MPT-1000V
Multi-Purpose In-Vehicle System

User's Manual

Version 1.0
(August 2021)

iBASE

Compliance

CE

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

FC

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

E24

This product holds the official approval as a certified in-vehicle device in accordance with ECE type approval E24.

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.
- Use this product in environments with ambient temperatures as indicated in the Specifications in this user's manual.

Care for your IBASE products:

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

WARNING

Attention during use:

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

Avoid Disassembly

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

CAUTION

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

Warranty Policy

- **IBASE standard products:**

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

- **3rd-party parts:**

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

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

Technical Support & Services

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

iBASE

Table of Contents

Chapter 1
General Information

The information provided in this chapter includes:

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

1.1 Introduction

The MPT-1000V is vehicle-mounted computing system supporting both car battery and DC power modes. Its robust and rugged design is based on the Intel® Atom™ E3950 processor (model MPT-1000V) and Intel® Celeron™ N3350 (MPT-1000VN) processor. Equipped with a variety of flexible I/O interface, the system offers high-speed data transmission and robust connections with reliable operation in harsh environments.

1.2 Features

- Fanless and ruggedized design
- SIM socket supports Dual WWAN
- Removable 2.5” device bay
- Ignition power-on signal control
- Isolated USB, serial, DIDO ports
- Onboard isolated CANbus
- Rich I/O interfaces for wireless, SSD, GPS, WWAN and add-on card expansion
- Add on card slots reset function
- Quick accessible RTC battery

1.3 Packing List

Your MPT-1000V 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.

Item	Q'ty	IBASE P/N
MPT-1000V	1	- -
Mounting Bracket	2	- -
Terminal block for power input and IGN 22 (3-pin) (Dinkle 5ESDVM-03P)	1	C12165ESD03105000P
Terminal block for CANbus (4-pin) (Dinkle EC381VM-04P)	1	C1216EC3804103000P
Flat head screw, M2*L4 (A2)	2	H0220351112200000P
Flat head screw, M3*3.8 (B21)	1	H0230321112200000P
Screw for SSD / HDD (if HDD not pre-installed)	4	H0230561B710BN000P

1.4 Specifications

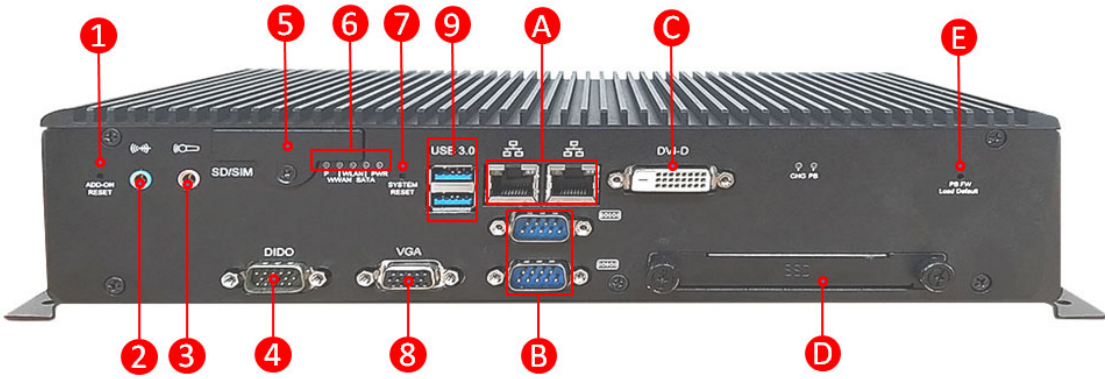
Product Name	MPT-1000V
System	
Motherboard	MBT-1002V
Operating System	• Windows 10 (64-bit) • Linux kernel 3.8.0 or above (64-bit)
CPU	• Intel® Atom™ E3950, TDP: 10W (Model MPT-1000V) • Intel® Celeron™ N3350, TDP: 6W (Model MPT-1000VN)
Memory	• 4GB SKU: MPT-1000VN (N3350) • 8GB SKU: MPT-1000V8G (E3950)
Graphics	Intel® HD graphics Gen. 7 with 4EU Supports DX 11, OGL 3.0, OCL 1.2, OGLES 2.0
Front I/O	• 1 x DVI-D connector for display connection • 1 x VGA DSUB15 • 2 x RJ45 for GbE LAN • 1 x 3,5mm blue color audio connector for line input. • 1 x 3,5mm pink color audio connector for MIC input. • 2 x external accessible SIM socket (covered) • 1 x external micro SD card slot (covered with SIM card slot) • 1 x Stack USB connector for 2 x USB3.1 (#1/#2) • 1 x DUSB9 for COM1 RS-232/422/485 port • 1 x DUB9 for COM2 RS-485 isolated port • 1 x DSUB15 for DIDO 6in / 6out isolated • Isolation voltage • 2500 VRMS • 1 x tack switch for system reset • 1 x tack switch for add-on cards reset • 1 x tack switch for power firmware load default • Indicators ■ 1 x Green color LED for power status ■ 1 x White color LED for storage activity status ■ 1 x Blue color LED for WLAN activity status (optional) ■ 1 x Orange color LED for WWAN activity status (optional)) ■ 1 x Green color LED, programmable ■ 1 x Red color LED for warning of wrong power voltage ■ 1 x Blue color LED for external battery connected

Rear I/O	• 2 x DSUB9 for COM3/4 RS232 ports • 1 x 4 pins terminal block for 2 masters CANbus • 1 x USB3.0 flag type A • 1 x audio jack for line out • 1 x tack switch for clear CMOS • 1 x open window for accessing RTC battery • 1 x 6-pin ATX connector for external battery, and 19VDC (Max:1A) output • 1 x blade fuse socket with cover • 1 x DIP switch for selecting power mode: Car / DC • 1 x 3-pin thermal block connector for power input and IGN
Storage	• 1 x accessible 2.5“ SATA space, at the front side • 1 x mSATA via miniPCIe (shared with SATA port) • 1 x micro SD slot
Expansion	• 1 x mPCIe socket, support MSATA • 1 x mPCIe socket, support SIM card 2nd • 1 x M.2 Type 3042 B Key socket, USB interface only, support SIM card 1st
Power Input	• 9V ~ 32VDC input without backup battery power • 12V ~ 32VDC input with backup battery • Ignition detection: enable(Car mode)/disable(DC mode) via dip switch • Reverse protection • Red LED for power status • Blue LED for battery charging
Dimensions (W x H x D)	• 300x158.4x66mm without side mount bracket • 332x158.4x72.5mm with side mount bracket
Environment	
Temperature	• **Operating:** -30°C ~70°C (w/o add on cards) MPT-1000V SKU -10°C ~60°C (w/o add on cards) MPT-1000VN SKU • **Storage:** --40 ~ 85 °C (-40 ~ 185 °F)
Relative Humidity	10 ~ 95% (non-condensing)
Vibration Protection	Operating/Non-operating: 2,26 Grms (5~500 Hz) / MIL-STD-810G composite wheeled condition (Z-axis only)
Shock Protection	Non-operating: Sawtooth: 40G, 11 msec (Z-axis) Operating: Sawtooth: 20G, 11msec (Z-axis)

All specifications are subject to change without prior notice.

1.5 Product View

Front View

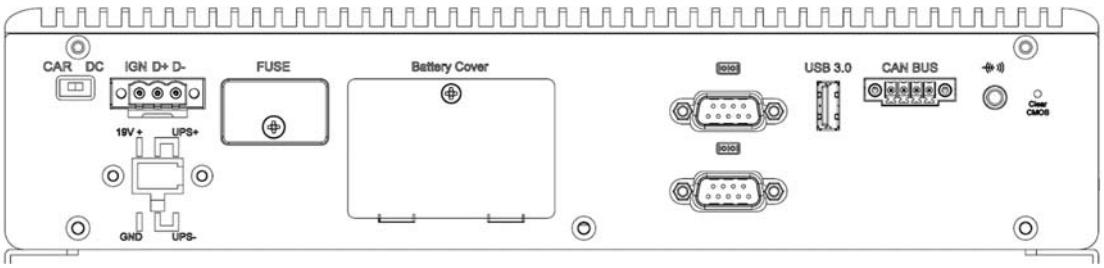


No.	Name	No.	Name
1	Switch for Add-on cards reset	8	VGA Connector
2	Line Input	9	2 x USB 3.1 (#1/$2)
3	Microphone Input	A	RJ45 for GbE
4	DSUB15 for DIDO	B	DSUB9 for COM1/COM2*
5	Micro SD card slot	C	DVI-D Connector
6	LED Indicators: Power (green), Storage (white), WLAN (blue), WWAN (orange/option), Warning of wrong voltage connected (red)	D	SSD Slot
7	System reset switch	E	Switch for power firmware load default

Remarks:
*COM1 (top) supports RS-232/422/485 and COM2 (bottom) supports RS-485 isolated port.

Rear View

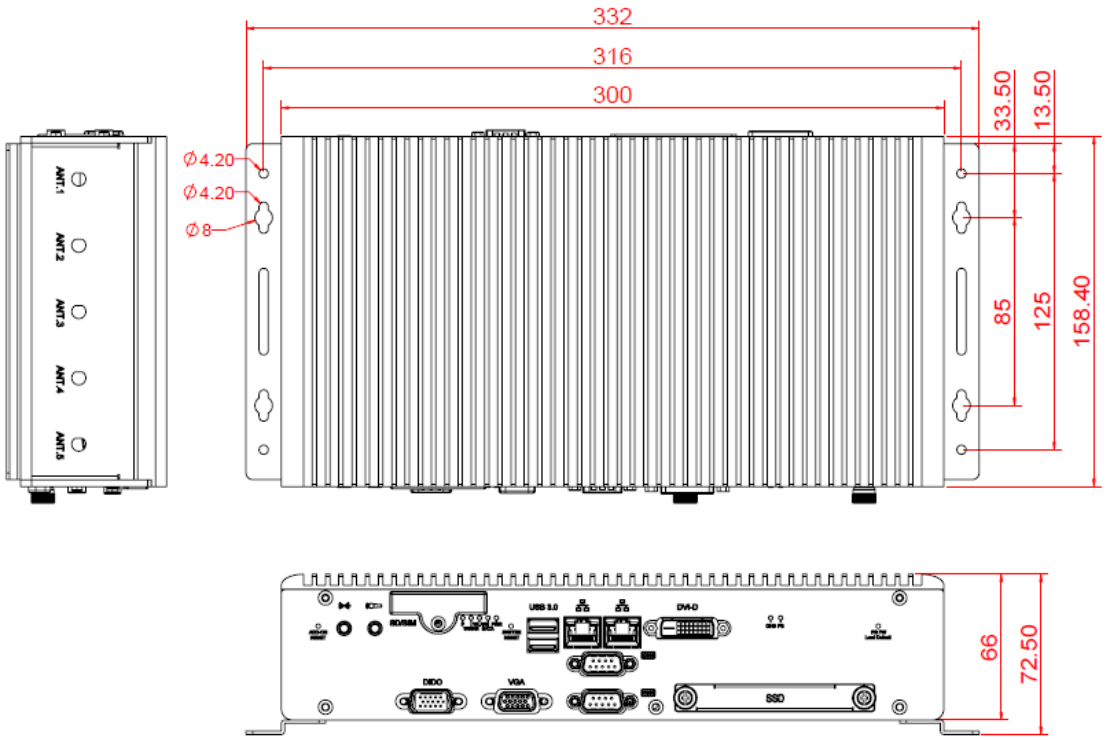
No.	Name	No.	Name
1	CAR/DC power mode switch	6	DSUB9 for COM3/4 RS232
2	Terminal block for power input and IGN 22	7	USB 3.0 port
3	ATX connector for external battery and 19VDC	8	Terminal block for 2 masters CANbus
4	Blade fuse socket with cover	9	Audio jack for line out
5	RTC battery access	0	Switch for clear CMOS



Oblique View

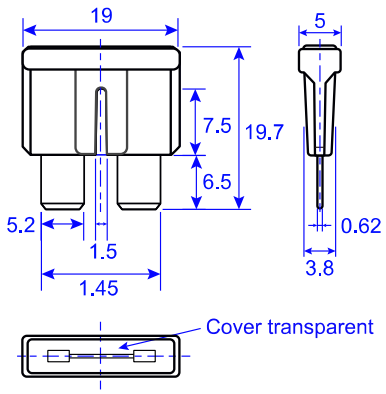
1.6 Dimensions

Unit: mm



1.7 Fuse Dimensions

Littlefuse® TAC ATO® Style Blade 15A, 58V DC



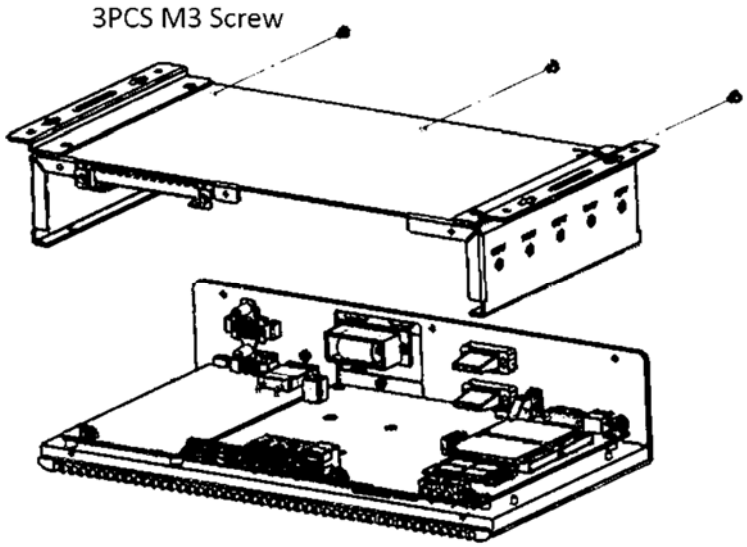
Chapter 2
Hardware Configuration

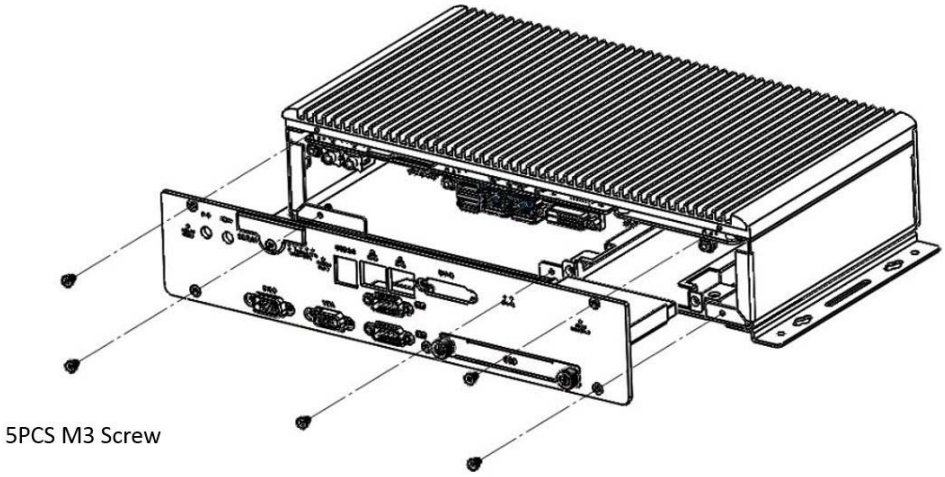
The information provided in this chapter includes:

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

2.1 Essential Installations

In general, in order to access the motherboard components, for example the memory slot, the mPCIe slots or M.2 slot, there are a total of seven screws that need to be removed first. After doing the necessary installation, replacement or changes, put the screws back and tighten them. Please refer to the two pictures below for the location of the screws to be removed.





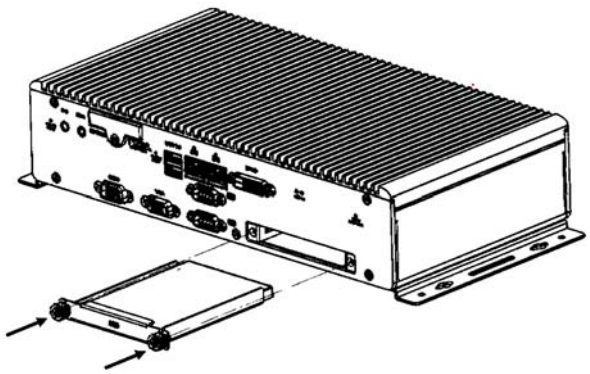
2.1.1 Memory Module Installation

The memory of the MPT-1000V is onboard type, there is no additional memory socket for expansion.

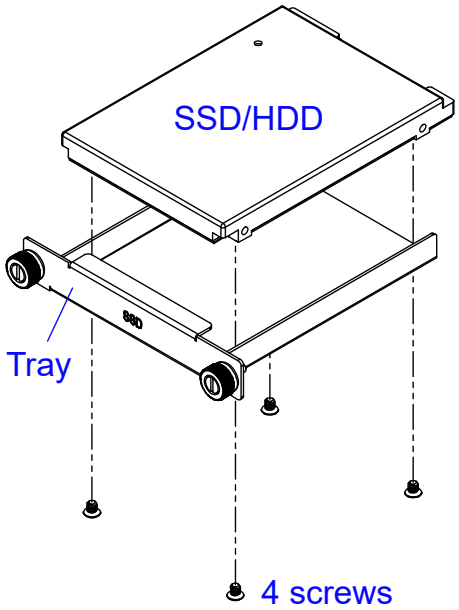
2.1.2 SSD Storage Installation

Installing the SSD drive

1. Release the two screws shown below and pull out the SSD tray.

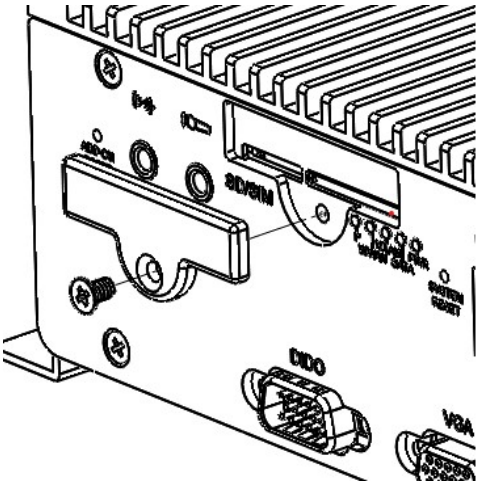


2. There are four screws that are to be loosen or tighten in the SSD drive during replacement as shown below.

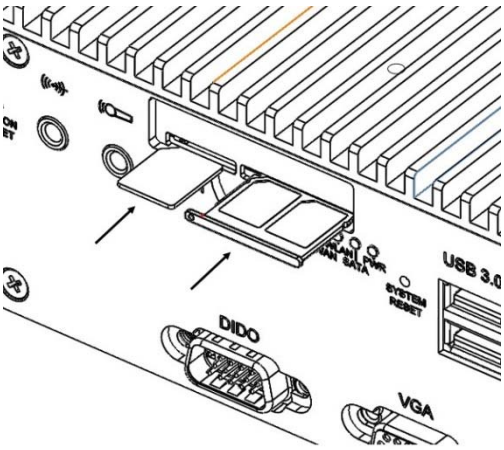


2.1.3 SIM and Micro-SD Card Installation

1. Release the screw of the cover of the SIM/Micro-SD card slots.

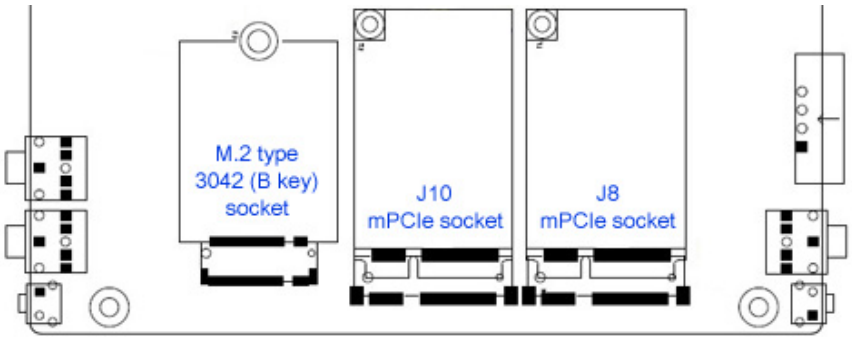


2. Insert the card(s) in their respective slots and push the card with your finger. Replace the cover when done. Or, to release the card(s), push the card again.

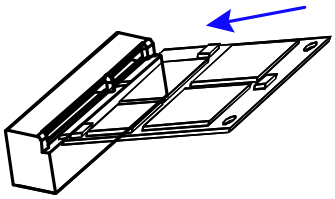


2.1.4 Mini-PCIe & M.2 Network Cards Installation

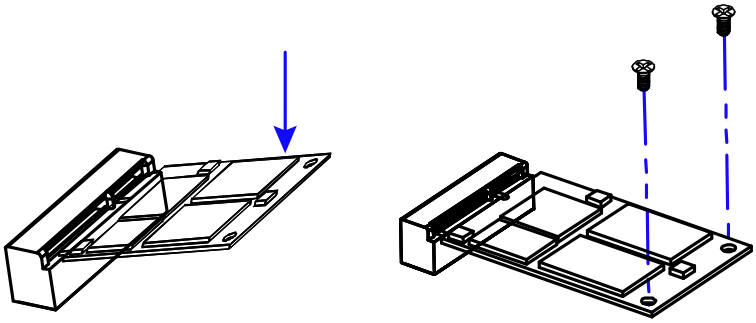
After you have removed the chassis top cover, you will be able to access the mini-PCIe slots and M.2 slot. Their locations are shown below.



1. Locate the mini-PCIe slot and align the key of the mini-PCIe card with that of the slot. Insert the card slantwise. (Insert the M.2 network card in the same way.)



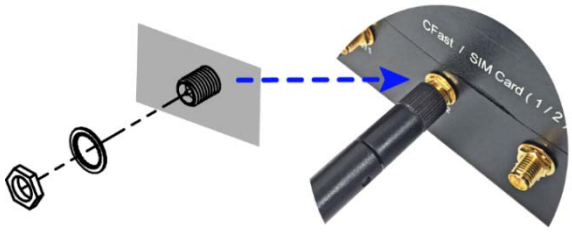
2. Push the mini-PCIe card down and hold it to the board with 2 flat head screws. (For the M.2 network card, use a round head screw.)



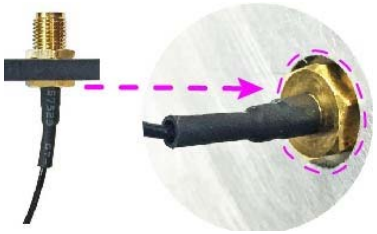
2.1.5 WiFi / 3G / 4G Antenna Installation

Thread the WiFi / 3G / 4G antenna extension cable through an antenna hole on the chassis and fasten the antenna as shown below. Apply adhesive to the edge of the hex nut behind the front I/O cover to prevent the extension cable from falling if the cable becomes loose.

1. Thread and fasten the hex nut and the washer. Then install the antenna.



2. Apply adhesive around here.



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

Location of antenna holes

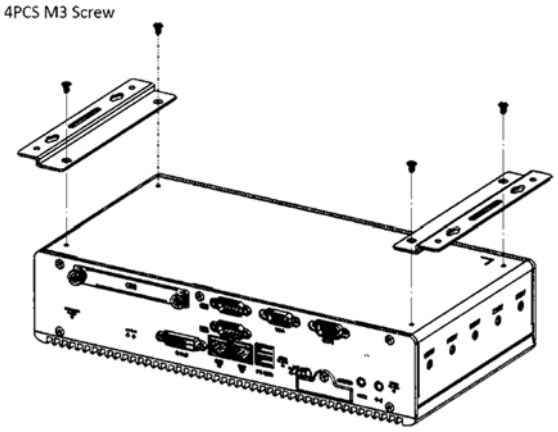
2.1.6 Mounting Brackets Installation

Requirements

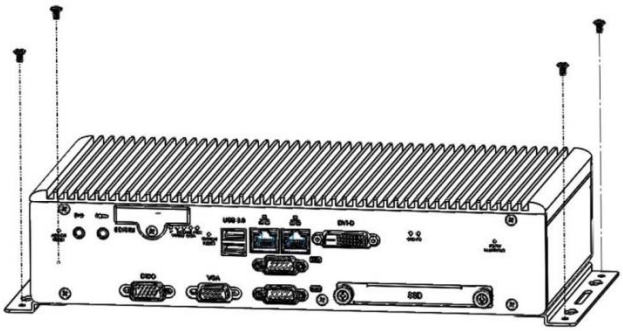
When mounting, ensure that you have enough room for power and signal cable routing. The method of mounting must be able to support weight of the MPT-1000V plus the suspend weight of all the cables to be attached to the system.

Installation instructions:

1. Turn your MPT-1000V upside down, and use the four screws shown below to attach or remove the mounting brackets from MPT-1000V.



2. To mount the MPT-1000V to the mounting surface, use four screws to secure the system.

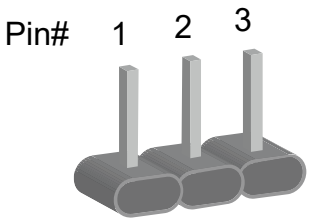


2.2 Setting the Jumpers

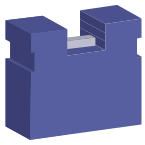
Set up and configure your MPT-1000V by using jumpers for various settings and features according to your needs and applications. Contact your supplier if you have doubts about the best configuration for your use.

2.2.1 How to Set Jumpers

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

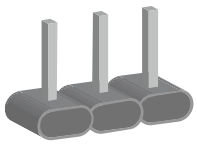
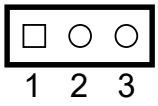
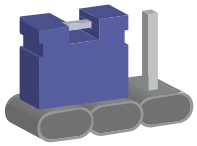
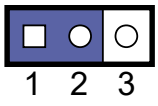
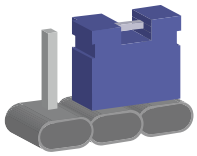
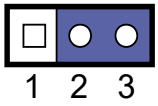


A 3-pin jumper



A jumper cap

Refer to the illustration below to set jumpers.

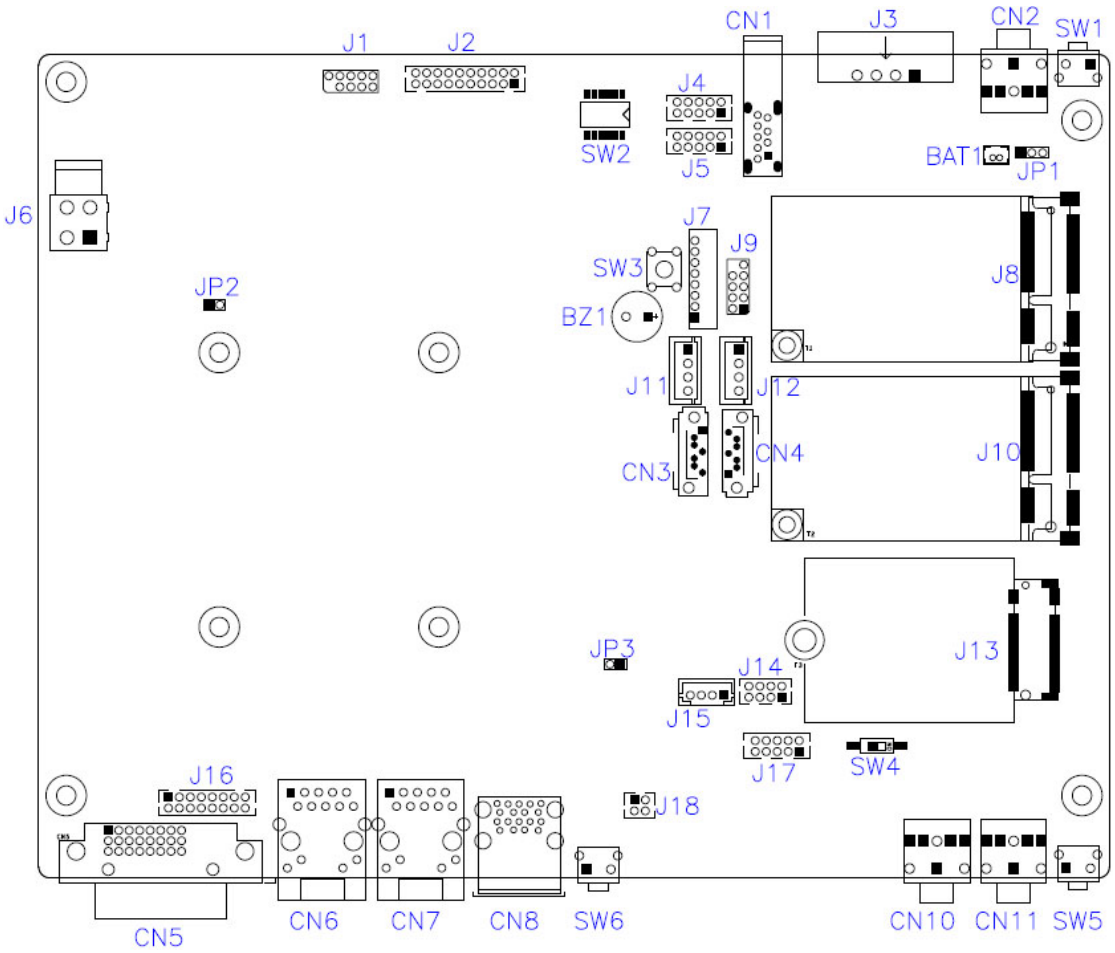
Pin closed	Oblique view	Schematic illustration in the manual
Open		
1-2		
2-3		

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

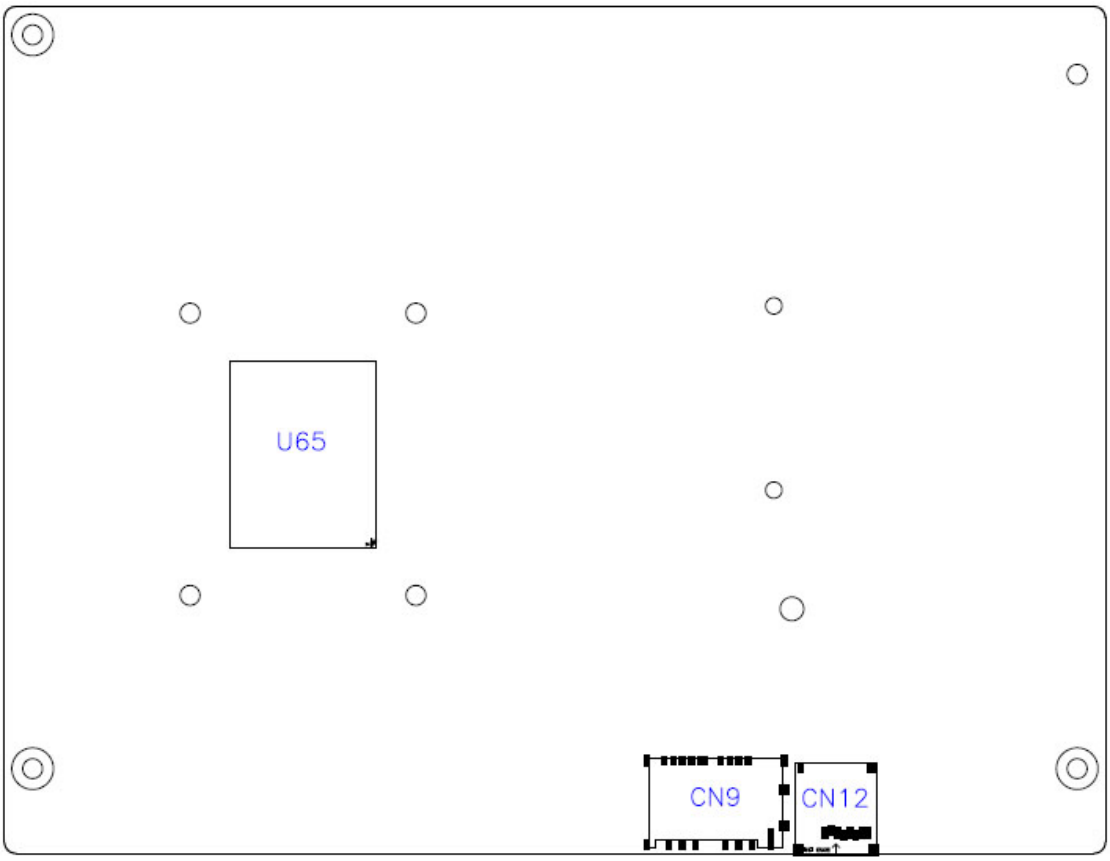
When a jumper cap is removed from two jumper pins, this jumper is **open**, i.e. turned **Off**.

2.3 Jumper & Connector Locations on Motherboard

Motherboard: MBT-1002



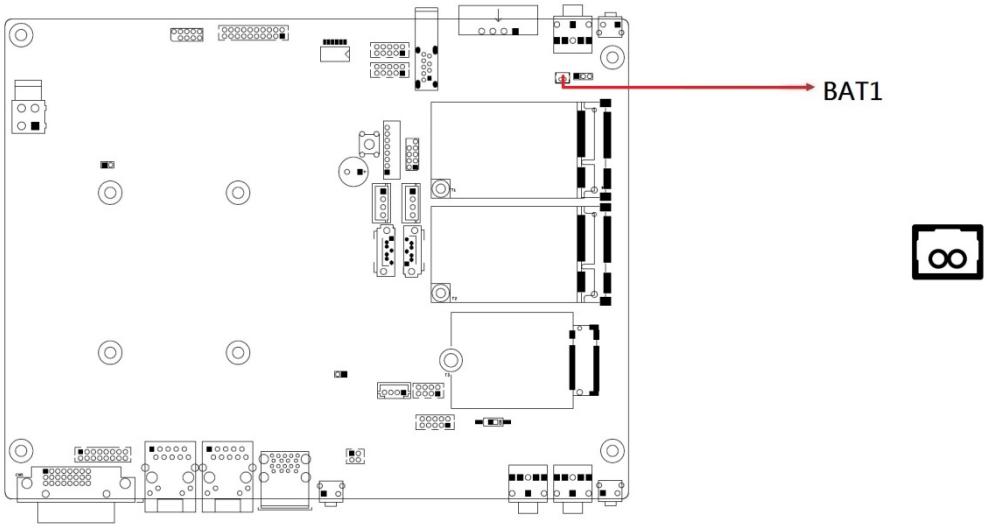
MBT-1002 – top view



MBT-1002 – bottom view

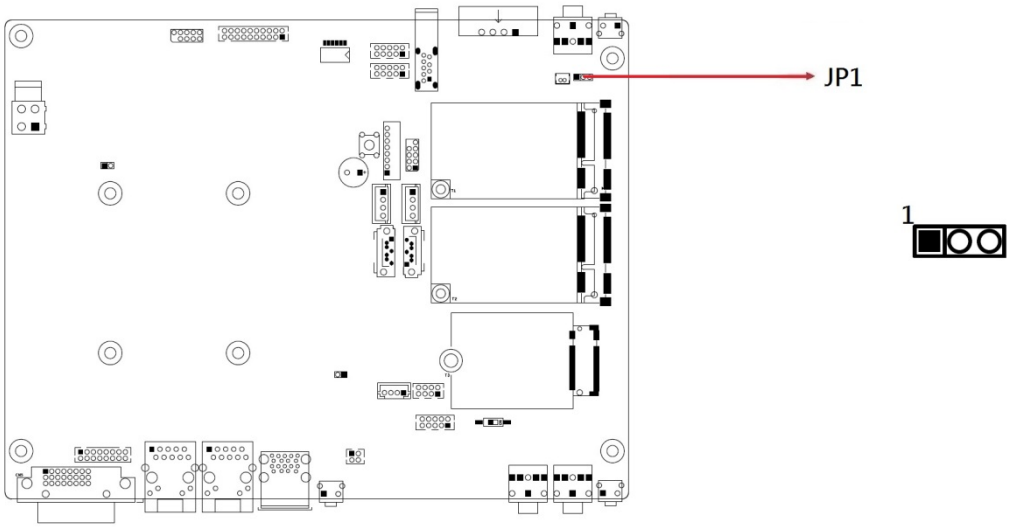
2.4 Jumpers Quick Reference

2.4.1 BAT1: RTC Battery



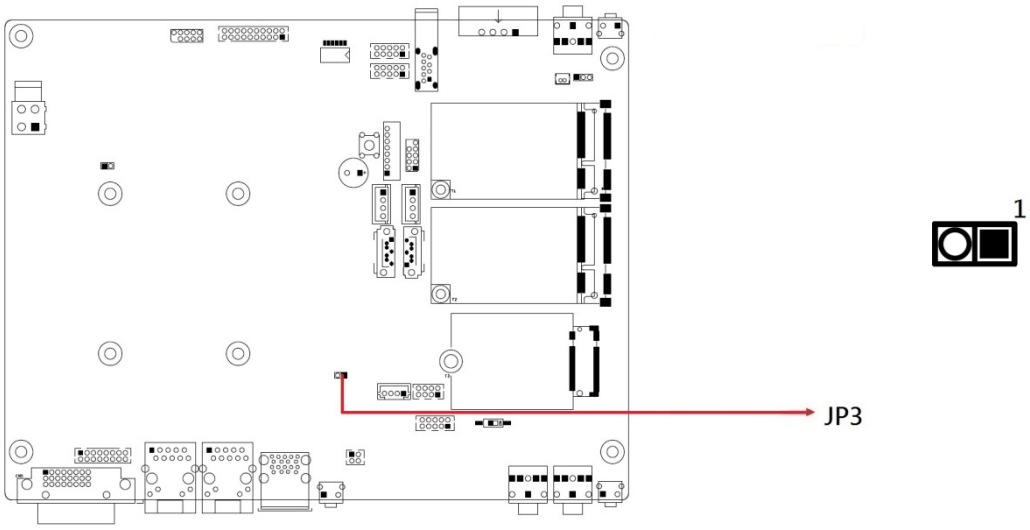
Pin	Assigment
1	Power
2	Ground

2.4.2 JP1: Clear ME Contents



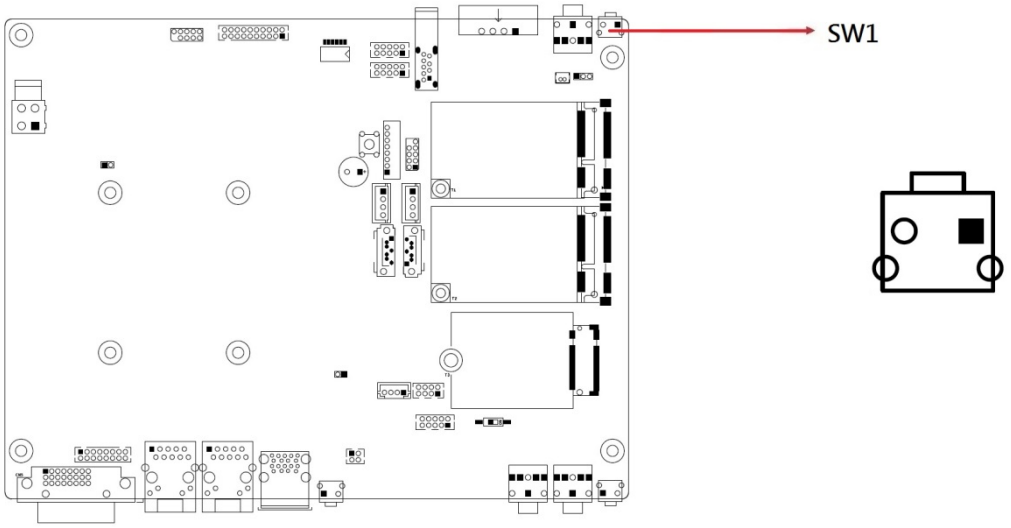
Function	Pin closed	Illustration
Normal (default)	1-2	1
Clear ME RTC Register	2-3	1

2.4.3 JP3: COM2 Terminator Enable

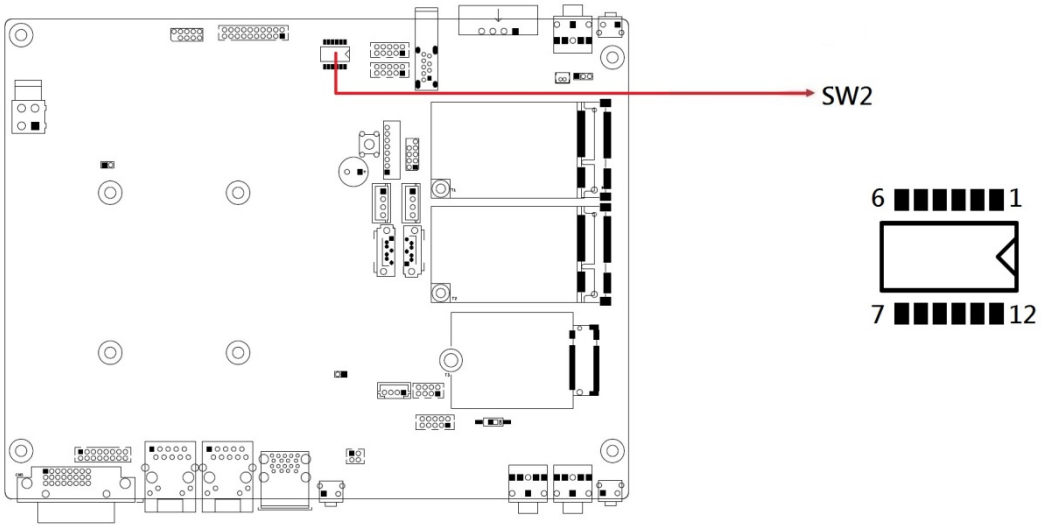


Function	Pin Setting
Disabled	Open
Enabled (default)	Short/Closed

2.4.4 SW1: RTC Reset

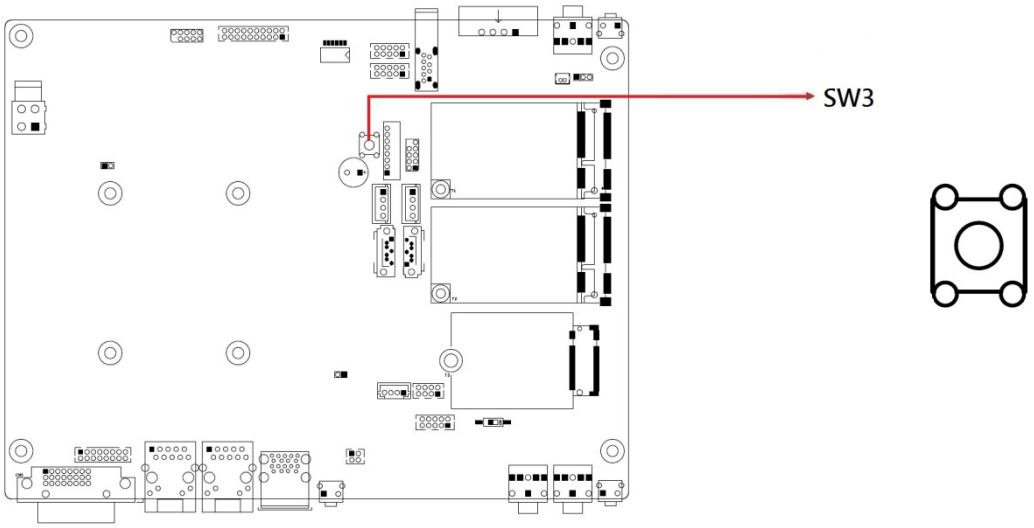


2.4.5 SW2: For DIO Verification Used(Default off)

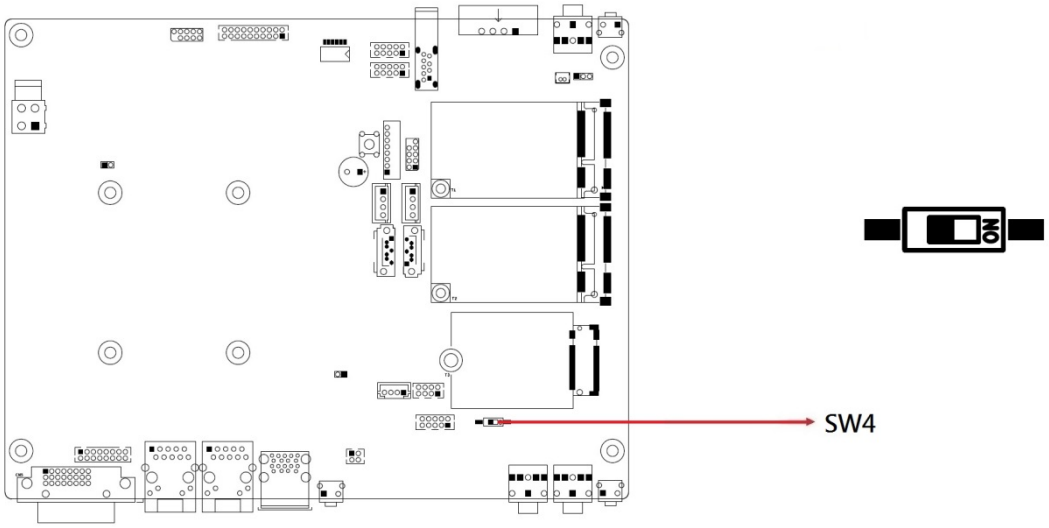


Pin	Assigment
1	DO0
2	DO1
3	DO2
4	DO3
5	DO4
6	DO5

2.4.6 SW3: ATX Power ON Switch

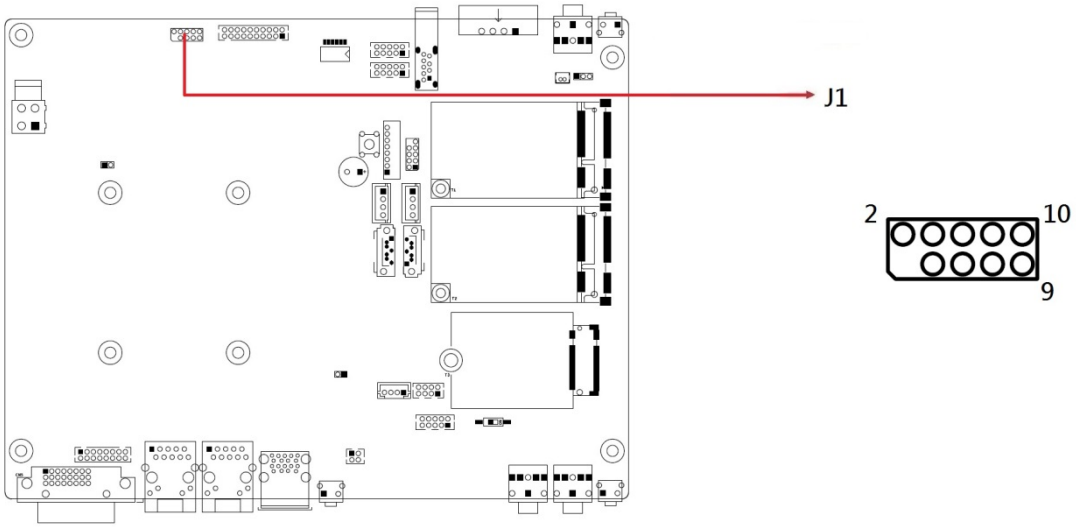


2.4.7 SW4: SIM Card Select

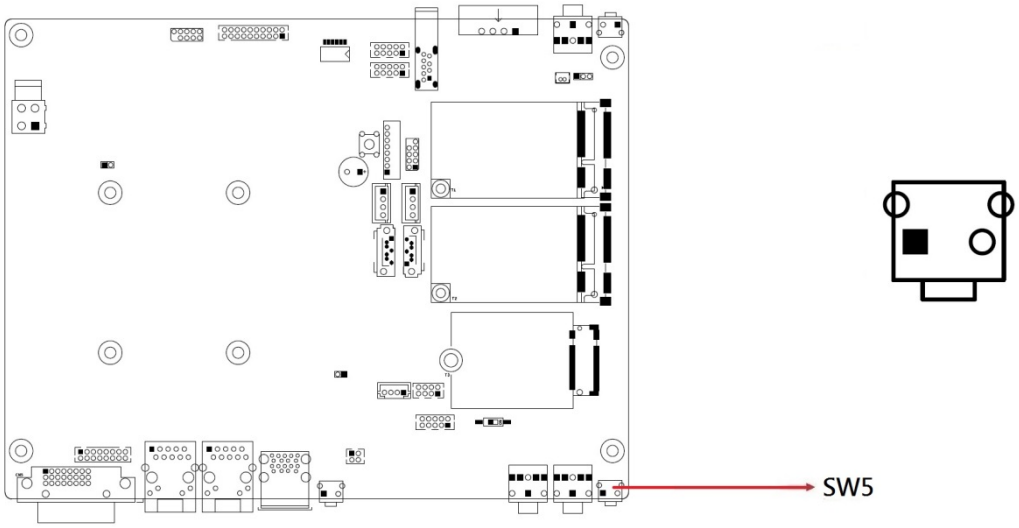


Function	Setting
From M.2(Default)	OFF
From Mini PCIe	ON

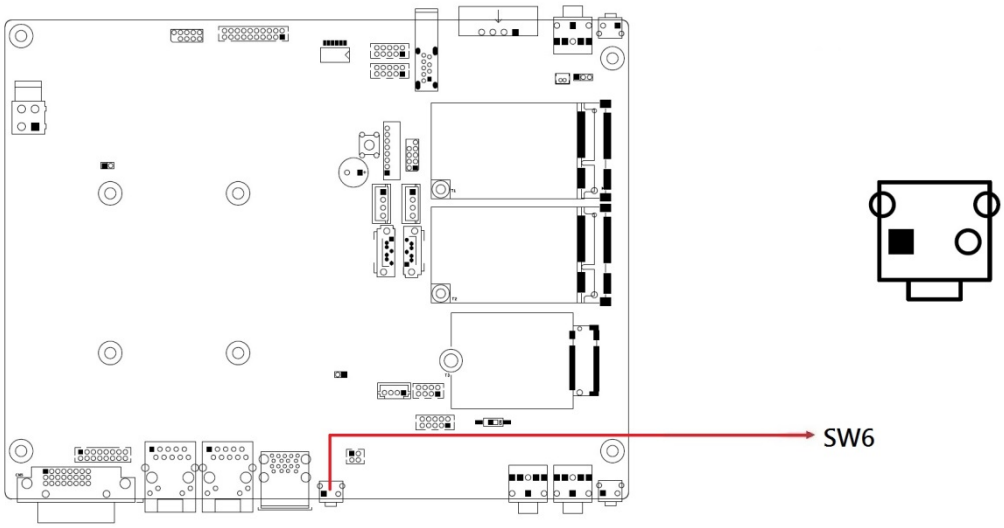
2.4.8 J1: Pin Header for SPI Debug



2.4.9 SW5: Expansion Sockets Reset Button

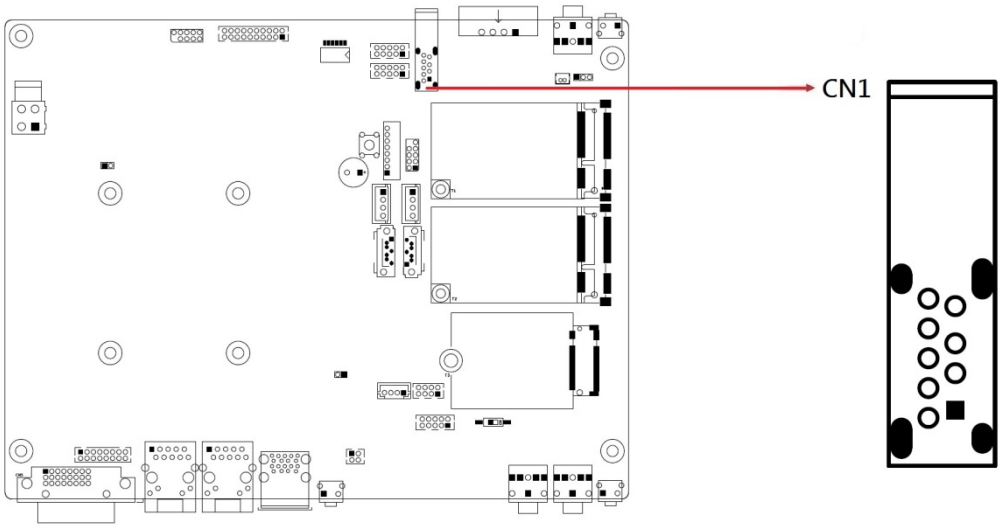


2.4.10 SW6: System Reset Button

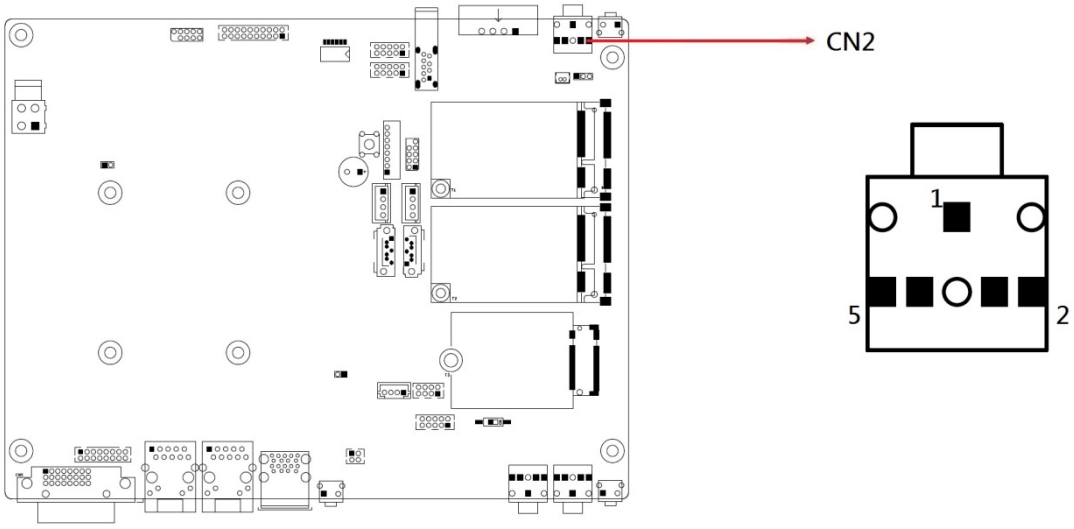


2.5 Connectors Quick Reference

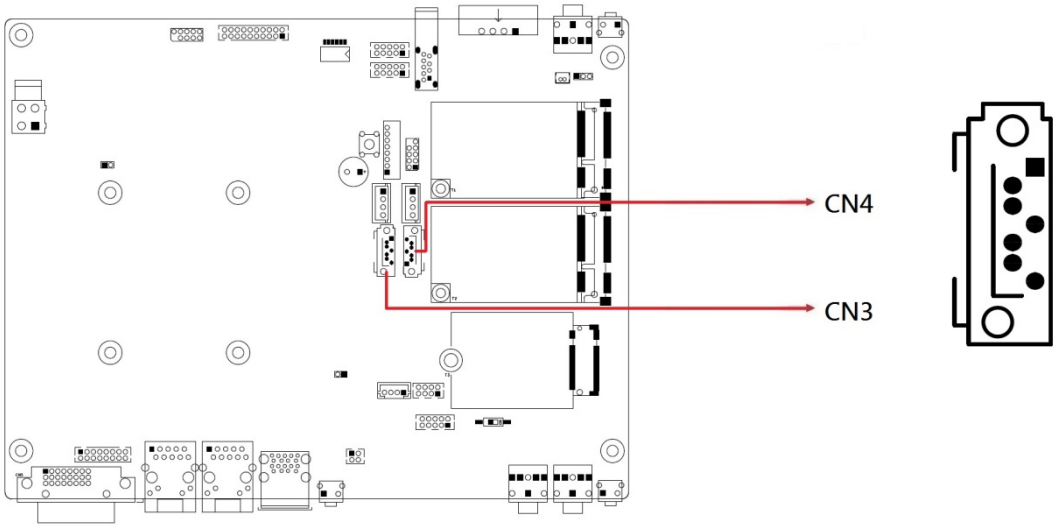
2.5.1 CN1: USB 3.0 Port



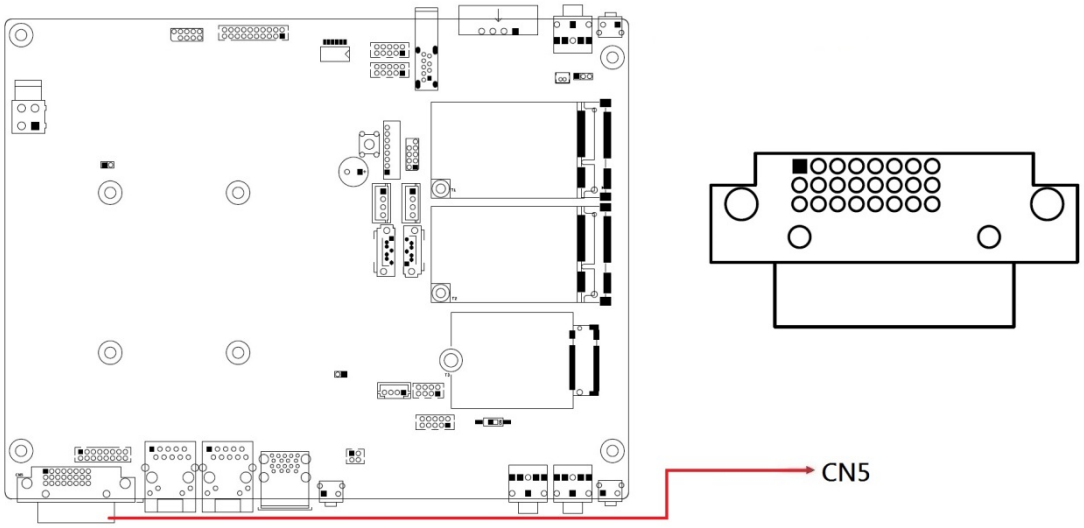
2.5.2 CN2: Audio Line Out



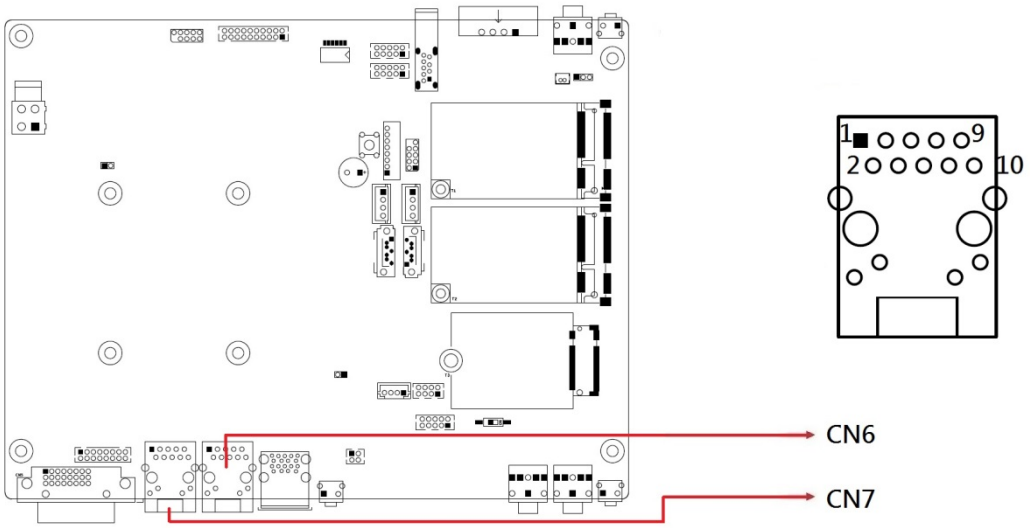
2.5.3 CN3, CN4: SATA Connectors



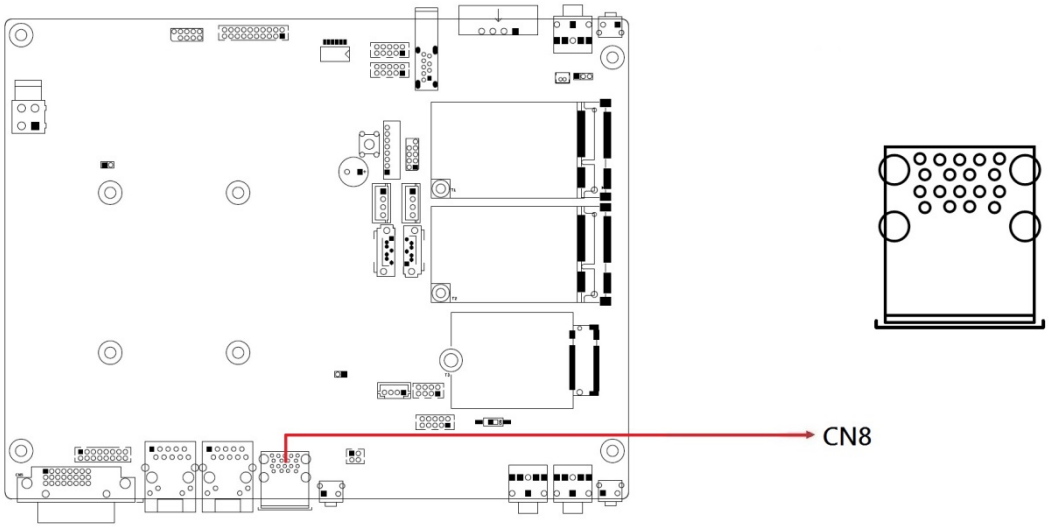
2.5.4 CN5: DVI-D Connector



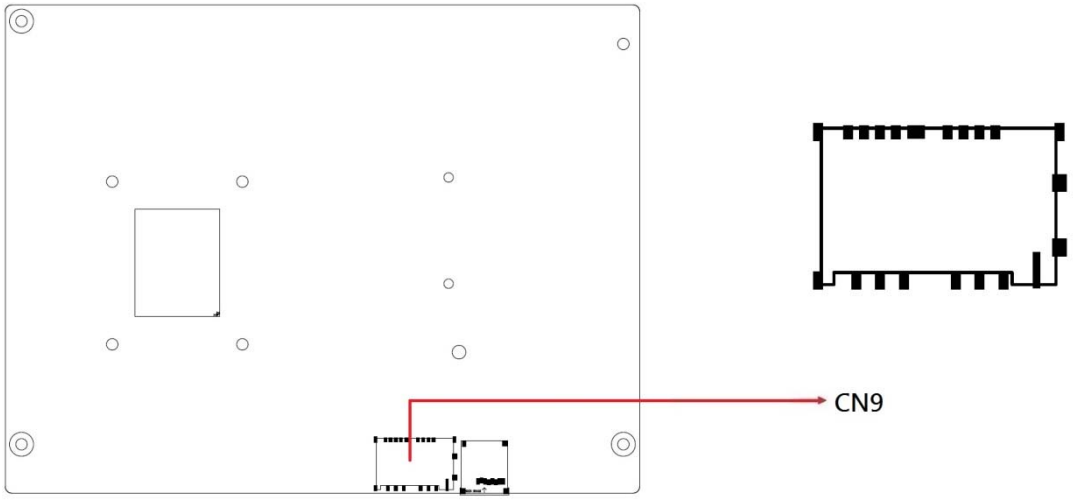
2.5.5 CN6, CN7: GbE LAN Port (Intel WGI210IT)



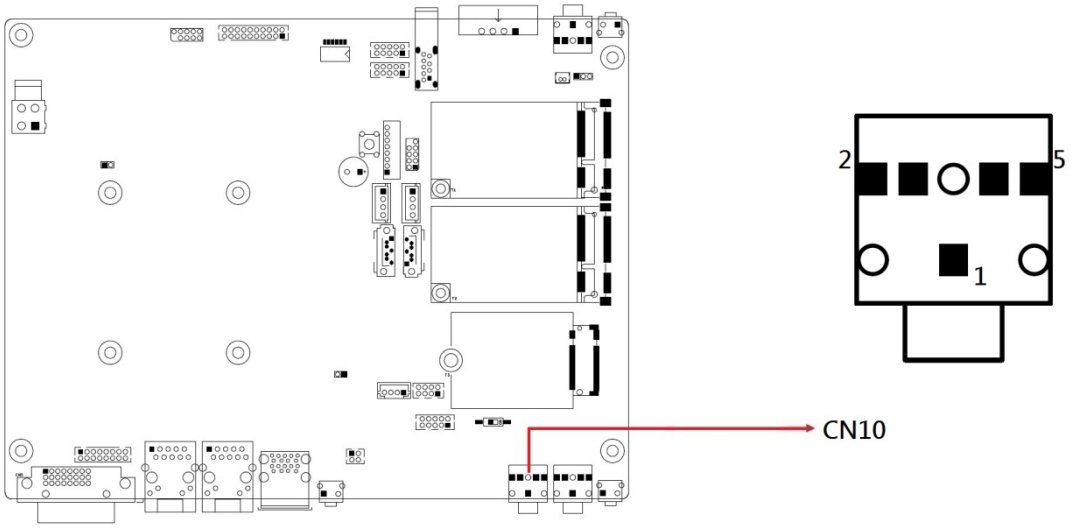
2.5.6 CN8: Dual USB 3.0 Ports



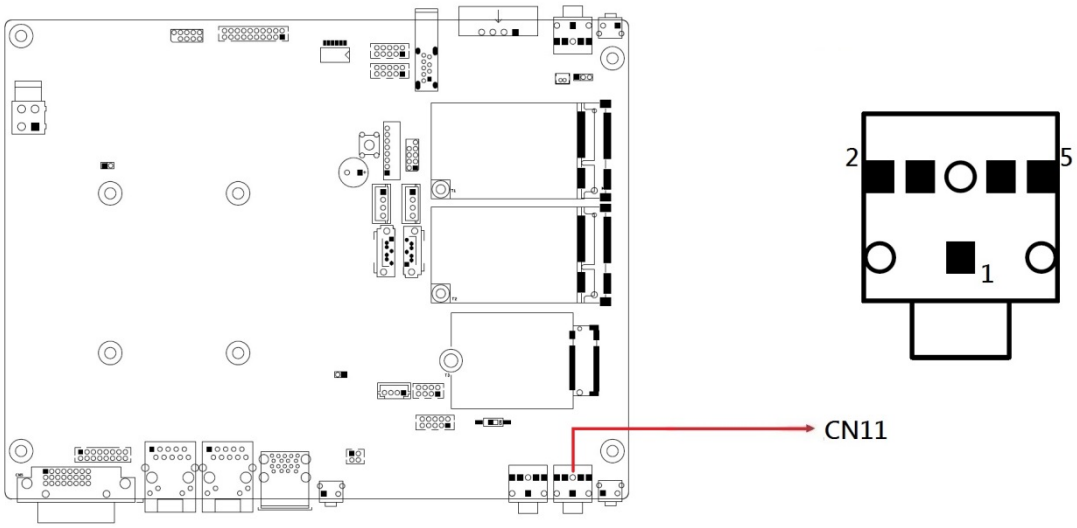
2.5.7 CN9: Dual Micro SIM Card Socket



2.5.8 CN10: Audio Mic In



2.5.9 CN11: Audio Line In



2.5.10 COM2 and DIDO Front Connectors

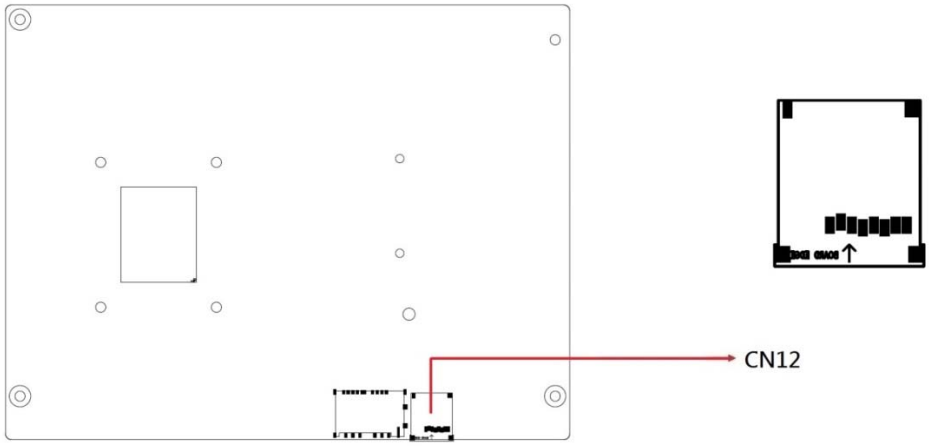
Pin assignments for COM2

Pin	Assigment
1	D-
2	D+
5	Ground

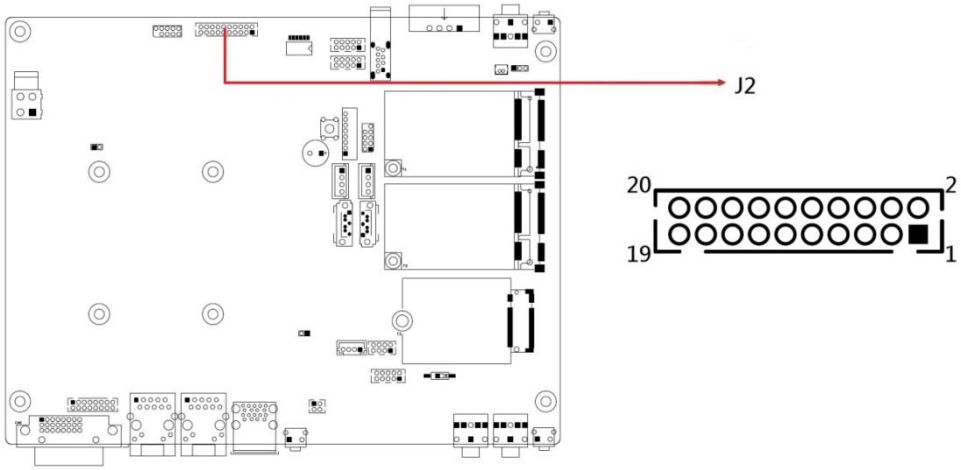
Pin assignments for DIDO

Pin	Assigment	Pin	Assigment
1	DI0	9	GND
2	DI1	10	DO5
3	DI2	11	DO0
4	DI3	12	DO1
5	DI4	13	DO2
6	DI5	14	DO3
7	GND	15	DO4
8	5V		

2.5.11 CN12: Micro SD Slot

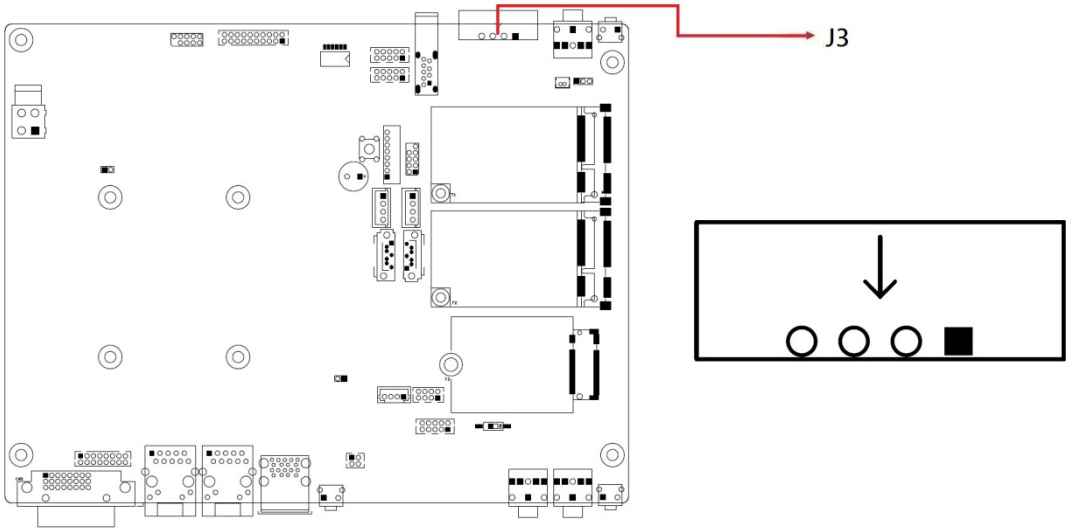


2.5.12 J2: Digital I/O Connector (Hirose DF11-20S-PA66H)



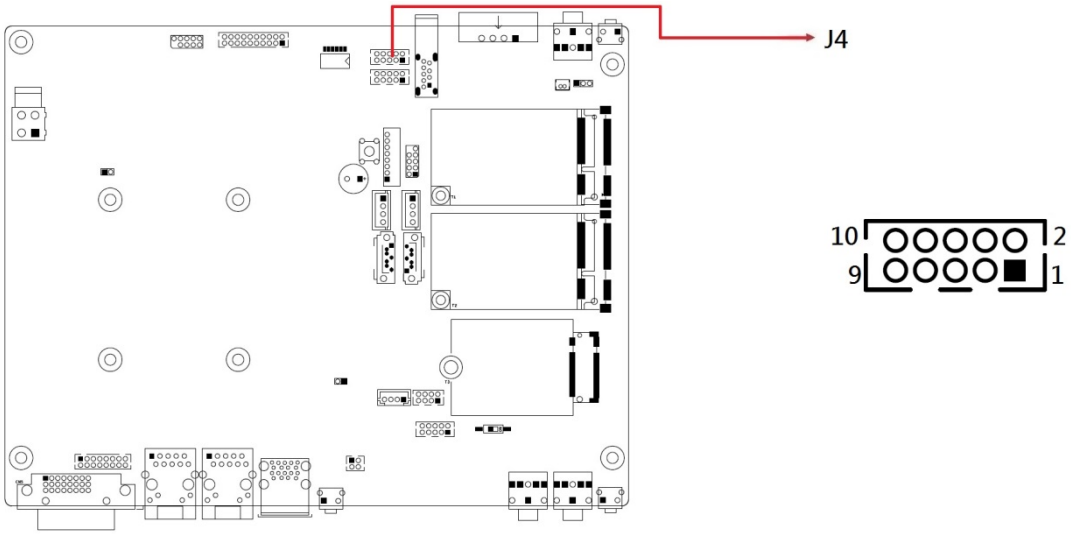
Pin	Assigment	Pin	Assigment
2	DO0	1	DI0
4	DO1	3	DI1
6	DO2	5	DI2
8	DO3	7	DI3
10	DO4	9	DI4
12	DO5	11	DI5
14	VCC_COM	13	GND_ISO_DIO
16	GND_ISO_DIO	15	GND_ISO_DIO
18	GND_ISO_DIO	17	GND_ISO_DIO
20	GND_ISO_DIO	19	GND_ISO_DIO

2.5.13 J3: CAN Bus Port (Dinkle ECH350RM-04P)



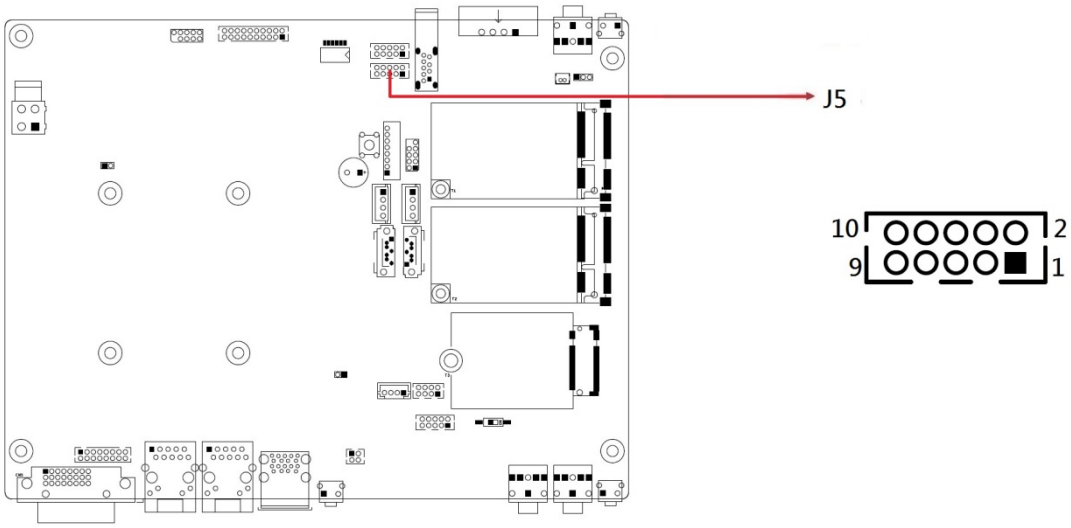
Pin	Assigment
1	CAN_DH1
2	CAN_DL1
3	CAN_DH2
4	CAN_DL2

2.5.14 J4: COM3 RS-232 Port (HRS_DF11-10DP-2DSA)



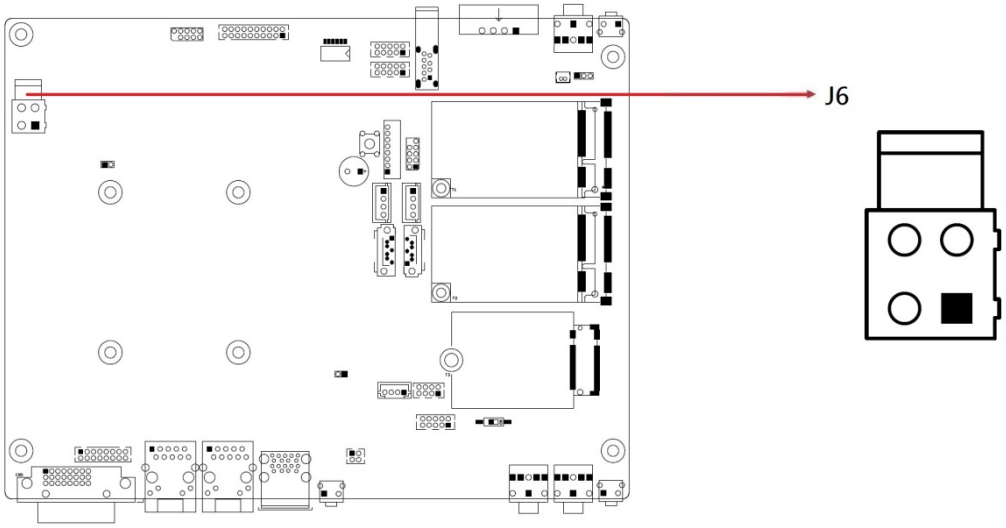
Pin	Assigment	Pin	Assigment
1	DCD	2	SIN
3	SOUT	4	DTR
5	GND	6	DSR
7	RTS	8	CTS
9	RI		

2.5.15 J5: COM4 RS-232 Port (HRS_DF11-10DP-2DSA)



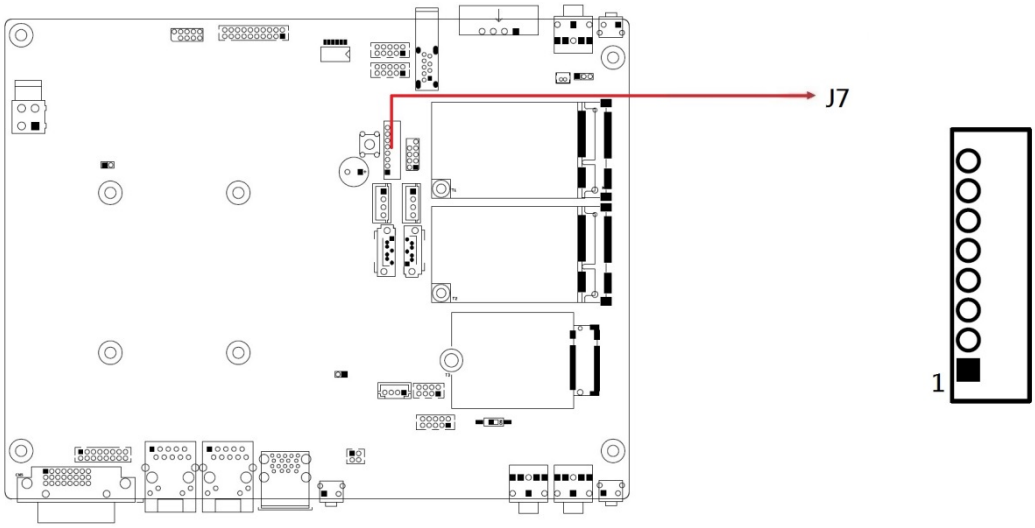
Pin	Assigment	Pin	Assigment
1	DCD	2	SIN
3	SOUT	4	DTR
5	GND	6	DSR
7	RTS	8	CTS
9	RI		

2.5.16 J6: 19VDUAL Input Connector (ATX4PT-NY46)



Pin	Assigment	Pin	Assigment
1	Ground	2	Ground
3	19VDUAL	4	19VDUAL

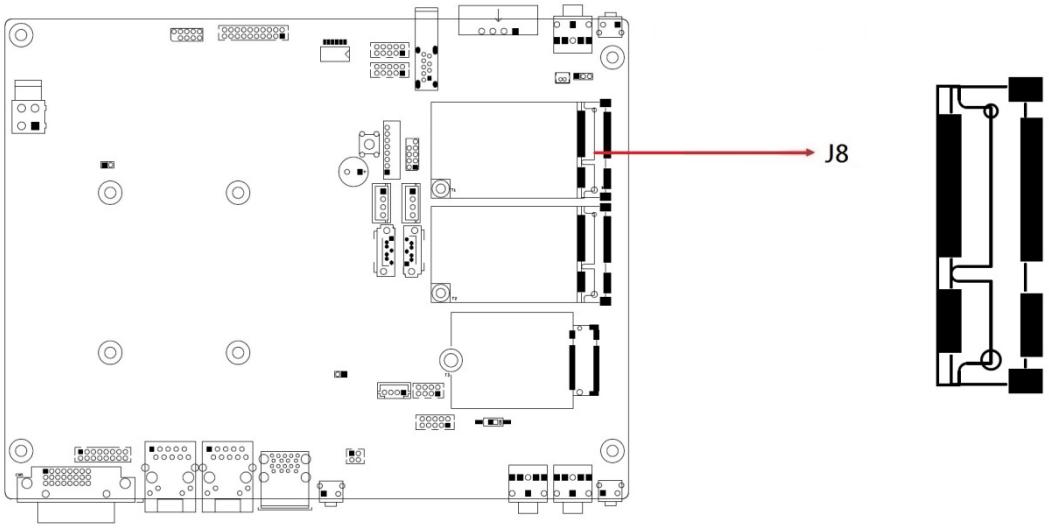
2.5.17 J7: Connector to Power Board (Hirose DF3-8P-2DSA)



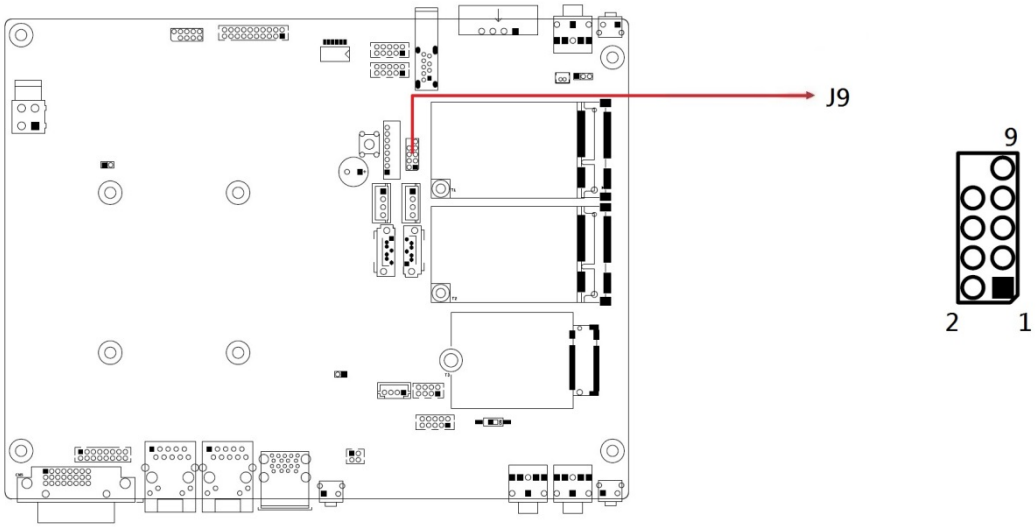
Pin	Assigment	Pin	Assigment
1	GND	5	SINF
2	PWR_BTN#	6	SOUTF
3	POWER_OK	7	VCC3_3 (2A)
4	PS_ON#		

2.5.18 J8: mPCIe Socket

J8 supports PCIe, USB 2.0 + SMBus + mSATA.

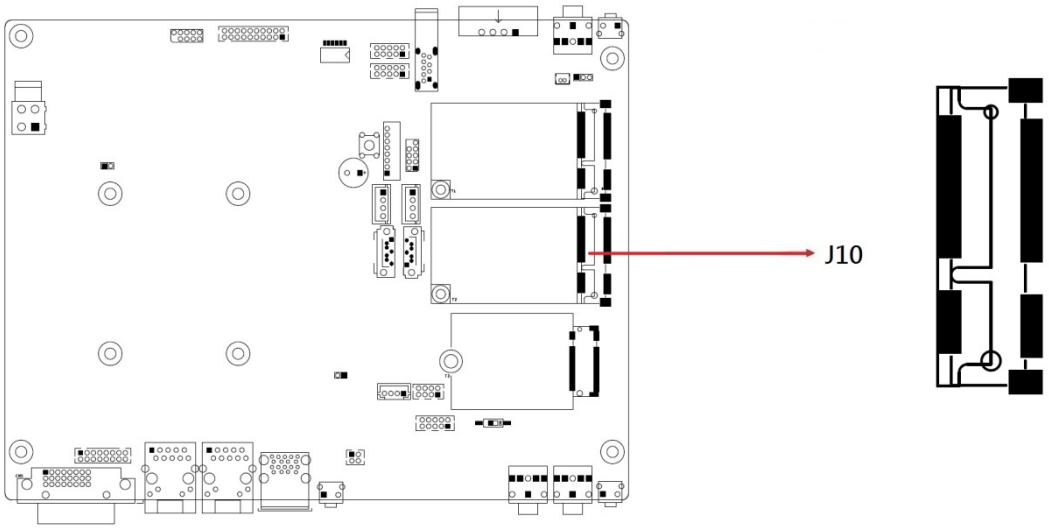


2.5.19 J9: 80 Port

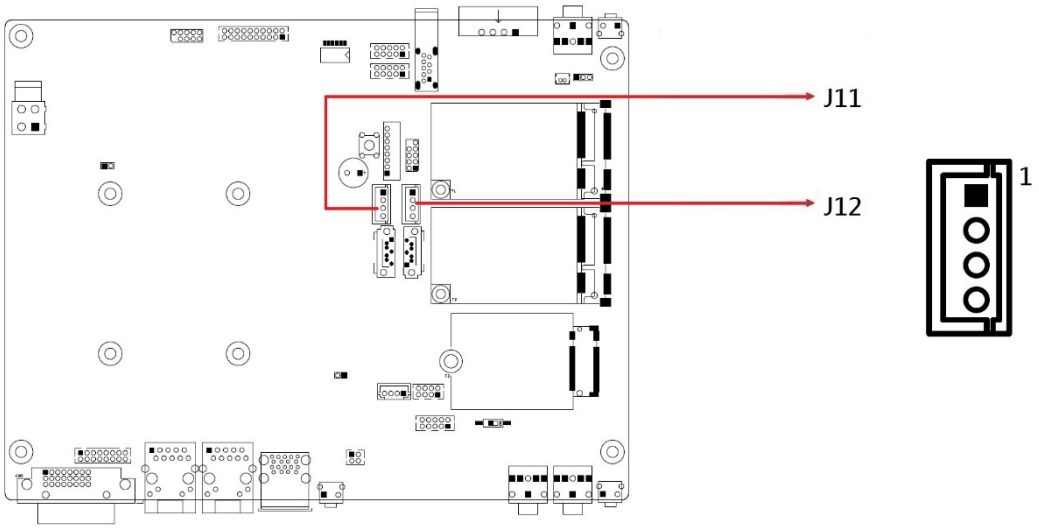


2.5.20 J10: mPCIe Socket

J10 supports PCIe, USB 2.0+ SMBus + SIM card.



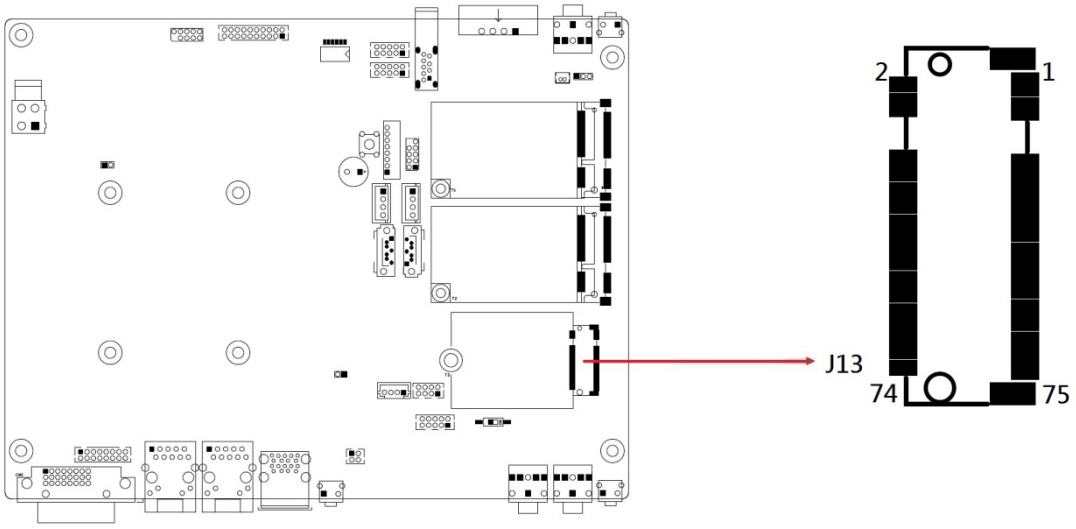
2.5.21 J11, J12: SATA Power Connectors (JST B4B-XH-A)



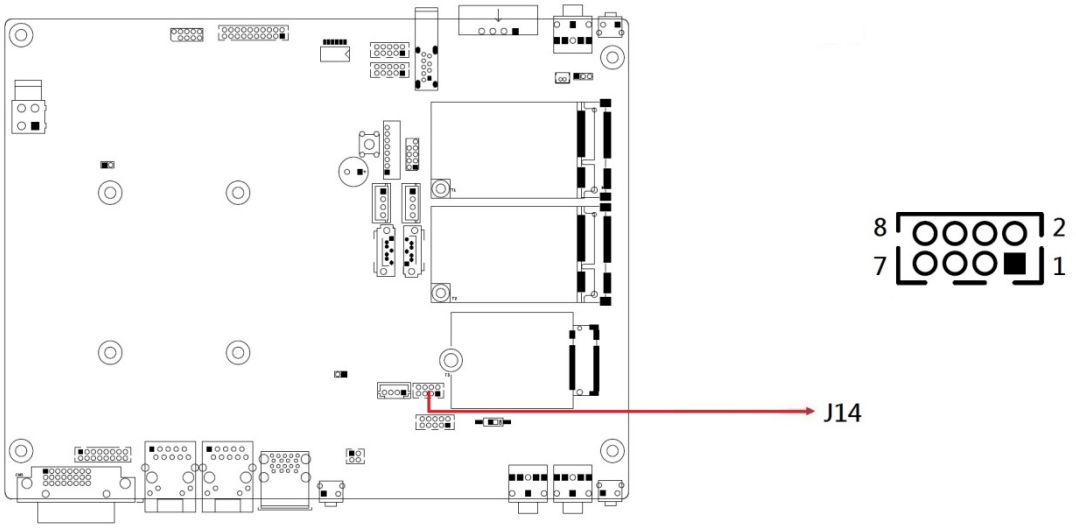
Pin	Assigment
1	VCC5 (1A)
2	GND
3	GND
4	VCC12 (1A)

2.5.22 J13: M.2 Type 3042 (B Key) Socket

J13 is for USB3.0 +USB 2.0 and 2 SIM cards.
For pin definition, refer to Sierra EM7455 and EM7565.

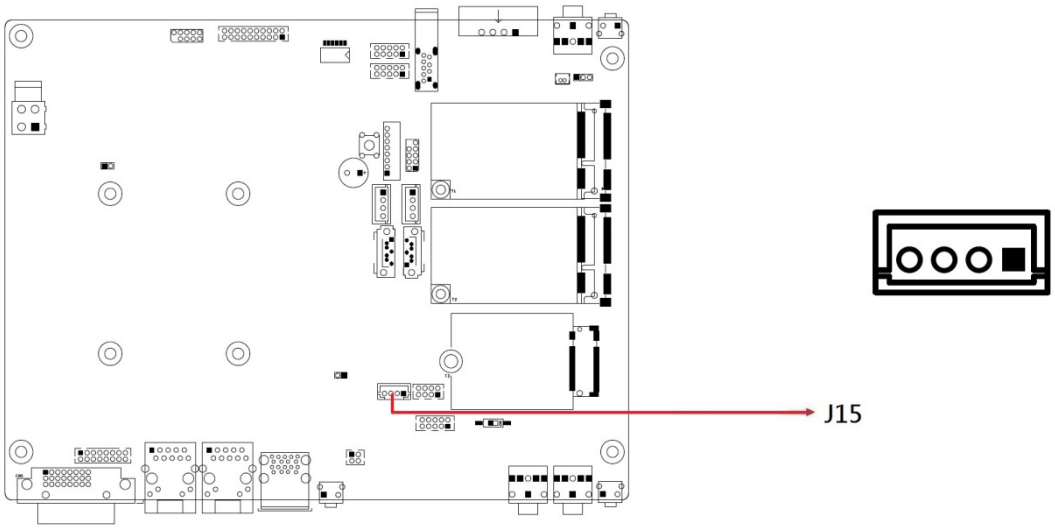


2.5.23 J14: USB 2.0 Connector (Hirose DF11-8S-PA66H)



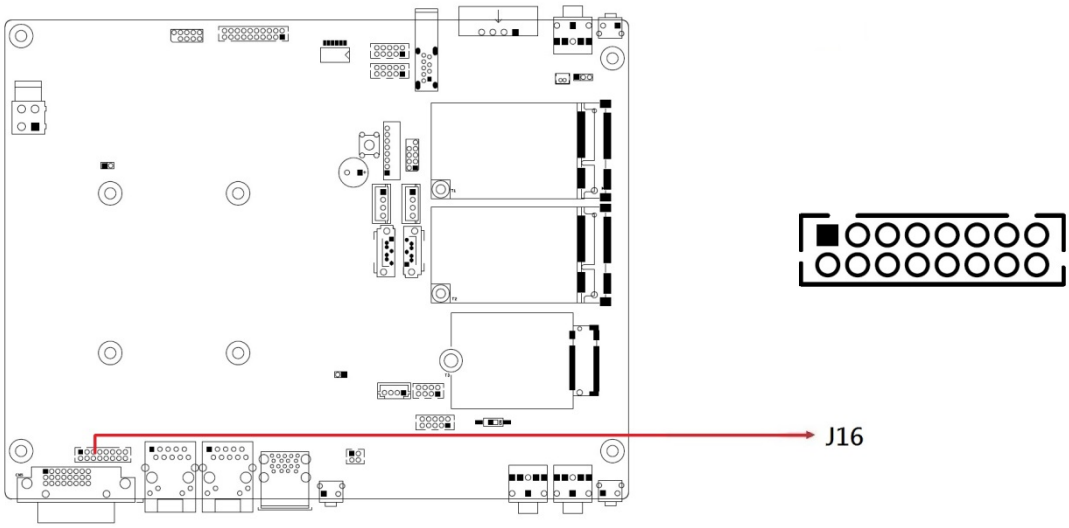
Pin	Assigment	Pin	Assigment
1	VCC (0.5A)	2	GND
3	D0-	4	D1+
5	D0+	6	D1-
7	GND	8	VCC (0.5A)

2.5.24 J15: USB 2.0 Connector (JST B4B-PH-K-S)



Pin	Assigment
1	VCC (0.5A)
2	D-
3	D+
4	GND

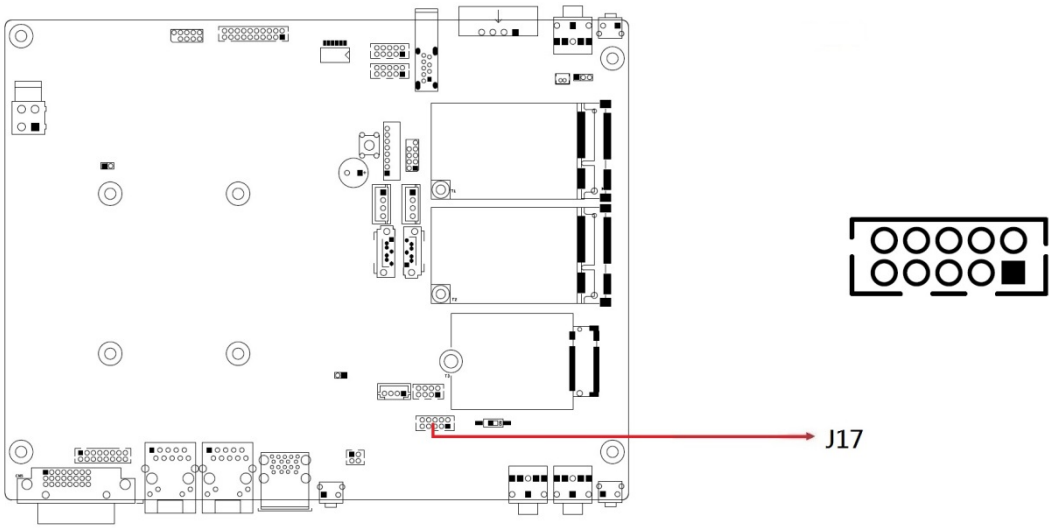
2.5.25 J16: VGA Connector (Hirose DF11-16DP-2DSA)



Pin	Assigment	Pin	Assigment
1	R	2	VCC
3	G	4	GND
5	B	6	NONE
7	NONE	8	DDC_DATA
9	GND	10	HSYNC
11	GND	12	VSYNC
13	GND	14	DDC_CLK
15	GND	16	NONE

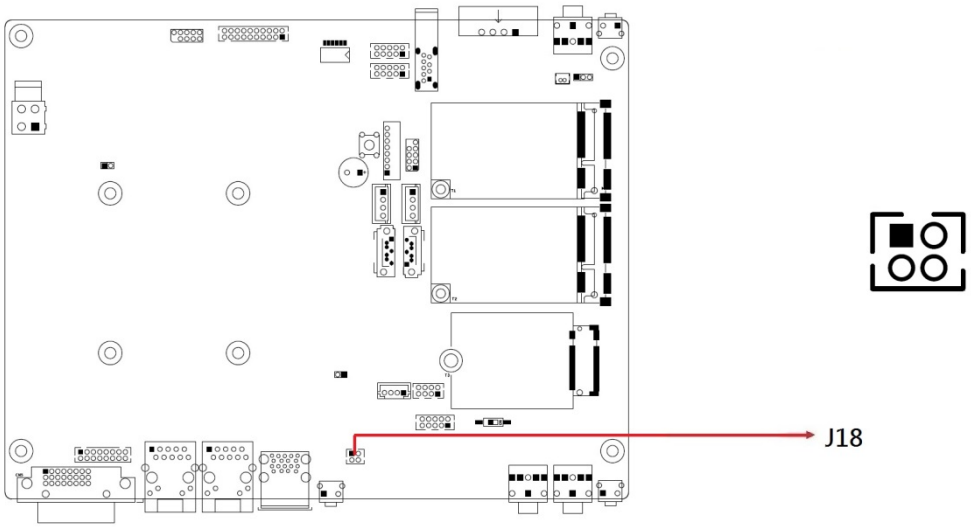
2.5.26 J17: COM1 RS-232/422/485 Port (Hirose DF11-10DP-2DSA)

J17 COM1 port is jumper-less and configurable in the BIOS.



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

2.5.27 J18: COM2 RS-485 Port (Hirose DF11-4DP-2DSA)



Pin	Assigment	Pin	Assigment
1	Data-	2	Data+
3	GND_ISO_COM2	4	GND_ISO_COM2

2.5.28 LED1: Power LED

2.5.29 LED2: SATA LED

2.5.30 LED3: Blue Color LED for WLAN Activity Status

2.5.31 LED4: Orange Color LED for WWAN Activity Status

2.5.32 LED5: Firmware Programming LED

2.5.33 LED6: Expansion Sockets Power Indicator

Chapter 3
Drivers Installation

The information provided in this chapter includes:

- Intel® Chipset Software Installation Utility
- VGA Driver Installation
- HD Audio Driver Installation
- LAN Driver Installation
- Intel® Trusted Execution Engine Installation
- Intel® Serial I/O Driver Installation
- G-Sensor Driver Installation

3.1 Introduction

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

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

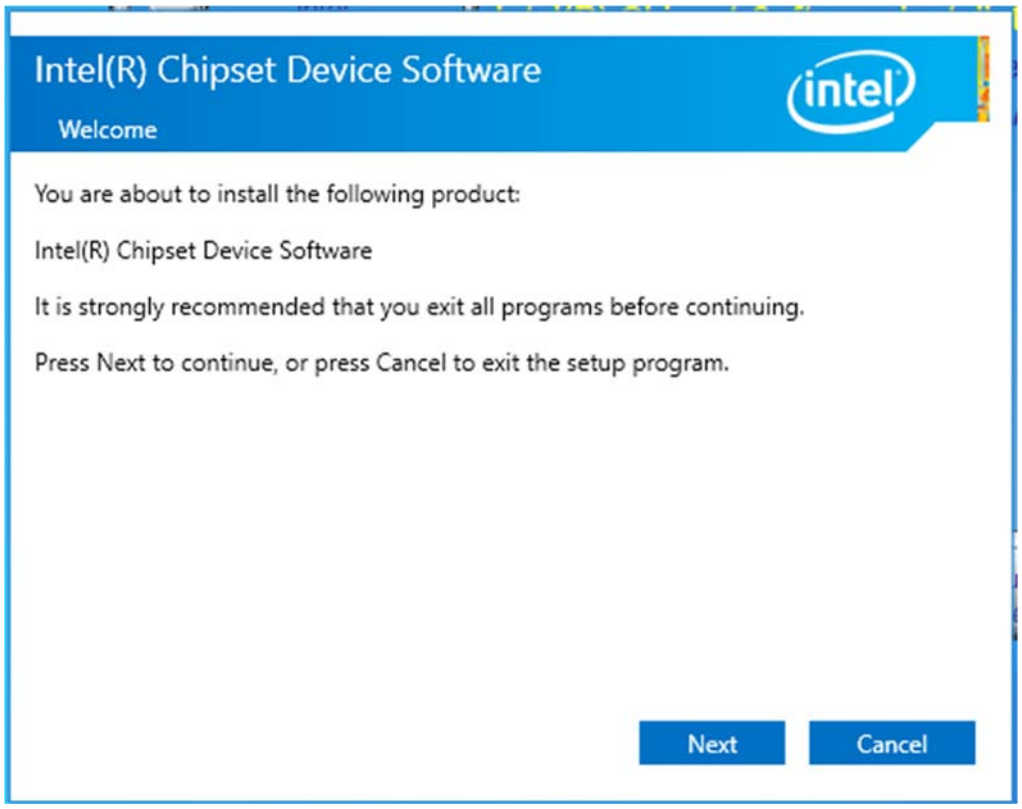
3.2 Intel® Chipset Software Installation Utility

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

1. Insert the drivers disk enclosed in the system package. In the initial screen, click **Intel** and then **Intel(R) Apollolake Chipset Drivers**.

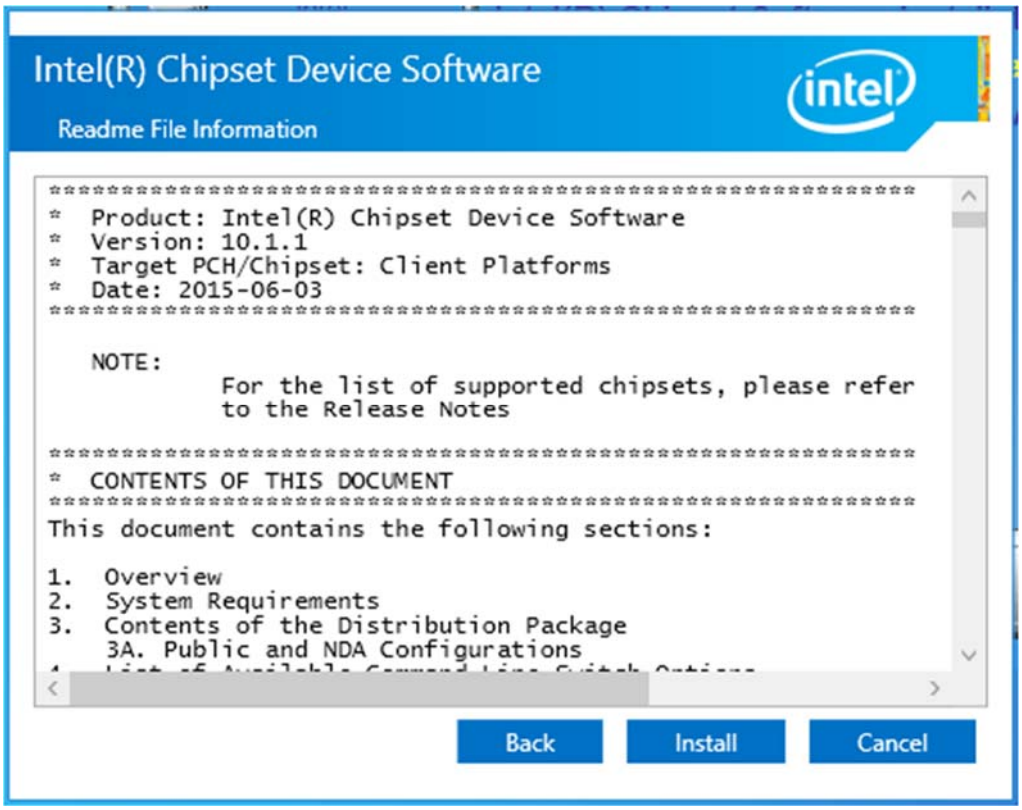
2. Click **Intel(R) Chipset Software Installation Utility**.

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.

5. After reading the *Readme File information*, click **Install**.



6. Click **Finish** to complete the setup process.

3.3 VGA Driver Installation

1. In the initial screen, click **Intel** and then **Intel(R) Apollolake Chipset Drivers**.

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

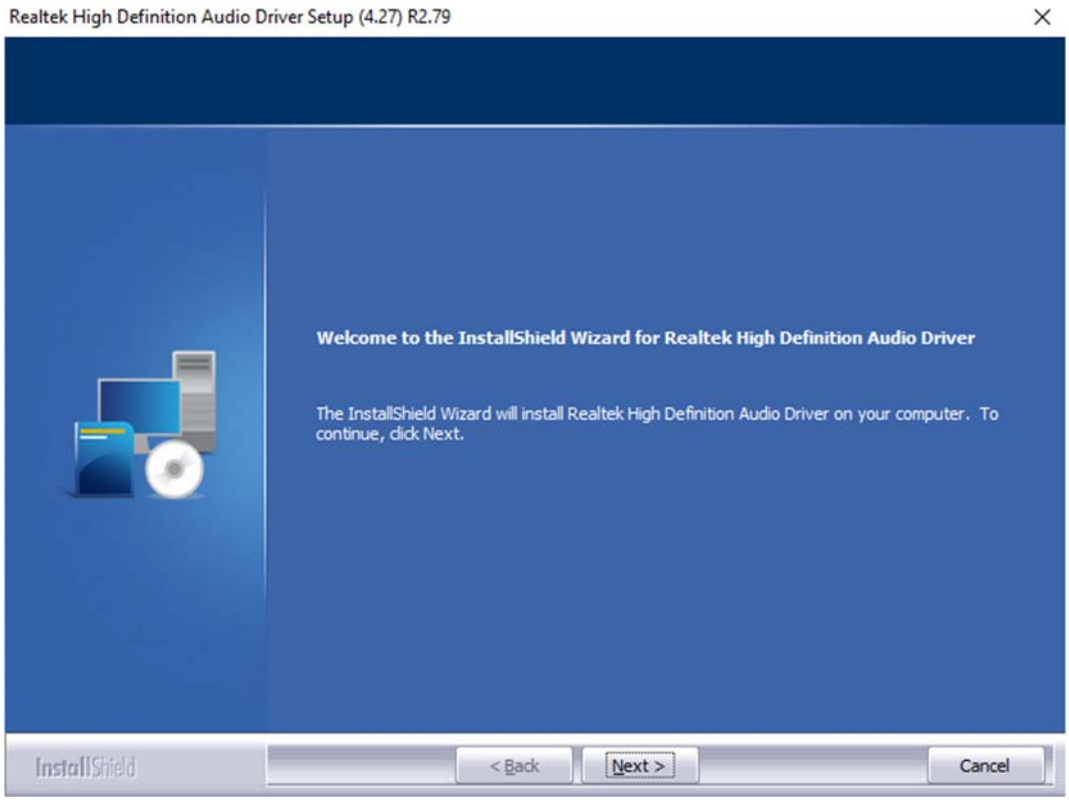
3. In the *Welcome to the Setup Program screen*, click **Next**.
4. In the *License Agreement* screen, click **Yes** to accept all the terms of the license agreement in order to continue the setup program.
5. In the *Readme File Information* screen, click **Next**.
6. In the *Setup Progress* screen, click **Next**.
7. When *Setup is Complete*, restart the system for the changes to take effect. Click **Finish**.

3.4 HD Audio Driver Installation

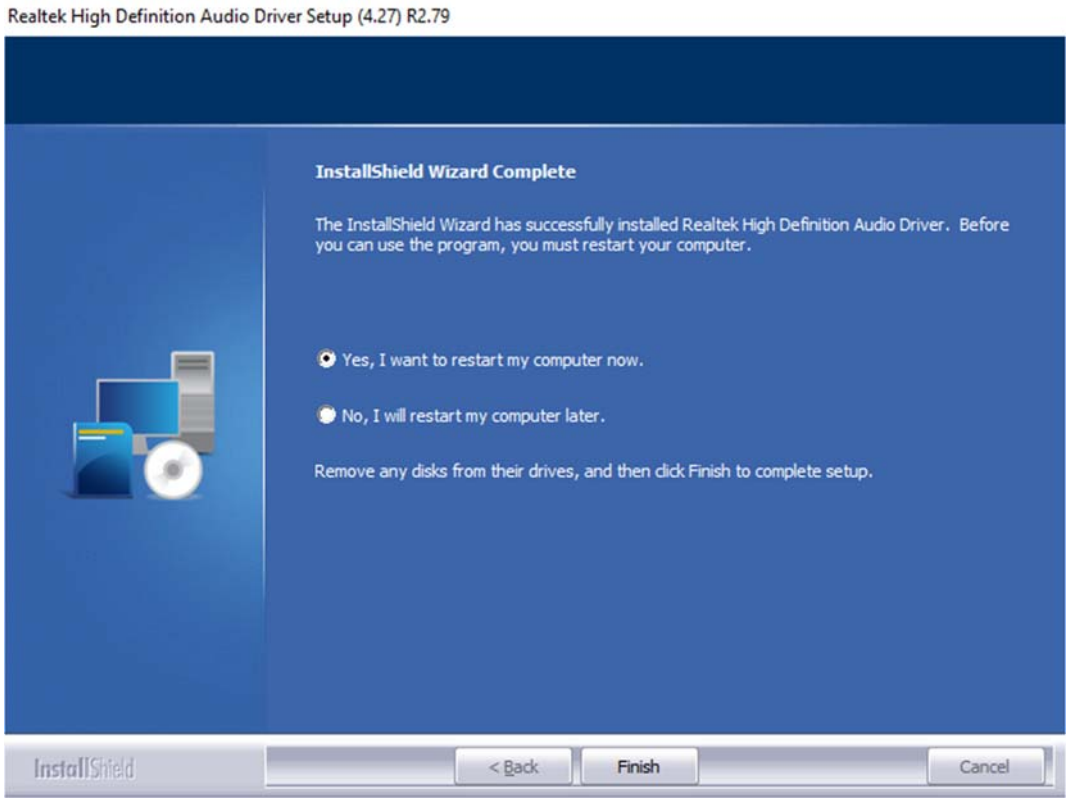
1. In the initial screen, click **Intel** and then **Intel(R) Apollolake Chipset Drivers**.

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

3. On the *Welcome to the InstallShield Wizard* screen, click **Next**.



4. Click **Finish** to complete setup.

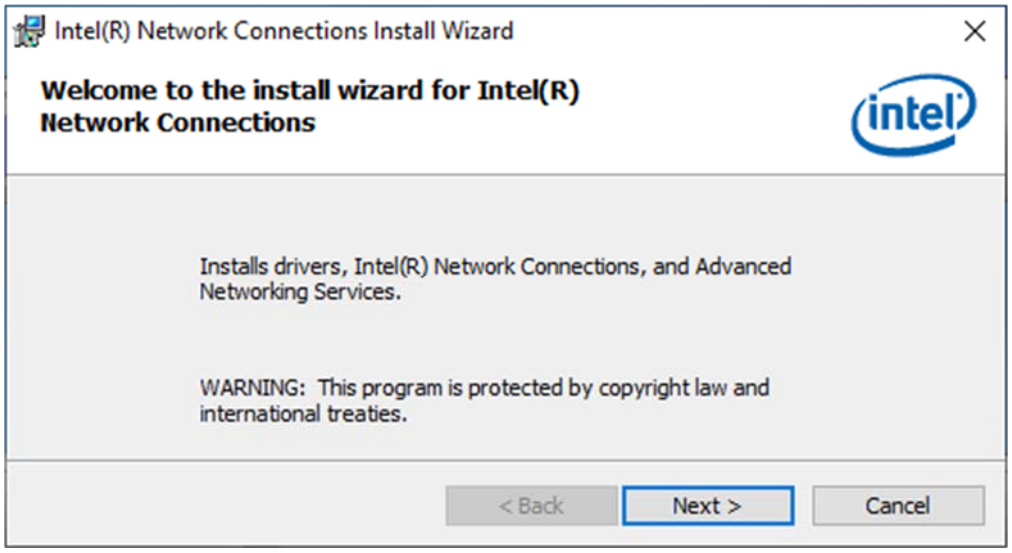


3.5 LAN Driver Installation

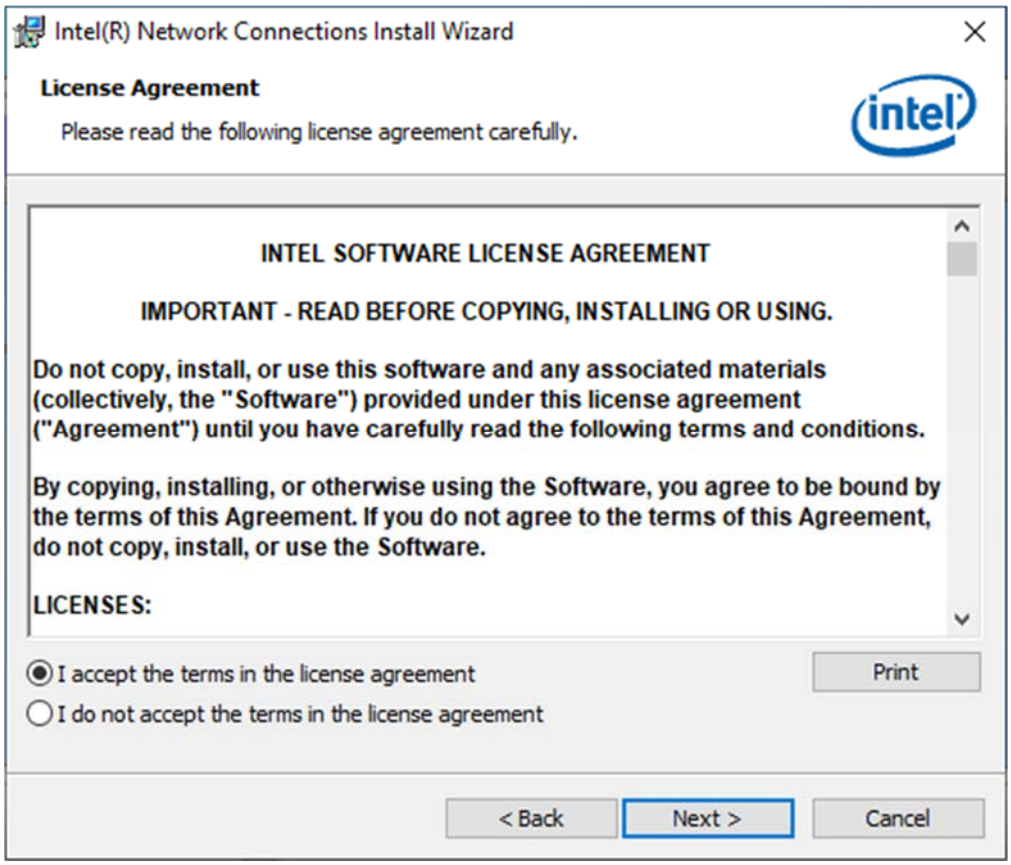
1. Click **Intel LAN Controller Drivers**.

2. Click **Intel(R) Gigabit Ethernet Drivers**.

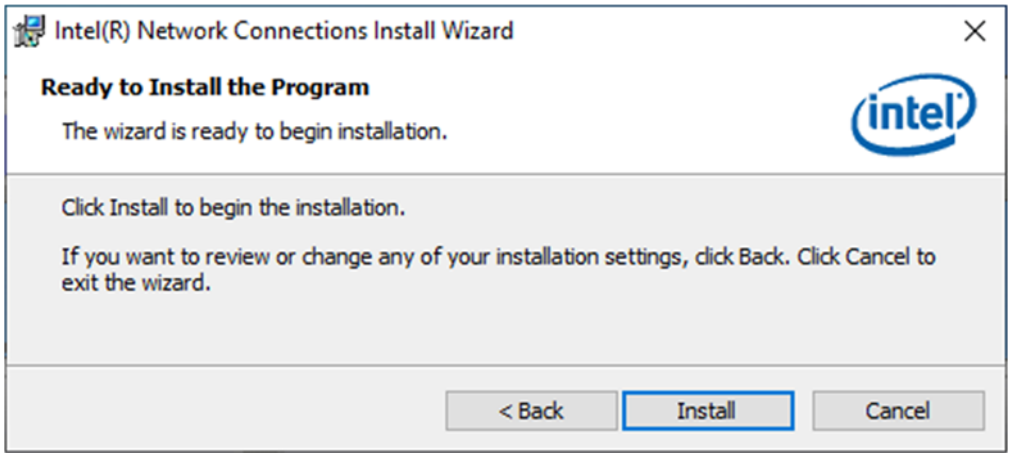
3. In the *Welcome* screen, click **Next**.



4. Accept the terms of the license agreement and click **Next**.



5. In the *Setup Options* screen, click **Next**.
6. In *Ready to Install the Program* screen, click **Install**.



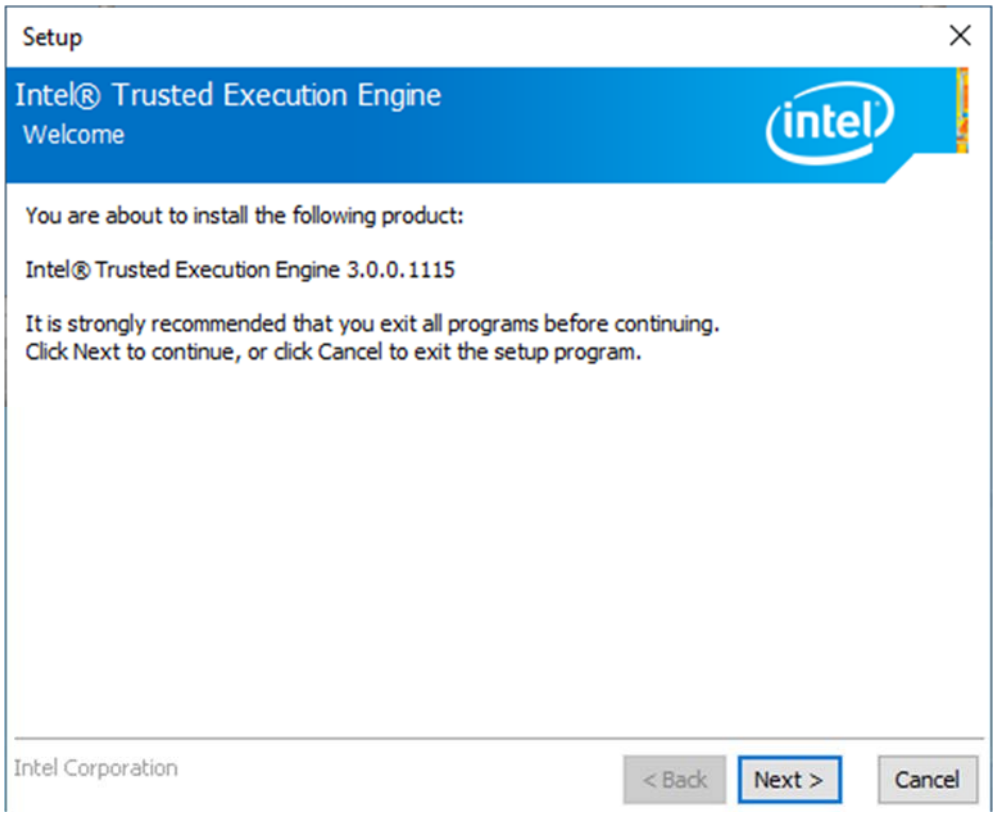
7. In the *Install wizard Completed screen, click* ***Finish***.

3.6 Intel® Trusted Execution Engine Installation

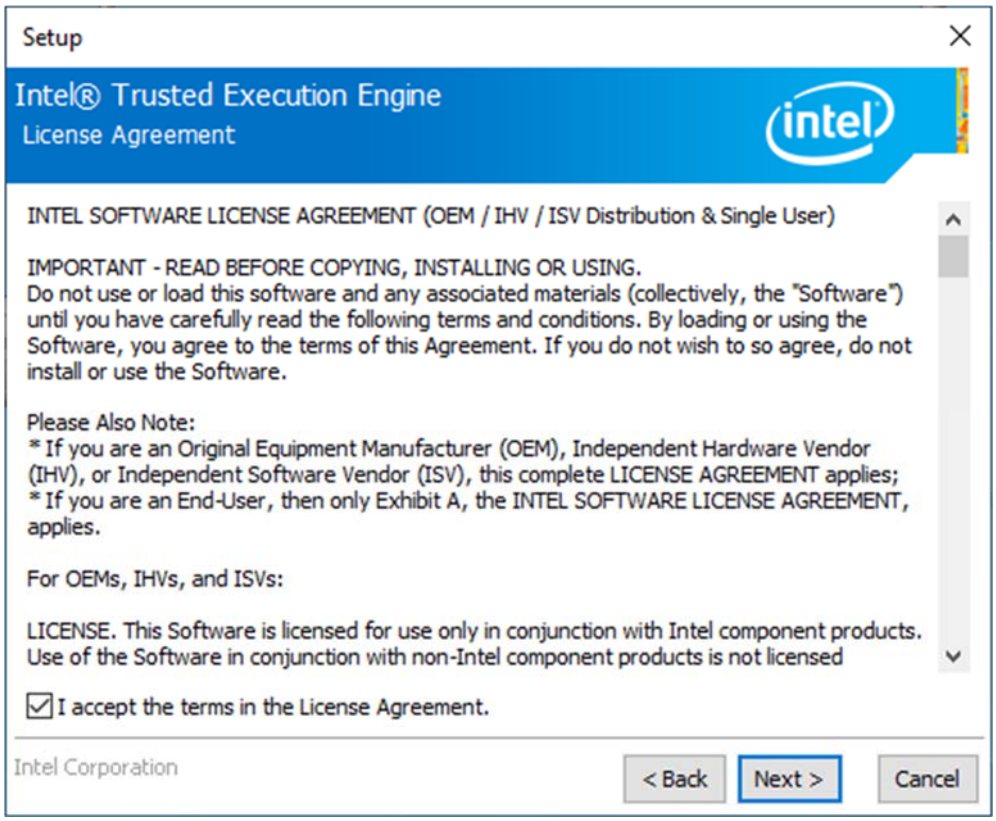
1. In the initial screen, click **Intel** and then **Intel(R) Apollolake Chipset Drivers**.

2. Click Intel(R) TXE Drivers.

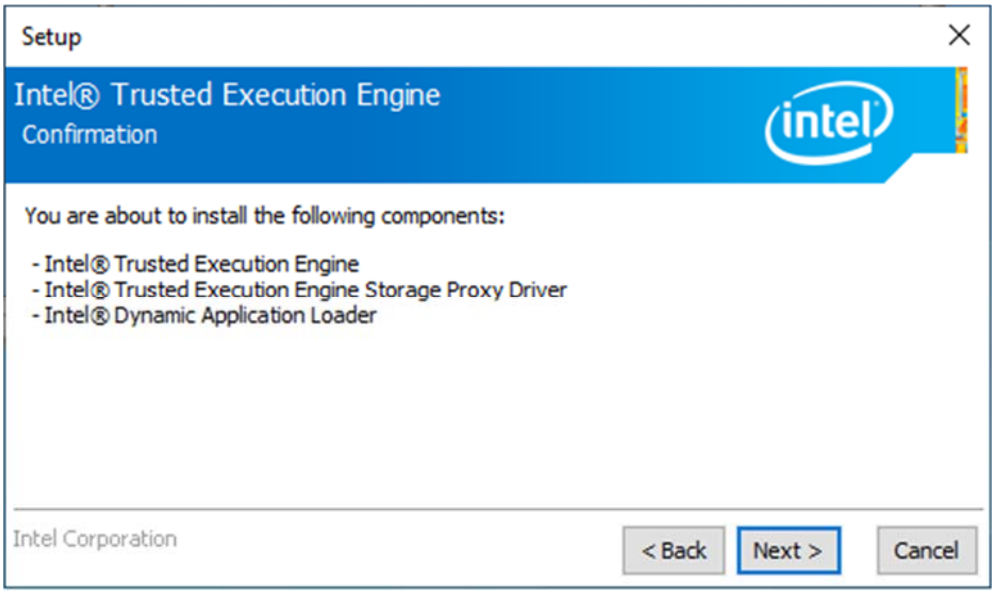
3. When the *Welcome* screen appears, click **Next**.



4. Click **Next** to accept the terms in the license agreement.



5. In the *Confirmation* screen, click **Next**.



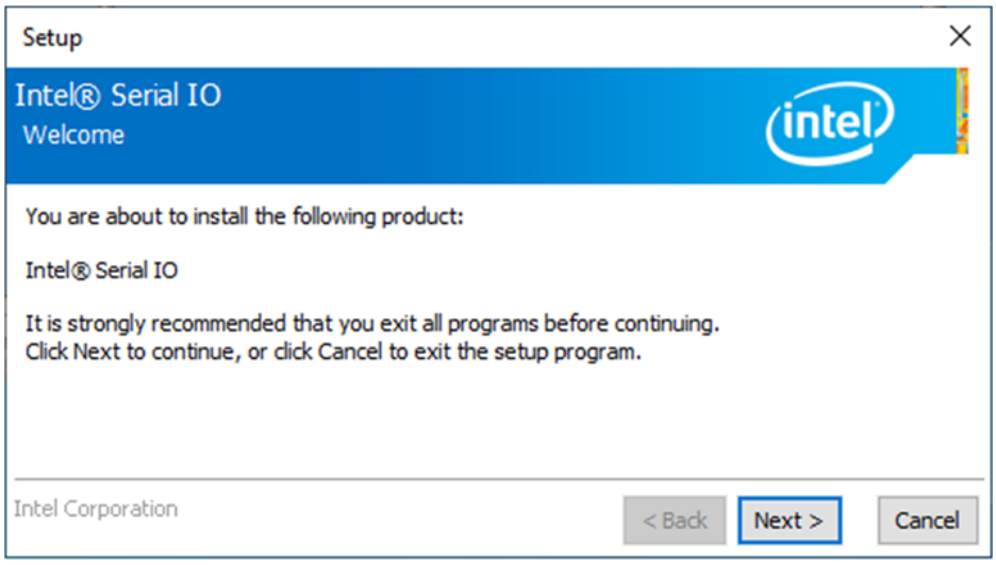
6. When installation is complete, click **Finish**.

3.7 Intel® Serial I/O Drivers Installation

1. In the initial screen, click **Intel** and then **Intel(R) Apollolake Chipset Drivers**.

2. Click **Intel** and then **Support Intel(R) Serial IO Driver**.

3. When the *Weocome* screen appears, click **Next** to contiune installation.



4. Click **Next** to accept the terms in the license agreement.
5. In the *Readme File Information* screen, click **Next**.
6. In the *Confirmation* screen, click **Next**.
7. Restart the system for the changes to take effect. Click **Finish**.

Chapter 4
BIOS Setup

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

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

4.1 Introduction

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

4.2 BIOS Setup

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

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

The following message will appear on the screen:

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

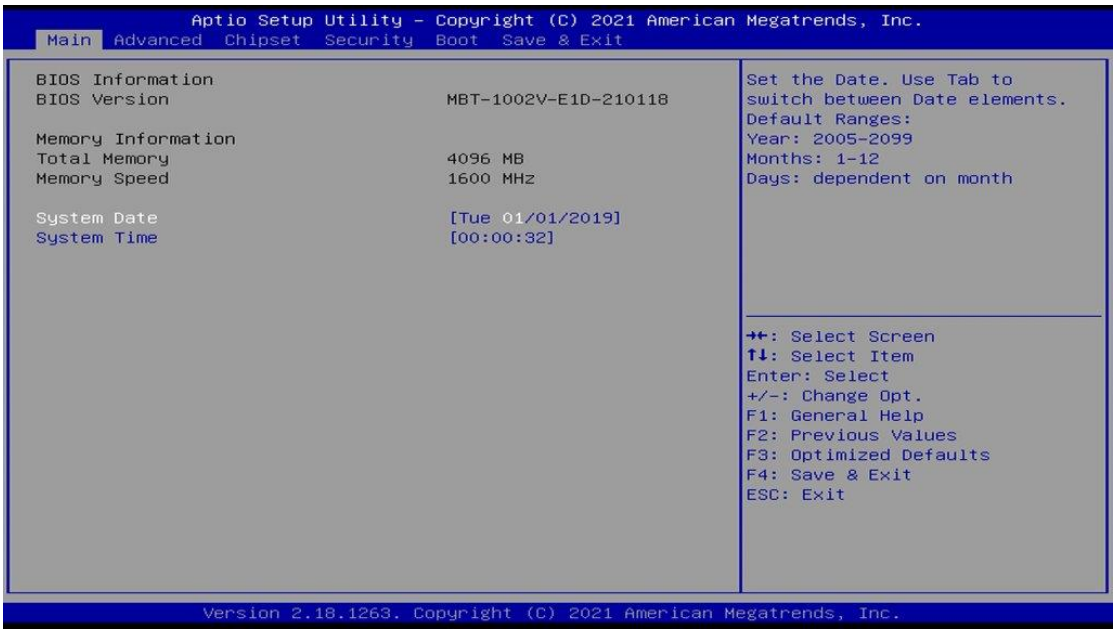
In general, press the arrow keys to highlight items, <Enter> to select, the <PgUp> and <PgDn> keys to change entries, <F1> for help, and <Esc> to quit.

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

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

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

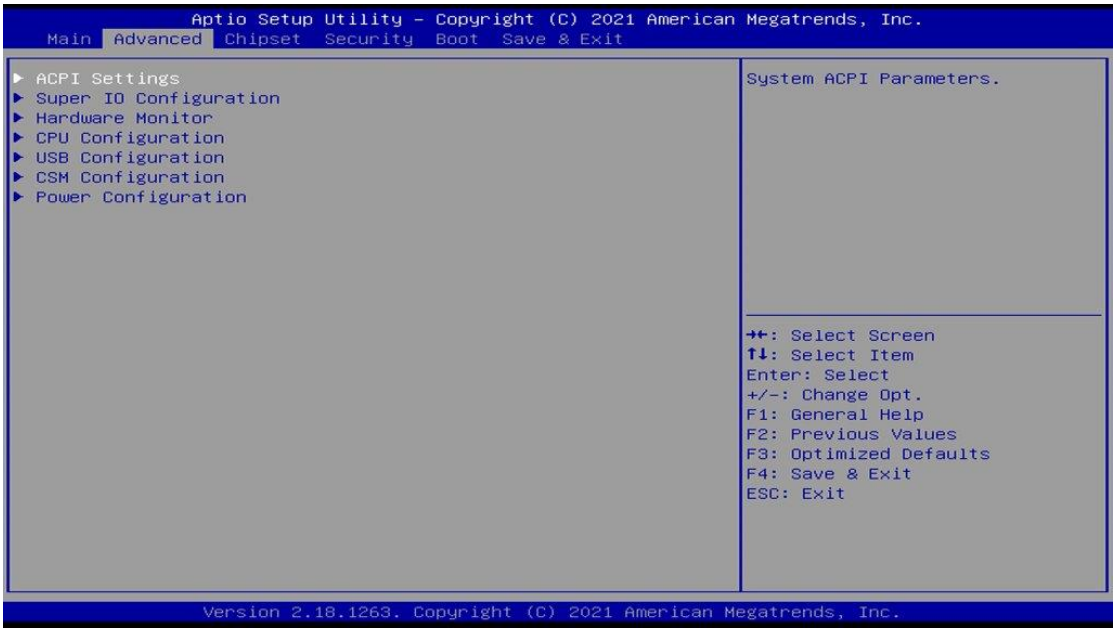
4.3 Main Settings



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

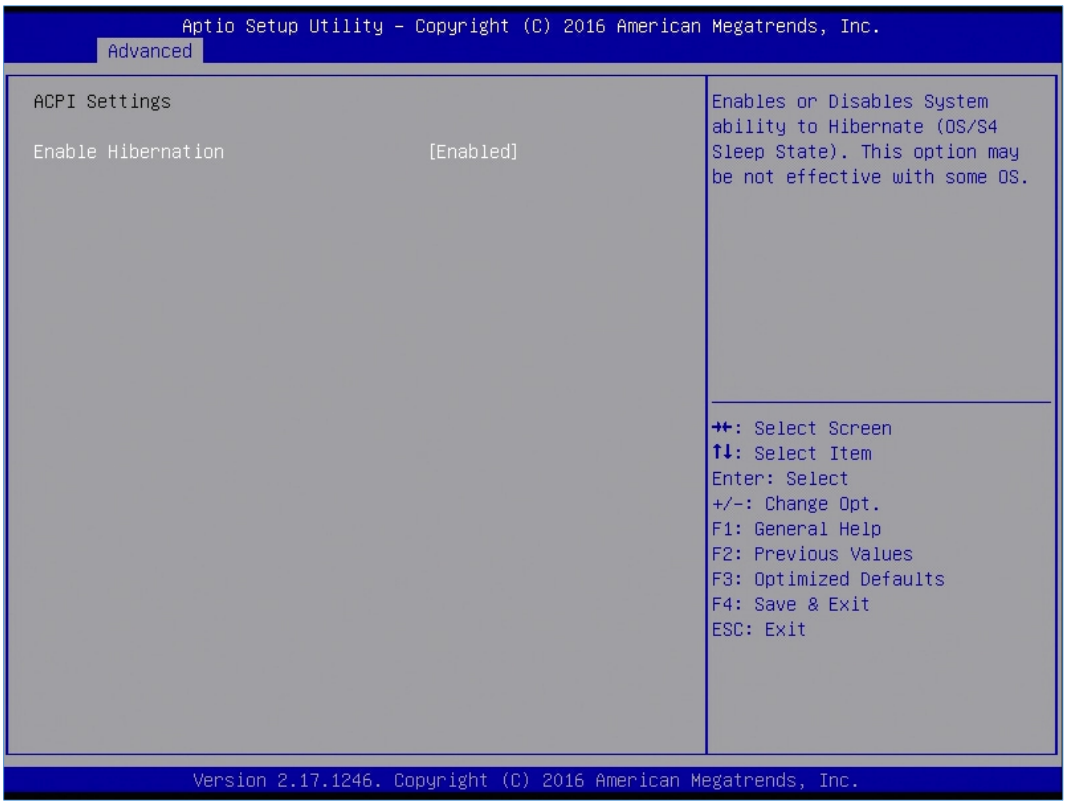
4.4 Advanced Settings

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



BIOS Setting	Description
ACPI Settings	System ACPI Parameters.

4.4.1 ACPI Settings

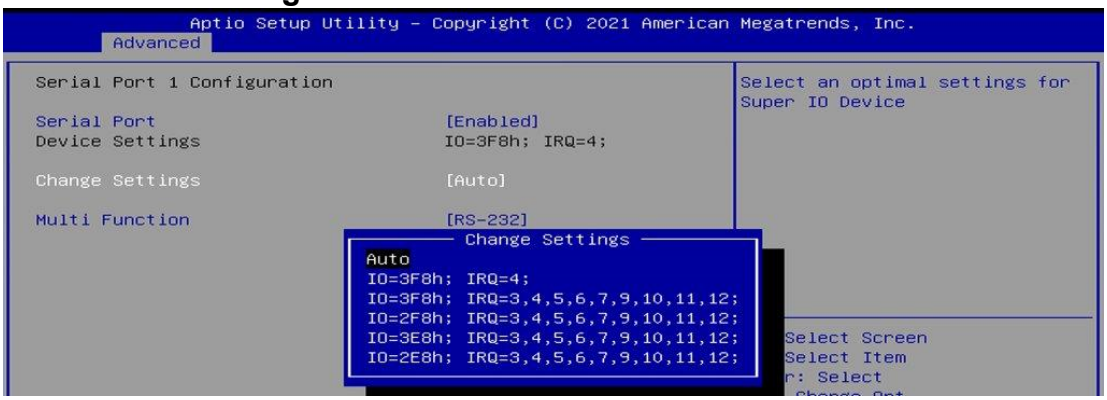


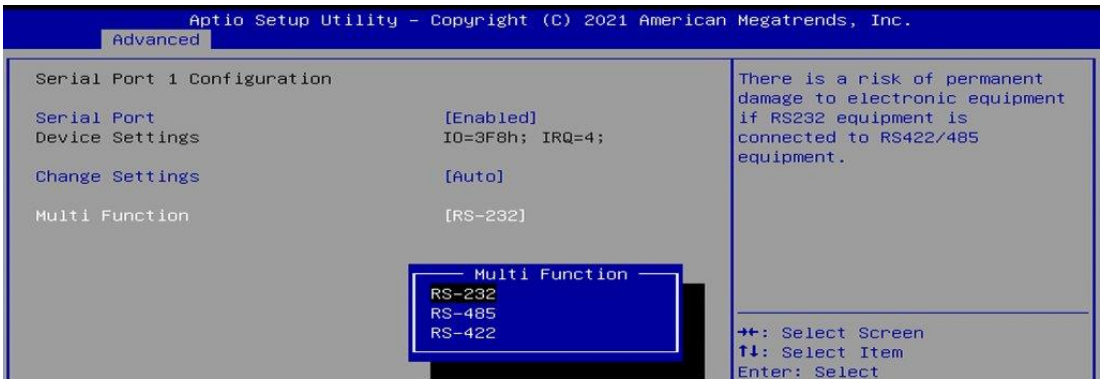
BIOS Setting	Description
Enable Hibernation	Enables or disables the System ability to Hibernate (OS/S4 Sleep State). This option may not be effective with some OS.

4.4.2 Super I/O Configuration

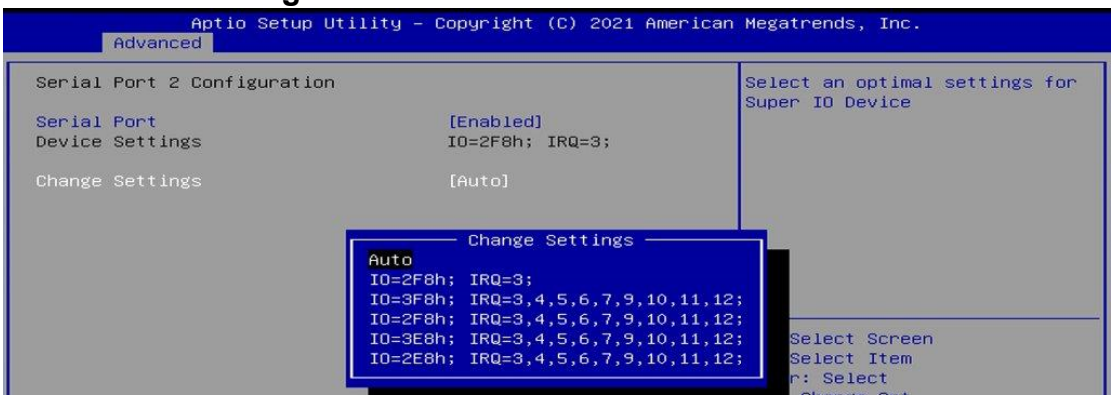
BIOS Setting	Description
Serial Port 1~4 Configuration	Set Parameters of Serial Port

Serial Port 1 Configuration

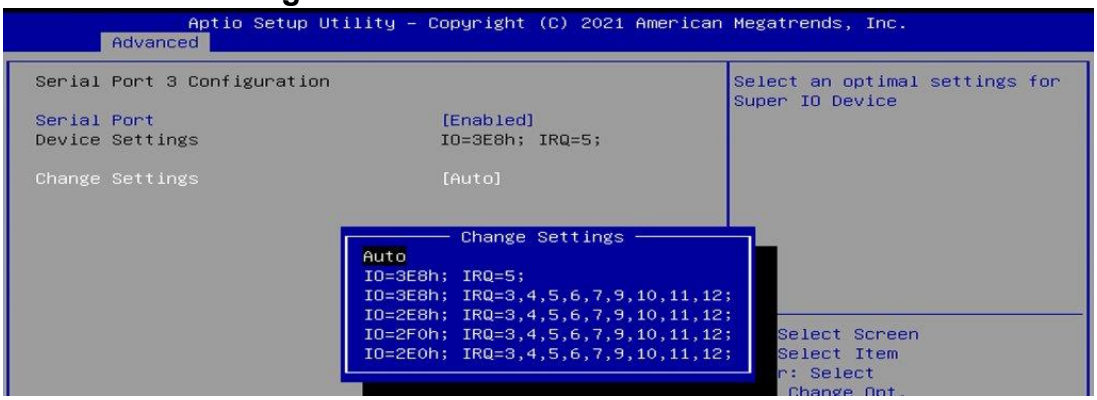




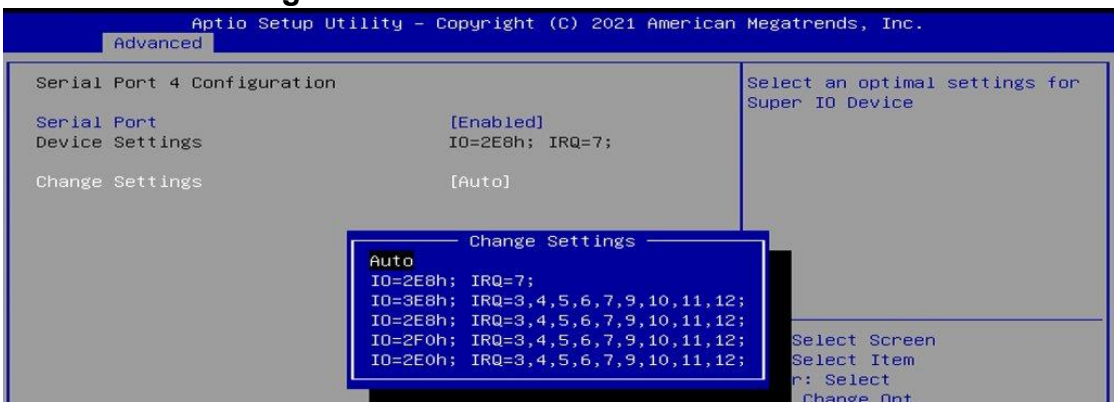
Serial Port 2 Configuration



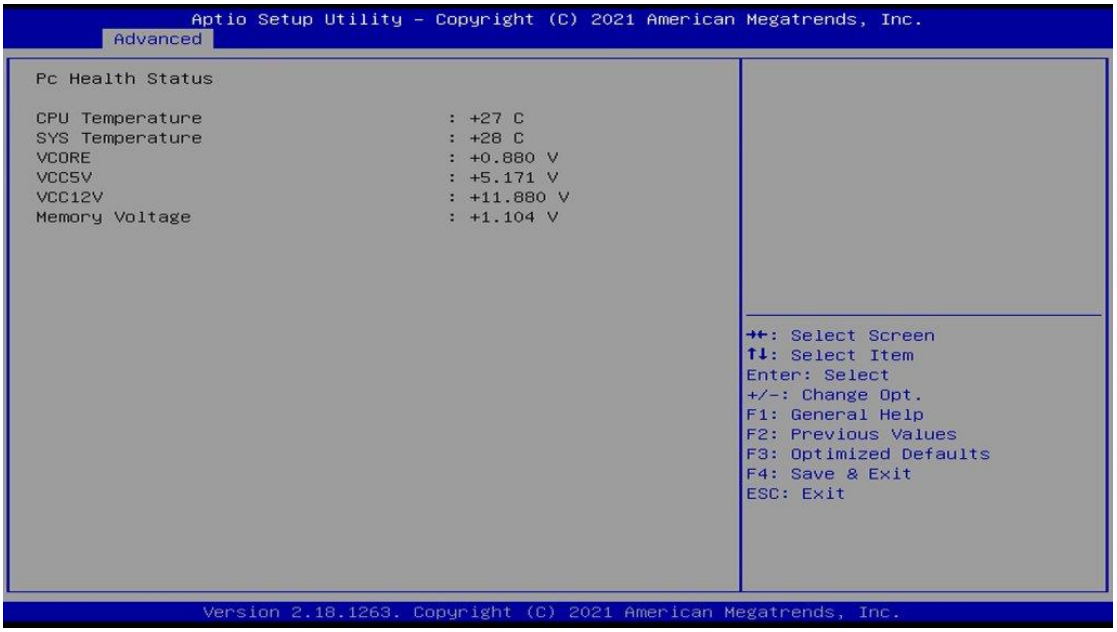
Serial Port 3 Configuration



Serial Port 4 Configuration

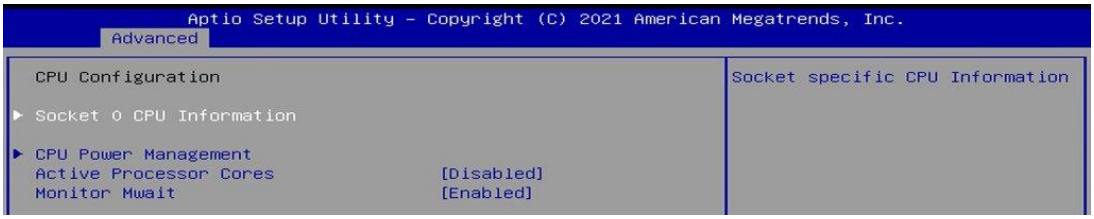


4.4.3 Hardware Monitor

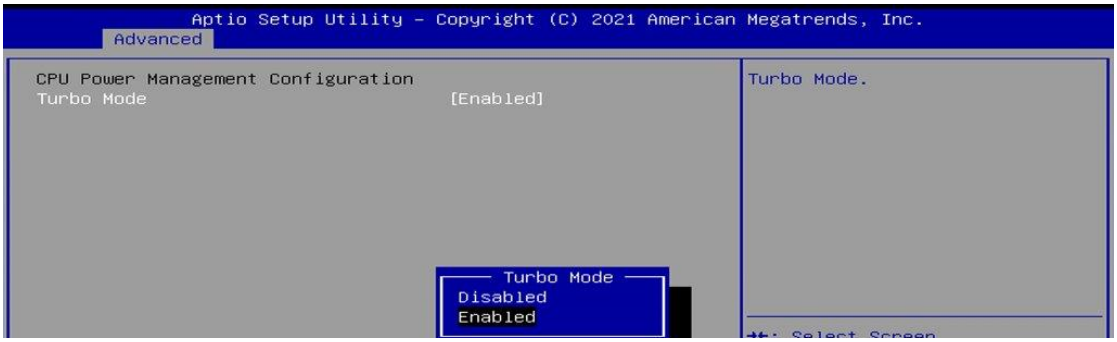


BIOS Setting	Description
Temperatures / Voltages	These fields are the parameters of the hardware monitoring function feature of the motherboard. The values are read-only values as monitored by the system and show the PC health status

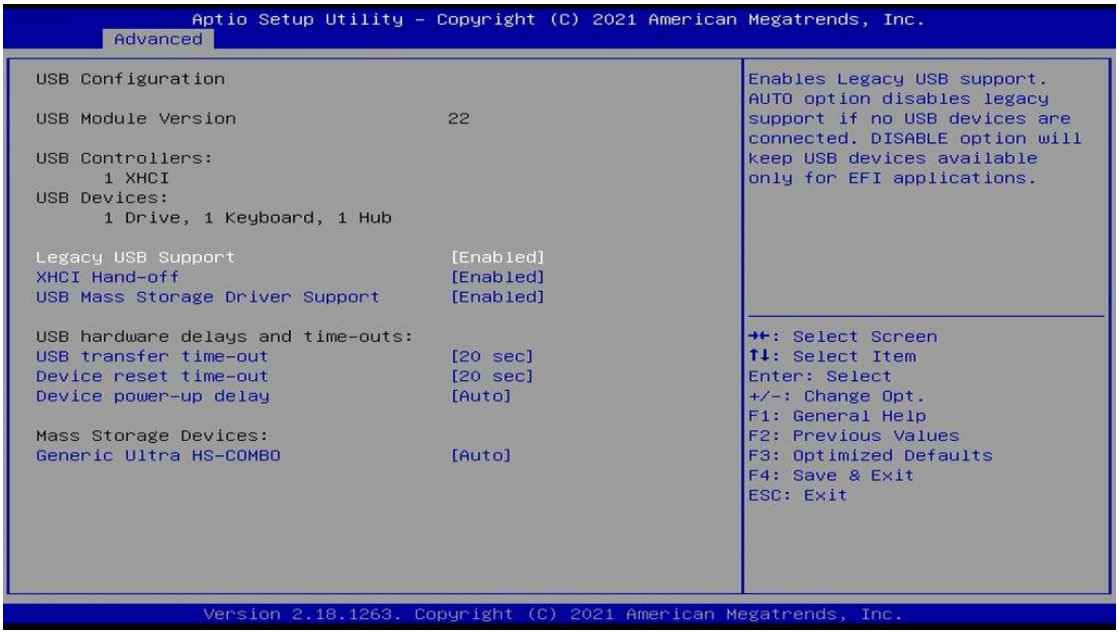
4.4.4 CPU Configuration



BIOS Setting	Description
Socket 0 CPU Information	Displays the socket specific CPU Information.
Turbo Mode	Options: Disabled, Enabled
Active Processor Cores	Number of cores to enable in each processor package Options: Disabled, Enabled
Monitor Mwait	Enable/Disable Monitor Mwait Options: Disabled, Enabled, Auto

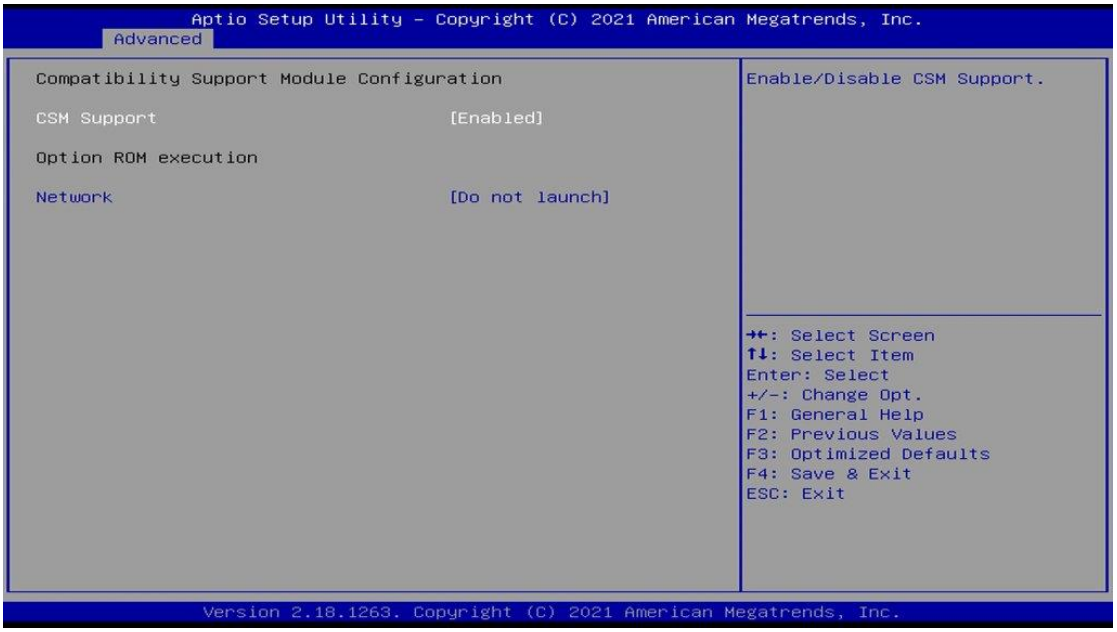


4.4.5 USB Configuration

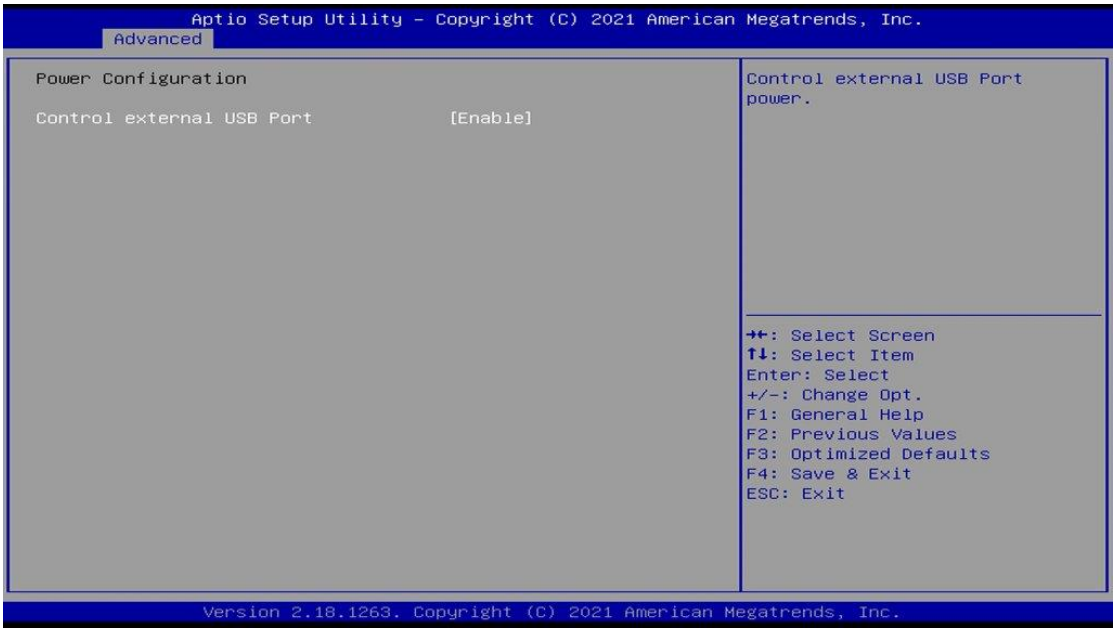


BIOS Setting	Description
Legacy USB Support	• **Enabled** enables Legacy USB support. • **Auto** disables legacy support if there is no USB device connected. • **Disabled** keeps USB devices available only for EFI applications.
XHCI Hand-off	This is a workaround for OSes without XHCI hand-off support. 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 (1, 5, 10, 20 secs) for Control, Bulk, and Interrupt transfers.
Device reset time-out	Gives seconds (10, 20, 30, 40 secs) to delay execution of Start Unit command to USB mass storage device.
Device power-up delay	The maximum time the device will take before it properly reports itself to the Host Controller. **Auto** uses default value for a Root port it is 100ms. But for a Hub port, the delay is taken from Hub descriptor.
Generic Ultra MS-COMBO	Mass storage device emulation type. “AUTO’ enumerates devices according to their media format. Optical drives are emulated as ‘CDROM’ drives with no media will be emulated according to a drive type.

4.4.6 CSM Configuration

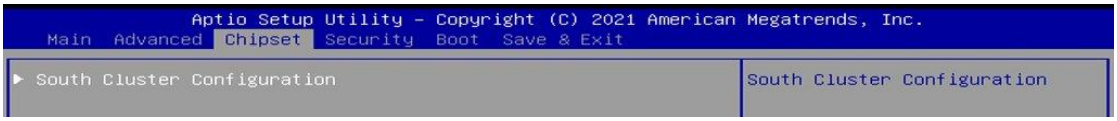


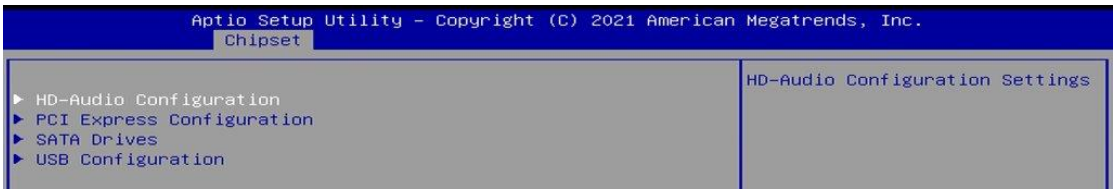
4.4.7 Power Configuration

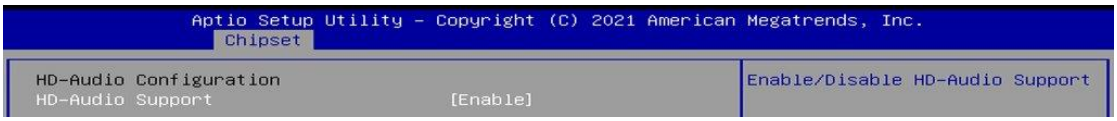


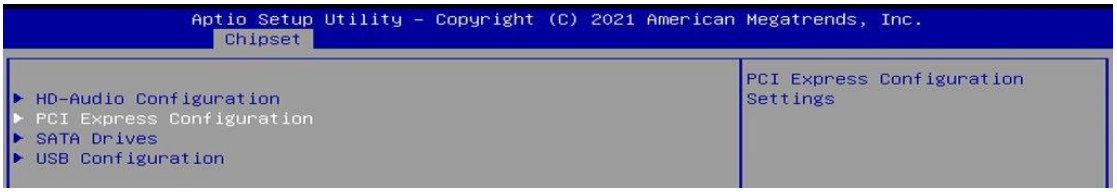
4.5 Chipset Settings

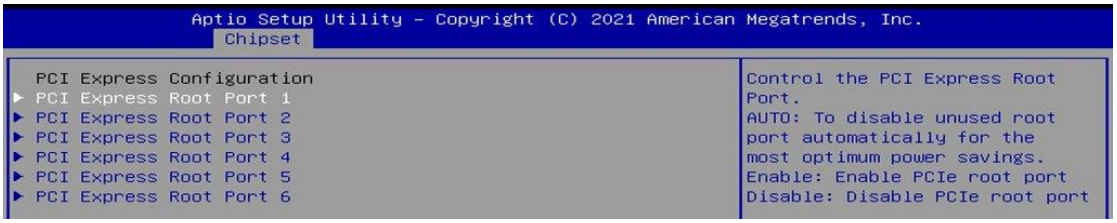
4.5.1 South Cluster Configuration



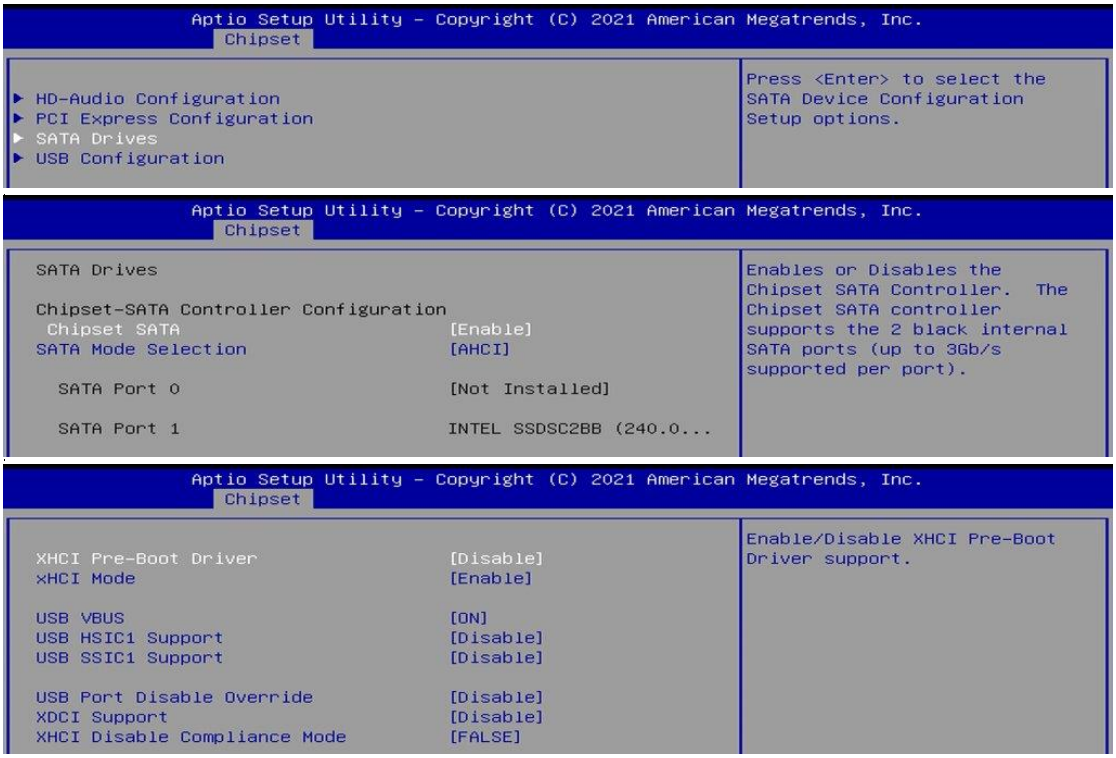






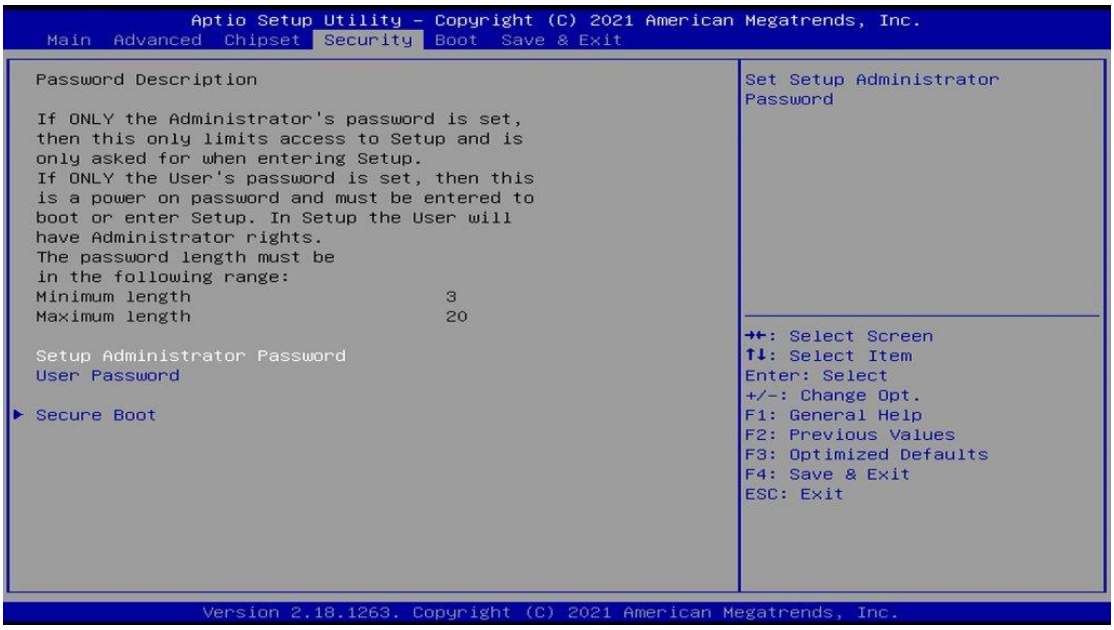


BIOS Setting	Description
ASPM	PCI Express Active State Power Management settings Options: Disable, L0s, L1, L0sL1, Auto
L1 Substates	PCI Express L1 Substates settings Options: L1.1, L1.2, L1.2 & L1.2
PME SCI	PCI Express PME SCI Enable/Disable
PCIe Speed	Configure PCIe Speed Options: Auto, Gen1, Gen2

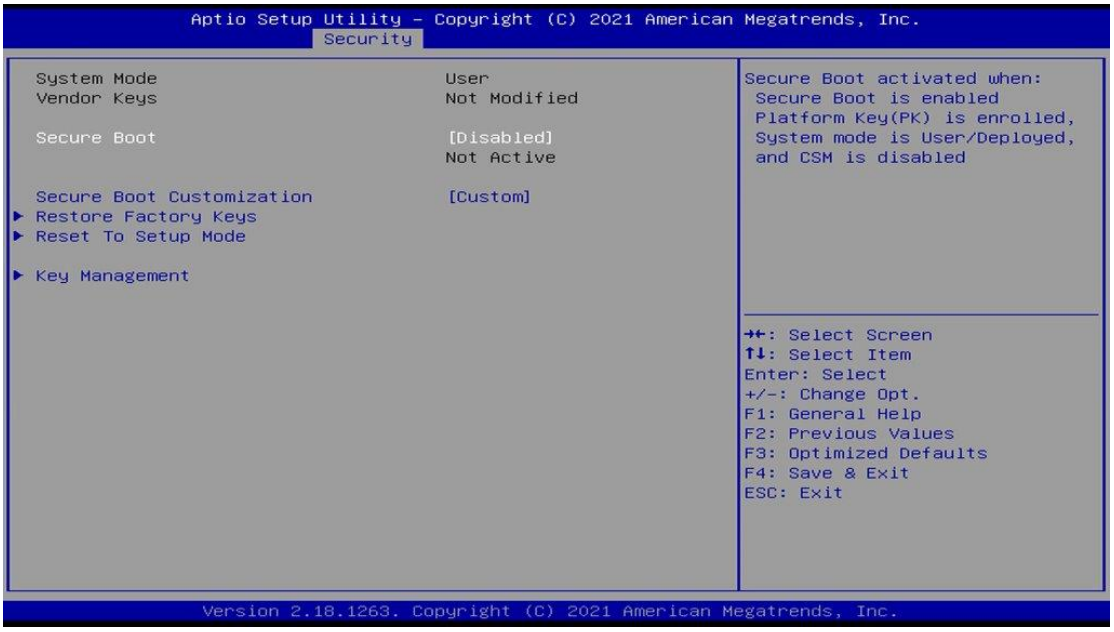


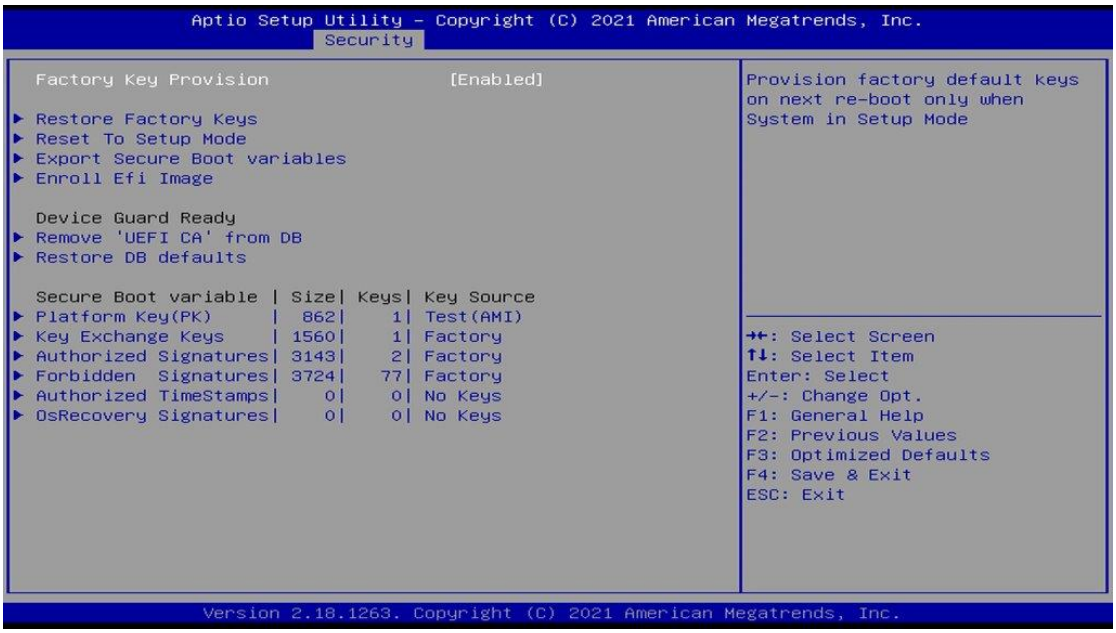
BIOS Setting	Description
SATA Mode Selection	Determines how SATA controller(s) operate
XHCI Pre-Boot Driver	Options: Enable, Disable
XHCI Mode	Once disabled, XHCI controller would be function disabled, none of the USB devices are detectable and usable during boot and in OS. Do not disable it unless for debug purpose.
USB VBUS	VBUS should be ON in MOST mode. It should be OFF in OTG device mode.
USB HSIC1 Support	Options: Enable, Disable
USB SSIC1 Support	Options: Enable, Disable
USB Port Disable Override	Selectively Enable/Disable the corresponding USB port from reporting a Device Connection to the controller.
XDCI Support	Options: Enable, Disable
XHCI Didable Compliance Mode	Options to disable XHCI Link Compliance Mode. Default is FALSE to not disable Compliance Mode. Set TRUE to disable Compliance Mode.

4.6 Security Settings



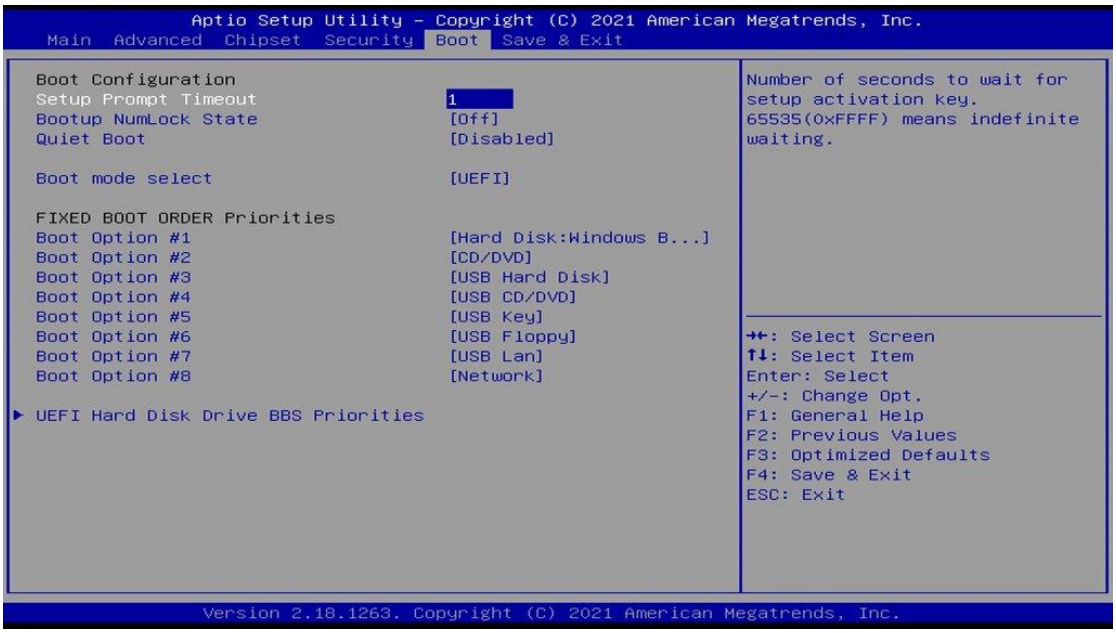
BIOS Setting	Description
Administrator Password	Sets an administrator password for the setup utility.
User Password	Sets a user password.
Secure Boot	Secure Boot feature is active if Secure Boot is enabled. Platform Key(PK) is enrolled and the system is in user mode. The mode change requires platform reset.





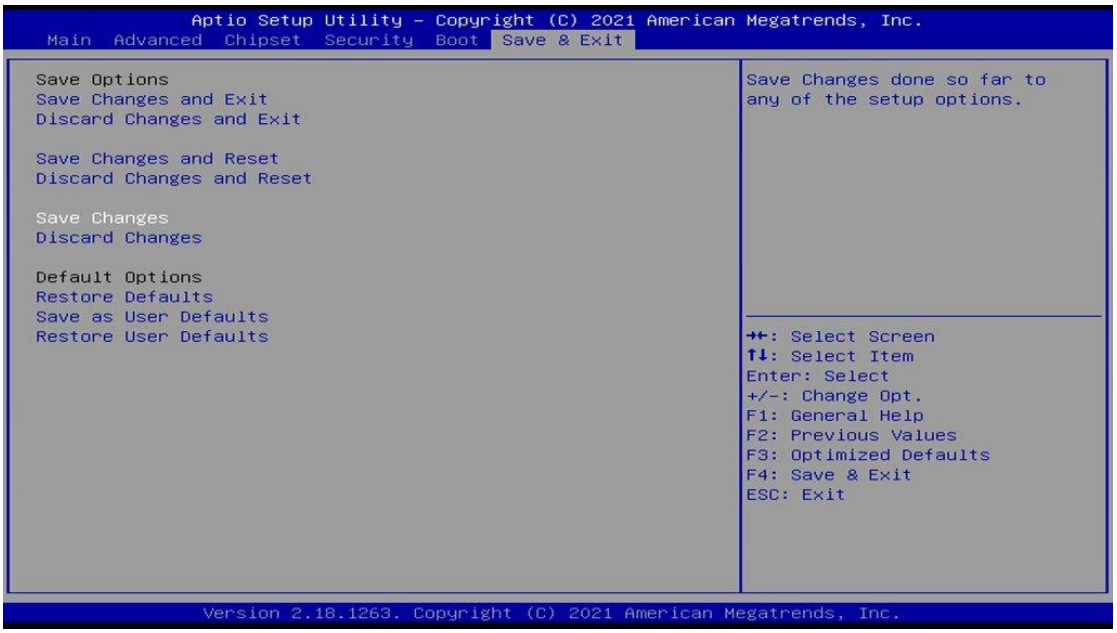
BIOS Setting	Description
Secure Boot Customization	Secure Boot Mode – Custom a Standard. Set UEFI Secure Boot Mode to STANDARD mode or CUSTOM mode. This change take effect after SAVE. Ater reset, th emode will return to STANDARD mode.
Restore Factory Keys	Force System to User Mode. Install factory default Secure Boot key databases.
Reset To Setup Mode	Delete all Secure Boot key databases from NVRAM
Factory Key Provision	Provision factory default keys on next re-boot only when System in Setup Mode.

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.
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.
UEFI Hard Disk Drive BBS Priorities	Specifies the Boot Device Priority sequence from available UEFI 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)
- Watchdog Timer Configuration
- Software Development Kit for WDT.DLL

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
0x0000002E-0x0000002F	Motherboard resources
0x0000004E-0x0000004F	Motherboard resources
0x00000061-0x00000061	Motherboard resources
0x00000063-0x00000063	Motherboard resources
0x00000065-0x00000065	Motherboard resources
0x00000067-0x00000067	Motherboard resources
0x00000070-0x00000070	Motherboard resources
0x00000070-0x00000070	System CMOS/real time clock
0x00000080-0x0000008F	Motherboard resources
0x00000092-0x00000092	Motherboard resources
0x000000B2-0x000000B3	Motherboard resources
0x00000680-0x0000069F	Motherboard resources
0x00000400-0x0000047F	Motherboard resources
0x00000500-0x000005FE	Motherboard resources
0x00000600-0x0000061F	Motherboard resources
0x0000164E-0x0000164F	Motherboard resources
0x0000E000-0x0000EFFF	PCI-to-PCI Bridge
0x000003F8-0x000003FF	Communications Port (COM1)
0x000002F8-0x000002FF	Communications Port (COM2)
0x000003E8-0x000003EF	Communications Port (COM3)
0x000002E8-0x000002EF	Communications Port (COM4)
0x000002E0-0x000002E7	Communications Port (COM5)
0x0000C000-0x0000CFFF	PCI-to-PCI Bridge
0x00000020-0x00000021	Programmable interrupt controller

Address	Device Description
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
0x0000F090-0x0000F097	Standard SATA AHCI Controller
0x0000F080-0x0000F083	Standard SATA AHCI Controller
0x0000F060-0x0000F07F	Standard SATA AHCI Controller
0x00000000-0x0000006F	PCI Express Root Complex
0x00000078-0x00000CF7	PCI Express Root Complex
0x00000D00-0x0000FFFF	PCI Express Root Complex
0x0000F040-0x0000F05F	SM Bus Controller
0x0000D000-0x0000DFFF	PCI-to-PCI Bridge
0x0000F000-0x0000F03F	Microsoft Basic Display Adapter
0x00000040-0x00000043	System timer
0x00000050-0x00000053	System timer

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 8	High precision event timer
IRQ 4	Communications Port (COM1)
IRQ 3	Communications Port (COM2)
IRQ 5	Communications Port (COM3)
IRQ 7	Communications Port (COM4)
IRQ 7	PCI Simple Communications Controller
IRQ 7	SM Bus Controller
IRQ 11	Communications Port (COM5)
IRQ 4294967282~87	Intel(R) I210 Gigabit Network Connection
IRQ 4294967294	Standard SATA AHCI Controller
IRQ 4294967288~93	Intel(R) I210 Gigabit Network Connection #2
IRQ 55~204	Microsoft ACPI-Compliant System
IRQ 256~511	Microsoft ACPI-Compliant System
IRQ 25	High Definition Audio Controller
IRQ 14	Intel(R) Serial IO GPIO Host Controller - INT3452
IRQ 27	Intel(R) Serial IO I2C Host Controller - 5AAC
IRQ 4294967281	Intel(R) USB 3.0 eXtensible Host Controller - 1.0 (Microsoft)
IRQ 0	System timer

C. 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.

Sample Code:

```
//---------------------------------------------------------------------------
//
// 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 "F81866.H"
//---------------------------------------------------------------------------
int main (int argc, char *argv[]); void EnableWDT(int);
void DisableWDT(void);
//---------------------------------------------------------------------------
int main (int argc, char *argv[])
{
unsigned char bBuf; unsigned char bTime; char **endptr;

char SIO;
printf("Fintek 81866 watch dog program\n"); SIO = Init_F81866();
if (SIO == 0)
{
printf("Can not detect Fintek 81866, program abort.\n"); return(1);
}//if (SIO == 0)

if (argc != 2)
{
printf(" Parameter incorrect!!\n"); return (1);
}

bTime = strtol (argv[1], endptr, 10);
printf("System will reset after %d seconds\n", bTime);

if (bTime)
{       EnableWDT(bTime);   } else
{       DisableWDT();   } return 0;
```

```
}
//-----------------------------------------------------------------------------
void EnableWDT(int interval)
{
unsigned char bBuf;

bBuf = Get_F81866_Reg(0x2B); bBuf &= (~0x20);
Set_F81866_Reg(0x2B, bBuf);   //Enable WDTO

Set_F81866_LD(0x07);      //switch to logic device 7
Set_F81866_Reg(0x30, 0x01);   //enable timer

bBuf = Get_F81866_Reg(0xF5); bBuf &= (~0x0F);
bBuf |= 0x52;
Set_F81866_Reg(0xF5, bBuf);           //count mode is second Set_F81866_Reg(0xF6, interval);
      //set timer
bBuf = Get_F81866_Reg(0xFA); bBuf |= 0x01;
Set_F81866_Reg(0xFA, bBuf);   //enable WDTO output

bBuf = Get_F81866_Reg(0xF5); bBuf |= 0x20;
Set_F81866_Reg(0xF5, bBuf);   //start counting
}
//-----------------------------------------------------------------------------
void DisableWDT(void)
{
unsigned char bBuf;
Set_F81866_LD(0x07);      //switch to logic device 7 bBuf = Get_F81866_Reg(0xFA);
bBuf &= ~0x01;
Set_F81866_Reg(0xFA, bBuf);   //disable WDTO output

bBuf = Get_F81866_Reg(0xF5); bBuf &= ~0x20;
bBuf |= 0x40;
Set_F81866_Reg(0xF5, bBuf);   //disable WDT
}
//-----------------------------------------------------------------------------
```

```
//---------------------------------------------------------------------------
//
// 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 "F81866.H"
#include <dos.h>
//---------------------------------------------------------------------------
unsigned int F81866_BASE; void Unlock_F81866 (void); void Lock_F81866 (void);
//---------------------------------------------------------------------------
unsigned int Init_F81866(void)
{
unsigned int result; unsigned char ucDid;

F81866_BASE = 0x4E;
result = F81866_BASE;

ucDid = Get_F81866_Reg(0x20);
if (ucDid == 0x07)      //Fintek 81866
{       goto Init_Finish; }

F81866_BASE = 0x2E;
result = F81866_BASE;

ucDid = Get_F81866_Reg(0x20);
if (ucDid == 0x07)      //Fintek 81866
{       goto Init_Finish; }

F81866_BASE = 0x00;
result = F81866_BASE;

Init_Finish:
return (result);
}
//---------------------------------------------------------------------------
void Unlock_F81866 (void)
{
outportb(F81866_INDEX_PORT, F81866_UNLOCK); outportb(F81866_INDEX_PORT,
F81866_UNLOCK);
}
//---------------------------------------------------------------------------
void Lock_F81866 (void)
{
outportb(F81866_INDEX_PORT, F81866_LOCK);
}
//---------------------------------------------------------------------------
void Set_F81866_LD( unsigned char LD)
{
Unlock_F81866();
outportb(F81866_INDEX_PORT, F81866_REG_LD);
outportb(F81866_DATA_PORT, LD);  Lock_F81866();
```

```
}
//---------------------------------------------------------------------------
void Set_F81866_Reg( unsigned char REG, unsigned char DATA)
{
Unlock_F81866(); outportb(F81866_INDEX_PORT, REG); outportb(F81866_DATA_PORT,
DATA); Lock_F81866();
}
//---------------------------------------------------------------------------
unsigned char Get_F81866_Reg(unsigned char REG)
{
unsigned char Result; Unlock_F81866();
outportb(F81866_INDEX_PORT, REG); Result = inportb(F81866_DATA_PORT);
Lock_F81866();
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       F81866_H
#define        F81866_H     1
//---------------------------------------------------------------------------
#defineF81866_INDEX_PORT   (F81866_BASE)
#defineF81866_DATA_PORT    (F81866_BASE+1)
//---------------------------------------------------------------------------
#defineF81866_REG_LD   0x07
//---------------------------------------------------------------------------
#define F81866_UNLOCK  0x87
#defineF81866_LOCK       0xAA
//---------------------------------------------------------------------------
unsigned int Init_F81866(void);
void Set_F81866_LD( unsigned char);
void Set_F81866_Reg( unsigned char, unsigned char); unsigned char
Get_F81866_Reg( unsigned char);
//---------------------------------------------------------------------------
#endif //      F81866_H
```

D. Software Development Kit for WDT.DLL

1.) OS Supported

Windows SP (32-bit / 64-bit) or above

2.) Driver Installation

- **For 32-bit environment:**

  Step 1:
  Copy the file **KMUI32_1K.SYS** to <%WINDIR%>\SYSTEM32\DRIVERS.

  Step 2:
  The following parameters must be written to your registry.

  HKLM,"System\CurrentControlSet\Services\KMUI32_1K","ErrorControl",%REG_DWORD%,0x00000001

  HKLM,"System\CurrentControlSet\Services\KMUI32_1K","Type",%REG_DWORD%,0x00000001

  HKLM,"System\CurrentControlSet\Services\KMUI32_1K","Start",%REG_DWORD%,0x00000000

  HKLM,"System\CurrentControlSet\Services\KMUI32_1K","DisplayName",%REG_SZ%,"KMUI32_1K"

  Step 3:
  Restart the system.

- **For 64-bit environment:**

  Step 1:
  Copy “KMUI64_1K.SYS” file to <%WINDIR%>\SYSTEM32\DRIVERS

  Step 2:
  The following parameters must be written to your registry.

  HKLM,"System\CurrentControlSet\Services\KMUI64_1K","ErrorControl",%REG_DWORD%,0x00000001

  HKLM,"System\CurrentControlSet\Services\KMUI64_1K","Type",%REG_DWORD%,0x00000001

  HKLM,"System\CurrentControlSet\Services\KMUI64_1K","Start",%REG_DWORD%,0x00000000

  HKLM,"System\CurrentControlSet\Services\KMUI64_1K","DisplayName",%REG_SZ%,"KMUI64_1K"

  Step 3:
  Restart the system.

Note: Do not install both of the 32-bit and 64-bit drivers on an operating system.

3.) Exportion from IB_WDT.DLL / IB_WDT.64.DLL

```
extern "C" __declspec(dllexport) int __stdcall InstallDriver(void);
extern "C" __declspec(dllexport) int __stdcall RemoveDriver(void);
extern "C" __declspec(dllexport) char* __stdcall GetWDTInfo(void);
extern "C" __declspec(dllexport) int __stdcall EnableWDT(int);
extern "C" __declspec(dllexport) int __stdcall DisableWDT(int);
extern "C" __declspec(dllexport) int __stdcall IsDioAvailable(int);
extern "C" __declspec(dllexport) int __stdcall SetDioInputMask(int);
extern "C" __declspec(dllexport) int __stdcall SetDioOutputMask(int);
extern "C" __declspec(dllexport) int __stdcall GetDioInput(int);
extern "C" __declspec(dllexport) int __stdcall SetDioOutput(int);
```

Note:

1. The **IB_WDT.DLL** is of 32-bit and works on 32-bit and 64-bit Windows operating system. The “IB_WDT.64.DLL” is for 64bit Windows only.
2. The routines are not thread-safe. Your software engineers should take the responsibility to avoid multi-entry condition.

● Driver Initialization & Deinitalization

```
extern "C" __declspec(dllexport) int __stdcall InstallDriver(void);
```

Input : None

Output : Return 1 if device driver loads successfully; otherwise return 0.

Note: This function should be invoked before Watchdog and Digital I/O routines.

```
extern "C" __declspec(dllexport) int __stdcall RemoveDriver(void);
```

Input : None

Output : Always return 1.

Note: This function should be invoked before the program closes and it will release the device driver and memory for **ib_wdt.dll**. If the program is closed without calling this routine, resource leak may occur.

- **Watchdog**

extern "C" __declspec(dllexport) char * __stdcall GetWDTInfo(void);

Input : None
Output : Return a string that describes the module information.

extern "C" __declspec(dllexport) int __stdcall EnableWDT(int);

Input : Timer interval, depending on the WDT chip
Output : Always return “0”

For further information, refer to the datasheet for WDT or contact your sales representative.

extern "C" __declspec(dllexport) int __stdcall DisableWDT(int);

Input : Dummy data and will be ignored.
Output : Always return “0”

- **Digital I/O**

extern "C" __declspec(dllexport) int __stdcall IsDioAvailable(int);

Input : Dummy data and will be ignored

Output : Return 1 if the digital I/O is available, otherwise return 0.

extern "C" __declspec(dllexport) int __stdcall SetDioInputMask(int);

Input: : Hardware parameter for digital I/O input function call.

For example, the 6 GPIO functions:

GPIO_0 to GPIO_2 are mapped as the input functions, and GPIO_4 to GPIO_6 are mapped as the output functions.

In this case, the parameter for “SetDioInputMask” is 0x07 and it indicates that GPIO_0 to GPIO_2 are the input functions.

The parameter for “SetDioOutputMask” is 0x70 and it indicates that GPIO_4 to GPIO_6 are the output functions.

Output : Dummy data and should be ignored.

extern "C" __declspec(dllexport) int __stdcall SetDioOutputMask(int);

Input : Hardware parameter for digital I/O output function call

For further information, refer to the following explanation of “SetDioInputMask” routine.

SetDioInputMask:

```
if ((*lpIsDioAvailable)(0))
        {
                int DioInput;
                printf(" Test for digital IO ....\n");

                //Please check digital IO setting in BIOS setup utility
                //Here example as follow:
                //    Digital GPIO [1..3] = Input
                //    Digital GPIO [4..6] = Output

                //set hardware information for GPIO chip
                //bit 0..2 : input for GPIO_0 to GPIO_2
                //bit 4..6 : output for GPIO_4 to GPIO_6
                (*lpSetDioInputMask)(0x07);
                (*lpSetDioOutputMask)(0x70);

                DioInput = (*lpGetDioInput)(0);
                printf(" Current GPIO_0 to GPIO_2 is 0x%X\n", DioInput);

        while(1)
        {
                        (*lpSetDioOutput)(0x00);
                        printf(" Set GPIO_4 to GPIO_6 to LOW\n");
                        (*lpSetDioOutput)(0x70);
                        printf(" Set GPIO_4 to GPIO_6 to HIGH\n");
                        (*lpSetDioOutput)(0x30);
                        printf(" Set GPIO_4 to GPIO_5 to HIGH\n");
                        (*lpSetDioOutput)(0x60);
                        printf(" Set GPIO_5 to GPIO_6 to HIGH\n");

                        Sleep(500);
                        if (_kbhit())
                        {
                break;
          }//if (kbhit())
                }//while(1)
          }//if ((*lpIsDioAvailable)(0))
```

Note: Be sure to set up the input / output for GPIO bits in BIOS identically with the GPIO hardware information above.

E. Motherboard MCU ISP Specifications

1.) Description

Security MCU provides following functionality

- **Getting Firmware Version**
  Software can get the firmware version, MCU provides commands to get current Major Version, Minor version and build version.

- **Setting Power-On-Delay Timer**
  MCU provides command to set power on delay timer, when arrived the setting time, MCU will send power button to let the system power on.

- **Setting Power-Off-Delay Timer**
  MCU provides command to set power off delay timer, when arrived the setting time, MCU will send power button to let the system power off.

- **Setting DC-Off Delay Timer**
  MCU provides command to set DC off delay timer, when arrived the setting time, MCU will cut off system power.

- **Setting Low Delay Timer**
  MCU provides command to set low delay timer, when car battery or UPS battery voltage is lower than UVP setting voltage, start low delay timer, when arrived the setting time, MCU will send power button to let the system power off.

- **Setting THL Delay Timer**
  MCU provides command to set low delay timer, when system temperature is lower than THL low setting temperature or higher than THL max setting temperature, start THL delay timer, when arrived the setting time, MCU will send power button to let the system power off.

- **Setting THL Max Temperature**
  MCU provides command to set high temperature protection, when system temperature is higher than THL max setting value, MCU trigger THL delay timer.

- **Setting THL Low Temperature**
  MCU provides command to set low temperature protection, when system temperature is lower than THL low setting value, MCU trigger THL delay timer.

- **Setting CB UVP Voltage**
  MCU provides command to set under voltage protection of cat battery, when car battery voltage is lower than CB UVP setting value, MCU will check the voltage of UPS battery whether change power source to UPS battery or not.

- **Setting PB UVP Voltage**
  MCU provides command to set under voltage protection of UPS battery, when UPS battery voltage lower than PB UVP setting value, MCU will check the voltage of car battery whether change power source to car battery or not.

- **Setting CB Good Voltage**
  MCU provides command to set CB good voltage protection of cat battery, when CB voltage level higher than CB good voltage start charging UPS battery.

- **Setting PB Good Voltage**
  MCU provides command to set PB good voltage protection of UPS battery, when system on and car battery voltage lower than CB good and UPS battery voltage higher than PB good. System will change power source to UPS battery.

- **Getting current CB & PB Voltage**
  MCU provides command to get current car battery voltage and UPS battery voltage.

- **Setting CHG STV Voltage**
  MCU provides command to set high of UPS battery charging voltage range, when system on and UPS battery voltage higher than CHG STV. System will charge UPS battery.

- **Setting CHG SPV Voltage**
  MCU provides command to set UPS battery charging range, when system on and UPS battery voltage lower than CHG SPV. System discharge UPS battery.

- **Setting Temperature Offset**
  MCU provides command to set temperature offset. It can adjust temperature disparity between system and power board.

- **Setting CB Voltage Offset**
  MCU provides command to set car battery voltage offset. It can adjust this voltage disparity between actual measurement and detected form MCU.

- **Setting PB Voltage Offset**
  MCU provides command to set UPS battery voltage offset. It can adjust this voltage disparity between actual measurement and detected form MCU.

- **Getting status**
  MCU provides command to get current status which includes the power on delay timer setting, power off delay timer setting, low delay setting value, DC off timer value, temperature protect setting value, voltage protect setting value, Car battery & UPS battery voltage offset setting value, UPS battery charge voltage setting value and current Car battery (CB) & UPS battery(PB) voltage and current temperature of power borad.

Note: Before using the ISP, your software engineer has to make sure the firmware version (GET_FORMWARE_VERSION) MUST be as follows to make the ISP function workable.

Major version is 0.

Minor version is 0.

Build version is 5 (or above).

2.) Protocol

- **Signal transmit format**

  Bandwidth
  Baud rate: 19200 bps

  Data Format
  Parity: No Parity
  1 start bit
  8 data bits
  1 stop bit

- **Packet Format**

Header	Size	Command	Data	CRC
2 bytes	1 byte	1 byte	0 – 64 bytes	2 bytes

Header bytes indicate start of the packet.

Size specifies number of bytes for data field.

Command identifies action, which is required to be performed on the data.

CRC verifies data integrity for header, size, command and data bytes.

- **CRC**

Protocol uses 16-bit CCITT CRC to verify data integrity.
$P(x) = X^{16}+X^{12}+X^{5}+1$

```
unsigned calc_crc(unsigned char *data, unsigned n, unsigned start) {
      unsigned I, k, q, c, crcval;
      crcval=start;
      for (I=0; I<n; I++) {
            c=data(I) & 0xFF;
            q=(crcval^c) & 0x0F;
            crcval=(crcval>>4)^(q*0x1081);
            q=(crcval^(c>>4)) & 0x0F;
            crcval=(crcval>>4)^(q*0x1081);
      }
      return crcval;
}
```

- **Communications flow**

  Communication between PC and Security MCU utilizes Master-Slave model, where PC is a master, and Security MCU is a slave. Master sends requests to the slave, and slave has to reply to them. Slave acts like a passive device and cannot send any requests to the master.

3.) Command and Reply Codes

- **Summary**

Code	Value	Description
GET_FIRMWARE_VERSION	0x01	Get Firmware version
SET_POWER_ON_DELAY_TIMER	0x24	Power on delay timer setting
SET_POWER_OFF_DELAY_TIMER	0x25	Power off delay timer setting
SET_DC_OFF_DELAT_TIMER S	0x26	DC off delay timer setting
SET_LOW_DELAY_TIMER	0x27	CB & PB protect voltage timer setting
SET_THL_DELAY_TIMER	0x28	Temperature protect timer setting
SET_THL_MAX	0xA0	High temperature protect setting
SET_THL_LOW	0xA1	Low temperature protect setting
GET_THL_NOW	0xA2	Get current temperature
SET_CB_UVP	0xB0	Setting Low voltage protect for CB(car battery)
SET_PB_UVP	0xB1	Setting Low voltage protect for PB(ups battery)
SET_CB_POWRE_GOOD	0xB2	Setting CB (car battery) good voltage
SET_PB_POWRE_GOOD	0xB3	Setting PB (ups battery) good voltage
GET_CURRENT_CB_PB_VOLTAGE	0xB4	Get current voltage for CB & PB

Code	Value	Description
SET_CHG_STV	0xB5	UPS charge voltage setting
SET_CHG_SPV	0xB6	UPS discharge voltage setting
SET_THL_OFFSET	0xC0	Adjust difference for temperature
SET_CB_OFFSET	0xC1	Adjust difference for CB voltage
SET_PB_OFFSET	0xC2	Adjust difference for PB voltage
GET_STATUS	0x20	Get Current Status

- **Getting Firmware Version**

Parameter: GET_FIRMWARE_VERSION

Reads version number of the security MCU firmware.

Request:

Header	Size	Command	Data	CRC
0xFF 0xEE	0x00	GET_FIRMWARE_VERSION	None	

Reply:

Header	Size	Command	Data	CRC
0xFF 0xEE	Size of Version structure	GET_FIRMWARE_VERSION	BSL Version Structure	

BSL Version Structure

Field	Type	Description
Major Version	Byte	Major version number (0x00)
Minor Version	Byte	Minor version number (0x00)
Build	Byte	Build version number (0x05)

Note: In this spec, MCU's build version must be 0x05.

- **Setting Power-On Delay Timer**

  Parameter: SET_POWER_ON_DELAY_TIMER

  Sets power on delay timer.

  Request:

Header	Size	Command	Data0	Data1	CRC
0xFF 0xEE	0x02	SET_POWER_ON_DELAY_TIMER	N (mins)	N (secs)	

  Reply

Header	Size	Command	Data	CRC
0xFF 0xEE	0x00	SET_POWER_ON_DELAY_TIMER	None	

  Note: Data1 range must be 0 ~ 59.

- **Setting Power-Off Delay Timer**

  Parameter: SET_POWER_OFF_DELAY_TIMER

  Sets power off delay timer.

  Request:

Header	Size	Command	Data0	Data1	CRC
0xFF 0xEE	0x02	SET_POWER_OFF_DELAY_TIMER	N (mins)	N (secs)	

  Reply

Header	Size	Command	Data	CRC
0xFF 0xEE	0x00	SET_POWER_OFF_DELAY_TIMER	None	

  Note: Data1 range must be 0 ~ 59.

- **Setting DC-Off-Delay Timer**

  Parameter: SET_DC_OFF_DELAY_TIMER

  Sets DC off delay timer.

  Request:

Header	Size	Command	Data0	Data1	CRC
0xFF 0xEE	0x02	SET_DC_OFF_DELAY_TIMER	N (mins)	N (secs)	

  Reply

Header	Size	Command	Data	CRC
0xFF 0xEE	0x00	SET_DC_OFF_DELAY_TIMER	None	

  Note: Data1 range must be 0 ~ 59.

- **Setting Low Delay Timer**

  Parameter: SET_LOW_ DELAY_TIMER

  Sets low delay timer.

  Request:

Header	Size	Command	Data0	Data1	CRC
0xFF 0xEE	0x02	SET_LOW_DELAY_TIMER	N (mins)	N (secs)	

  Reply

Header	Size	Command	Data	CRC
0xFF 0xEE	0x00	SET_LOW_DELAY_TIMER	None	

  Note: Data1 range must be 0 ~ 59.

- **Setting THL Delay Timer**

  Parameter: SET_THL_ DELAY_TIMER

  Sets THL delay timer.

  Request:

Header	Size	Command	Data0	Data1	CRC
0xFF 0xEE	0x02	SET_THL_DELAY_TIMER	N (mins)	N (secs)	

  Reply

Header	Size	Command	Data	CRC
0xFF 0xEE	0x00	SET_THL_DELAY_TIMER	None	

  Note: Data1 range must be 0 ~ 59.

- **Setting THL Max**

  Parameter: SET_THL_ MAX_TEMPERATURE_VALUE

  Sets max. temperature protection.

  Request:

Header	Size	Command	Data	CRC
0xFF 0xEE	0x01	SET_THL_MAX	N (degree)	

  Reply

Header	Size	Command	Data	CRC
0xFF 0xEE	0x00	SET_THL_MAX	None	

  Note: Data range must be +85~ (-50).

- **Setting THL Low**

  Parameter: SET_THL_ LOW_TEMPERATURE_VALUE

  Sets the low temperature protection.

  Request:

Header	Size	Command	Data	CRC
0xFF 0xEE	0x01	SET_THL_LOW	N (degree)	

  Reply

Header	Size	Command	Data	CRC
0xFF 0xEE	0x00	SET_THL_LOW	None	

  Note: Data range must be +85~ (-50).

- **Getting THL Now**

  Parameter: GET_THL_ NOW

  Gets the current temperature.

  Request:

Header	Size	Command	CRC
0xFF 0xEE	0x00	SET_THL_MOW	

  Reply

Header	Size	Command	Data	CRC
0xFF 0xEE	0x01	SET_THL_NOW	N (degree)	

- **Setting CB UVP**

Parameter: SET_CB_UVP

Sets under voltage protection for car battery.

Request:

Header	Size	Command	Data0	Data1	CRC
0xFF 0xEE	0x02	SET_CB_UVP	High Byte	Low Byte	

Reply

Header	Size	Command	Data	CRC
0xFF 0xEE	0x00	SET_CB_UVP	None	

Round off to the 1st decimal place.
Ex. Setting 11.2V mapping to 11200 (11.2 x 1000) = 2BC0 (hex)
Data0 = 2B (hex)
Data1 = C0 (hex)

- **Setting PB UVP**

Parameter: SET_PB_UVP

Sets under voltage protection for UPS battery.

Request:

Header	Size	Command	Data0	Data1	CRC
0xFF 0xEE	0x02	SET_PB_UVP	High Byte	Low Byte	

Reply

Header	Size	Command	Data	CRC
0xFF 0xEE	0x00	SET_PB_UVP	None	

Round off to the 1st decimal place.
Ex. Setting 11.2V mapping to 11200 (11.2 x 1000) = 2BC0 (hex)
Data0 = 2B (hex)
Data1 = C0 (hex)

- **Setting CB Good**

Parameter: SET_CB_GOOD

Sets Power Good voltage for car battery.

Request:

Header	Size	Command	Data0	Data1	CRC
0xFF 0xEE	0x02	SET_CB_GOOD	High Byte	Low Byte	

Reply

Header	Size	Command	Data	CRC
0xFF 0xEE	0x00	SET_CB_GOOD	None	

Round off to the 1st decimal place.
Ex. Setting 11.2V mapping to 11200 (11.2 x 1000) = 2BC0 (hex)
Data0 = 2B (hex)
Data1 = C0 (hex)

- **Setting PB Good**

Parameter: SET_PB_GOOD

Sets Power Good voltage for UPS battery.

Request:

Header	Size	Command	Data0	Data1	CRC
0xFF 0xEE	0x02	SET_PB_GOOD	High Byte	Low Byte	

Reply

Header	Size	Command	Data	CRC
0xFF 0xEE	0x00	SET_PB_GOOD	None	

Round off to the 1st decimal place.
Ex. Setting 11.2V mapping to 11200 (11.2 x 1000) = 2BC0 (hex)
Data0 = 2B (hex)
Data1 = C0 (hex)

- **Getting Current CB & PB Value**

  Parameter: GET_CURRENT_CB_PB_VOLTAGE

  Gets current voltage form car battery and UPS battery.

  Request:

Header	Size	Command	CRC
0xFF 0xEE	0x00	GET_CURRENT_CB_PB_VOLTAGE	

  Reply

Header	Size	Command	Data0	Data1	Data2	Data3	CRC
0xFF 0xEE	0x04	GET_CURRENT_ CB_PB_VOLTAGE	CB (High Byte)	CB (Ligh Byte)	PB (High Byte)	PB (Ligh Byte)	

  Round off to the 1st decimal place.
  Ex. Setting 11.2V mapping to 11200 (11.2 x 1000) = 2BC0 (hex)
  Data0 = 2B (hex)
  Data1 = C0 (hex)
  Setting PB 13.2V mapping to 13200 (13.2 x 1000) = 3390 (hex)
  Data2 = 33 (hex)
  Data3 = 90 (hex)

- **Setting CHG STV**

  Parameter: SET_CHG_STV

  Sets voltage value for start charging UPS battery.

  Request:

Header	Size	Command	Data0	Data1	CRC
0xFF 0xEE	0x02	SET_CHG_STV	High Byte	Low Byte	

  Reply

Header	Size	Command	Data	CRC
0xFF 0xEE	0x00	SET_CHG_STV	None	

  Round off to the 1st decimal place.
  Ex. Setting 13.2V mapping to 13200 (13.2 x 1000) = 3390 (hex)
  Data0 = 33 (hex)
  Data1 = 90 (hex)

- **Setting CHG SPV**

Parameter: SET_CHG_SPV

Sets voltage value for stop charging UPS battery.

Request:

Header	Size	Command	Data0	Data1	CRC
0xFF 0xEE	0x02	SET_CHG_SPV	High Byte	Low Byte	

Reply

Header	Size	Command	Data	CRC
0xFF 0xEE	0x00	SET_CHG_SPV	None	

Round off to the 1st decimal place.
Ex. Setting 11.2V mapping to 11200 (11.2 x 1000) = 2BC0 (hex)
Data0 = 2B (hex)
Data1 = C0 (hex)

- **Setting THL Offset**

Parameter: SET_THL_OFFSET

Sets offset to adjust temperature difference between system and power board.

Request:

Header	Size	Command	Data	CRC
0xFF 0xEE	0x01	SET_THL_OFFSET	N (degree)	

Reply

Header	Size	Command	Data	CRC
0xFF 0xEE	0x00	SET_THL_OFFSET	None	

Note: Data range must be +40~ (-40).

- **Setting CB Offset**

Parameter: SET_CB_OFFSET

Sets offset to adjust CB voltage difference between actual measurement and detected form MCU.

Request:

Header	Size	Command	Data	CRC
0xFF 0xEE	0x01	SET_CB_OFFSET	N (V)	

Reply

Header	Size	Command	Data	CRC
0xFF 0xEE	0x00	SET_CB_OFFSET	None	

Note: Data range must be +5V~ (-5V).

- **Setting PB Offset**

Parameter: SET_CB_OFFSET

Sets offset to adjust PB voltage difference between actual measurement and detected form MCU.

Request:

Header	Size	Command	Data	CRC
0xFF 0xEE	0x01	SET_PB_OFFSET	N (V)	

Reply

Header	Size	Command	Data	CRC
0xFF 0xEE	0x00	SET_PB_OFFSET	None	

Note: Data range must be +5V~ (-5V).

- **Getting Status**

  Gets the current power on delay timer setting, power off delay timer setting, low delay setting value, DC off timer value, temperature protect setting value, voltage protect setting value, Car battery & UPS battery voltage offset setting value, UPS battery charge voltage setting value and current Car battery (CB) & UPS battery(PB) voltage and current temperature of power board.

  Request:

Header	Size	Command	Data	CRC
0xFF 0xEE	0x00	GET_STATUS	None	

  Reply:

Header	Size	Command	Data	CRC
0xFF 0xEE	0x05	GET_STATUS	Status structure	

  Status Structure

Field	Type	Description
Power on delay timer	Word	Byte0 + Byte1
Power off delay timer	Word	Byte2 + Byte3
DC off delay timer	Word	Byte4 + Byte5
Low delay timer	Word	Byte6 + Byte7
THL delay timer	Word	Byte8 + Byte9
THL Max	Byte	Byte10
THL Low	Byte	Byte11
THL Now	Byte	Byte12
CB uvp	Word	Byte13 + Byte14
PB uvp	Word	Byte15 + Byte16
CB good	Word	Byte17 + Byte18
PB good	Word	Byte19 + Byte20
CHG stv	Word	Byte21 + Byte22
CHG spv	Word	Byte23 + Byte24
CB NOW	Word	Byte25 + Byte26
PB NOW	Word	Byte27 + Byte28
THL offset	Byte	Byte29
CB offset	Byte	Byte30
PB offset	Byte	Byte31