init_NCT5523D(void);
void Set_NCT5523D_LD( unsigned char);
void Set_NCT5523D_Reg( unsigned char, unsigned char);
unsigned char Get_NCT5523D_Reg( unsigned char);
//-----------------------------------------------------------------------------
#endif      //__NCT5523D_H
```

## 2. 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 "NCT5523D.H"
//---------------------------------------------------------------------------
int main (void);

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

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

	SIO = Init_NCT5523D();
	if (SIO == 0)
	{
		printf("Can not detect Nuvoton NCT5523D, program abort.\n");
		return(1);
	}

	WDTInitial();

	WDTEnable(10);

	WDTDisable();

	return 0;
}
//---------------------------------------------------------------------------
void WDTInitial(void)
{
	unsigned char bBuf;
	Set_NCT5523D_LD(0x08);                          //switch to logic device 8
	bBuf = Get_NCT5523D_Reg(0x30);
	bBuf &= (~0x01);
	Set_NCT5523D_Reg(0x30, bBuf);                   //Enable WDTO
}
//---------------------------------------------------------------------------
```

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

	Set_NCT5523D_LD(0x08);				//switch to logic device 8
	Set_NCT5523D_Reg(0x30, 0x01);			//enable timer

	bBuf = Get_NCT5523D_Reg(0xF0);
	bBuf &= (~0x08);
	Set_NCT5523D_Reg(0xF0, bBuf);			//count mode is second

	Set_NCT5523D_Reg(0xF1, NewInterval);		//set timer
}
//-------------------------------------------------------------------------
void WDTDisable(void)
{
	Set_NCT5523D_LD(0x08);				//switch to logic device 8
	Set_NCT5523D_Reg(0xF1, 0x00);			//clear watchdog timer
	Set_NCT5523D_Reg(0x30, 0x00);			//watchdog disabled
}
//-------------------------------------------------------------------------
```

## 3. Sample Code: The file NCT5523D.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 "NCT5523D.H"
#include <dos.h>
//---------------------------------------------------------------------------
unsigned int NCT5523D_BASE;
void Unlock_NCT5523D (void);
void Lock_NCT5523D (void);
//---------------------------------------------------------------------------
unsigned int Init_NCT5523D(void)
{
        unsigned int result;
        unsigned char ucDid;

        NCT5523D_BASE = 0x4E;
        result = NCT5523D_BASE;

        ucDid = Get_NCT5523D_Reg(0x20);
        if (ucDid == 0xC4)                              //NCT5523D??
        {       goto Init_Finish;}

        NCT5523D_BASE = 0x2E;
        result = NCT5523D_BASE;

        ucDid = Get_NCT5523D_Reg(0x20);
        if (ucDid == 0xC4)                              //NCT5523D??
        {       goto Init_Finish;}

        NCT5523D_BASE = 0x00;
        result = NCT5523D_BASE;

Init_Finish:
        return (result);
}
//---------------------------------------------------------------------------
void Unlock_NCT5523D (void)
{
        outportb(NCT5523D_INDEX_PORT, NCT5523D_UNLOCK);
        outportb(NCT5523D_INDEX_PORT, NCT5523D_UNLOCK);
}
//---------------------------------------------------------------------------
```

```
void Lock_NCT5523D (void)
{
	outportb(NCT5523D_INDEX_PORT, NCT5523D_LOCK);
}
//----------------------------------------------------------------------
void Set_NCT5523D_LD( unsigned char LD)
{
	Unlock_NCT5523D();
	outportb(NCT5523D_INDEX_PORT, NCT5523D_REG_LD);
	outportb(NCT5523D_DATA_PORT, LD);
	Lock_NCT5523D();
}
//----------------------------------------------------------------------
void Set_NCT5523D_Reg( unsigned char REG, unsigned char DATA)
{
	Unlock_NCT5523D();
	outportb(NCT5523D_INDEX_PORT, REG);
	outportb(NCT5523D_DATA_PORT, DATA);
	Lock_NCT5523D();
}
//----------------------------------------------------------------------
unsigned char Get_NCT5523D_Reg(unsigned char REG)
{
	unsigned char Result;
	Unlock_NCT5523D();
	outportb(NCT5523D_INDEX_PORT, REG);
	Result = inportb(NCT5523D_DATA_PORT);
	Lock_NCT5523D();
	return Result;
}
//----------------------------------------------------------------------
```