ASB200-918

Fanless System with 3.5” Disk-Size SBC System

User’s Manual

Version 1.0
(October 2020)

iBASE

Compliance

CE

This product has passed CE tests for environmental specifications and limits, 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 -20˚C ~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 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

Do not 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

THIS PAGE IS INTENTIONALLY LEFT BLANK.

Chapter 1
General Information

The information provided in this chapter includes:

- Features
- Packing List
- Optional Accessories
- Specifications
- Product View
- Dimensions

1.1 Introduction

The ASB200-918 compact fanless system supports the AMD Ryzen™ Embedded V1000/R1000 SoC featuring Vega GPU for stunning 4K Ultra High-Definition resolution and driving two HDMI (2.0a) graphics display. It supports two SO-DIMM sockets supporting fast data transfers with up to 32GB DDR4-2400 memory and ECC for data integrity and flexible I/O connectivity and expansion ports. The system comes with an operating temperature range of 0°C to 45°C.

1.2 Features

- Fanless system with IBASE IB918 3.5'' Disk-Size SBC
- AMD Ryzen™ Embedded V1000 /R1000 on board
- 2x DDR4 SO-DIMM, Max. 32GB, supports ECC
- Supports 2x M.2 sockets (E-Key / M-Key, NVMe)
- 2x HDMI(2.0a), 2x Intel® Gigabit LAN
- 4 x COM ports, External GPIO
- 4 x USB3.1, 1x USB2.0
- 12V~24V DC-in Power Input
- Operating temperature 0°C~45°C (32°F~113°F)

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 purchased the product.

- ASB200-918……x 1
- 3-Pin Terminal Block (for power)……x 1
- 10-Pin Terminal Block (for GPIO connector)……x 1
- Side Brackets……..x 2
- Screws for Side Brackets……..x 4
- Screws for M key and E key M.2 ……x 2
- Adapter for ASB200-918……x 1
- Power cord……x 1

1.4 Optional Accessories

IBASE provide optional accessories as follows. Please contact us or your dealer if you need any.

- WIFI / BT module Kit
- DC Power Jack
- Power Adapter and Power Cord (for DC power jack)

1.5 Specifications

Product Name	ASB200-918-1605	ASB200-918-1202	ASB200-918-1606G	ASB200-918-1505G
Motherboard	IB918F-1605	IB918F-1202	IB918F-1606G	IB918F-1505G
MB Form Factor	3.5” disk-size SBC			
System				
Operating System	• Windows 10 • Linux Ubuntu / Fedora			
AMD Embedded Ryzen™ SoC*	V1605B	V1202B	R1606G	R1505G
Memory	2x 8GB DDR4-2666 SO-DIMM Expandable to 32 GB, ECC supported			
Storage	1 x 2.5” HDD or SSD drive bay			
Graphics	AMD V1000 series APU built-in Radeon Vega graphics (V1605B Support Vega 8)			
Power Supply	90W power adaptor (Optional)			
Chassis	Aluminum & steel, black			
Mounting	Desktop mount or Wall mount bracket			
Dimensions (W x H x D)	240mm (W) x 162mm (D) x 42mm (H)			
Weight	1.8 kg (3.97 lb)			
Certificate	CE / LVD / FCC Class B			

AMD Embedded Ryzen™ SoC Supported:

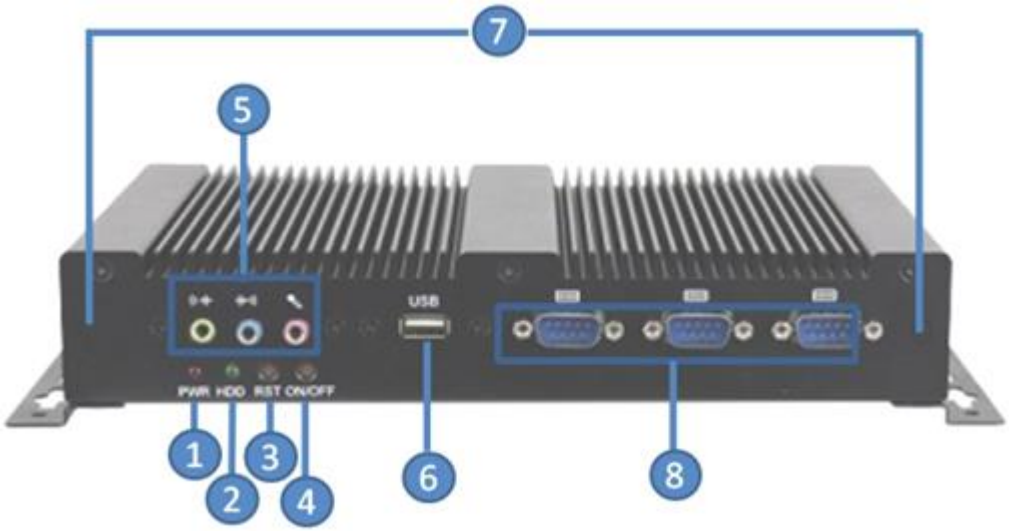
- AMD V1605B /QC/**2.0GHz**~3.6GHz/2MB L2 cache/12W~25W TDP [C018RYZEG**200**10100P] For IB918F-1605
- AMD V1202B /DC/**2.3GHz**~3.3GHz/1MB L2 cache/12W~25W TDP [C018RYZEG**230**10100P] For IB988F-1202
- AMD R1606G /DC/2.6GHz~3.5GHz/1MB L2 cache/12W~25W TDP [C018RYZEG20010100P] For IB988F-1606G
- AMD R1505G /DC/**2GHz**~3.3GHz/1MB L2 cache/12W~25W TDP [C018RYZEG**230**10100P] For IB988F-1505G

I/O Interface	
Front Panel I/O	- 1 x Power button/RST/HDD/On/Off - 2 x Antenna holes - 1 x Audio Jack for Line-out/Line-in/Mic-in - 3 x DB9 for COM#2~#4 (RS232 only) - 1 x USB port 2.0/3.0
Rear Panel I/O	- 1 x 3-pin terminal block connector for external digital I/O - 1x GPIO (4-in/4-out) - 1 x DB9 for COM#1 (RS232/422/485), - 2 x RJ-45 GbE Connector - 2 x HDMI - 2 x Antenna holes
Storage	- 1 x M.2 128GB TLC Toshiba - 1 x 2.5'' SSD;2.5" SATA3 64G TLC 7mm [PHISON SSBP064GTB3C0-S11] (Default)
Environmental	
Operating Temperature	0°C~45°C (32°F~113°F)
Storage Temperature	-20°C~80°C (-4°F~176°F)
Humidity	5%~90%@45°C (non-condensing)
Vibration	Operating : 3Grms / 5~500Hz
Shock	Operating : 20G / 11ms Non-operating : 40G / 11ms

All specifications are subject to change without prior notice.

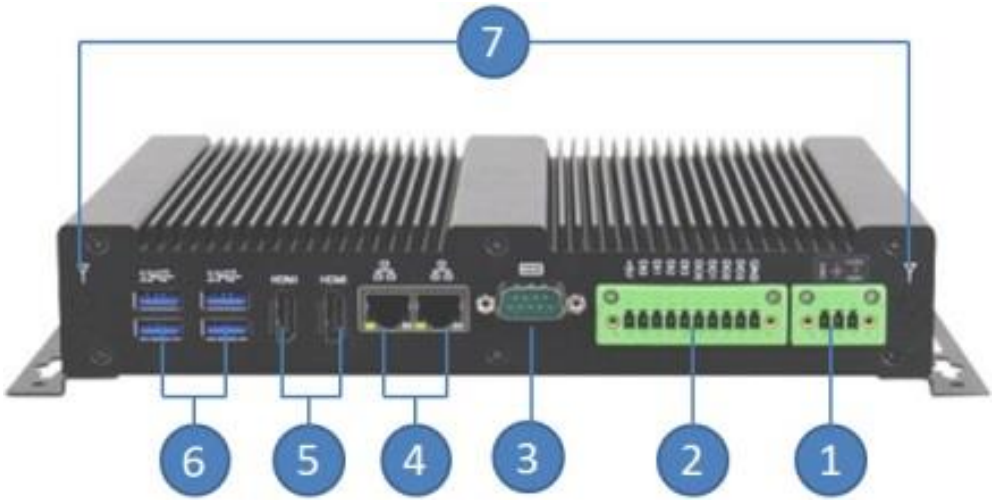
1.6 Product View

Front View



No.	Name	No.	Name
1	PWR LED Indicator	5	Audio Jacks (From left to right: Mic-In, Line-In, Lin-Out)
2	HDD LED Indicator	6	USB 3.0 (ASB200-918-1605 & ASB200-918-1202 only) USB 2.0 (ASB200-918-1606 & ASB200-918-1505 only)
3	Power Button	7	COM2 ~ COM4 Ports (COM2 ~ COM4: RS-232)
4	Reset Button	8	2 Antenna Holes

Rear View

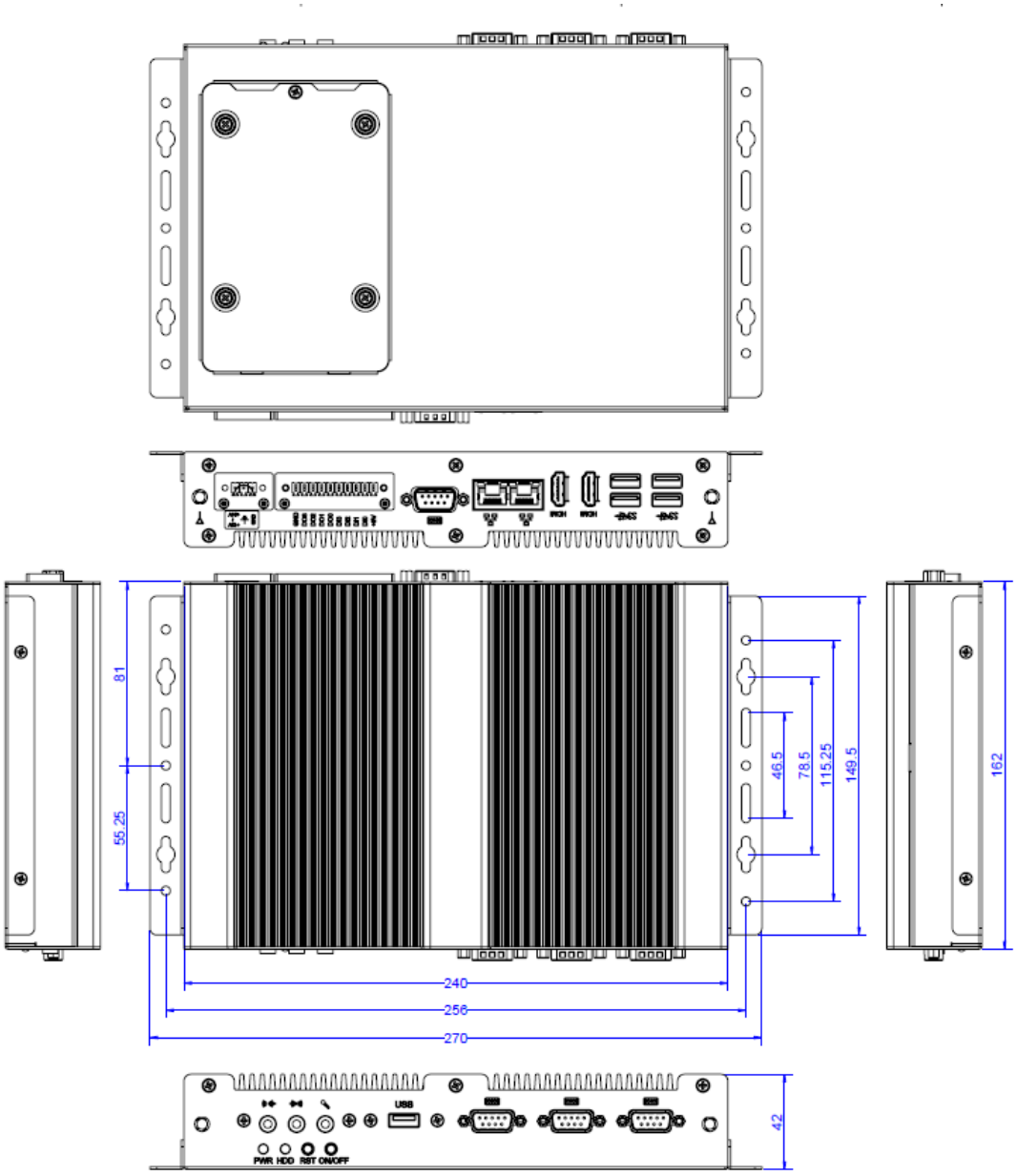


No.	Name	No.	Name
1	DC-In Power Connector	5	HDMI Port
2	Digital I/O Connector (4-In / 4-Out)	6	USB 3.0 Ports
3	COM1 (COM1: RS-232/422/485)	7	2 Antenna Holes
4	GbE LAN Ports		

Oblique View

1.7 Dimensions

Unit: mm



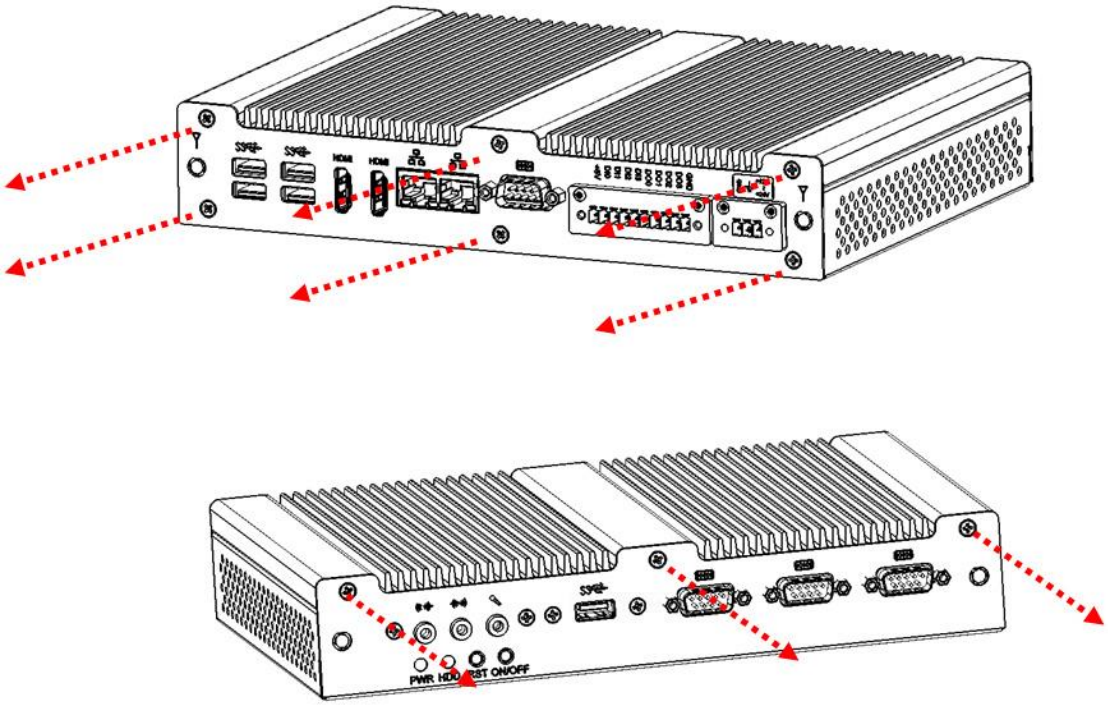
Chapter 2
Hardware Configuration

The information provided in this chapter includes:

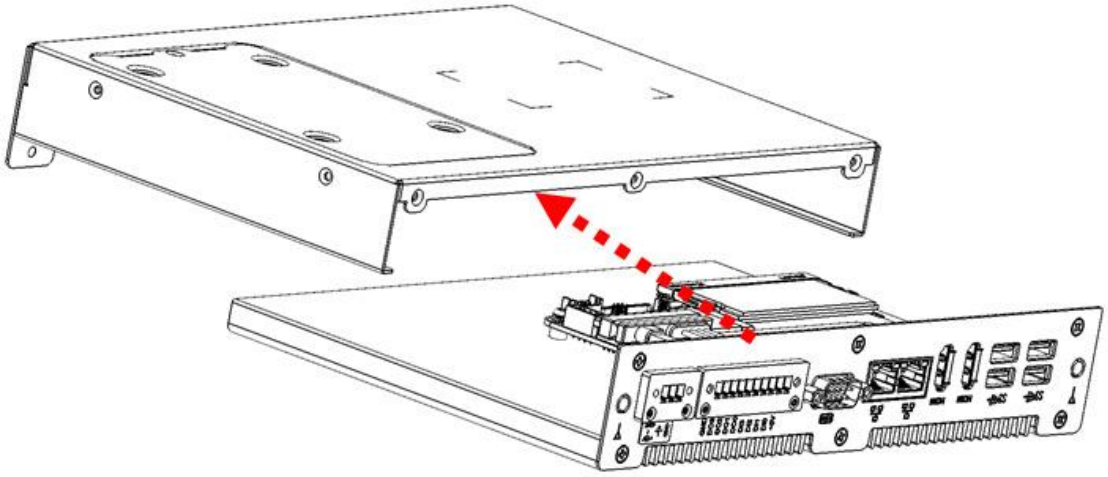
- Installations
- Information and locations of connectors

2.1 Installations

Before installations, you need to remove the bottom chassis cover by loosening the screws shown below.

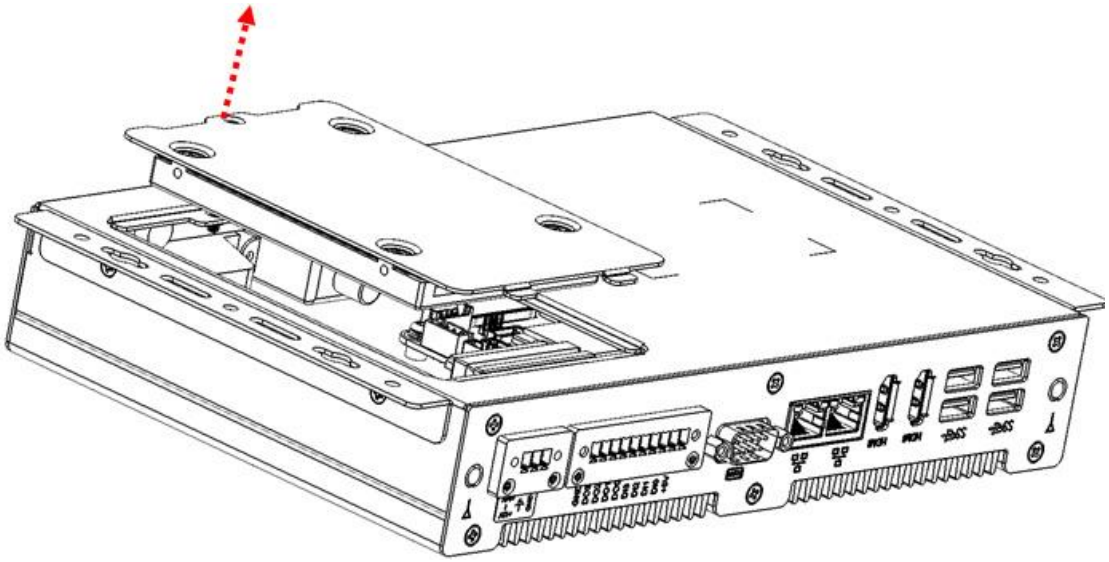


After removing the above screws, remove the bottom chassis cover.

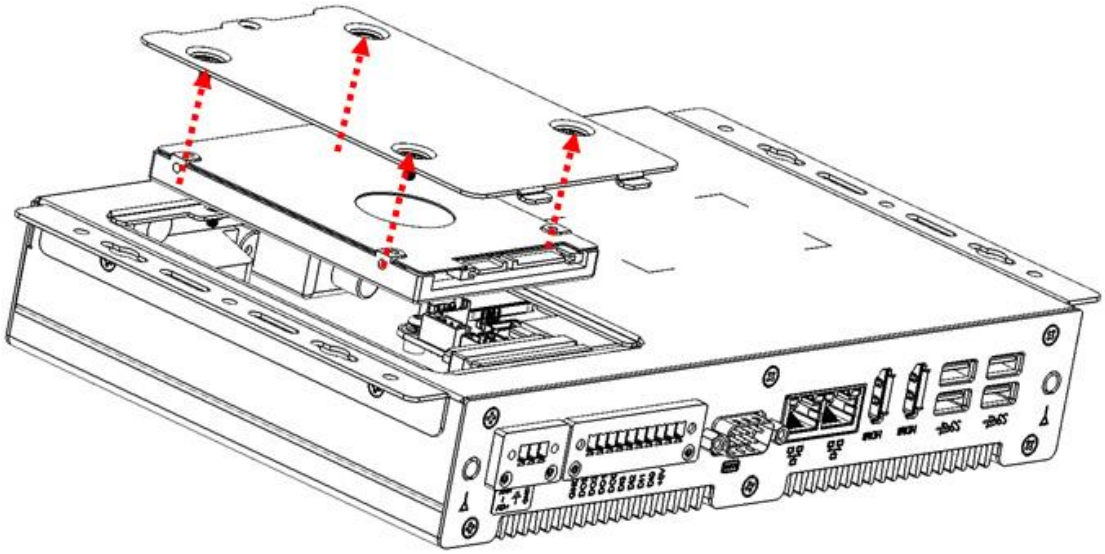


2.1.1 HDD Installation

If you need to install or replace an SSD or a HDD, remove the single screw shown below to remove the tray, and then loosen the four screws. Attach the SSD / HDD and tighten these screws.



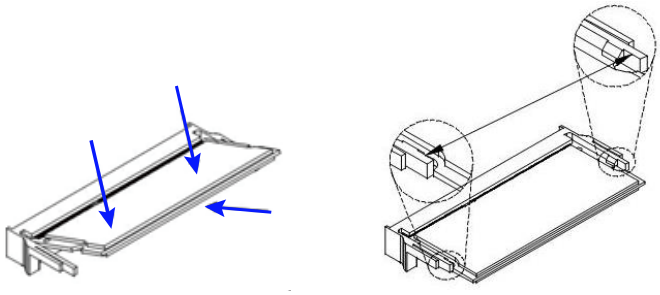
Remove the single screw holding the tray.



Remove the four screws holding the HDD/SSD.

2.1.2 Memory Installation

If you need to install or replace a memory module, perform the following steps:



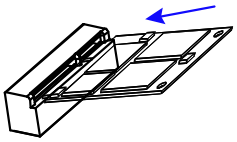
1. Locate the memory slot on the board.
2. Align the key of the memory module with that on the memory slot and insert the module slantwise.
3. Gently push the module down in an upright position until the clips of the slot close to hold the module in place when the module touches the bottom of the slot.

To remove the module, press the clips outwards with both hands, and the module will pop-up.

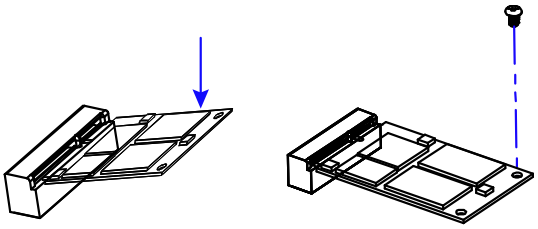
2.1.3 Mini-PCIe Card Installation

If you need to use a mini-PCIe card for expansion slots, follow the instructions below for installation after you remove the device bottom chassis cover.

1. Align the key of the mini-PCIe card to the mini-PCIe interface, and insert the card slantwise.

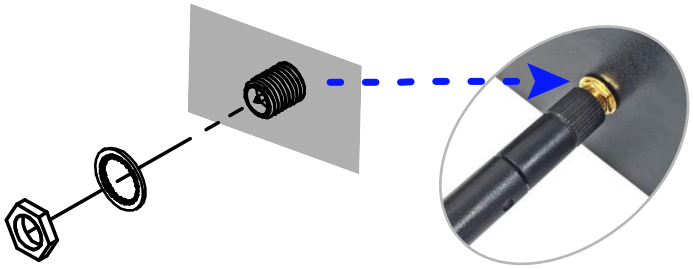


2. Push the mini-PCIe card down, fix it onto the standoff with a screw.



2.1.4 WiFi / 3G / 4G Antenna Installation

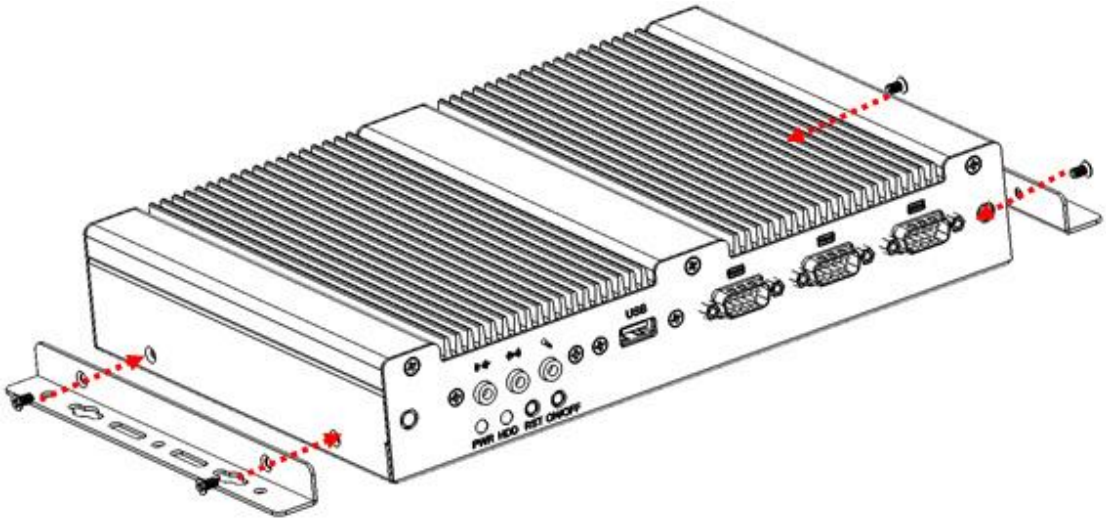
Thread the WiFi / 3G / 4G antenna cable through an antenna hole. Then fasten the antenna as shown below.



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

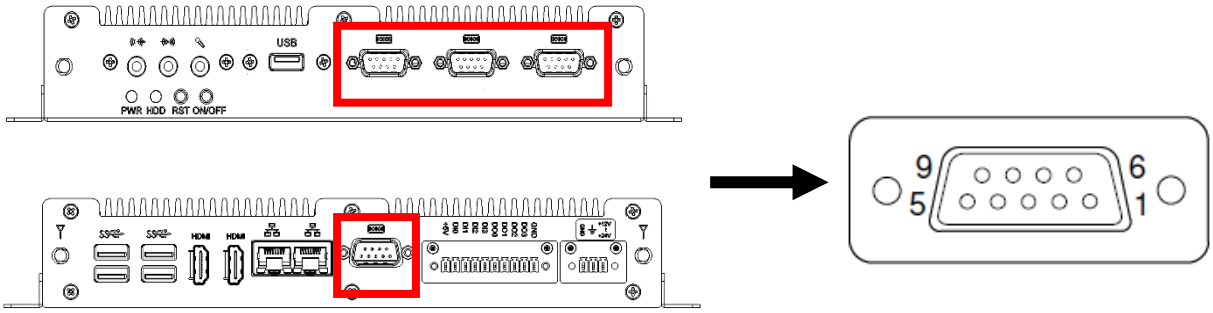
2.1.5 Side Bracket Installation

Install the side bracket by using the four screws supplied with the package.



2.1.6 Pinout for COM Ports, DC Power & Digital I/O Connectors

- **RS232/422/485 COM Port**

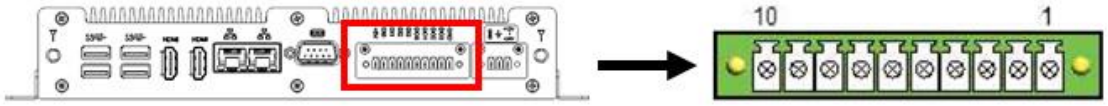


Pin	Signal	Pin	Signal
1	DCD, Data carrier detect	6	DSR, Data set ready
2	RXD, Receive data	7	RTS, Request to send
3	TXD, Transmit data	8	CTS, Clear to send
4	DTR, Data terminal ready	9	RI, Ring indicator
5	Ground		

- **DC Power Input Connector (terminal block)**

Pin	Signal	Pin	Signal
1	Ground	3	+12V ~ +24V
2	Chassis Ground		

- **Digital I/O Connector (terminal block)**



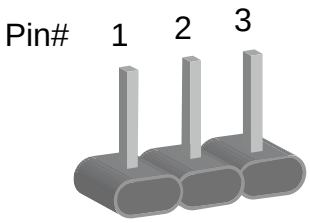
Pin	Signal	Pin	Signal
1	Ground	6	DI3
2	DO3	7	DI2
3	DO2	8	DI1
4	DO1	9	DI0
5	DO0	10	+5V

2.2 Setting the Jumpers

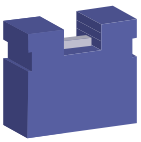
Set up and configure your product by using jumpers for various settings and features according to your needs and applications. Contact your supplier if you have doubts about the best configuration for your use.

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.

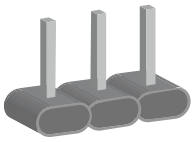
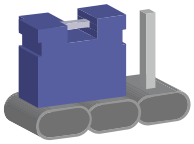
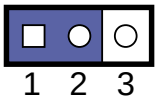
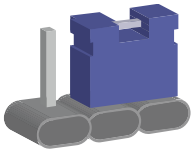
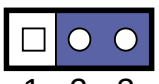


A 3-pin jumper



A jumper cap

Refer to the illustration below to set jumpers.

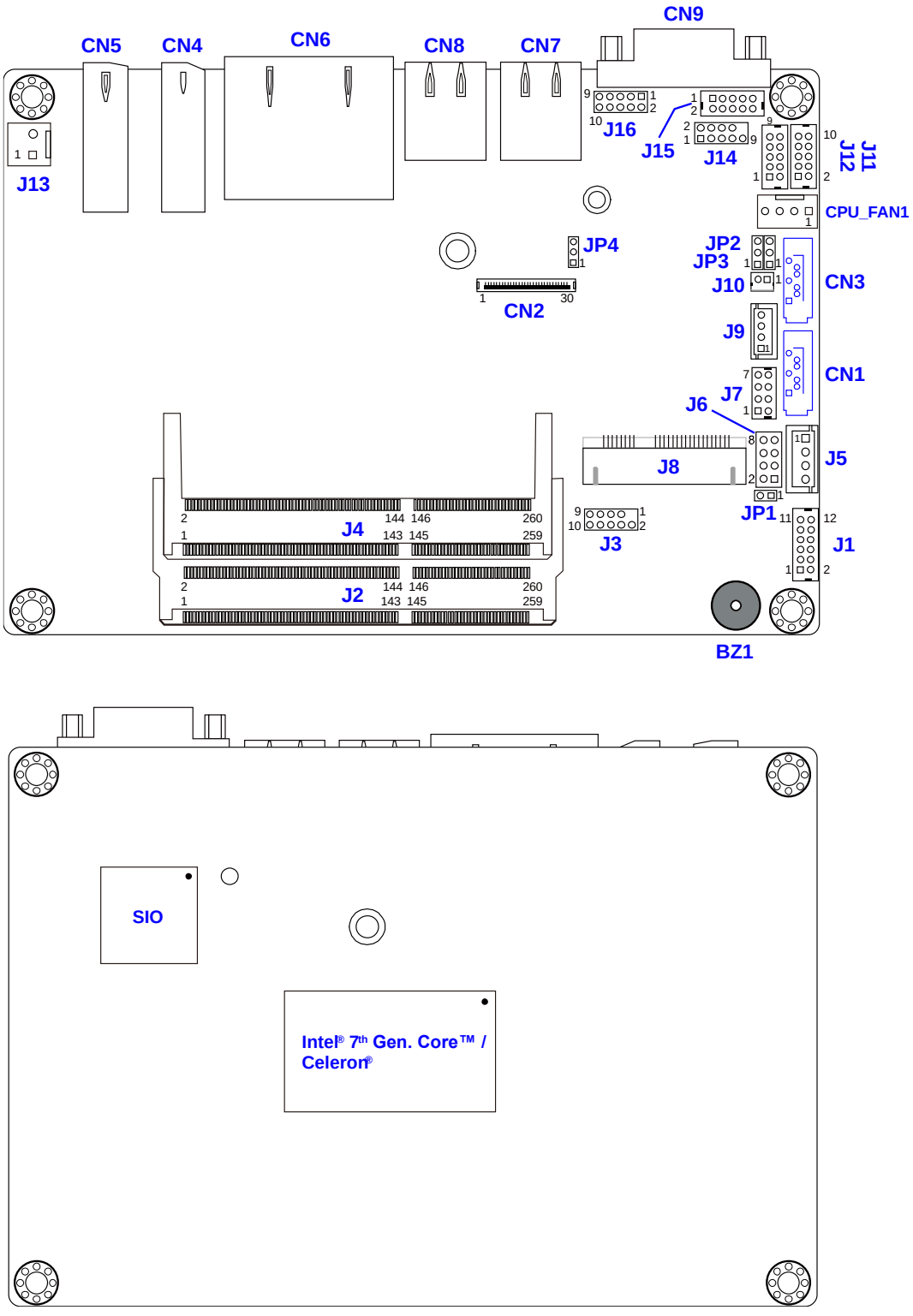
Pin closed	Oblique view	illustration
Open		
1-2		
2-3		

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**.

2.3 Jumper & Connector Locations on Motherboard

Motherboard: IB918

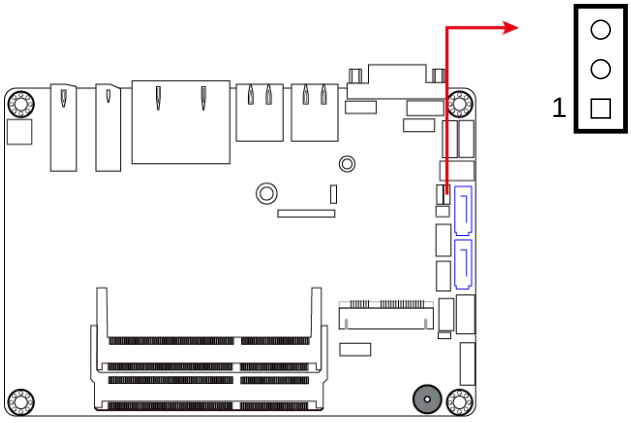


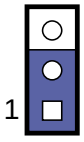
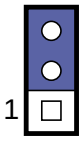
Board diagram of IB918

2.4 Jumpers Quick Reference

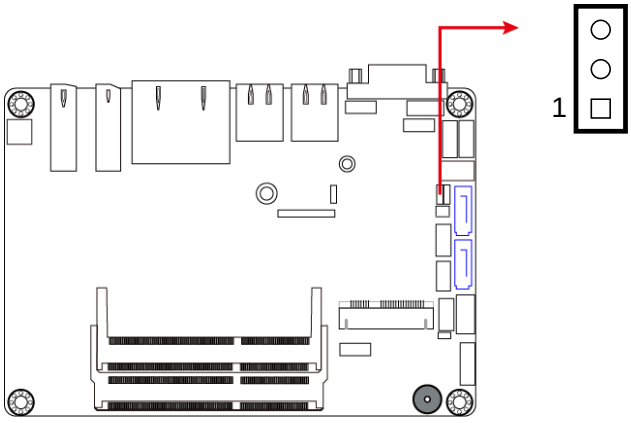
Function	Connector	Page
CMOS Data Clearance	JP2	17
ME Register Clearance	JP3	18
eDP Panel Power Selection	JP4	18
Factory Use Only	JP1	- -

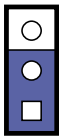
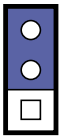
2.4.1 Clear CMOS Data (JP2)



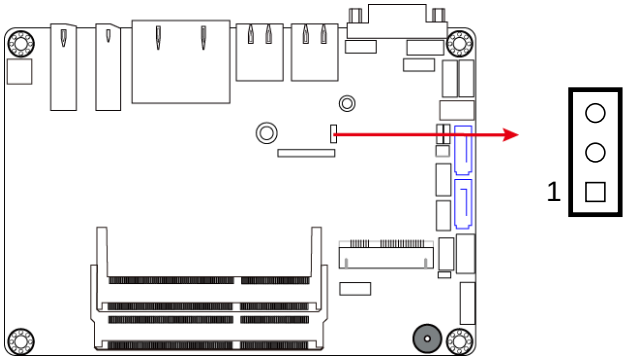
Function	Pin closed	JP2
Normal (default)	1-2	
Clear CMOS	2-3	

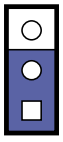
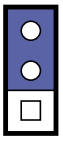
2.4.2 Clear ME Register (JP3)



Function	Pin closed	JP3
Normal (default)	1-2	
Clear ME	2-3	

2.4.3 eDP Panel Power Selection (JP4)



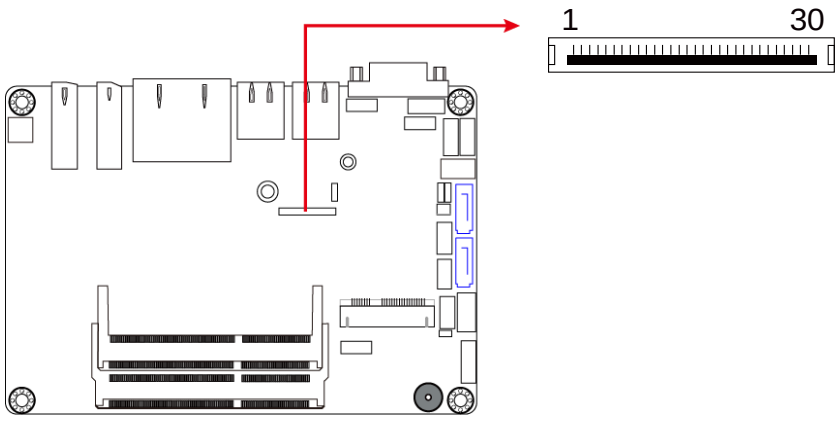
Function	Pin closed	JP4
3.3V (default)	1-2	
5V	2-3	

2.5 Connectors Quick Reference

Function	Connector	Page
eDP Connector	CN2	20
Audio Connector	J1	21
SATA HDD Power Connector	J5	21
Front Panel Setting Connector	J6	22
USB 2.0 Connector	J7	23
Battery Connector	J10	23
DC Power Input Connector	J13	24
Digital I/O Connector	J16	24
COM1 RS-232/422/485 Port [1]	CN9	--
COM2, COM3, COM4 RS-232 Ports [1]	J15, J12, J11	--
SATA III Port	CN1, CN3	--
HDMI 1.4 Port	CN4	--
Display Port	CN5	--
GbE LAN Ports	CN6	--
USB 3.0 Port	CN7, CN8	--
DDR4 SO-DIMM Slot	J2, J4	--
Mini-PCIe / mSATA Slot	J8	--
Factory Use Only	J3, J9, J14	--

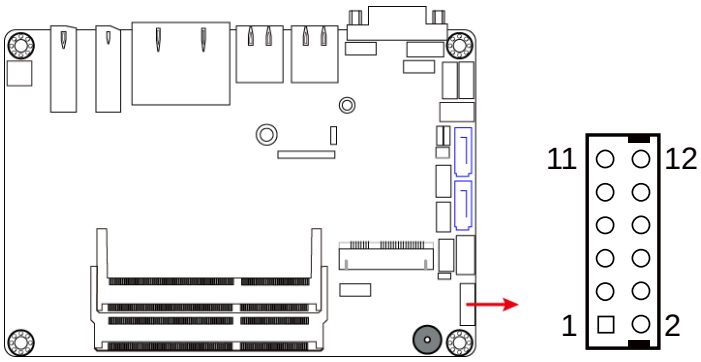
[1]: Refer to ***錯誤! 找不到參照來源。 錯誤! 找不到參照來源。***.

2.5.1 eDP Connector (CN2)



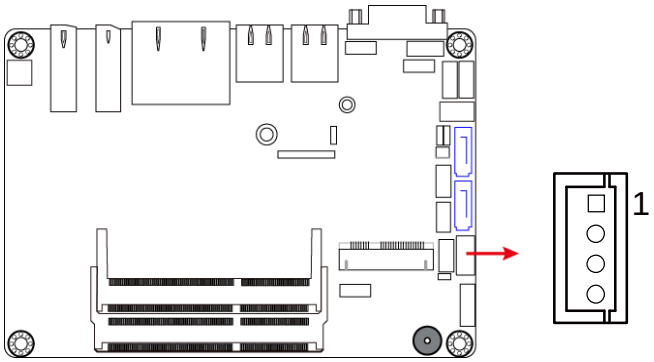
Pin	Signal	Pin	Signal
1	NC	16	Ground
2	BL_Power	17	NC
3	BL_Power	18	Panel_VDD
4	BL_Power	19	Panel_VDD
5	BL_Power	20	Ground
6	NC	21	AUX_N
7	NC	22	AUX_P
8	Brightness	23	Ground
9	Bklt_en	24	TX0_P
10	Ground	25	TX0_N
11	Ground	26	Ground
12	Ground	27	TX1_P
13	Ground	28	TX1_N
14	HPD	29	Ground
15	Ground	30	NC

2.5.2 Audio Connector (J1)



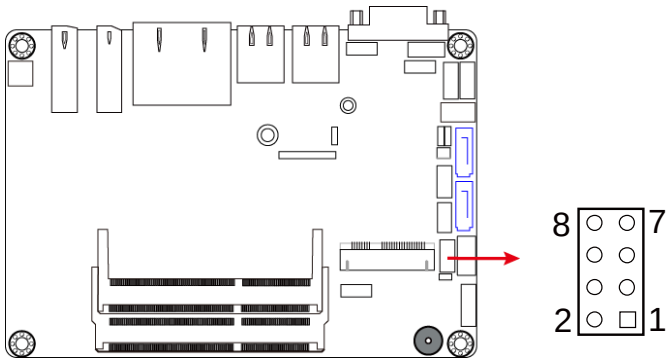
Pin	Signal	Pin	Signal
1	Lineout_L	2	Lineout_R
3	JD_FRONT	4	Ground
5	LINEIN_L	6	Linein_R
7	JD_LINEIN	8	Ground
9	MIC_L	10	MIC-R
11	JD_MIC1	12	Ground

2.5.3 SATA HDD Power Connector (J5)



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

2.5.4 Front Panel Connector (J6)

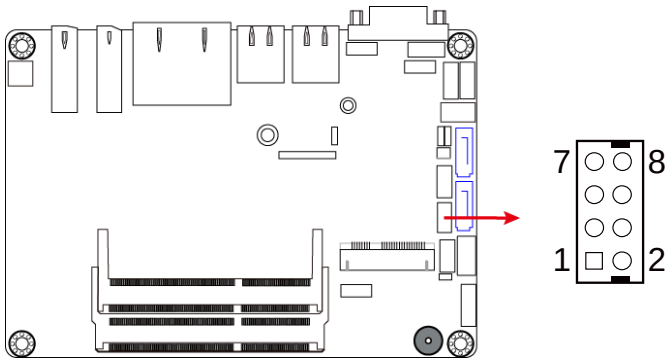


Pin	Signal	Pin	Signal
1	Ground	2	PWR_BTN
3	3.3V	4	HDD Active
5	Ground	6	Reset
7	+5V	8	Ground

J6 is utilized for system indicators to provide light indication of the computer activities and switches to change the computer status. It provides interfaces for the following functions.

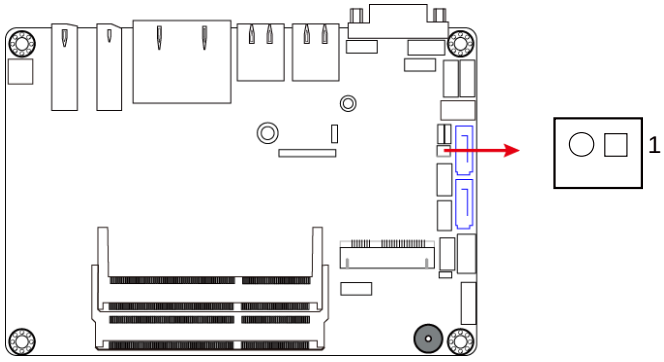
- **ATX Power ON Switch (Pins 1 and 2)**
  The 2 pins make an “ATX Power Supply On/Off Switch” for the system that connects to the power switch on the case. When pressed, the power switch will force the system to power on. When pressed again, it will power off the system.
- **Hard Disk Drive LED Connector (Pins 3 and 4)**
  This connector connects to the hard drive activity LED on control panel. This LED will flash when the HDD is being accessed.
- **Reset Switch (Pins 5 and 6)**
  The reset switch allows you to reset the system without turning the main power switch off and then on again. Orientation is not required when making a connection to this header.
- **Power LED: Pins 7 and 8**
  This connector connects to the system power LED on control panel. This LED will light when the system turns on.

2.5.5 USB 2.0 Connector (J7)



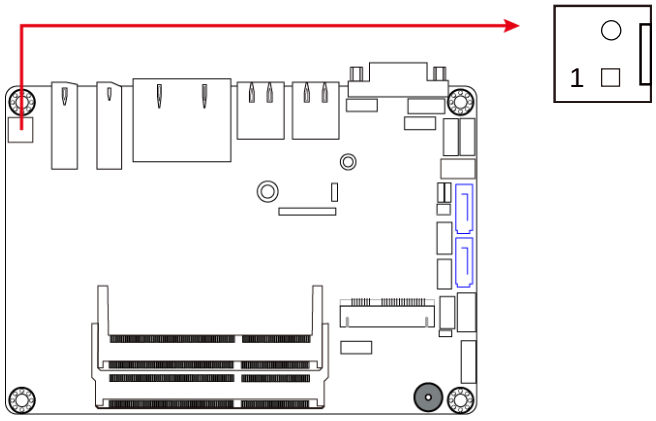
Pin	Signal	Pin	Signal
1	VCC	2	Ground
3	D0-	4	D1+
5	D0+	6	D1-
7	Ground	8	VCC

2.5.6 Battery Connector (J10)



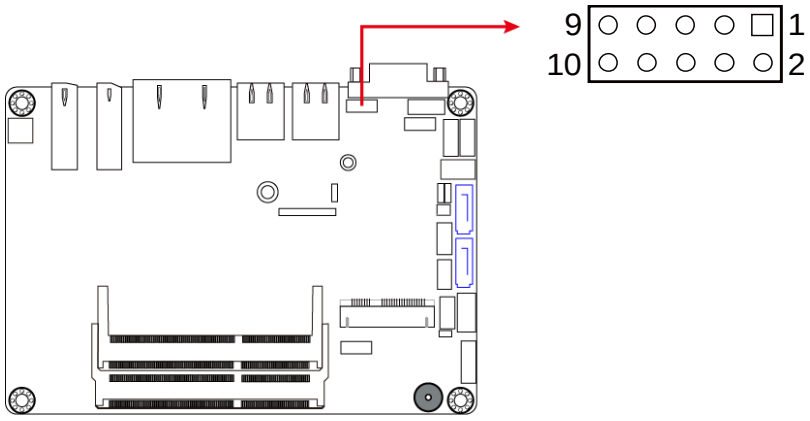
Pin	Signal	Pin	Signal
1	Battery+	2	Ground

2.5.7 DC Power Input Connector (J13)



Pin	Signal	Pin	Signal
1	+9V ~ +24V	2	Ground

2.5.8 Digital I/O Connector (J16)



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

Chapter 3
Driver Installation

The information provided in this chapter includes:

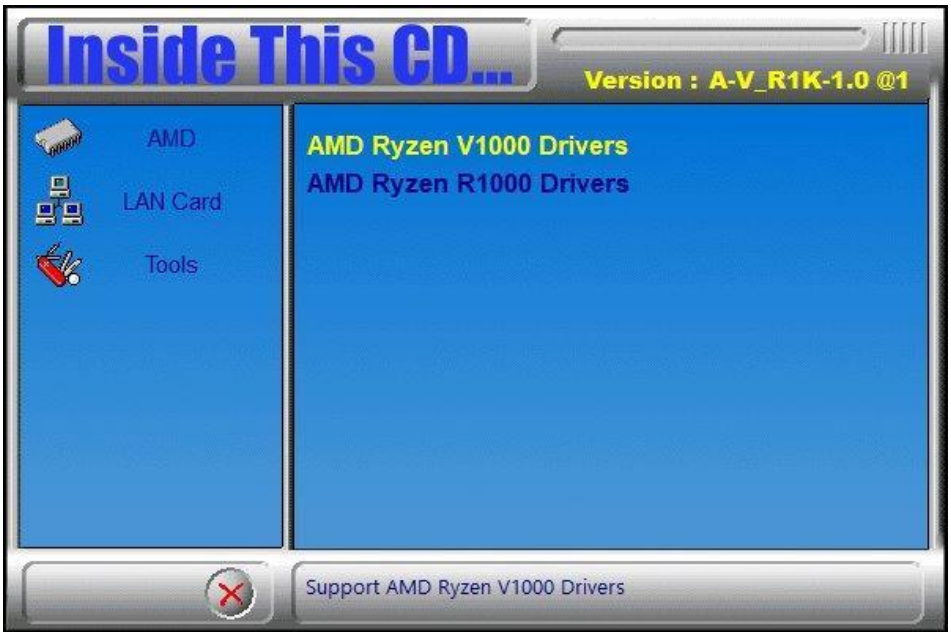
- Introduction
- AMD Ryzen™ V1000 Graphics Drivers
- AMD Ryzen™ R1000 Graphics Drivers
- HD Audio Driver Installation
- LAN Driver Installation
- Observer

3.1 Introduction

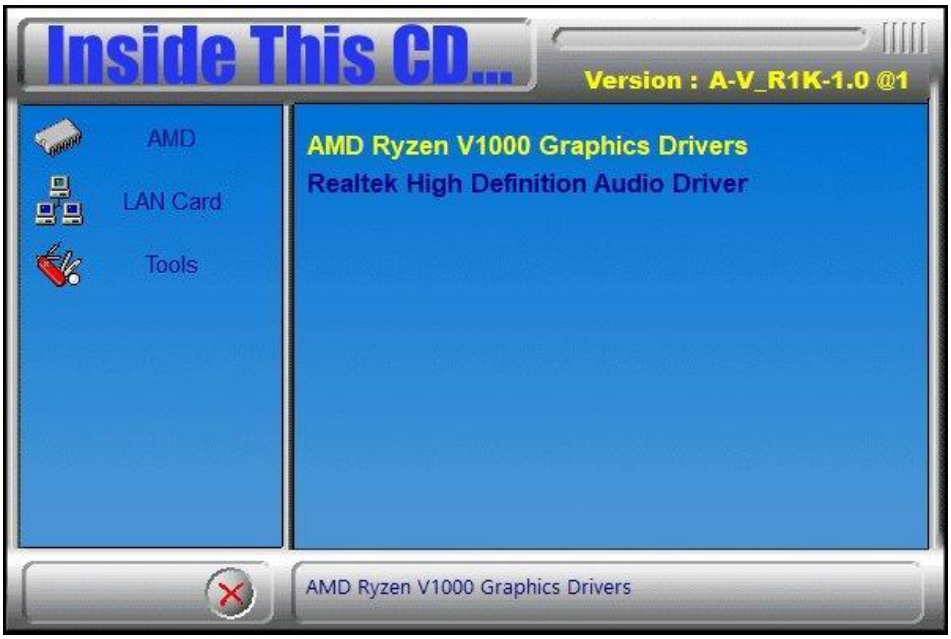
This section describes the installation procedures for software and drivers. The contents of this section include the following:

3.2 AMD Ryzen™ V1000 Graphics Drivers

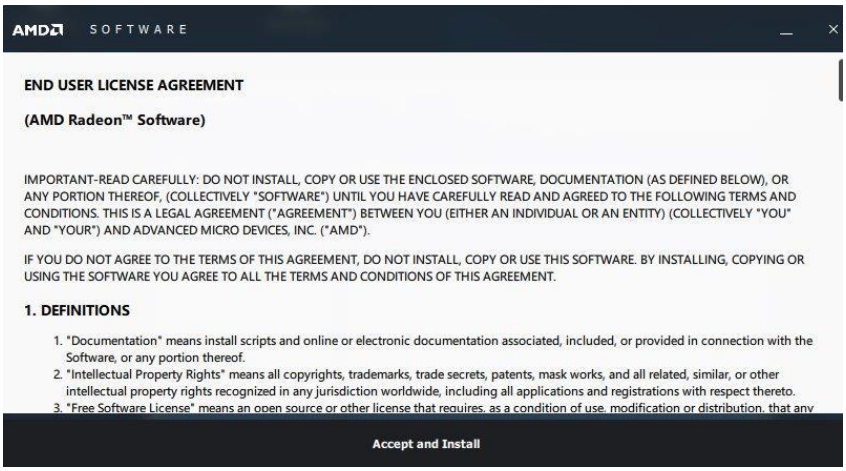
1. Insert the disk enclosed in the package. Click **AMD** on the left pane and then **AMD Ryzen V1000 Drivers** on the right pane.



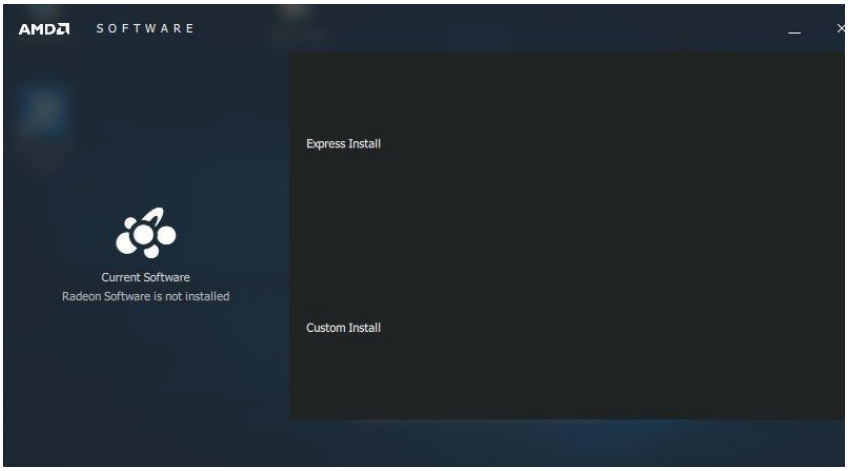
2. Click **AMD Ryzen V1000 Graphics Drivers**.



3. Now click **Accept and Install** to proceed.



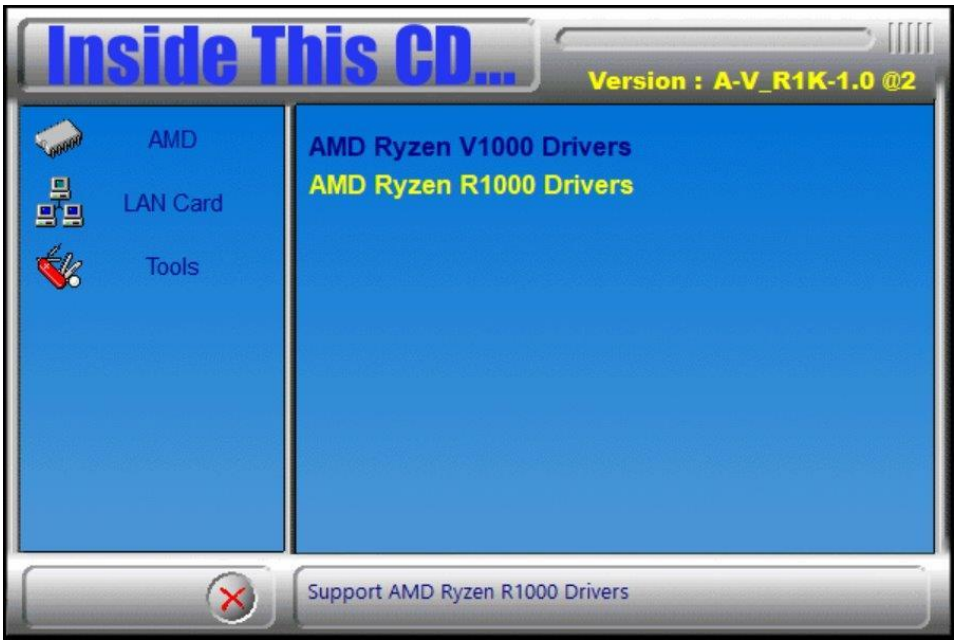
4. Click on either **Express Install** or **Custom Install**.



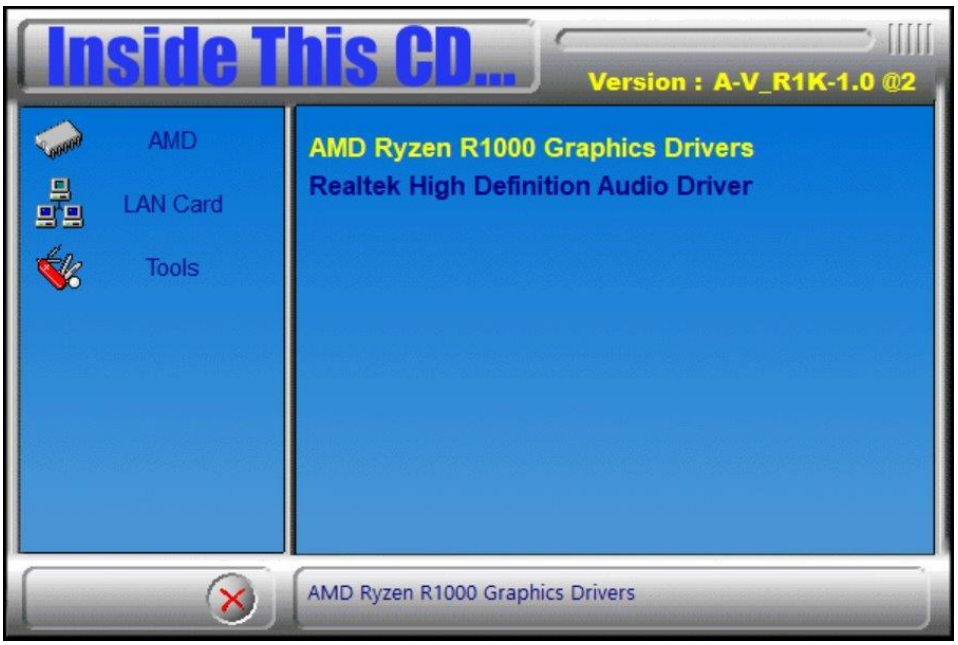
5. When the software driver has been installed, restart the computer for changes to take effect.

3.3 AMD Ryzen™ R1000 Graphics Drivers

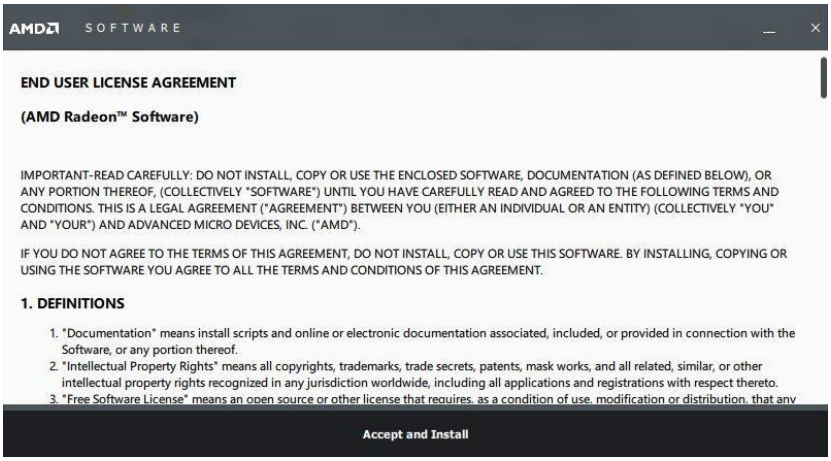
1. Insert the disk enclosed in the package. Click **AMD** on the left pane and then **AMD Ryzen R1000 Drivers** on the right pane.



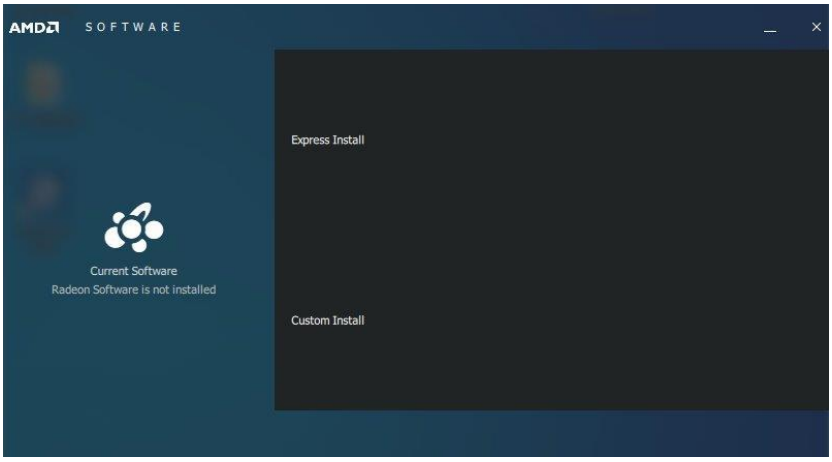
2. Click **AMD Ryzen R1000 Graphics Drivers**.



3. Read the software license agreement and click **Accept and Install** to proceed.



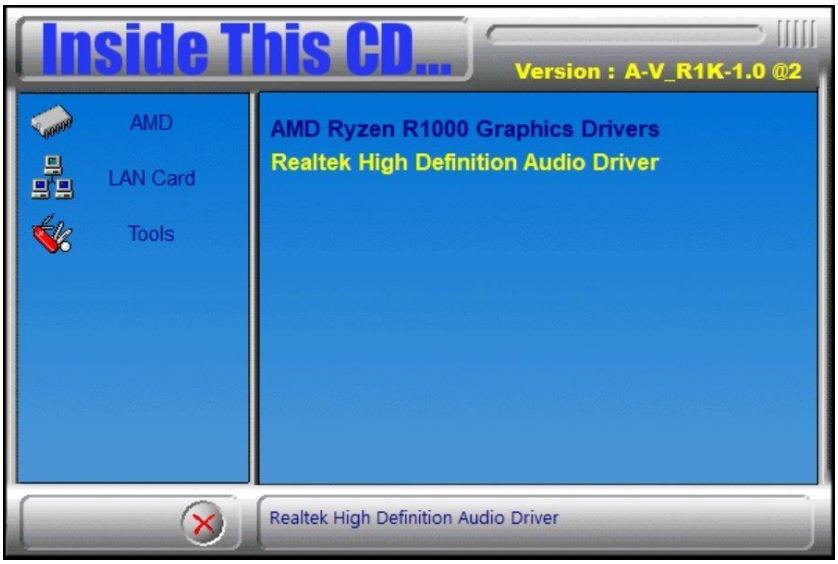
4. Click on either **Express Install** or **Custom Install**.



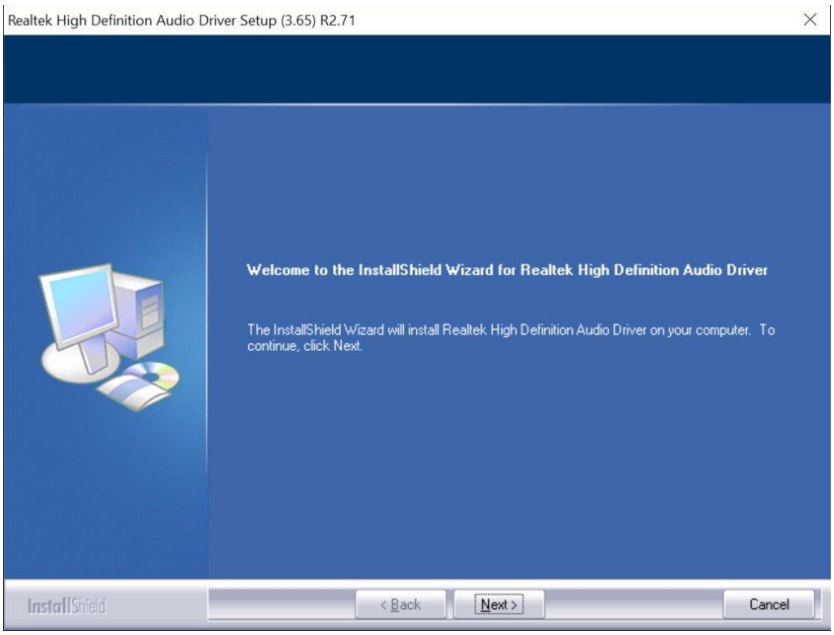
5. When the software driver has been installed, restart the computer for changes to take effect.

3.4 HD Audio Driver Installation

1. Insert the disk enclosed in the package with the board. Click **AMD** on the left pane and then **Realtek High Definition Audio Driver** on the right pane.



2. On the *Welcome* screen of the InstallShield Wizard, click **Next**.

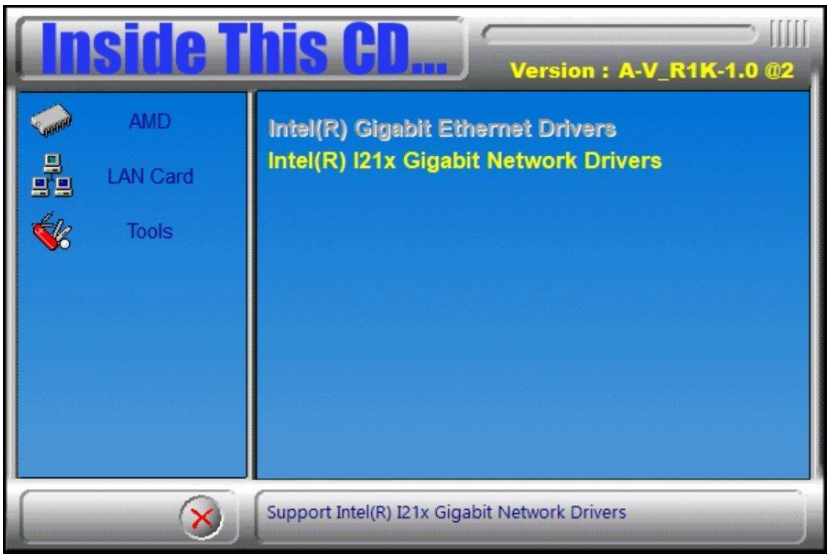


3. Follow the steps accordingly. When Setup has successfully installed the Audio Driver, restart your computer for changes to take effect.

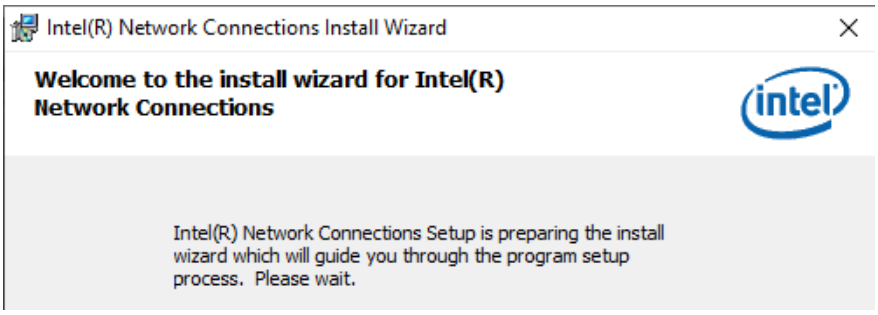
3.5 LAN Driver Installation

1. Insert the disk enclosed in the package. Click **LAN Card** on the left pane and **Intel LAN Controller Drivers** on the right pane.

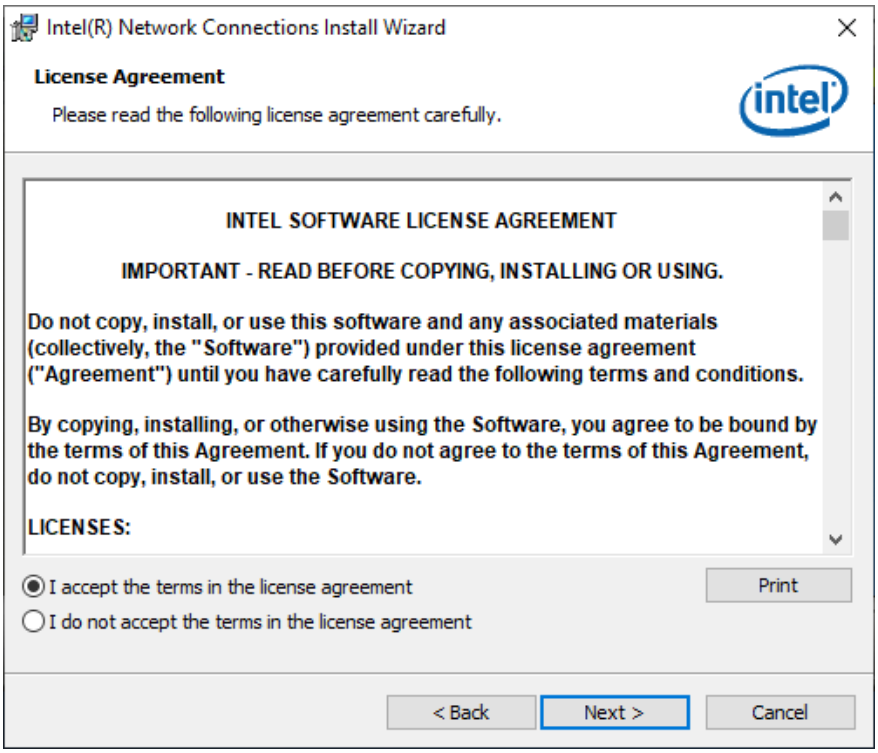
2. Click **Intel(R) I21x Gigabit Networks Drivers**.



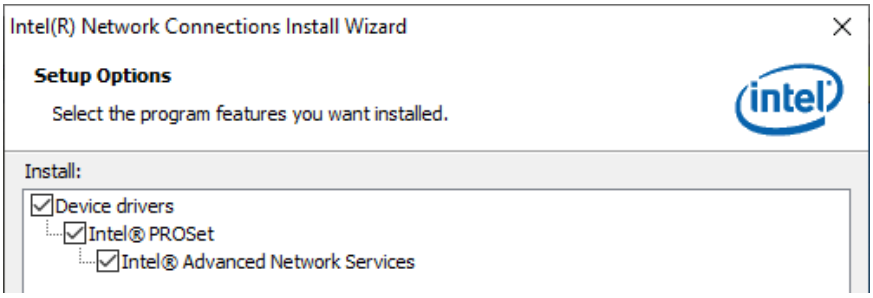
3. When the Welcome screen appears, click **Next**.



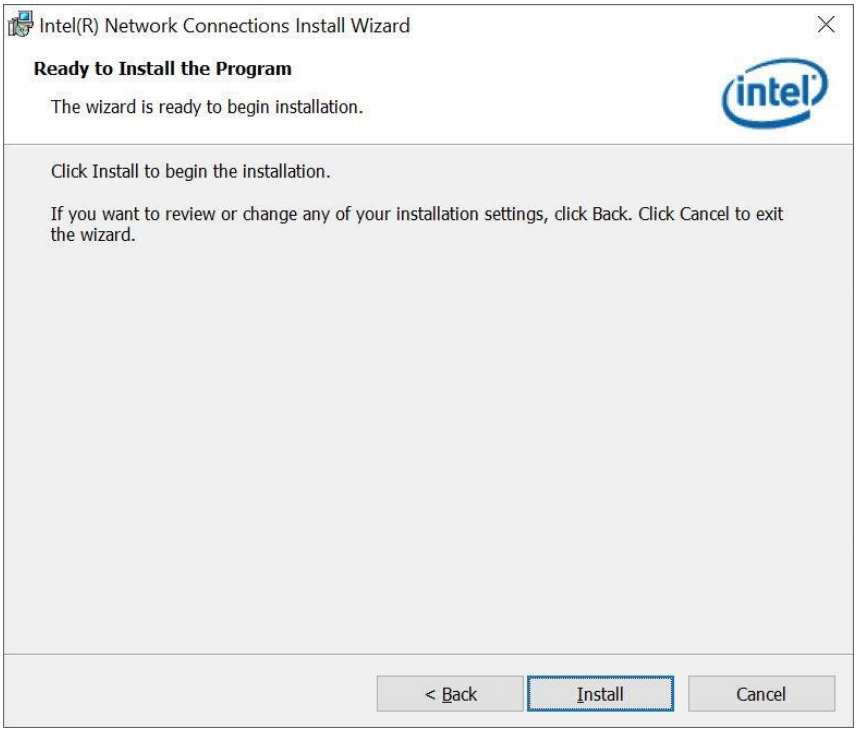
4. Choose “I accept the terms in the license agreement” and click **Next**.



5. Click **Next** after selecting the program features as shown below.



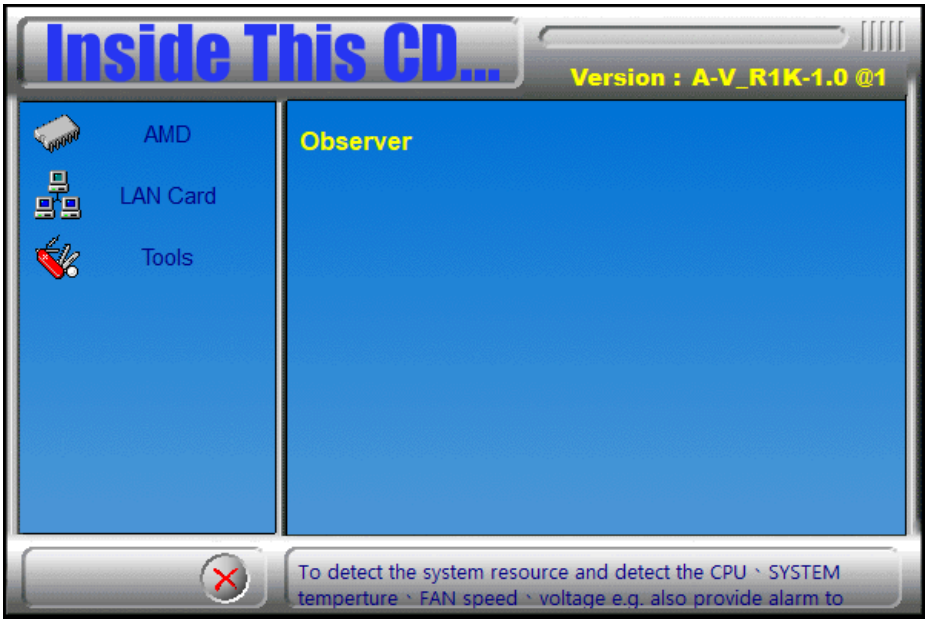
6. On the next screen, click **Install** to install the program.



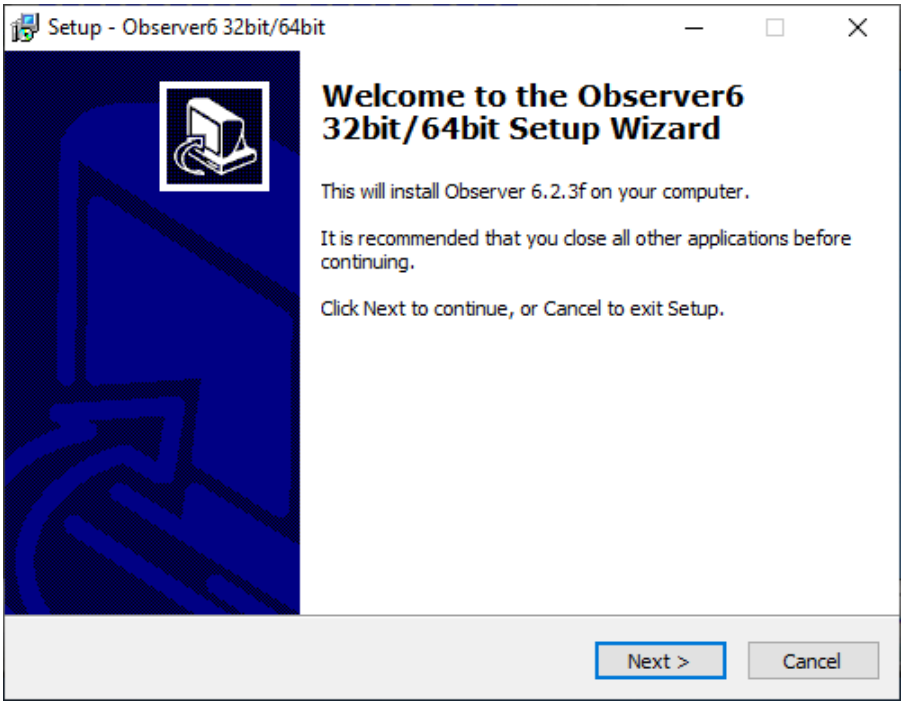
7. When the wizard has successfully installed the Program, restart your computer for changes to take effect.

3.6 Observer

1. Insert the disk enclosed in the package with the board. Click **Tools** on the left pane and then **Observer** on the right pane.

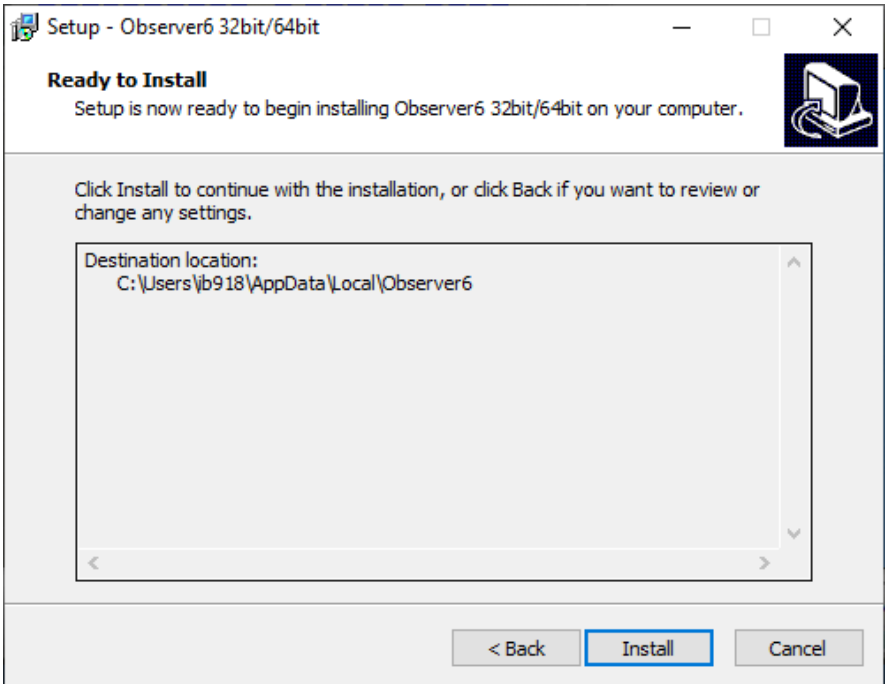


2. Click **Next** when the Welcome screen appears and to continue installing the Observer on your computer.



3. The next screen shows the destination location where Observer will be installed. Click **Next**.

4. Setup is now ready to begin installing Observer on your computer. Click **Install**.



5. To complete the installation of Observer, restart the computer (click **Finish**) 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

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
```

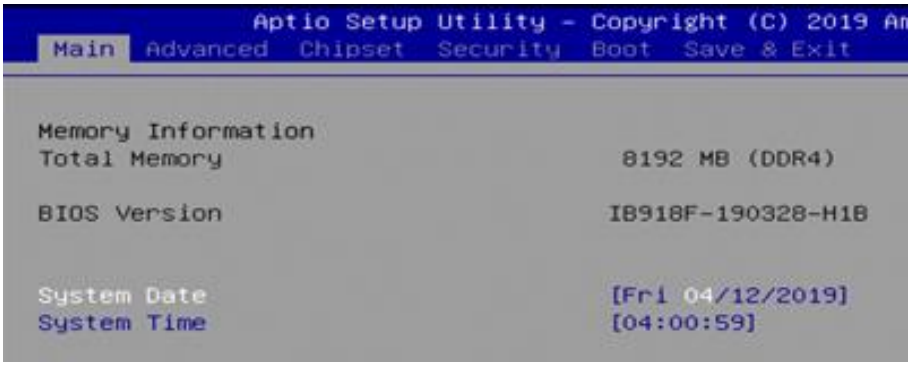
In general, press the arrow keys to highlight items, <Enter> to select, the <PgUp> and <PgDn> keys to change entries, <F1> for help, and <Esc> to quit.

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.

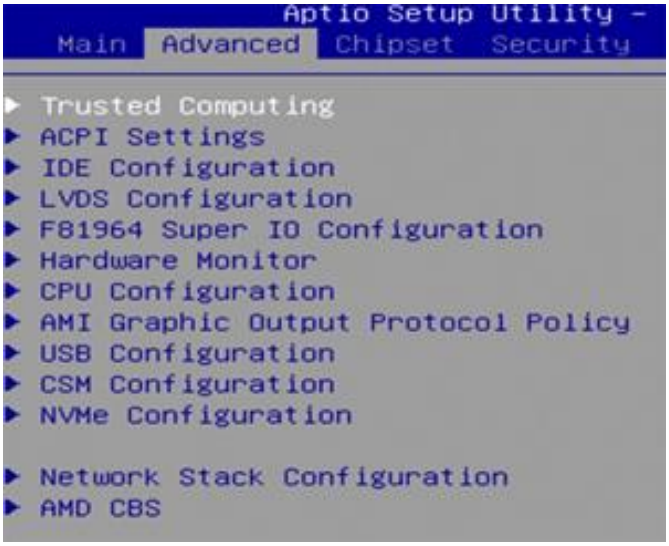
4.3 Main Settings



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 Trusted Computing Settings

BIOS Setting	Description
Security Device Support	Enables or Disables BIOS support for security device. OS will now show Security Device. TCG EFI protocol and INT1A interface will not be available **(Default: Enable)**
SHA-1 PCR Bank	Enable or Disable **(Default: Enable)**
SHA-256 PCR Bank	Enable or Disable **(Default: Enable)**
Pending operation	Schedule an operation for the security device. Note: Your computer will reboot durring restart in order to change state of security device. Options: None, TPM Clear **(Default: None)**
Platform HIerarchy	Enable or Disable **(Default: Enable)**
Storage HIerarchy	Enable or Disable **(Default: Enable)**

BIOS Setting	Description
Endorsement HIerarchy	Enable or Disable **(Default: Enable)**
TPM2.0 UEFI Spec Version	Select the tCG2 spec version support. TCG_1_2: the compatible mode for Win8/Win10 TCG_2: Support new tCG2 protocol and event format for Win10 or later. **(Default: TCG_2)**
Physical Presence Spect Version	Select to tell OS to support PPI spec version 1.2 or 1.3. Note some HCK tests might not support 1.3. **(Default: 1.3)**
Device Select	TPM 1.2 will restrict support to TPM 1.2 deevices. TPM 2.0 will restrict support to TPM 2.0 devices. Auto will support both with the default set to TPM 2.0 devices if not found. TPM 1.2 devices will be enumerated. **(Default: Auto)**

4.4.2 ACPI Settings

BIOS Setting	Description
Enable ACPI Auto Configuration	Enables / **Disables** BIOS ACPI auto configuration.
Enable Hibernation	**Enables** / Disables the system ability to hibernate (OS/S4 Sleep State). This option may be not effective with some OS.
ACPI Sleep State	Selects an ACPI sleep state where the system will enter when the Suspend button is pressed. **(Default: S3 (Suspend to RAM)**

4.4.3 IDE Configuration

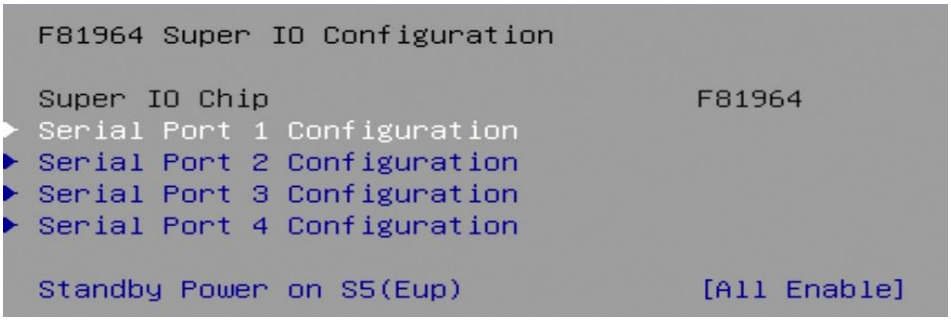
BIOS Setting	Description
SATA Ports	Detects the connection of SATA0 and SATA1.

4.4.4 LVDS Configuration

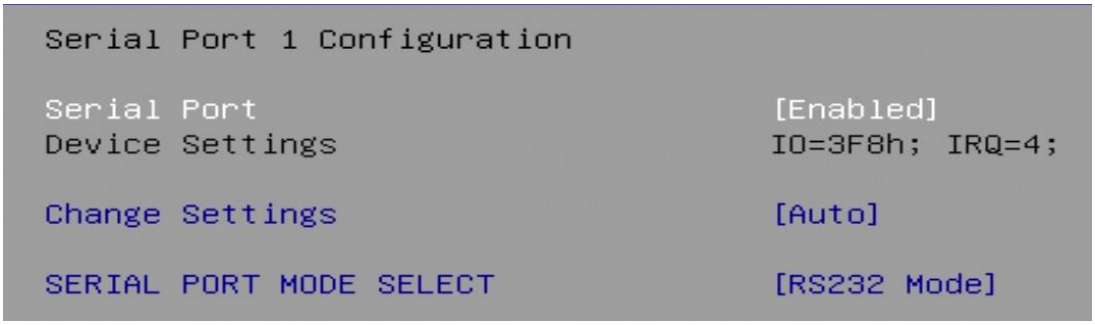
BIOS Setting	Description
LVDS Control	Enables / **Disables** the LVDS function.

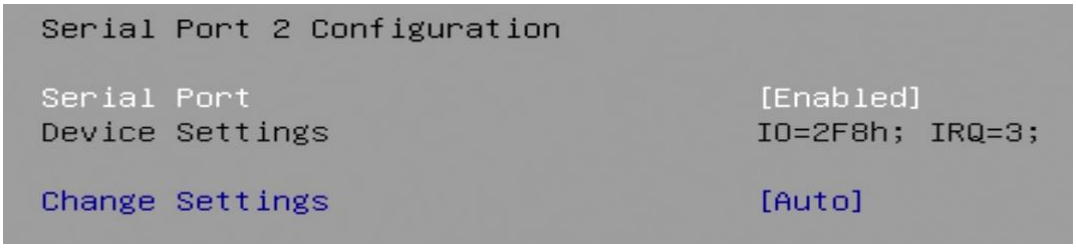
When enabled, LVDS Control will show the following items with their options:
1. Panel Color Depth: 18 BIT / 24 BIT
2. LVDS Channel Type: Single / Dual
3. Panel Type: 800x480, 800x600, 1024x768, 1280x768, 1280x800, 1280x960, 1280x1024, 1366x768, 1440x900, 1600x900, 1600x1200, 1680x1050, 1920x1080, 1920x1200
4. LVDS Backlight Control: 0(Min), 1, 2, 3, 4, 5, 6, 7(Max

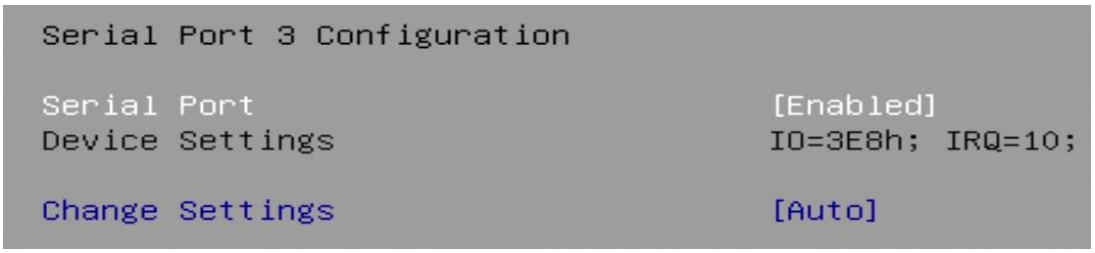
4.4.5 F81964 Super IO Configuration

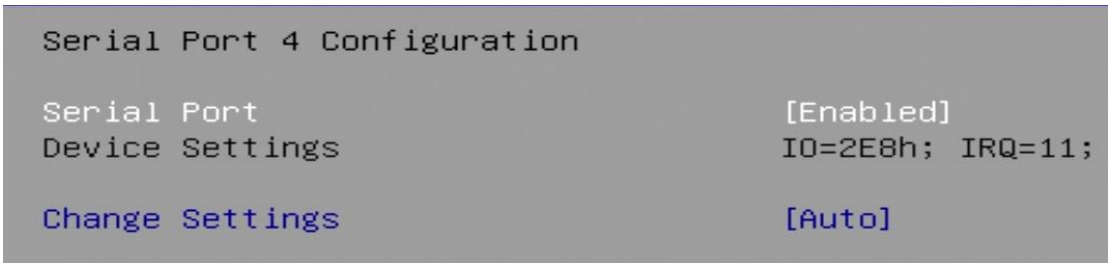


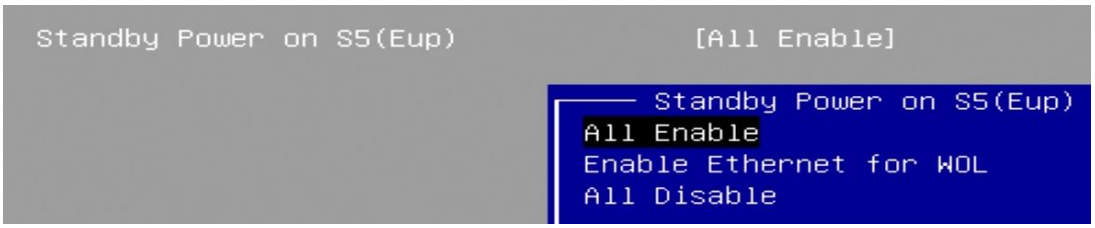
BIOS Setting	Description
Serial Port 1/2/3/4 Configuration	Set parameters of Serial Port 1/2/3/4
Standby Power on S5(EuP)	[Enable] Provide the standby power for devices. [Disable] Shutdown the standby power.


Serial Port 1 Configuration
Serial Port [Enabled]
Device Settings IO=3F8h; IRQ=4;
Change Settings [Auto]
SERIAL PORT MODE SELECT [RS232 Mode]

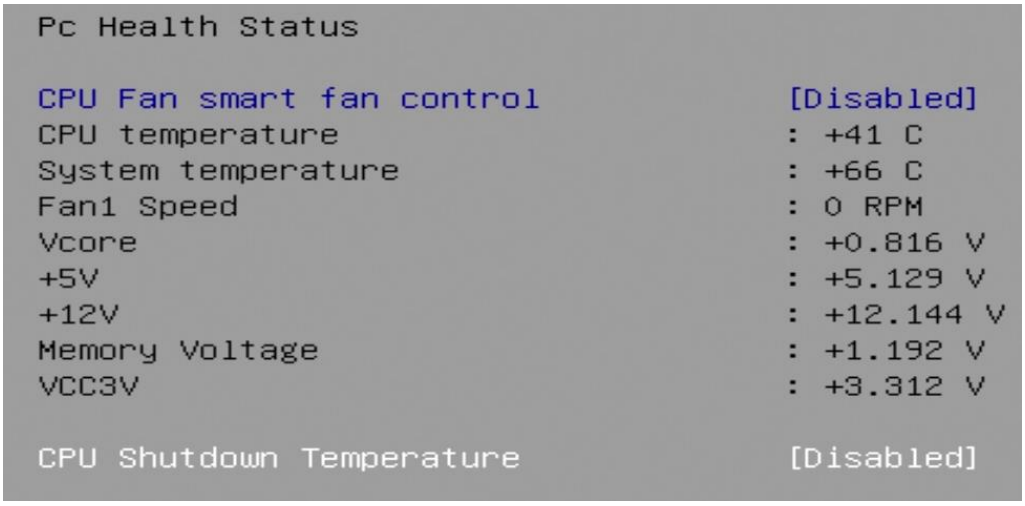

Serial Port 2 Configuration
Serial Port [Enabled]
Device Settings IO=2F8h; IRQ=3;
Change Settings [Auto]


Serial Port 3 Configuration
Serial Port [Enabled]
Device Settings IO=3E8h; IRQ=10;
Change Settings [Auto]


Serial Port 4 Configuration
Serial Port [Enabled]
Device Settings IO=2E8h; IRQ=11;
Change Settings [Auto]

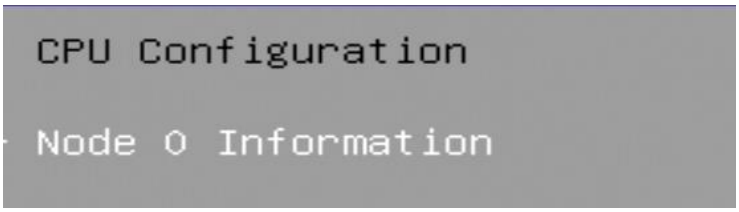

Standby Power on S5(Eup) [All Enable]
Standby Power on S5(Eup)
All Enable
Enable Ethernet for WOL
All Disable

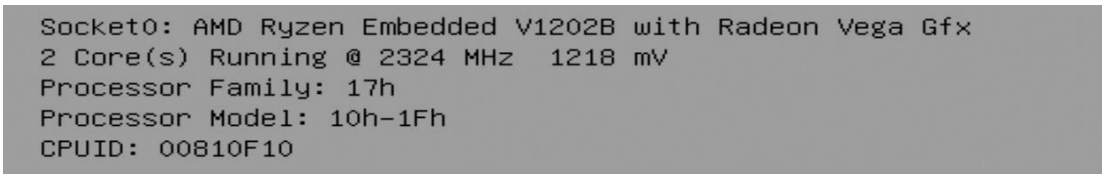
4.4.6 Hardware Monitor



BIOS Setting	Description
CPU Fan Smart Fan Control	Enables / Disables the CPU smart fan feature.
Temperatures / Fan Speed / Voltages	The values are read-only values as monitored by the system and show the PC health status.
CPU Shutdown Temperature	Enables / Disables the CPU shutdown temperature function.

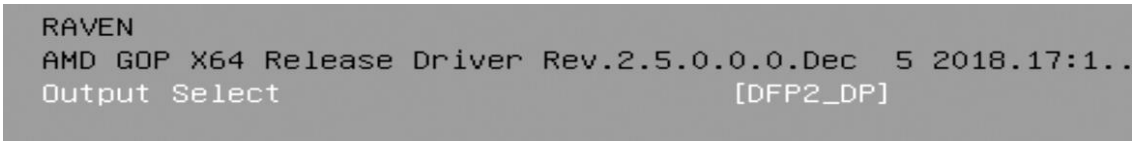
4.4.7 CPU Configuration





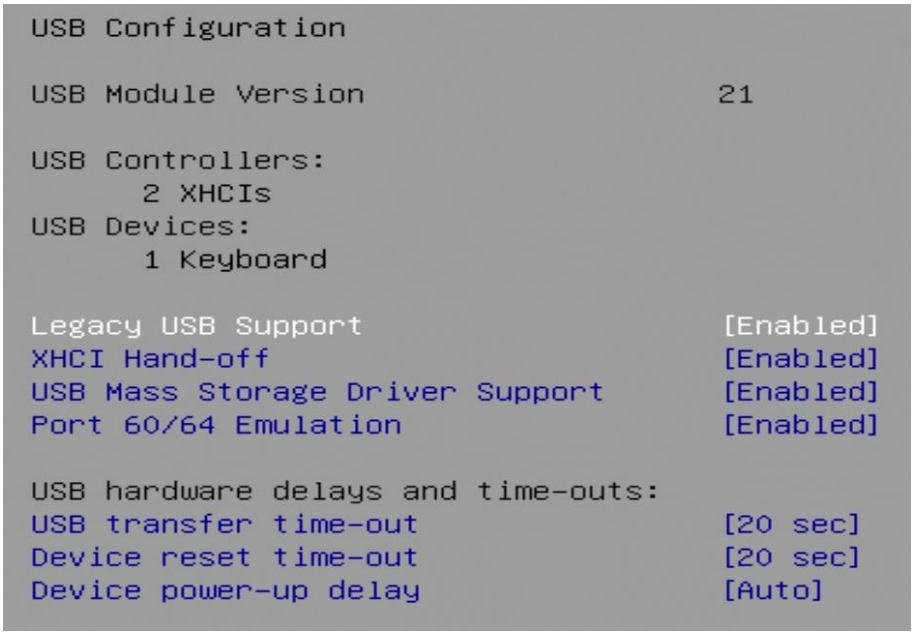
BIOS Setting	Description
Node 0 Information	Displays the memory information related to Node 0.

4.4.8 AMI Graphic Output Protocol Policy



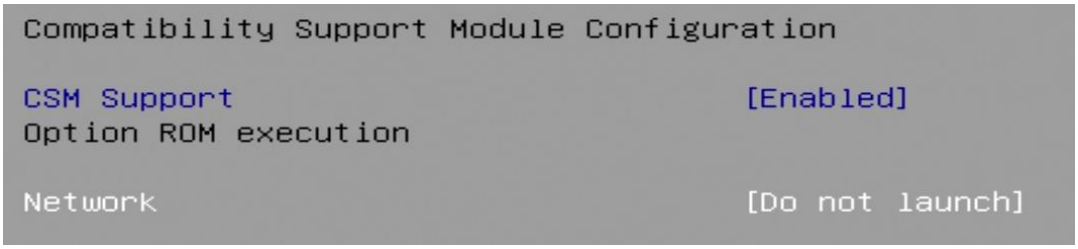
BIOS Setting	Description
Output Select	Allows you to select an output interface.

4.4.9 USB Configuration



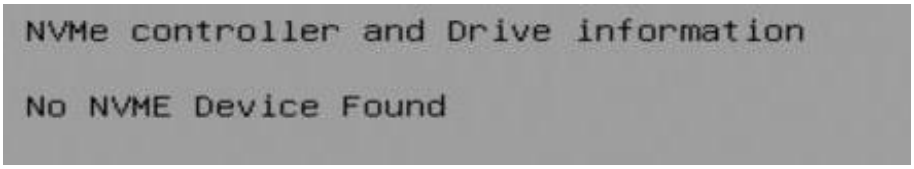
BIOS Setting	Description
Legacy USB Support	Enables Legacy USB support. • **Auto** disables legacy support if there is no USB device connected. • **Disable** keeps USB devices available only for EFI applications.
XHCI Hand-off	This is a workaround for OSes without XHCI hand-off support. The XHCI ownership 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. Options: 1 sec / 5 sec / 10 sec / 20 sec
Device reset time-out	Seconds of delaying execution of start unit command to USB mass storage device. Options: 10 sec / 20 sec / 30 sec / 40 sec
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 100ms. But for a Hub port, the delay is taken from Hub descriptor. Options: Auto / Manual

4.4.10 CSM Configuration

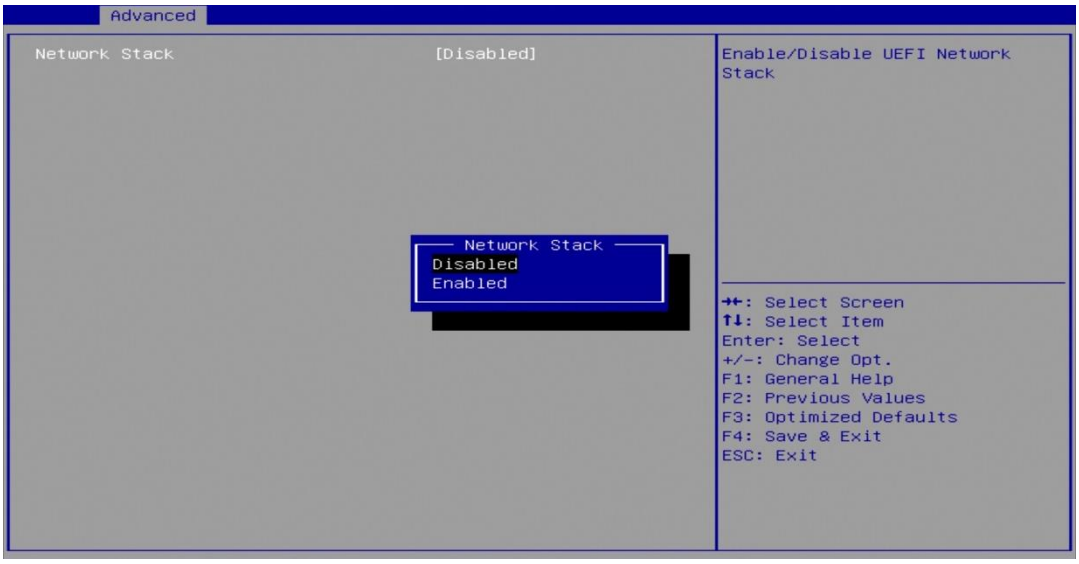


BIOS Setting	Description
CSM Support	Enables / Disables CSM support.
Network	Controls the execution of UEFI and Legacy PXE OpROM. Options: Do not launch UEFI / Legacy

4.4.11 NVMe Configuration

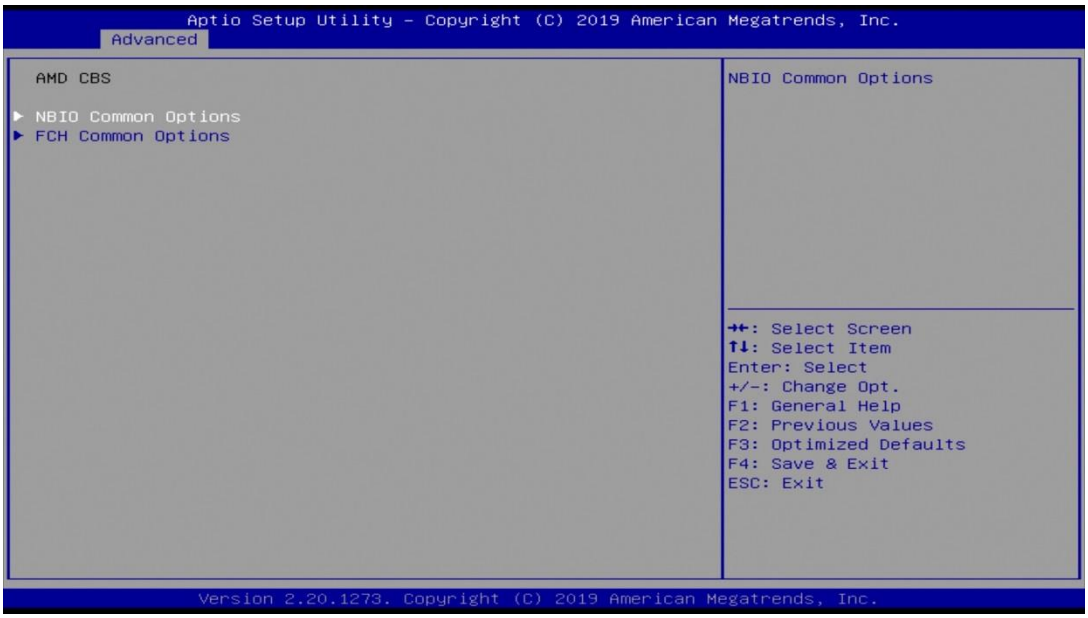


4.4.12 Network Stack Configuration

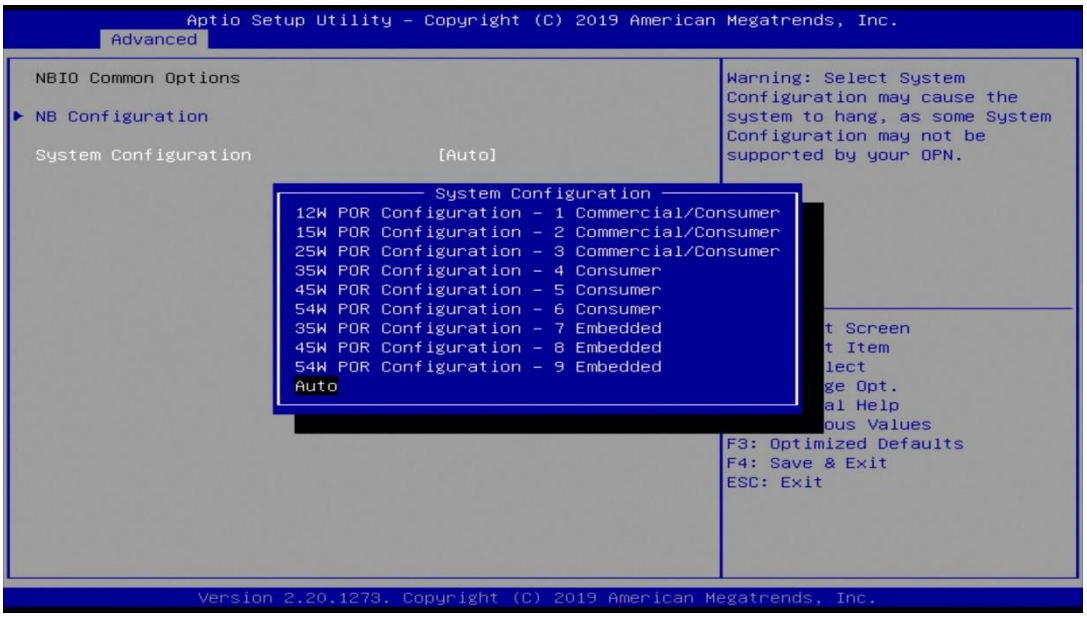


BIOS Setting	Description
Network Stack	Enables / Disables UEFI Network Stack.

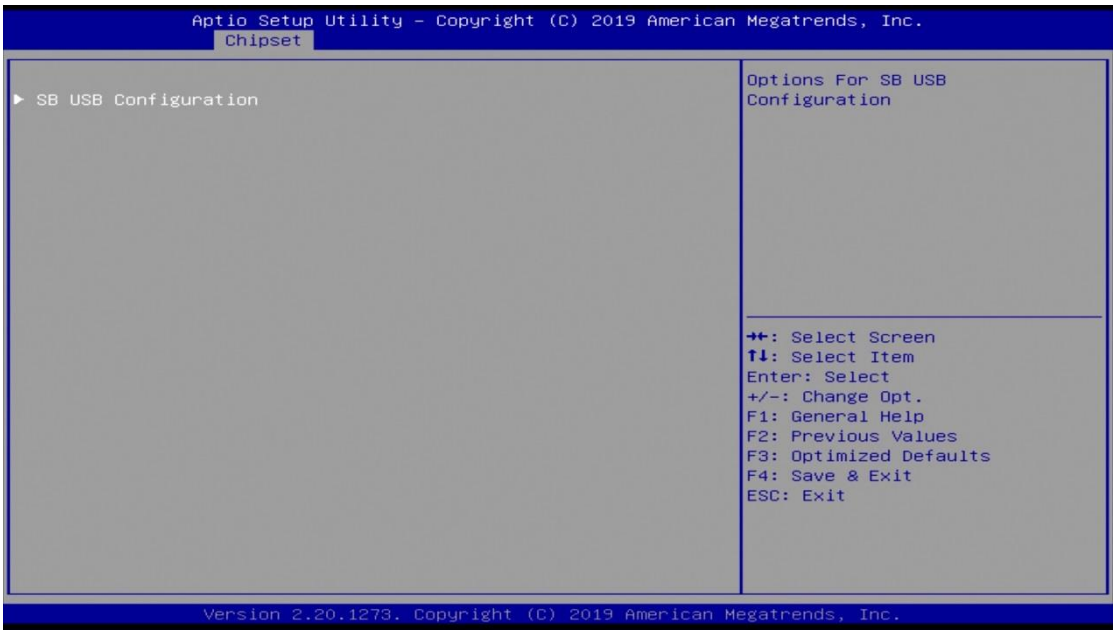
4.4.13 AMD CBS



BIOS Setting	Description
NB Common Options	This is for the NB configuration for IOMMU settings and system configuration.
FCH Common Options	Here you can set the AC loss control method with options of: **Always Off** / Always On



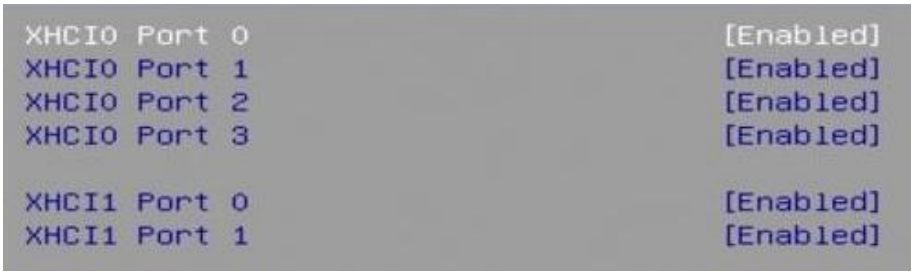
4.5 Chipset Settings



4.5.1 SB USB Configuration

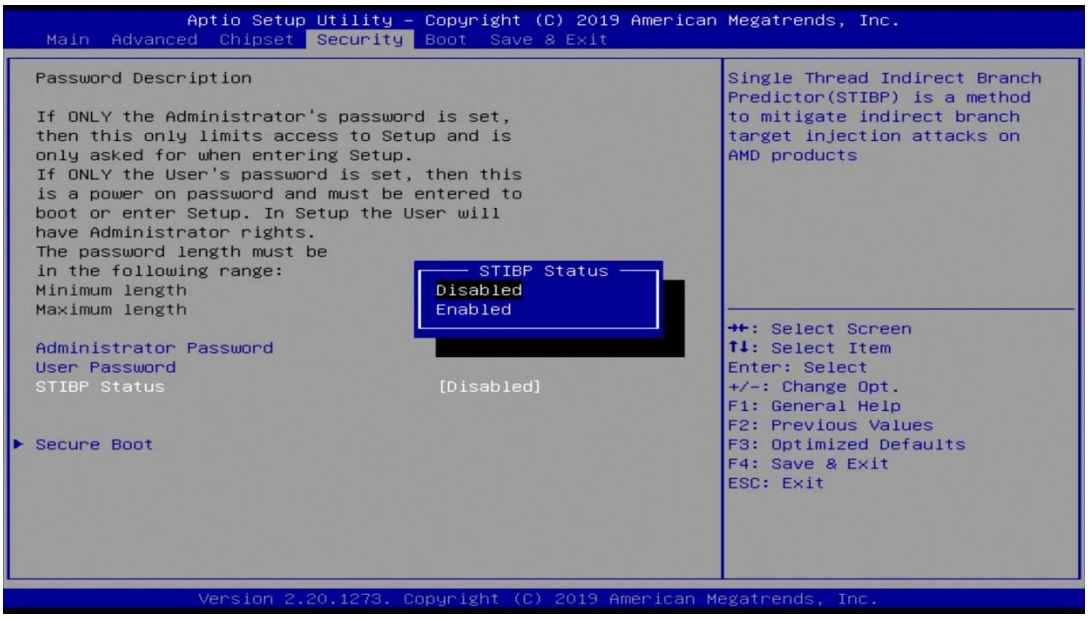
BIOS Setting	Description
SB USB Configuration	Options for SB USB Configuration.

4.5.1.1. XHCI Ports



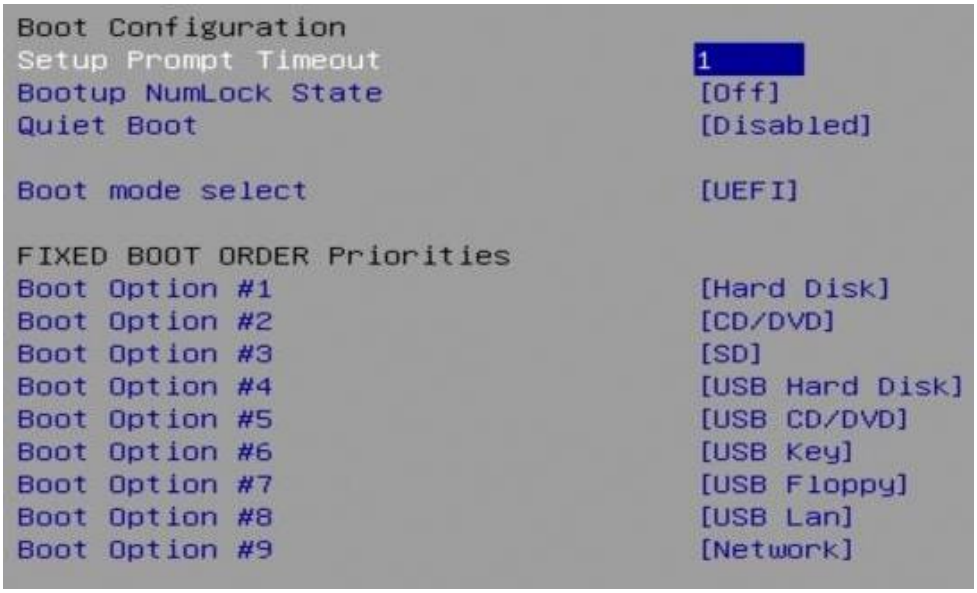
BIOS Setting	Description
XHCI0 & XHCI1 Ports	Enables / Disables the XHCI0 & XHCI1 ports (XHCI/EMCI).

4.6 Security Settings

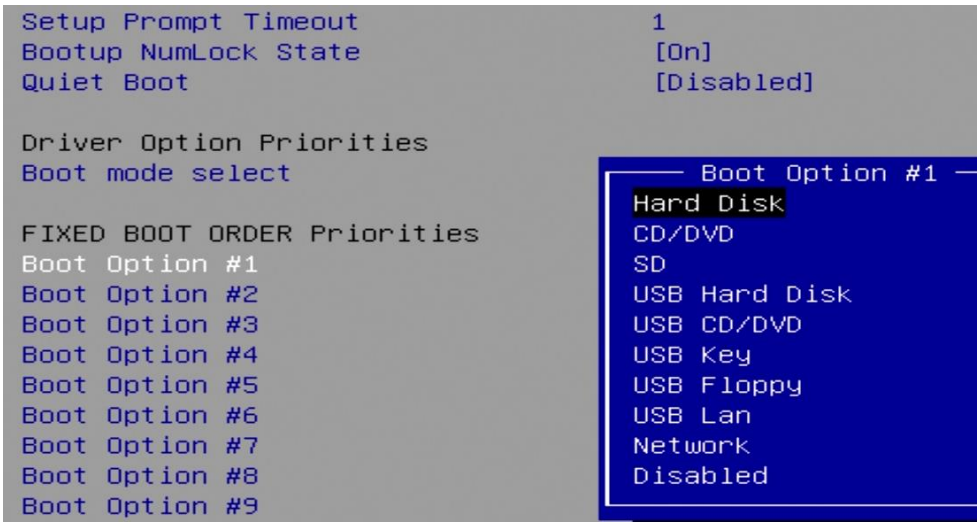


BIOS Setting	Description
Administrator Password	Sets an administrator password for the setup utility.
User Password	Sets a user password.
STIBP Status	Single Thread Indirect Branch Predictor (STIBP) is a method to mitigate indirect branch target injection attacks on AMD products. Options: **Disabled**, Enabled
Secure Boot	Secure Boot feature is Active if Secure Boot is enabled. Platform Key (PK) is enrolled and the system is in user mode. The mode change requires platform reset.
Secure Boot Mode	Secure Boot mode options: **Standard** or Custom. In custom mode, secure boot policy variables can be configured by a physically present user without full authentication.
Secure Boot	Secure Boot feature is Active if Secure Boot is enabled. Platform Key (PK) is enrolled and the system is in user mode. The mode change requires platform reset.

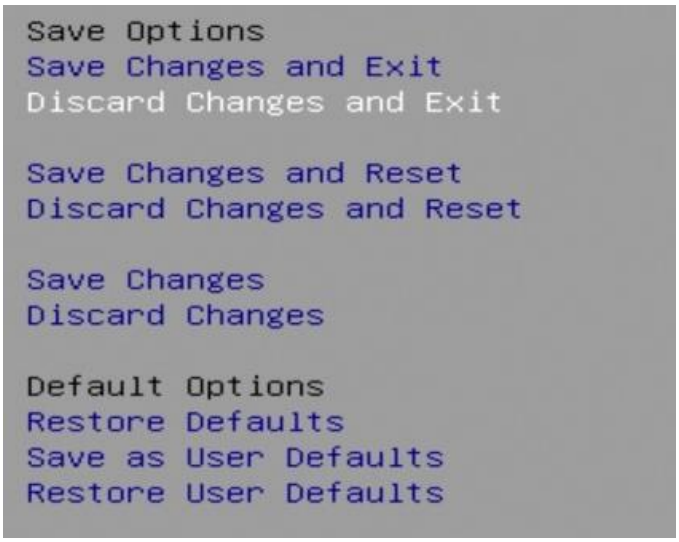
4.7 Boot Settings



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.
Quiet Boot	Enables / Disables Quiet Boot option.
Boot mode select	Selects a Boot mode, Legacy / UEFI.
Boot Option Priorities	Sets the system boot order.



4.8 Save & Exit



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
0x00000A00-0x00000A0F	Motherboard resources
0x00000A10-0x00000A1F	Motherboard resources
0x00000A20-0x00000A2F	Motherboard resources
0x00000A30-0x00000A3F	Motherboard resources
0x00000A40-0x00000A4F	Motherboard resources
0x00000070-0x00000071	System CMOS/real time clock
0x0000D000-0x0000DFFF	PCI Express Root Port
0x0000D000-0x0000DFFF	AMD Radeon(TM) Vega 8 Graphics
0x000003F8-0x000003FF	Communications Port (COM1)
0x000002F8-0x000002FF	Communications Port (COM2)
0x000003E8-0x000003EF	Communications Port (COM3)
0x000002E8-0x000002EF	Communications Port (COM4)
0x00000238-0x0000023F	Communications Port (COM5)
0x00000338-0x0000033F	Communications Port (COM6)
0x00000020-0x00000021	Programmable interrupt controller
0x000000A0-0x000000A1	Programmable interrupt controller
0x00000000-0x000003AF	PCI Express Root Complex
0x00000000-0x000003AF	Direct memory access controller
0x000003E0-0x00000CF7	PCI Express Root Complex
0x000003B0-0x000003DF	PCI Express Root Complex
0x00000D00-0x0000FFFF	PCI Express Root Complex
0x0000F000-0x0000FFFF	PCI Express Root Port
0x0000E000-0x0000EFFF	PCI Express Root Port
0x00000040-0x00000043	System timer
0x00000010-0x0000001F	Motherboard resources
0x00000022-0x0000003F	Motherboard resources
0x00000063-0x00000063	Motherboard resources
0x00000065-0x00000065	Motherboard resources
0x00000067-0x0000006F	Motherboard resources
0x00000072-0x0000007F	Motherboard resources
0x00000080-0x00000080	Motherboard resources
0x00000084-0x00000086	Motherboard resources
0x00000088-0x00000088	Motherboard resources
0x0000008C-0x0000008E	Motherboard resources

Address	Device Description
0x00000090-0x0000009F	Motherboard resources
0x000000A2-0x000000BF	Motherboard resources
0x000000B1-0x000000B1	Motherboard resources
0x000000E0-0x000000EF	Motherboard resources
0x000004D0-0x000004D1	Motherboard resources
0x0000040B-0x0000040B	Motherboard resources
0x000004D6-0x000004D6	Motherboard resources
0x00000C00-0x00000C01	Motherboard resources
0x00000C14-0x00000C14	Motherboard resources
0x00000C50-0x00000C51	Motherboard resources
0x00000C52-0x00000C52	Motherboard resources
0x00000C6C-0x00000C6C	Motherboard resources
0x00000C6F-0x00000C6F	Motherboard resources
0x00000CD0-0x00000CD1	Motherboard resources
0x00000CD2-0x00000CD3	Motherboard resources
0x00000CD4-0x00000CD5	Motherboard resources
0x00000CD6-0x00000CD7	Motherboard resources
0x00000CD8-0x00000CDF	Motherboard resources
0x00000800-0x0000089F	Motherboard resources
0x00000B00-0x00000B0F	Motherboard resources
0x00000B20-0x00000B3F	Motherboard resources
0x00000900-0x0000090F	Motherboard resources
0x00000910-0x0000091F	Motherboard resources
0x00000061-0x00000061	System speaker
0x00000081-0x00000083	Direct memory access controller
0x00000087-0x00000087	Direct memory access controller
0x00000089-0x0000008B	Direct memory access controller
0x0000008F-0x0000008F	Direct memory access controller
0x000000C0-0x000000DF	Direct memory access controller

B. Interrupt Request Lines (IRQ)

C. 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 4294967292	PCI Express Root Port
IRQ 0	High precision event timer
IRQ 0	System timer
IRQ 8	High precision event timer
IRQ 7	AMD GPIO Controller
IRQ 4294967265~72	AMD USB 3.10 eXtensible Host Controller - 1.10 (Microsoft)
IRQ 4	Communications Port (COM1)
IRQ 3	Communications Port (COM2)
IRQ 5	Communications Port (COM3)
IRQ 5	Communications Port (COM4)
IRQ 6	Communications Port (COM5)
IRQ 6	Communications Port (COM6)
IRQ 53	High Definition Audio Controller
IRQ 53	AMD Audio CoProcessor
IRQ 54	AMD High Definition Audio Controller
IRQ 54~55	Microsoft ACPI-Compliant System
IRQ 55	AMD SFH KMDF I2C
IRQ 56 ~ IRQ 204	Microsoft ACPI-Compliant System
IRQ 256 ~ IRQ 511	Microsoft ACPI-Compliant System
IRQ 4294967293~94	PCI Express Root Port
IRQ 4294967291	PCI Express Root Port
IRQ 4294967273~75	AMD Radeon(TM) Vega 8 Graphics
IRQ 4294967290	Standard SATA AHCI Controller
IRQ 4294967257~64	AMD USB 3.10 eXtensible Host Controller - 1.10 (Microsoft)
IRQ 4294967282~87	Intel(R) I211 Gigabit Network Connection #5
IRQ 4294967276~81	Intel(R) I211 Gigabit Network Connection #6
IRQ 4294967288~89	AMD PSP 10.0 Device

E. Digital I/O Sample Code

1. DIO Sample Code: The file F81846.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 "F81846.H"
#include <dos.h>
//=====================================================================
unsigned int F81846_BASE;
void Unlock_F81846 (void);
void Lock_F81846 (void);
//=====================================================================
unsigned int Init_F81846(void)
{
	unsigned int result;	//
	unsigned char ucDid;

	F81846_BASE = 0x4E;
	result = F81846_BASE;

	ucDid = Get_F81846_Reg(0x20);
	if ((ucDid == 0x07) || (ucDid == 0x10) || (ucDid == 0x15)) //Fintek
81865/81846/81846/81946/81846
	{	goto Init_Finish;}

	F81846_BASE = 0x2E;
	result = F81846_BASE;

	ucDid = Get_F81846_Reg(0x20);
	if ((ucDid == 0x07) || (ucDid == 0x10) || (ucDid == 0x15))	//Fintek
81865/81846/81846/81946/81846
	{	goto Init_Finish;}

	F81846_BASE = 0x00;
	result = F81846_BASE;

Init_Finish:
	return (result);
}
//=====================================================================
void Unlock_F81846 (void)
{
	outportb(F81846_INDEX_PORT, F81846_UNLOCK);
	outportb(F81846_INDEX_PORT, F81846_UNLOCK);
}
```

```
//=============================================================================
void Lock_F81846 (void)
{
	outportb(F81846_INDEX_PORT, F81846_LOCK);
}
//=============================================================================
void Set_F81846_LD( unsigned char LD)
{
	Unlock_F81846();
	outportb(F81846_INDEX_PORT, F81846_REG_LD);
	outportb(F81846_DATA_PORT, LD);
	Lock_F81846();
}
//=============================================================================
void Set_F81846_Reg( unsigned char REG, unsigned char DATA)
{
	Unlock_F81846();
	outportb(F81846_INDEX_PORT, REG);
	outportb(F81846_DATA_PORT, DATA);
	Lock_F81846();
}
//=============================================================================
unsigned char Get_F81846_Reg(unsigned char REG)
{
	unsigned char Result;
	Unlock_F81846();
	outportb(F81846_INDEX_PORT, REG);
	Result = inportb(F81846_DATA_PORT);
	Lock_F81846();
	return Result;
}
//=============================================================================
```

2. DIO Sample Code: The file F81846.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 __F81846_H
#define __F81846_H              1
//=====================================================================
#define     F81846_INDEX_PORT           (F81846_BASE)
#define     F81846_DATA_PORT            (F81846_BASE+1)
//=====================================================================
#define     F81846_REG_LD               0x07
//=====================================================================
#define F81846_UNLOCK               0x87
#define     F81846_LOCK                 0xAA
//=====================================================================
unsigned int Init_F81846(void);
void Set_F81846_LD( unsigned char);
void Set_F81846_Reg( unsigned char, unsigned char);
unsigned char Get_F81846_Reg( unsigned char);
//=====================================================================
#endif      //__F81846_H
```

3. DIO 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 "F81846.H"
//---------------------------------------------------------------------------
void ClrKbBuf(void);
int main (int argc, char *argv[]);
//---------------------------------------------------------------------------
int main (int argc, char *argv[])
{
        unsigned char result;
        char SIO;

        SIO = Init_F81846();
        if (SIO == 0)
        {
                printf("Can not detect Fintek F81846, program abort.\n");
                return(1);
        }

           Set_F81846_LD(0x06);
//switch to logic device 6

           result = ((Get_F81846_Reg(0xE2)) & 0x04) ? 0x01 : 0x00;          //result = 0x00 GPI is
Low / result = 0x01 GPI is High

        return (result);
}
```

F. 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 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 NCT6116D.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 __NCT6116D_H
#define __NCT6116D_H                    1
//-----------------------------------------------------------------------------
#define   NCT6116D_INDEX_PORT       (NCT6116D_BASE)
#define   NCT6116D_DATA_PORT        (NCT6116D_BASE+1)
//-----------------------------------------------------------------------------
#define   NCT6116D_REG_LD           0x07
//-----------------------------------------------------------------------------
#define NCT6116D_UNLOCK             0x87
#define   NCT6116D_LOCK             0xAA
//-----------------------------------------------------------------------------
unsigned int Init_NCT6116D(void);
void Set_NCT6116D_LD( unsigned char);
void Set_NCT6116D_Reg( unsigned char, unsigned char);
unsigned char Get_NCT6116D_Reg( unsigned char);
//-----------------------------------------------------------------------------
#endif    //__NCT6116D_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 "NCT6116D.H"
//---------------------------------------------------------------------------
int main (void);

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

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

	SIO = Init_NCT6116D();
	if (SIO == 0)
	{
		printf("Can not detect Nuvoton NCT6116D, program abort.\n");
		return(1);
	}

	WDTInitial();

	WDTEnable(10);

	WDTDisable();

	return 0;
}
//---------------------------------------------------------------------------
void WDTInitial(void)
{
	unsigned char bBuf;
```

```
	Set_NCT6116D_LD(0x08);			//switch to logic device
8
	bBuf = Get_NCT6116D_Reg(0x30);
	bBuf &= (~0x01);
	Set_NCT6116D_Reg(0x30, bBuf);		//Enable WDTO
}
//-----------------------------------------------------------------------
void WDTEnable(unsigned char NewInterval)
{
	unsigned char bBuf;

	Set_NCT6116D_LD(0x08);			//switch to logic device
8
	Set_NCT6116D_Reg(0x30, 0x01);		//enable timer

	bBuf = Get_NCT6116D_Reg(0xF0);
	bBuf &= (~0x08);
	Set_NCT6116D_Reg(0xF0, bBuf);		//count mode is second

	Set_NCT6116D_Reg(0xF1, NewInterval);	//set timer
}
//-----------------------------------------------------------------------
void WDTDisable(void)
{
	Set_NCT6116D_LD(0x08);			//switch to logic device
8
	Set_NCT6116D_Reg(0xF1, 0x00);		//clear watchdog timer
	Set_NCT6116D_Reg(0x30, 0x00);		//watchdog disabled
}
//-----------------------------------------------------------------------
```

3. Sample Code: The file NCT6116D.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 "NCT6116D.H"
#include <dos.h>
//---------------------------------------------------------------------------
unsigned int NCT6116D_BASE;
void Unlock_NCT6116D (void);
void Lock_NCT6116D (void);
//---------------------------------------------------------------------------
unsigned int Init_NCT6116D(void)
{
	unsigned int result;
	unsigned char ucDid;

	NCT6116D_BASE = 0x4E;
	result = NCT6116D_BASE;

	ucDid = Get_NCT6116D_Reg(0x20);
	if (ucDid == 0xC4)				//NCT6116D??
	{	goto Init_Finish;	}

	NCT6116D_BASE = 0x2E;
	result = NCT6116D_BASE;

	ucDid = Get_NCT6116D_Reg(0x20);
	if (ucDid == 0xC4)				//NCT6116D??
	{	goto Init_Finish;	}

	NCT6116D_BASE = 0x00;
	result = NCT6116D_BASE;

Init_Finish:
	return (result);
}
//---------------------------------------------------------------------------
void Unlock_NCT6116D (void)
{
	outportb(NCT6116D_INDEX_PORT, NCT6116D_UNLOCK);
	outportb(NCT6116D_INDEX_PORT, NCT6116D_UNLOCK);
}
//---------------------------------------------------------------------------
```

```
void Lock_NCT6116D (void)
{
	outportb(NCT6116D_INDEX_PORT, NCT6116D_LOCK);
}
//-----------------------------------------------------------------------
void Set_NCT6116D_LD( unsigned char LD)
{
	Unlock_NCT6116D();
	outportb(NCT6116D_INDEX_PORT, NCT6116D_REG_LD);
	outportb(NCT6116D_DATA_PORT, LD);
	Lock_NCT6116D();
}
//-----------------------------------------------------------------------
void Set_NCT6116D_Reg( unsigned char REG, unsigned char DATA)
{
	Unlock_NCT6116D();
	outportb(NCT6116D_INDEX_PORT, REG);
	outportb(NCT6116D_DATA_PORT, DATA);
	Lock_NCT6116D();
}
//-----------------------------------------------------------------------
unsigned char Get_NCT6116D_Reg(unsigned char REG)
{
	unsigned char Result;
	Unlock_NCT6116D();
	outportb(NCT6116D_INDEX_PORT, REG);
	Result = inportb(NCT6116D_DATA_PORT);
	Lock_NCT6116D();
	return Result;
}
//-----------------------------------------------------------------------
```

```
//---------------------------------------------------------------------------
//
// 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 "F81846.H"
#include <dos.h>
//---------------------------------------------------------------------------
unsigned int F81846_BASE; void Unlock_F81846 (void); void Lock_F81846 (void);
//---------------------------------------------------------------------------
unsigned int Init_F81846(void)
{
	unsigned int result;
	unsigned char ucDid;

	F81846_BASE = 0x4E;
	result = F81846_BASE;

	ucDid = Get_F81846_Reg(0x20);
	if (ucDid == 0x07)			//Fintek 81846
	{	goto Init_Finish;}

	F81846_BASE = 0x2E;
	result = F81846_BASE;

	ucDid = Get_F81846_Reg(0x20);
	if (ucDid == 0x07)			//Fintek 81846
	{	goto Init_Finish;}

	F81846_BASE = 0x00;
	result = F81846_BASE;

Init_Finish:
	return (result);
}
//---------------------------------------------------------------------------
void Unlock_F81846 (void)
{
	outportb(F81846_INDEX_PORT, F81846_UNLOCK);
	outportb(F81846_INDEX_PORT, F81846_UNLOCK);
}
//---------------------------------------------------------------------------
void Lock_F81846 (void)
{
	outportb(F81846_INDEX_PORT, F81846_LOCK);
}
//---------------------------------------------------------------------------
void Set_F81846_LD( unsigned char LD)
{
	Unlock_F81846();
```

```
	outportb(F81846_INDEX_PORT, F81846_REG_LD);
	outportb(F81846_DATA_PORT, LD); Lock_F81846();
}
//----------------------------------------------------------------------------
void Set_F81846_Reg( unsigned char REG, unsigned char DATA)
{
	Unlock_F81846();
	outportb(F81846_INDEX_PORT, REG);
	outportb(F81846_DATA_PORT, DATA);
	Lock_F81846();
}
//----------------------------------------------------------------------------
unsigned char Get_F81846_Reg(unsigned char REG)
{
	unsigned char Result;
	Unlock_F81846();
	outportb(F81846_INDEX_PORT, REG);
	Result = inportb(F81846_DATA_PORT);
	Lock_F81846();
	return Result;
}
//----------------------------------------------------------------------------
```

```
//----------------------------------------------------------------------------
//
// 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	F81846_H
#define	F81846_H	1
//----------------------------------------------------------------------------
#define	F81846_INDEX_PORT	(F81846_BASE)
#define	F81846_DATA_PORT	(F81846_BASE+1)
//----------------------------------------------------------------------------
#define	F81846_REG_LD	0x07
//----------------------------------------------------------------------------
#define F81846_UNLOCK 0x87
#define	F81846_LOCK	0xAA
//----------------------------------------------------------------------------
unsigned int Init_F81846(void);
void Set_F81846_LD( unsigned char);
void Set_F81846_Reg( unsigned char, unsigned char); unsigned char
Get_F81846_Reg( unsigned char);
//----------------------------------------------------------------------------
#endif //	F81846_H
```