MBN501R
MBN501D

Networking Motherboards

with Intel® Celeron® N3350 / Pentium® N4200

User's Manual

Version 1.0
(July 2018)

iBASE

Disclaimer

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

Trademarks

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

Compliance

CE

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

FC

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

WEEE

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

Green IBASE

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

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

Important Safety Information

Carefully read the precautions before using the device.

Environmental conditions:

- Lay the device horizontally on a stable and solid surface in case the device may fall, causing serious damage.
- 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 at ambient temperatures 0˚C ~ 60˚C.
- DO NOT LEAVE THIS DEVICE IN AN ENVIRONMENT WHERE THE STORAGE TEMPERATURE MAY GO BELOW -20˚C OR ABOVE 80˚C. This could damage the device. The device must be used in a controlled environment.

Care for your IBASE products:

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

WARNING

Attention during use:

- Do not use this product near water.
- Do not spill water or any other liquids on your device.
- Do not place heavy objects on the top of the device.
- Operate this device from the type of power indicated on the marking label. If you are not sure of the type of power available, consult your distributor or local power company.
- Do not walk on the power cord or allow anything to rest on it.
- If you use an extension cord, make sure that the total ampere rating of the product plugged into the extension cord does not exceed its limits.
- When handling processor chips or memory modules, avoid touching their pins or gold fingers. Put modules or peripherals back into antistatic bags when they are not in use or not installed in the chassis.

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. Under no circumstances should the Lithium battery cell be shorted; otherwise the battery cell may heat up or cause potential burn hazards.

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.

Table of Contents

Chapter 1
General Information

The information provided in this chapter includes:

- Features
- Packing List
- Optional Accessories
- Specifications
- Overview
- Dimensions

1.1 Introduction

MBN501 series is specifically designed for the network security and management market. There are plenty of applications adopting network security or network management as listed below.

MBN501 series includes the following models:

- **MBN501R** (for rackmount form factor)
- **MBN501D** (for desktop form factor)

Network Security Applications:

- Firewall
- Unified Threat Management (UTM)
- Virtual Private Network (VPN)
- Proxy Server
- Caching Server

Network Management Applications:

- Load balancing
- Quality of Service
- Remote Access Service

The networking appliance product line covers the spectrum from offering platforms designed for:

- SOHO
- SMB
- Enterprise

Each product is designed to address the distinctive requirements of its respective market segment from cost effective entry-level solutions to high throughput and performance-bound systems for the enterprise level.

1.2 Features

- Entry level platform designed with Intel® Celeron® N3350 / Pentium® N4200 processor
- 2 x DDR3L SO-DIMM 1867 MHz, expandable up to 16 GB (non-ECC)
- 6 x PCIe GbE LAN ports with 1 advanced LAN Bypass pair included
- 1 x CF slot
- 1 x Mini-PCIe expansion slot with PCIe and USB 2.0 signals

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.

- MBN501R / MBN501D

1.4 Optional Accessories

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

- Console Cable (160 cm, PK1-51)
- VGA Cable (40 cm, VGA21A, for MBN501R only)

1.5 Specifications

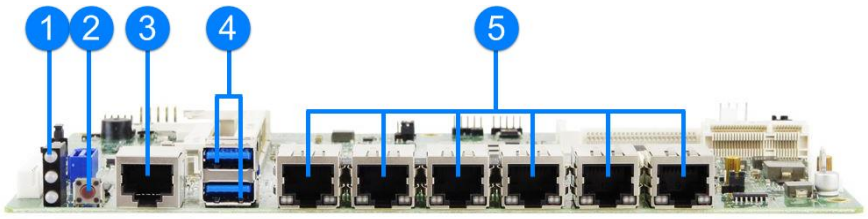
Product Name	MBN501R	MBN501D
System		
Motherboard	MBN501R-4C/2C	MBN501D-4C/2C
Operating System	• Windows 10 (64-bit) • Linux Ubuntu 16.04	
CPU	• Intel® Pentium® N4200 Quad-Core, 2.5 GHz • Intel® Celeron® N3350 Dual-Core, 2.4 GHz	
Cache	Up to 2 MB	
Chipset	Integrated	
Graphics	Integrated	
Memory	2 x DDR3L SO-DIMM up to 1867 MHz, expandable to 16 GB (non-ECC)	
Storage	• 2.5” HDD / SSD (1 bracket for HDD / SSD) • CF card (1 onboard slot)	
Network	6 x Intel® I211AT GbE	
Bypass	1 x segments (LAN5 / LAN6)	
NIC Slot	N/A	
Super I/O	Fintek F81964D-I	
BIOS	AMI BIOS	
TPM	1.2	
Watchdog	Watchdog Timer 256 segments, 0, 1, 2…255 sec/min	
Chassis	Steel with textured black color paint	
Dimensions	216 x 138 mm (8.5” x 5.43)	
Certificate	CE / FCC Class B	
I/O Ports		
Console	1 x Console port	
Traffic LAN Port	6 x RJ45 GbE LAN ports	6 x RJ45 GbE LAN ports with additional respective LEDs
USB	2 x USB 3.0	
CF	1 x CF slot (onboard)	
Micro-SIM	1 x Micro-SIM card slot	
Display	1 x VGA (via an onboard pin header)	N/A
Expansion	1 x Mini-PCIe slot with PCIe and USB 2.0 signals	

Environment	
Temperature	• **Operating:** 0 ~ 60 °C (32 ~ 140 °F) • **Storage:** -20~ 80 °C (-4 ~ 176 °F)
Relative Humidity	5 ~ 90%

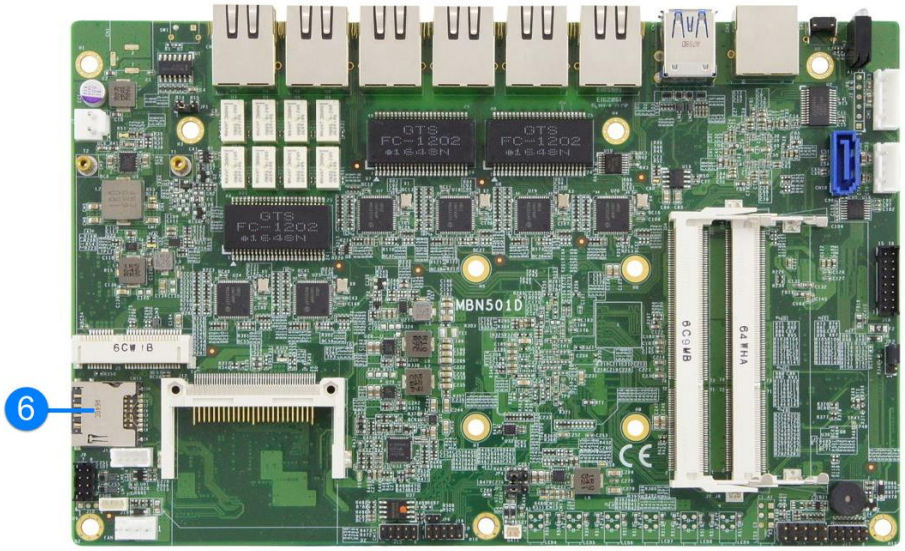
All specifications are subject to change without prior notice.

1.6 Overview – MBN501R

I/O View



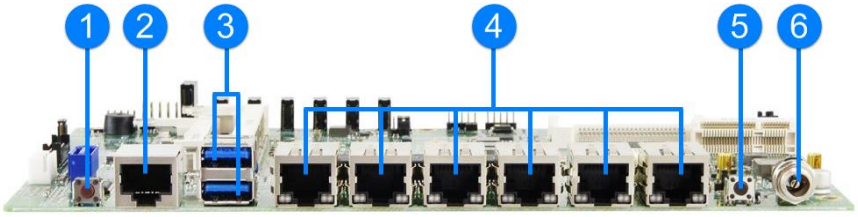
Top View



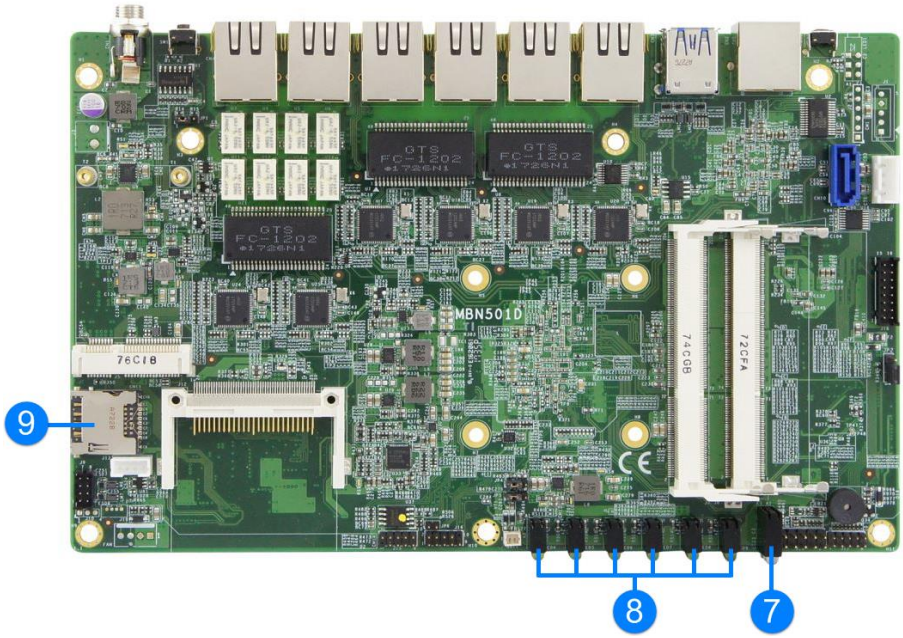
No.	Name	No.	Name
1	LED Indicators (Top to bottom: GPIO Status, HDD & Power)	4	USB 3.0 Ports
2	User Self-defined Button (with GPI Signal)	5	GbE LAN 6 Ports
3	Console Port	6	Micro-SIM Slot

1.7 Overview – MBN501D

I/O View



Top View



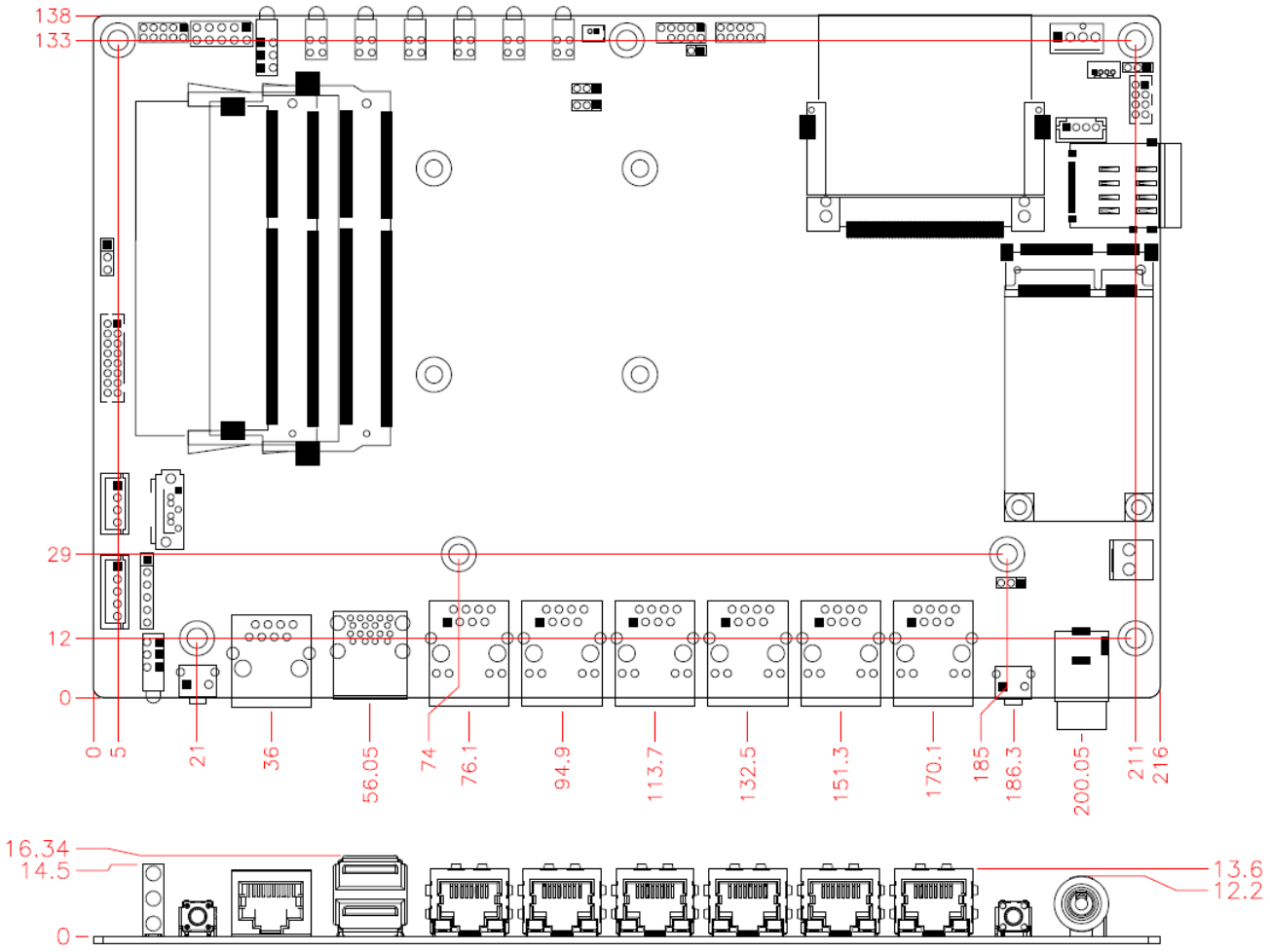
No.	Name	No.	Name
1	User Self-defined Button (with GPI Signal)	6	DC-12V Jack
2	Console Port	7	LED Indicators (Top to bottom: GPIO Status, HDD & Power)
3	USB 3.0 Ports	8	Additional LED Indicators for 6 LAN Ports (Top to bottom: Link, Speed)
4	GbE LAN 6 Ports [1]	9	Micro-SIM Slot
5	Power Button		

[1]: On MBN501D, each LAN port has additional corresponding LEDs that indicate the link and speed of the port. Refer to the description for item 8 above.

1.8 Dimensions

Unit: mm

Motherboard: MBN501 Series



Chapter 2
Hardware Configuration

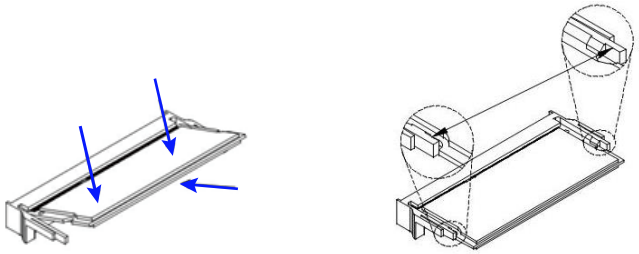
The information provided in this chapter includes:

- Installations
- Information and locations of connectors

2.1 Installations

2.1.1 Memory Installation / Replacement

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



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

2.1.2 CF Card Installation

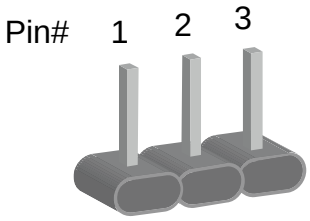
Locate the CF slot, hold the card with your thumb and index finger to insert the CF card. To remove the CF card, pull it out with your thumb and index finger directly.

2.2 Setting the Jumpers

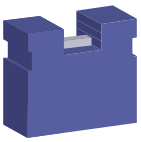
Set up and configure your product by using jumpers for various settings and features according to your needs and applications. Contact your supplier if you have doubts about the best configuration for your use.

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.

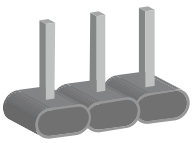
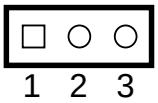
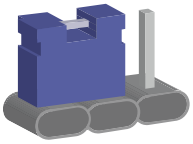
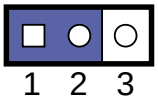
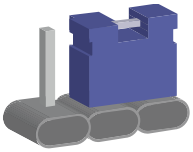
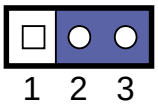


A 3-pin jumper



A jumper cap

Refer to the illustration below to set jumpers.

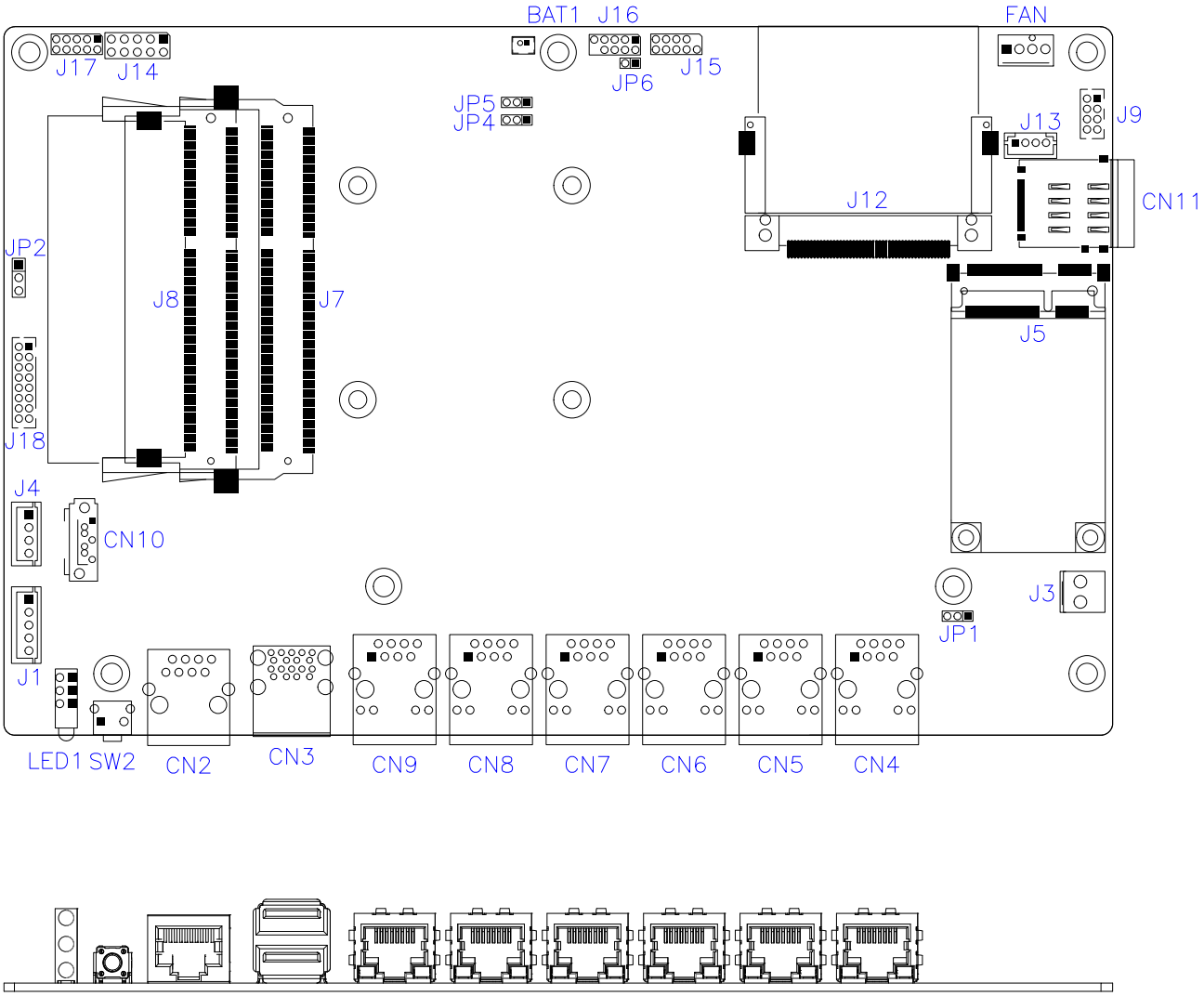
Pin closed	Oblique view	Schematic illustration in the manual
Open		
1-2		
2-3		

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

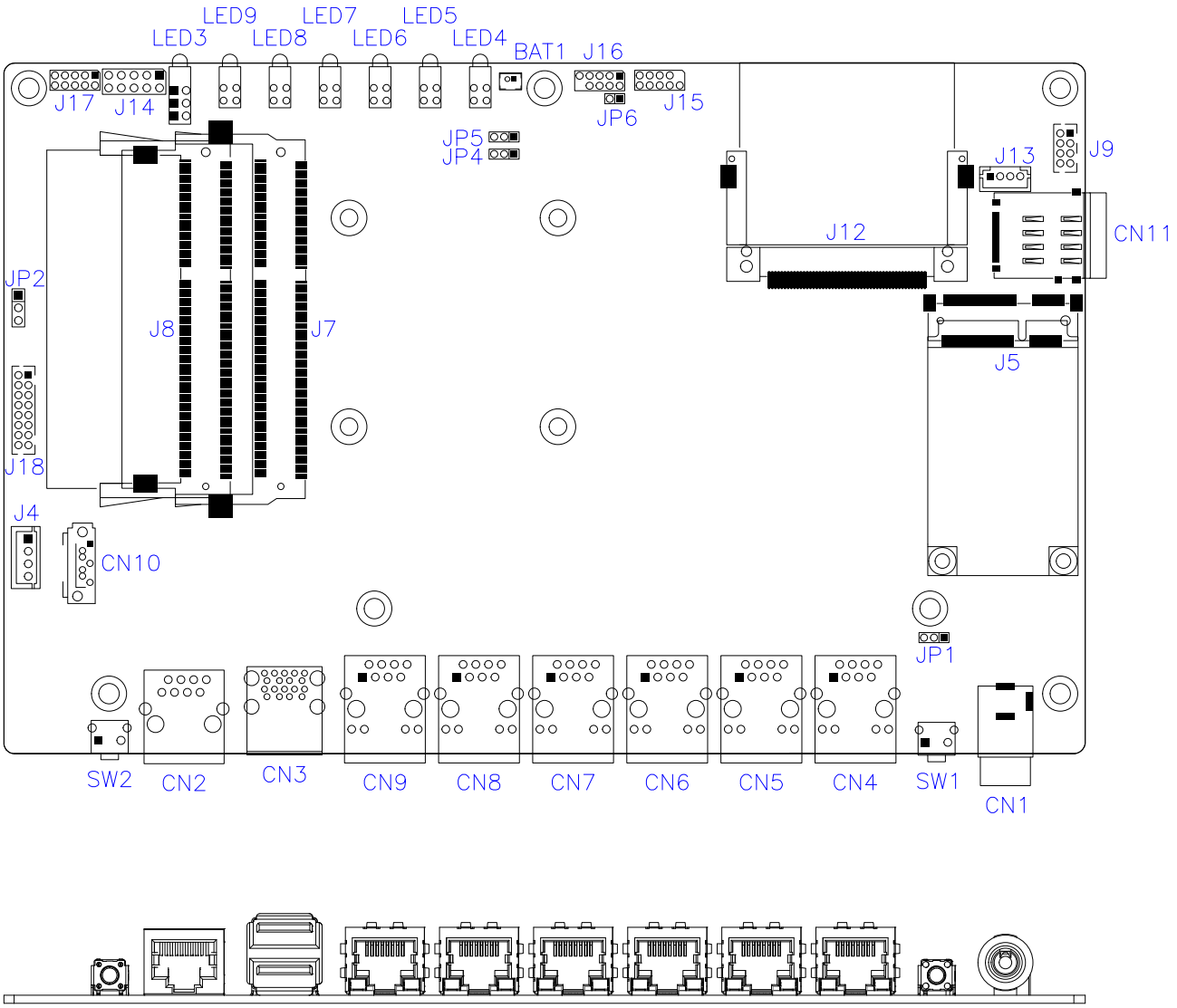
When a jumper cap is removed from two jumper pins, this jumper is **open**, i.e. turned **Off**.

2.3 Jumper & Connector Locations on Motherboard

Motherboard: MBN501**R**



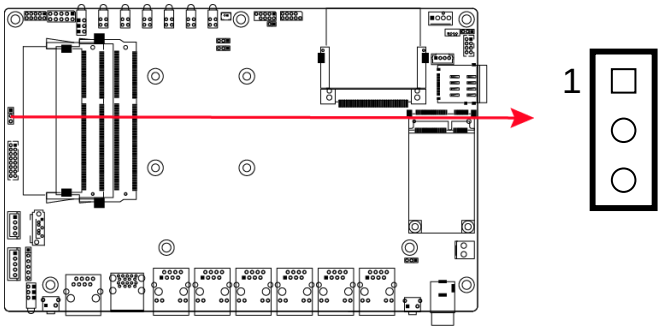
Motherboard: MBN501**D**

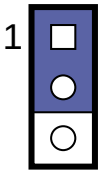
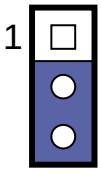


2.4 Jumpers Quick Reference

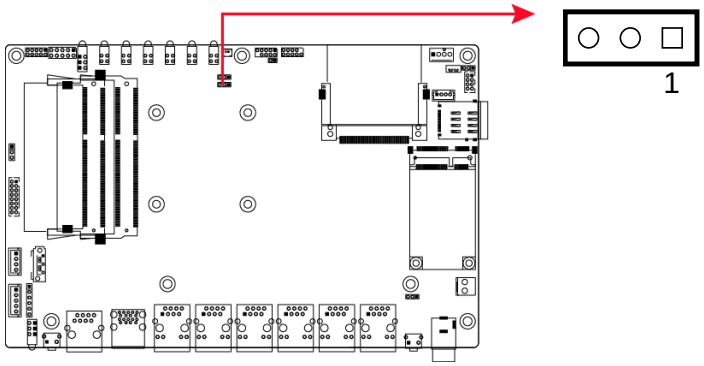
Function	Connector Name	Page
AT & ATX Mode Selection	JP2	14
ME Register Clearance	JP4	15
CMOS Data Clearance	JP5	15
Factory Use Only	JP1, JP6	--

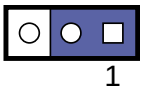
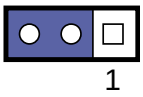
2.4.1 AT & ATX Mode Selection (JP2)



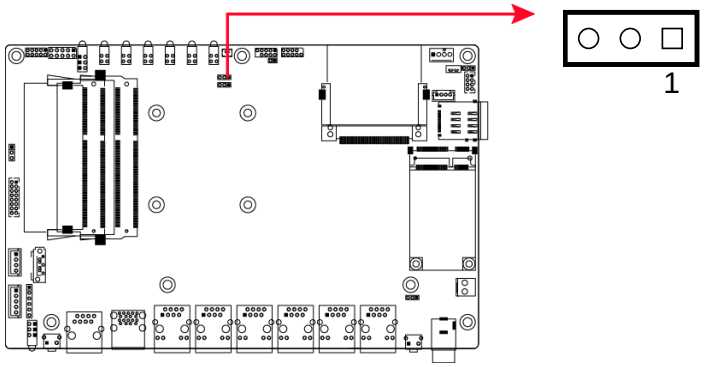
Function	Pin closed	Illustration
AT Mode (default)	1-2	
ATX Mode	2-3	

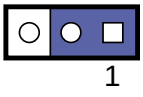
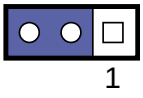
2.4.2 ME Register Clearance (JP4)



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

2.4.3 CMOS Data Clearance (JP5)



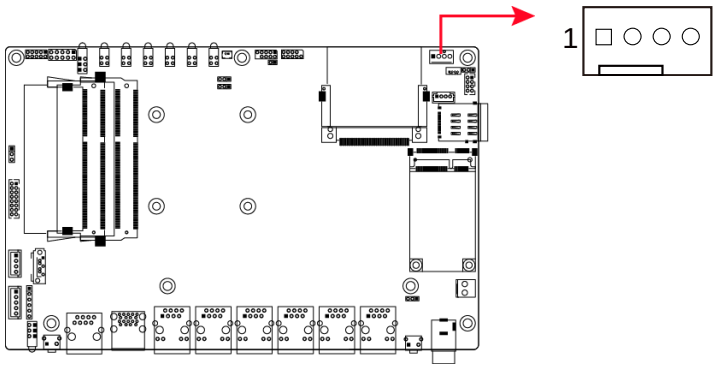
Function	Pin closed	Illustration
Normal (default)	1-2	
Clear CMOS	2-3	

2.5 Connectors Quick Reference

Function	Connector Name	Page
System Fan Power Connector	FAN	17
AT 12V Power Connector	J3 * Available only for **MBN501R**.	17
External SATA Power Connector	J4	18
VGA Connector	J18	18
USB 2.0 Port	J9	19
Digital I/O Connector	J17	19
System Function Connector	J14	20
Console Port	CN2	21
LCM COM2 Connector	J1	--
Mini-PCIe (x1)	J5	--
DDR3L SO-DIMM Slot	J7, J8	--
CF Slot	J12	--
DC 12V Power Jack	CN1 * Available only for **MBN501D**.	--
USB 3.0 Port	CN3	--
GbE LAN Port [1]	CN4, CN5, CN6, CN7, CN8, CN9	--
SATA 3.0 Port	CN10	--
SIM Slot	CN11	--
LED Connectors for **MBN501R**	LED1 (top to bottom: GPIO status, HDD, Power), LED2 (CF Card Power), LED10 (LAN Bypass)	--
LED Connectors for **MBN501D**	LED2 (CF Card Power), LED3 (top to bottom: GPIO status, HDD, power), LED4/5/6/7/8/9 for CN4/5/6/7/8/9 (top to bottom: link, speed), LED10 (LAN Bypass)	--
System Power Button	SW1 * Available only for **MBN501D**.	--
User Self-defined Button with GPI Signal	SW2	--
Factory Use Only	J13, J15, J16	--

[1]: On MBN501D, each LAN port has additional corresponding LEDs that indicate the link and speed of the port.

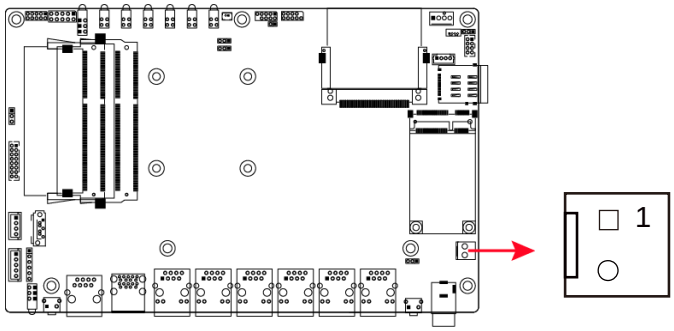
2.5.1 System Fan Power Connector (FAN)



FAN connector is available only for MBN501R.

Pin	Assigment	Pin	Assigment
1	Ground	3	Rotation detection
2	+12V	4	Rotation control

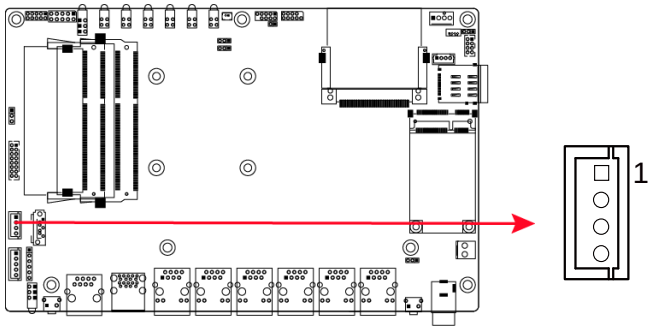
2.5.2 AT 12V Power Connector (J3)



J3 is available only for MBN501R.

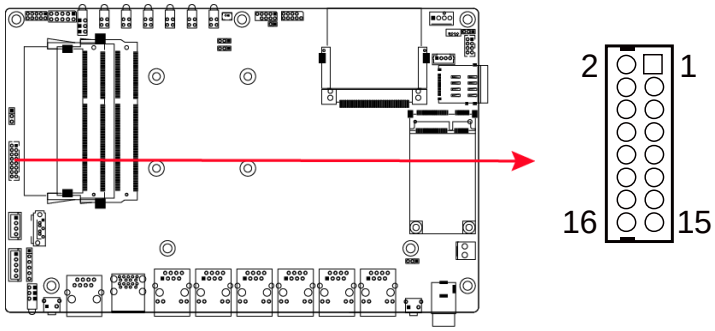
Pin	Assigment	Pin	Assigment
1	+12V	2	Ground

2.5.3 External SATA Power Connector (J4)



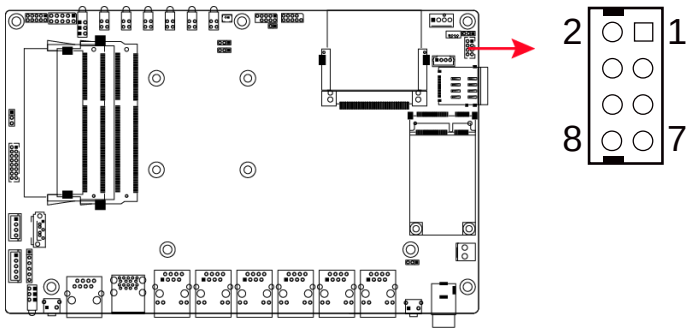
Pin	Assigment	Pin	Assigment
1	+5V	3	Ground
2	Ground	4	+12V

2.5.4 VGA Connector (J18)



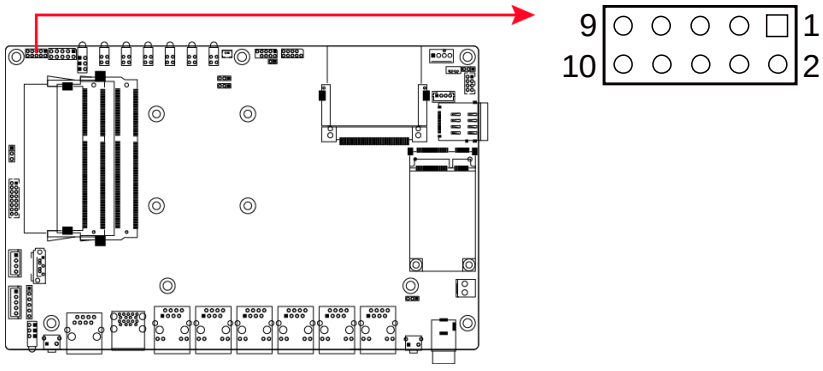
Pin	Assigment	Pin	Assigment
1	VGA_R	2	VGA_PWR
3	VGA_G	4	Ground
5	VGA_B	6	NC
7	NC	8	VGADDCDATA
9	Ground	10	HSYNC
11	Ground	12	VSYNC
13	Ground	14	VGADDCCLK
15	Ground		

2.5.5 USB 2.0 Port (J9)



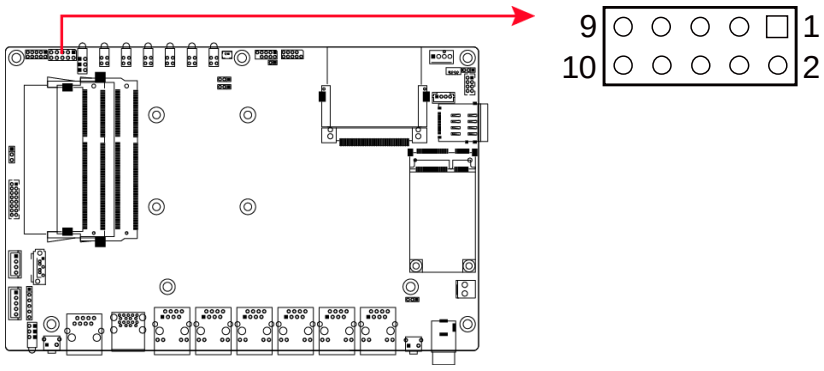
Pin	Assigment	Pin	Assigment
1	+5V	2	Ground
3	P4-	4	P5+
5	P4+	6	P5-
7	Ground	8	+5V

2.5.6 Digital I/O Connector (J17)



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

2.5.7 System Function Connector (J14)

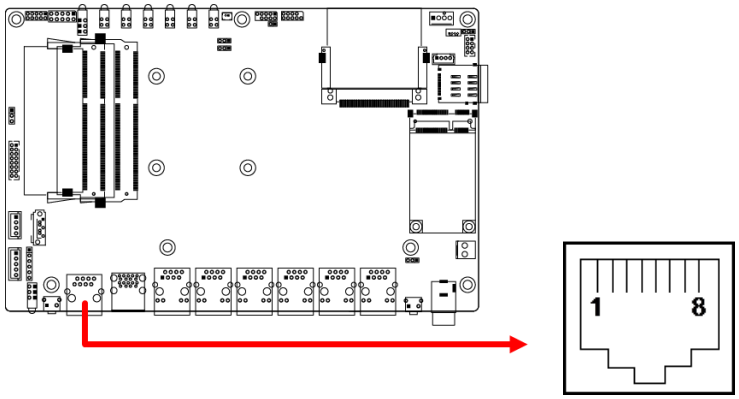


Pin	Assigment	Pin	Assigment
1	Ground	2	Power_ON
3	Ground	4	PM_SYSRST#
5	+3.3V	6	Ground
7	+3.3V	8	-HDD_LED
9	+3.3V	10	BYPASSLED

J14 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.
- **Reset Switch (Pins 3 and 4)**
  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 5 and 6)**
  This connector connects to the system power LED on control panel. This LED will light when the system turns on.
- **Hard Disk Drive LED Connector (Pins 7 and 8)**
  This connector connects to the hard drive activity LED on control panel. This LED will flash when the HDD is being accessed.
- **Bypass LED1 (Pins 9 and 10)**
  The two pins make an external LED connector light for LAN bypass.

2.5.8 Console Port (CN2)



Pin	Assigment	Pin	Assigment
1	RTS	5	Ground
2	DTR	6	RXD
3	TXD	7	DSR
4	Ground	8	CTS

Chapter 3
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

3.1 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.

3.2 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
```

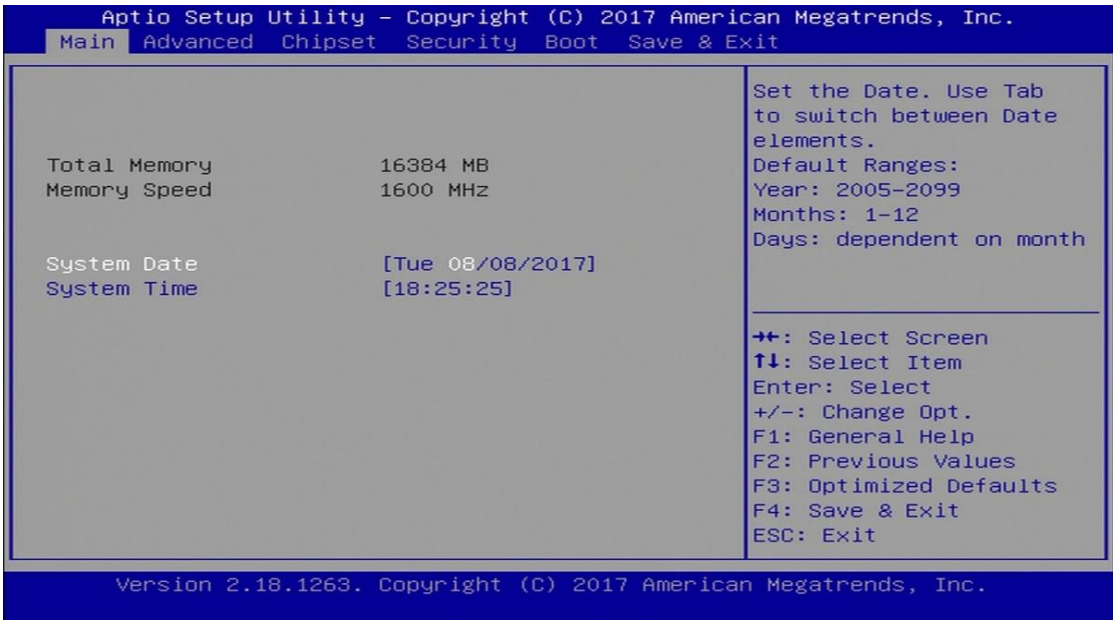
In general, press the arrow keys to highlight items, <Enter> to select, the <PgUp> and <PgDn> keys to change entries, <F1> for help, and <Esc> to quit.

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.

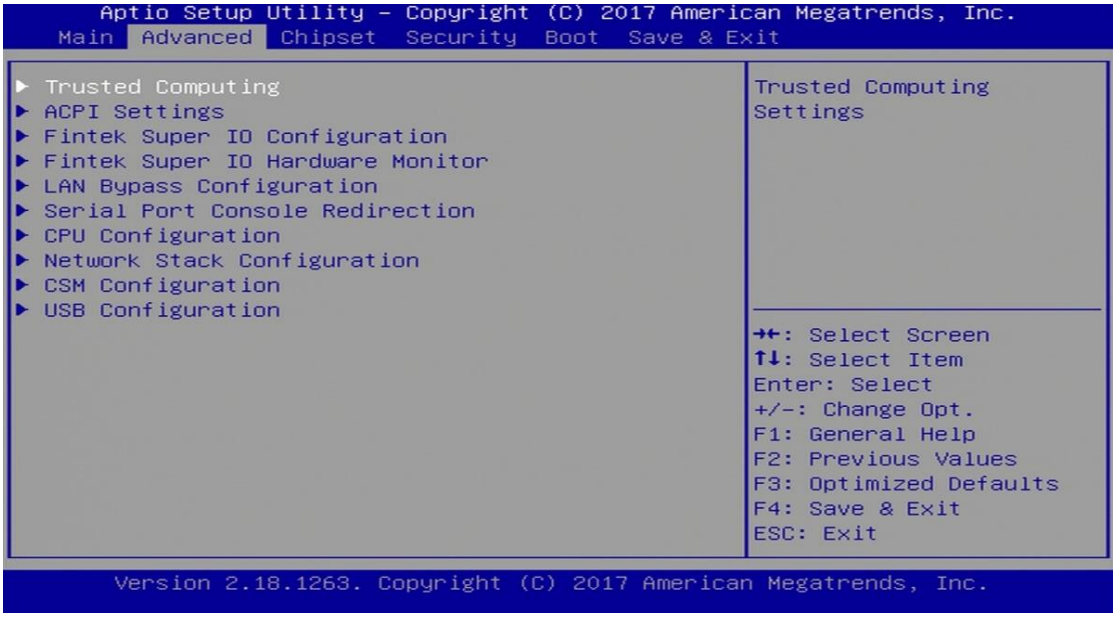
3.3 Main Settings



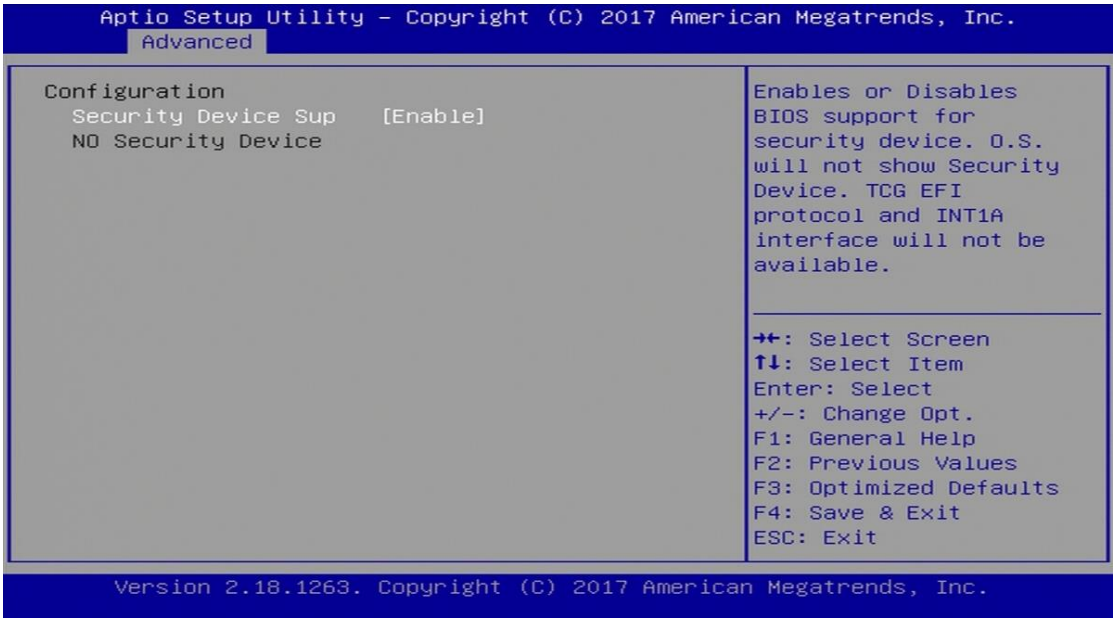
BIOS Setting	Description
System Date	Sets the date. Use the <Tab> key to switch between the data elements.
System Time	Set the time. Use the <Tab> key to switch between the data elements.

3.4 Advanced Settings

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

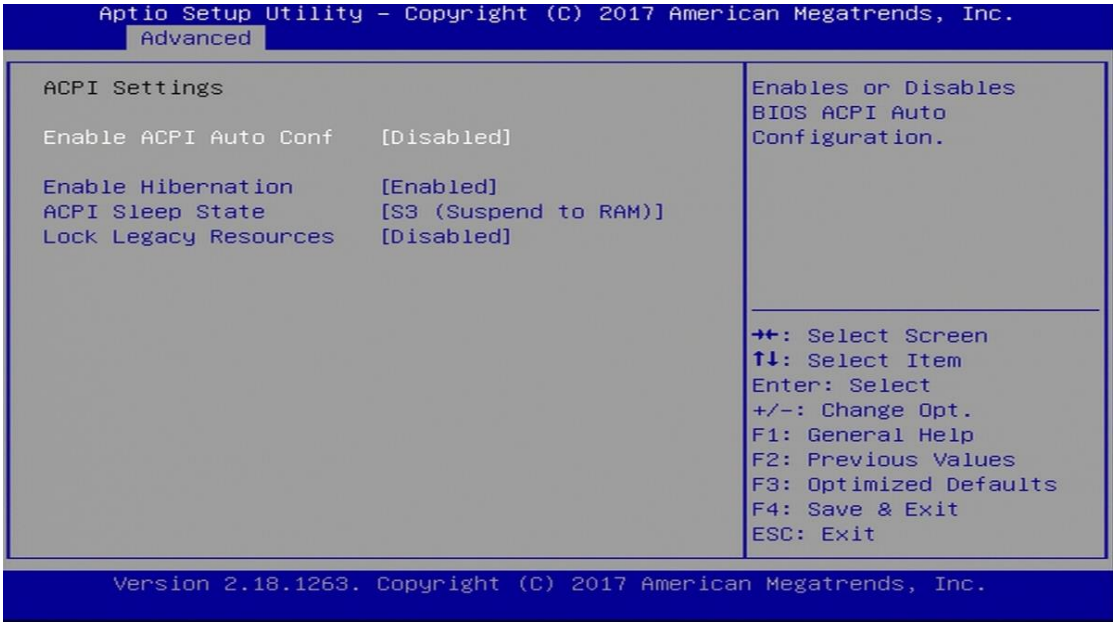


3.4.1 Trusted Computing



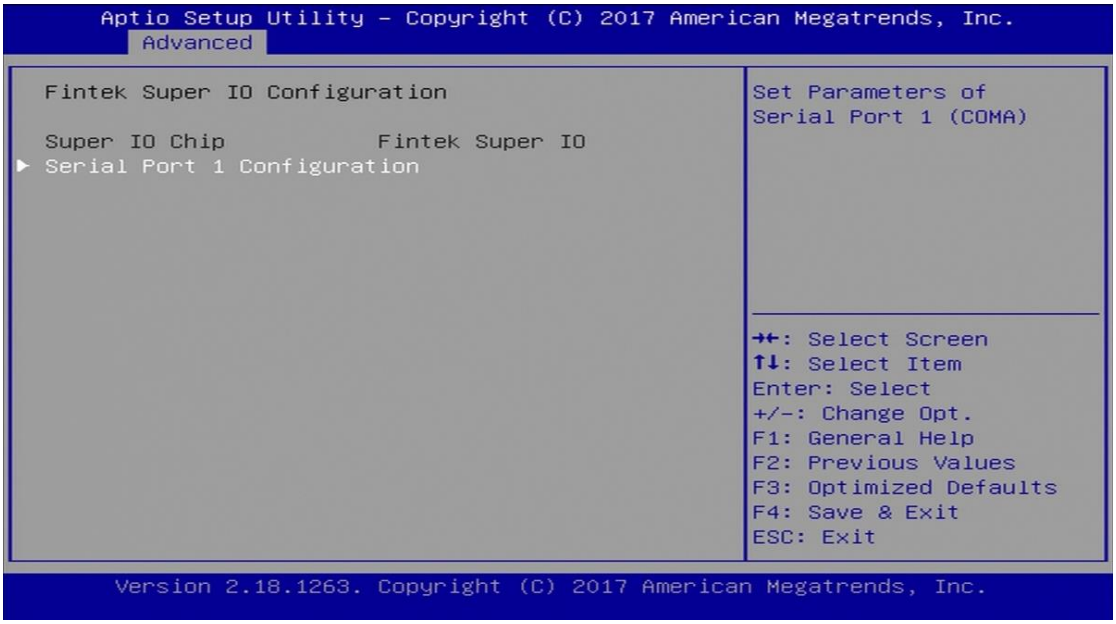
BIOS Setting	Description
Security Device Support	Enables / Disables BIOS support for security device. OS will not show the security device. TCG EFI protocol and INT1A interface will not be available.

3.4.2 ACPI Settings



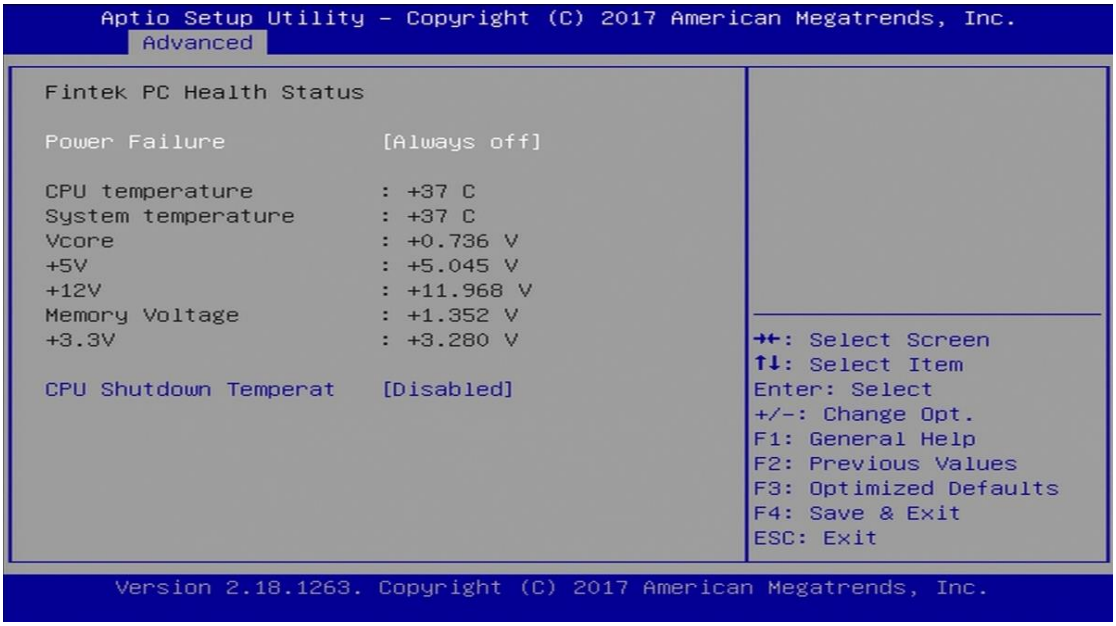
BIOS Setting	Description
Enable ACPI Auto Configuration	Enables / Disables BIOS ACPIU auto configuration.
Enable Hibernation	Enables / Disables the system ability to hibernate (OS/S4 Sleep State). This option may not be effective with some OS.
ACPI Sleep State	Selects a ACPI sleep state for the system to enter. Options: Suspend Disabled, S3 (Suspend to RAM)
Lock Legacy Resources	Enables / Disables Lock of Legacy Resources.

3.4.3 Fintek Super IO Configuration



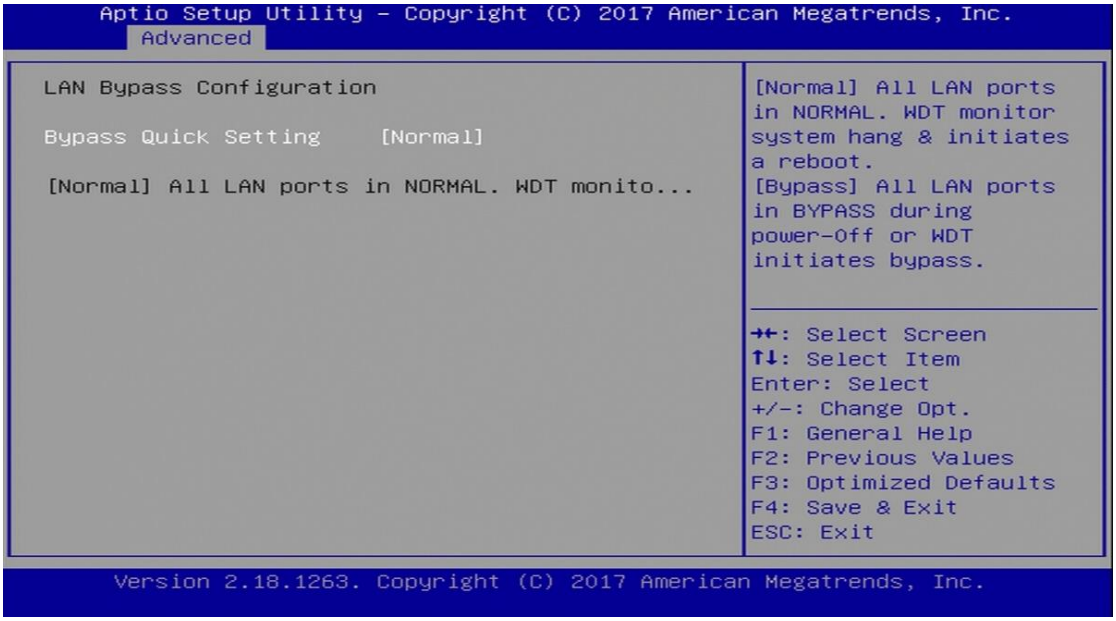
BIOS Setting	Description
Serial Port Configuration	Sets Parameters of Serial Port(s). You can enable / disable the serial port and select an optimal settings for the Super IO device.

3.4.4 Fintek Super IO Hardware Monitor



BIOS Setting	Description
Power Failure	Options: Bypass mode, Always on, Always off.
Temperatures / Voltages	These fields are the parameters of the hardware monitoring function feature of the motherboard. The values are read-only as monitored by the system and showing the PC health status
Shutdown Temperature	This field enables or disables the Shutdown Temperature

3.4.5 LAN Bypass Configuration

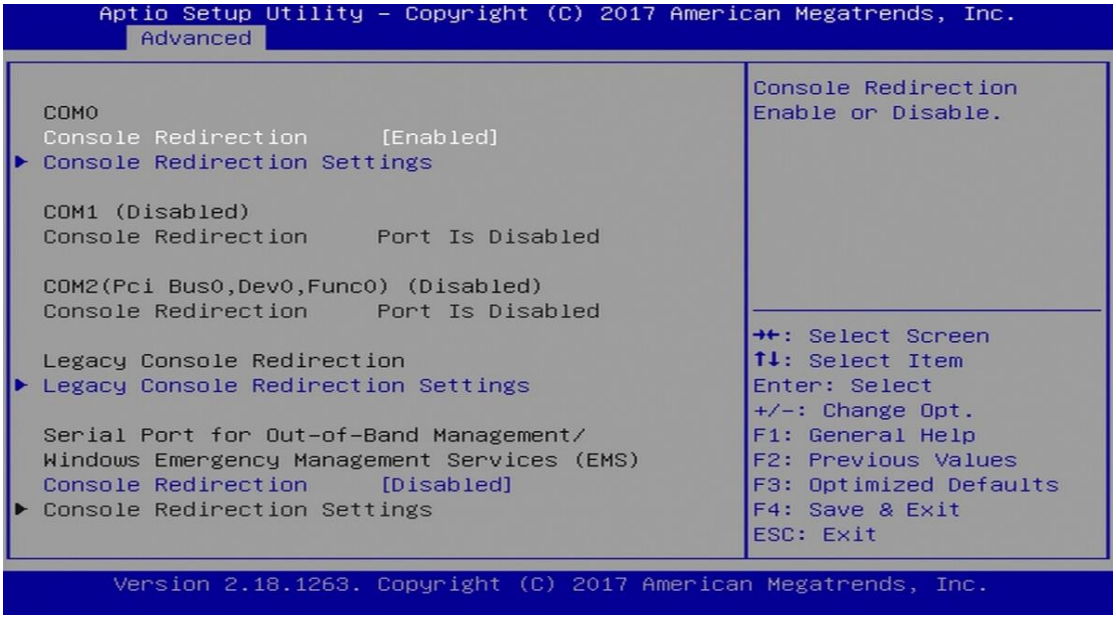


BIOS Setting	Description
Bypass Quick Setting	**Normal:** All LAN ports are in a normal state. WDT monitor system hang and initiates a reboot. **Bypass:** The paired LAN ports in Bypass during power-off or WDT initiates bypass. Options: Bypass, Normal, Firewall, Custom Define

You can verify the LAN bypass function via BIOS by referring to the steps below.

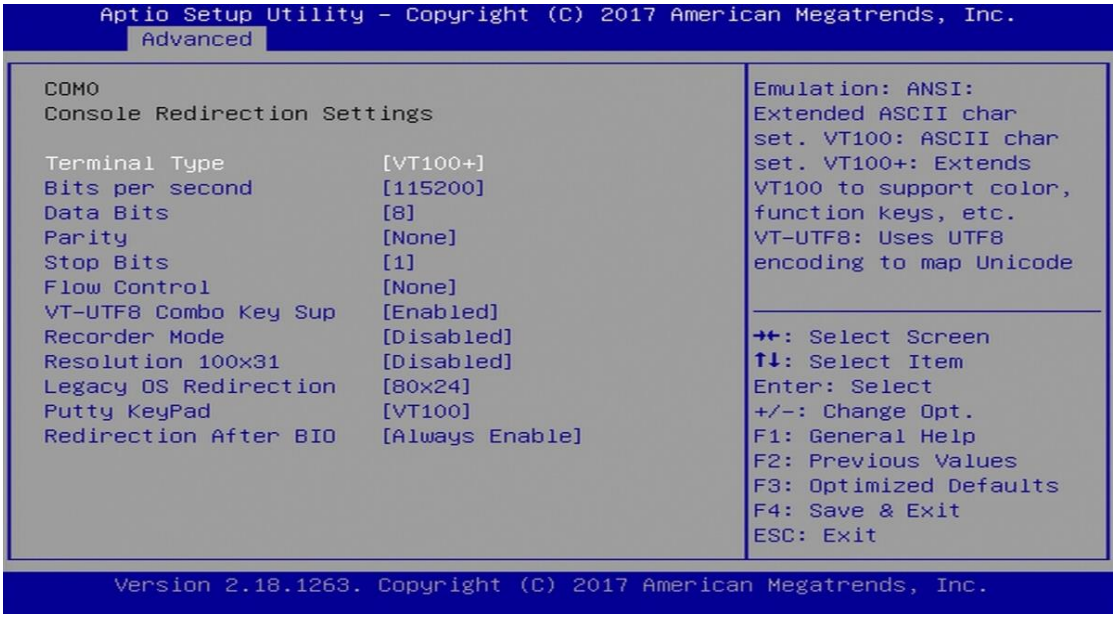
1. Select Bypass mode for the above LAN Bypass Configuration in BIOS.
2. Save the setting and exit.
3. Run your test program under MS-DOS.
4. The lit Bypass LED indicates the bypass function runs properly.

3.4.6 Serial Port Console Redirection



BIOS Setting	Description
Console Redirection	Enables / Disables console redirection.
Console Redirection Settings	Sets up the terminal type, bits, parity, flow control, combo key, recorder mode, resolution and putty keypad.
Legacy Console Redirection Settings	Sets up Legacy Console Redirection.
Console Redirection	Enables / Disables Console Redirection.

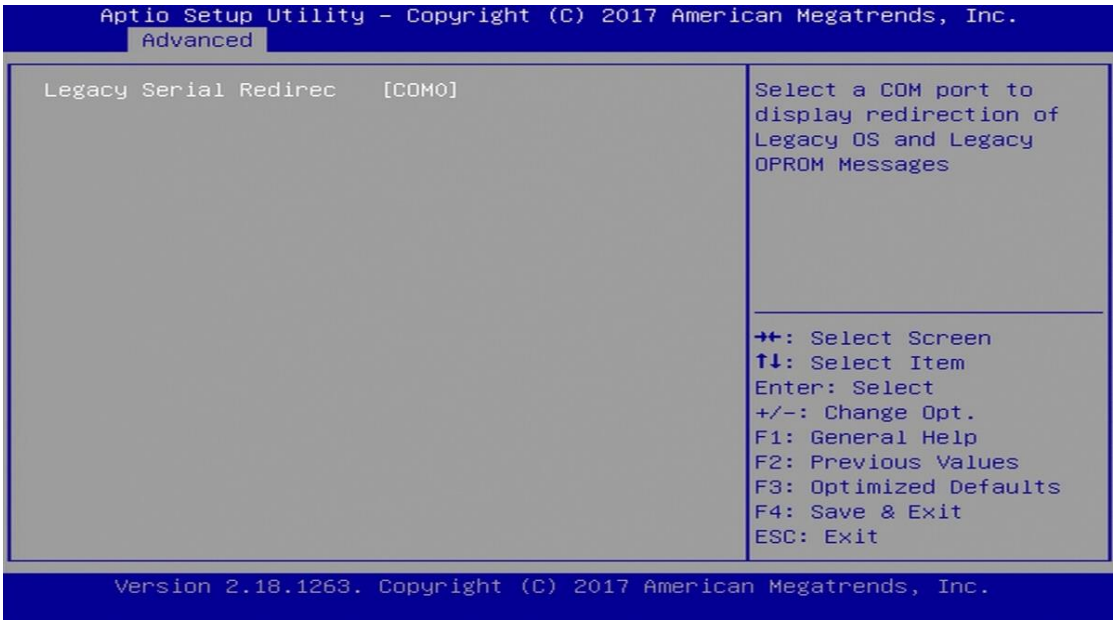
3.4.6.1. Console Redirection Settings



BIOS Setting	Description
Therminal Type	Sets the terminal type as VT100, VT100+, VT-UTF8, or ANSI.
Bits per second	Selects serial port transmission speed. The speed msut be matched on the other side. Long or noisy lines may require lower speeds. Options: 9600 19200, 38400, 57600, 115200
Data Bits	Options: 7, 8
Parity	A parity bit can be sent with the data bits to detect some transmission errors. Options: None, Even, Odd, Mark, Space
Stop Bits	Stop bits indicate the end of a serial data packet. (A start bit indicates the beginning.) The standard setting is 1 stop bit. Options: 1, 2
Flow Control	Flow control can prevent data loss from buffer overflow. When sending data, if the receiving buffers are full, a “stop” signal can be sent to stop the data flow. Options: None, Hardware RTS/CTS
VT-UTFB Combo Key Support	Enables / Disables VT-UTFB combination key support for ANSI/VT100 terminals.
Recorder Mode	Enabling this mode allows only text to be sent. This is to capture terminal data.

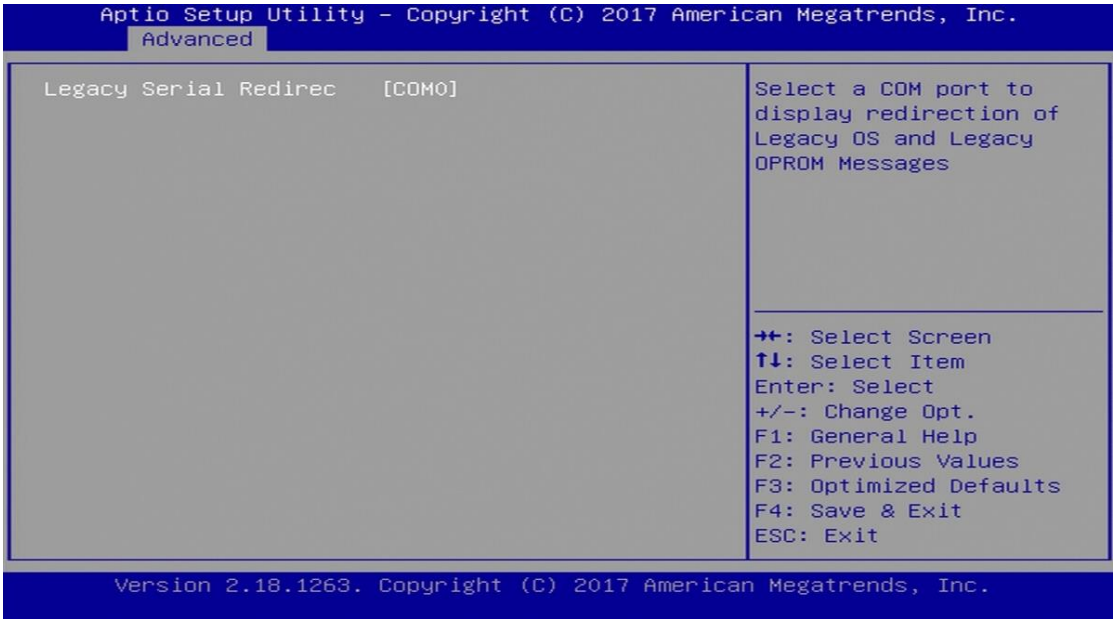
BIOS Setting	Description
Resolution 100x31	Enables / Disables extended terminal resolution.
Legacy OS Redirection	On Legacy OS, the number of rows and columns supported redirection. Options: 80 x 24, 80 x 25
Putty Keypad	Selects function eky and keypad on Putty. Options: VT100, Linux, XTERMR6, SC0, ESCN, VT400
Redirection After BIO	When Bootloader is selected, the Legacy Console Redirection is disabled before booting to legacy OS. Options: Always Enable, Bootloader

3.4.6.2. Legacy Console Redirection Settings



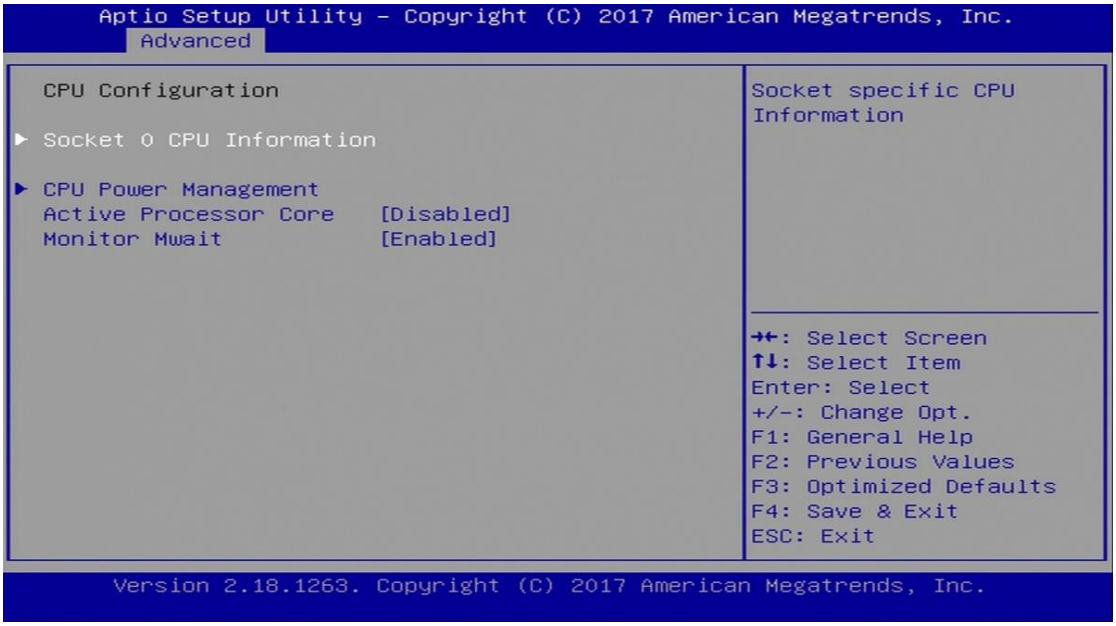
BIOS Setting	Description
Legacy Serial Redirec	Selects a COM Port to display redirection of Legacy OS and Legacy OPROM Messages. Options: COMO, COM1 (Disabled), COM2 (Pci Bus0, Dev0, Func0) (Disabled)

3.4.6.3. Console Redirection



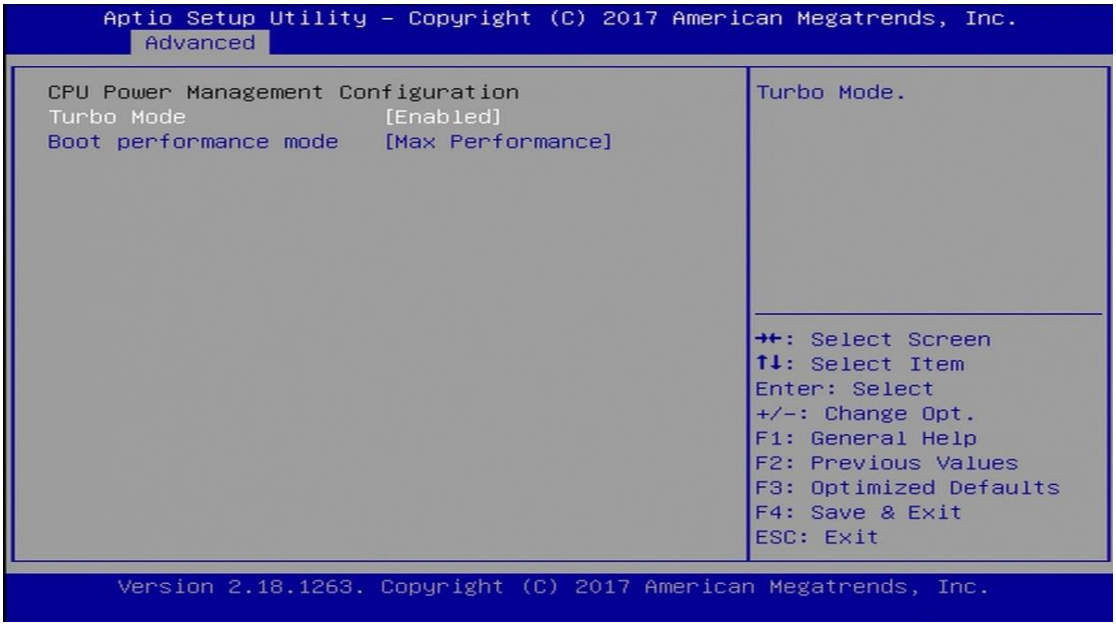
BIOS Setting	Description
Legacy Serial Redirec	Selects a COM Port to display redirection of Legacy OS and Legacy OPROM Messages. Options: COM0, COM1 (Disabled), COM2 (Pci Bus0, Dev0, Func0) (Disabled)

3.4.7 CPU Configuration



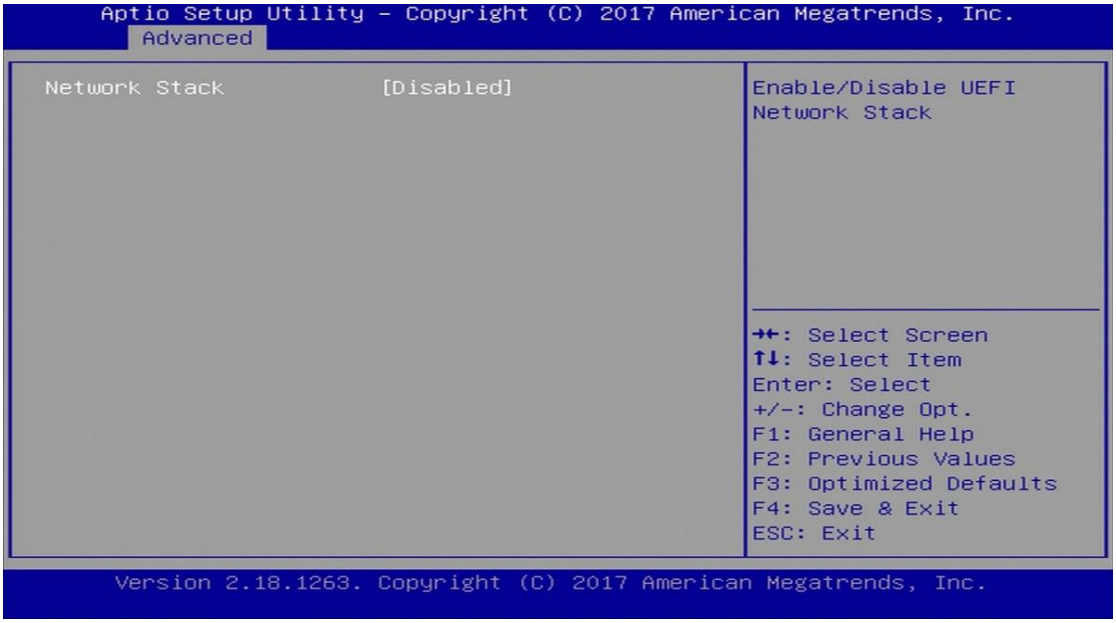
BIOS Setting	Description
Socket 0 CPU Information	Displays the socket specific CPU information.
CPU Power Management	Configures CPU power management.
Active Processor Core	Enables / Disables number of cores in each processor package.
Monitor Mwait	Enables / Disables the Monitor Mwait.

3.4.7.1. CPU Power Management



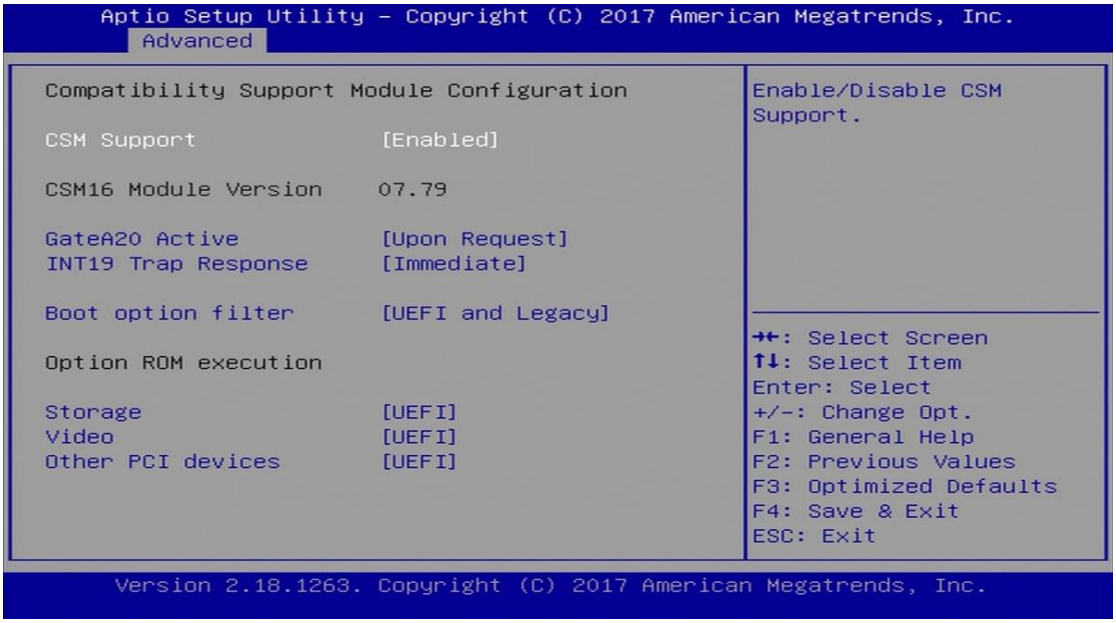
BIOS Setting	Description
Turbo Mode	Enables / Disables the turbo mode.
Boot performance mode	Selects the performance state that the BIOS will set before OS handoff. Options: Max Performance, Max Battery

3.4.8 Network Stack Configuration



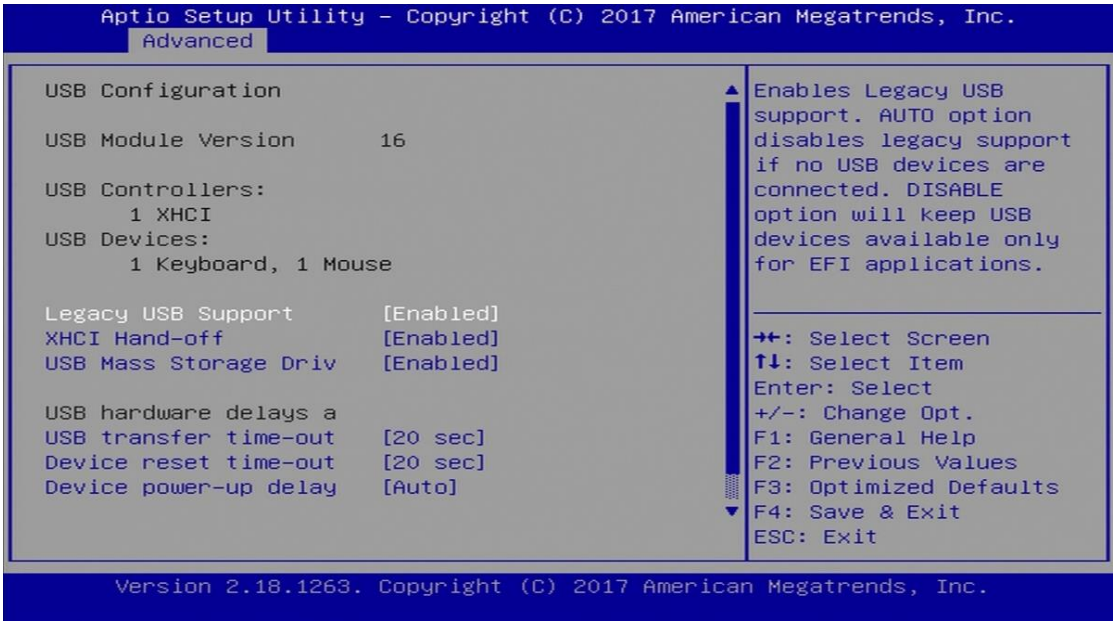
BIOS Setting	Description
Network Stack	Enables / Disables UEFI Network Stack.

3.4.9 CSM Configuration



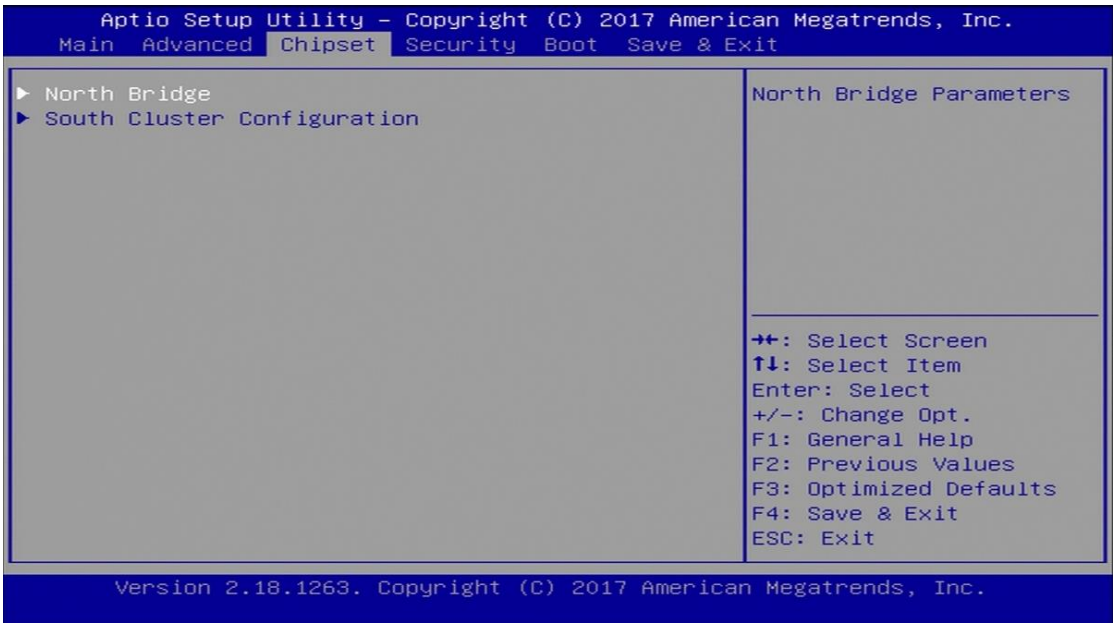
BIOS Setting	Description
CSM Support	Enables / Disables CSM support.
GateA20 Active	• Upon Request disables GA20 when using BIOS services. • Always cannot disable GA20, but is useful when any RT code is executed above 1 MB.
INT19 Trap Response	Sets how BIOS reacts on INT19 trap by Option ROM. • Immediate executes the trap right away. • Postponed executes the trap during legacy boot.
Boot option filter	Controls the priority of Legacy and UEFI ROMs. Options: UEFI and Legacy / Legacy only / UEFI only
Storage	Controls the execution of UEFI and Legacy Storage OpROM. Options: Do not lanuch / UEFI / Legacy
Video	Controls the execution of UEFI and Legacy Video OpROM. Options: Do not lanuch / UEFI / Legacy
Other PCI devices	Determines OpROM execution policy for devices other than network, storage or video. Options: Do not lanuch / UEFI / Legacy

3.4.10 USB Configuration



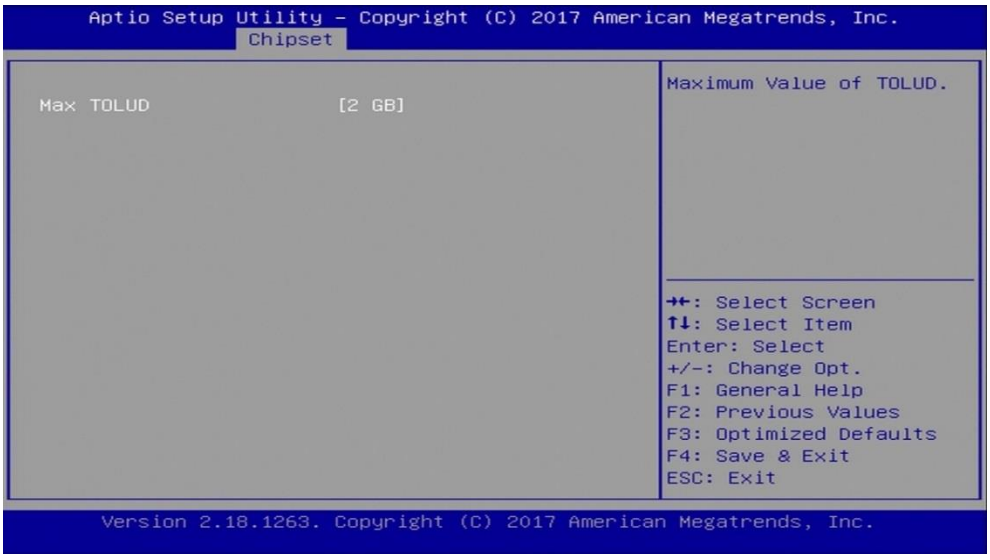
BIOS Setting	Description
Legacy USB Support	Enables / Disables Legacy USB support. • **Auto** disables legacy support if there is no USB device connected. • **Disable** keeps USB devices available only for EFI applications.
XHCI Hand-pff	This is a workaround for OSes without XHCI hand-off support. The XHCI ownership change should be claimed by XHCI driver.
USB Mass Storage Driver Support	Enables / Disables USB mass storage driver support.
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

3.5 Chipset



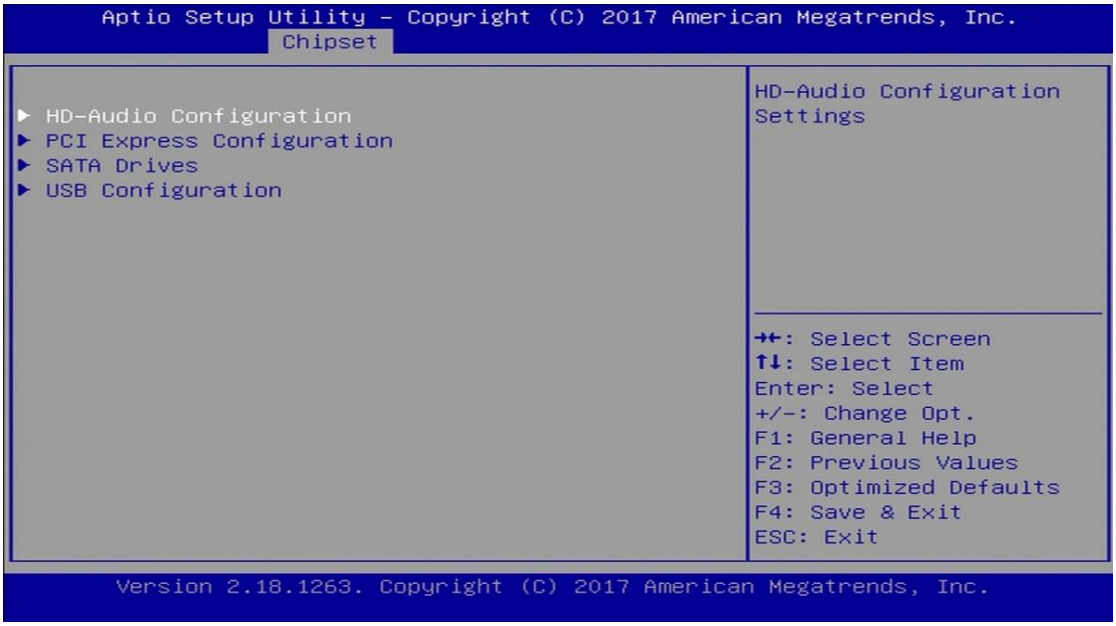
BIOS Setting	Description
North Bridge	North Bridge Parameters
South Cluster Configuration	South Cluster Configuration

3.5.1 North Bridge



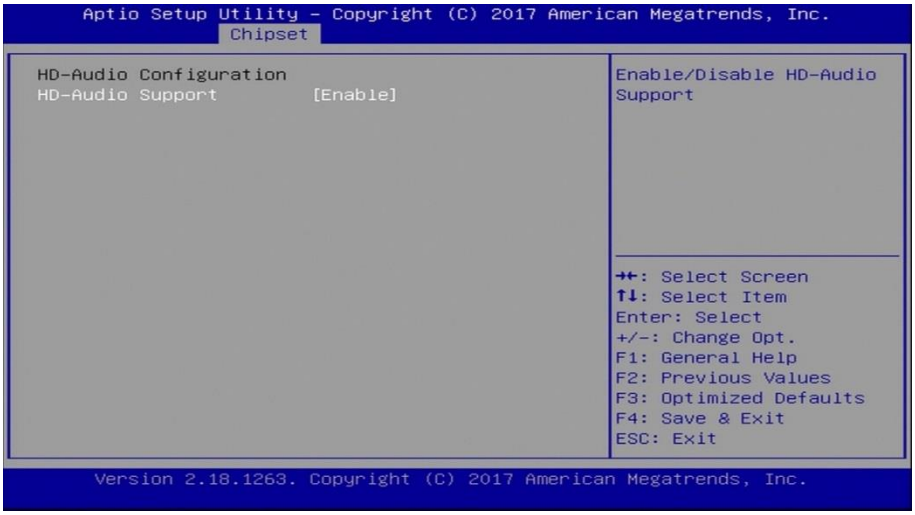
BIOS Setting	Description
Max TOLUD	Maximum Value of TOLUD. Options: 2 GB, 2.25 GB, 2.5 GB, 2.75 GB, 3 GB

3.5.2 South Cluster Configuration



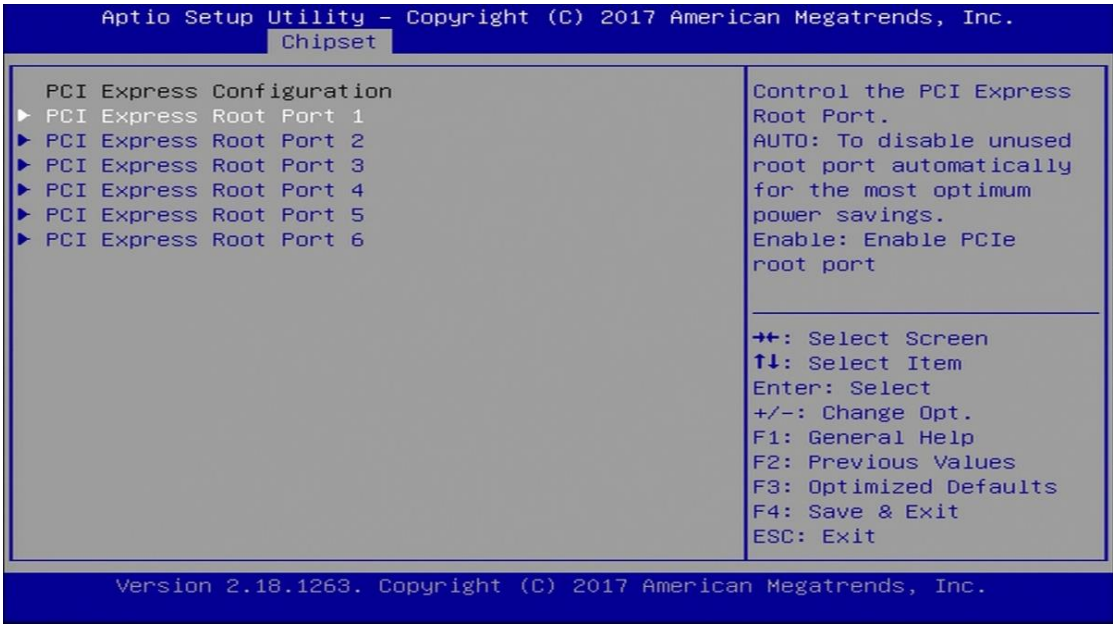
BIOS Setting	Description
HD-Audio Configuration	HD-Audio configuration settings
PCI Express Configuration	PCIe configuration settings
SATA Drivers	Press Enter to Select the SATA device configuration setup options.
USB Configuration	USB configuration settings

3.5.2.1. HD-Audio Configuration

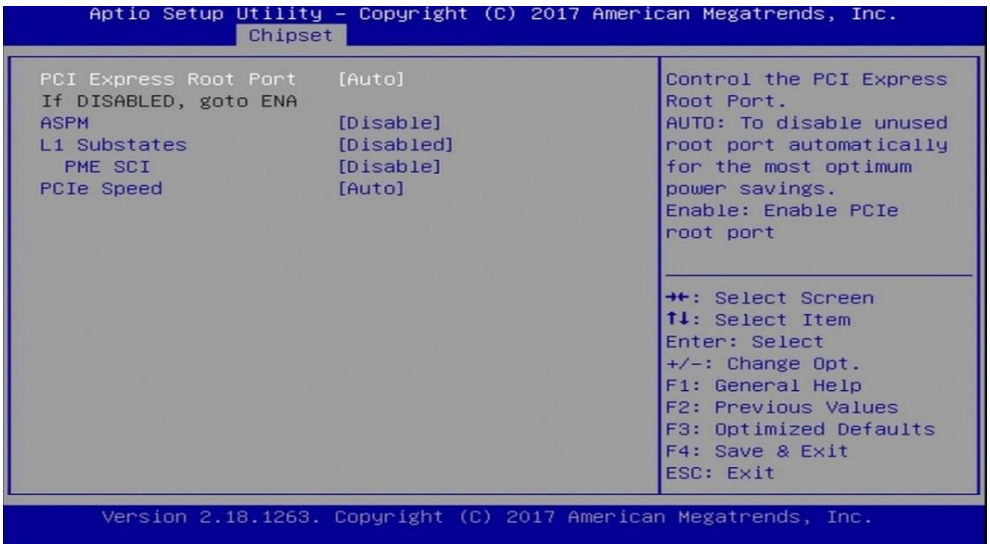


BIOS Setting	Description
HD-Audio Support	Enables / Disables HD-Audio support.

3.5.2.2. PCI Express Configuration



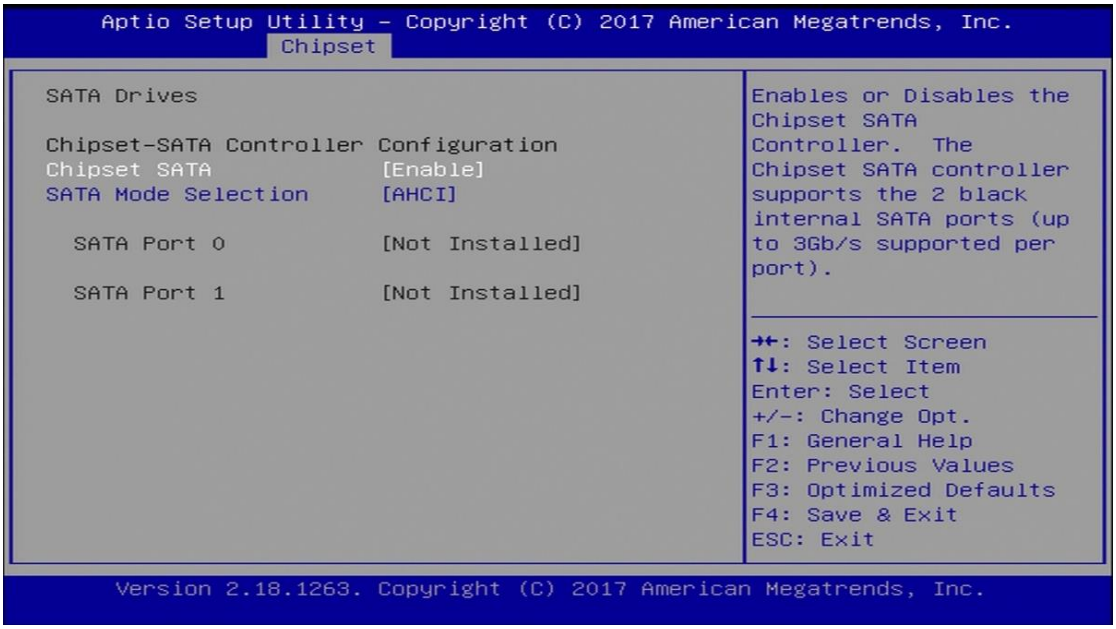
BIOS Setting	Description
PCI Express Root Ports	Controls the PCIe root port(s). “Auto” disables the unused root port automatically for the most optimumpower savings.



BIOS Setting	Description
PCI Express Root Ports	Controls the PCIe root port(s). “Auto” disables the unused root port automatically for the most optimumpower savings. Options: Enable / Disable / Auto

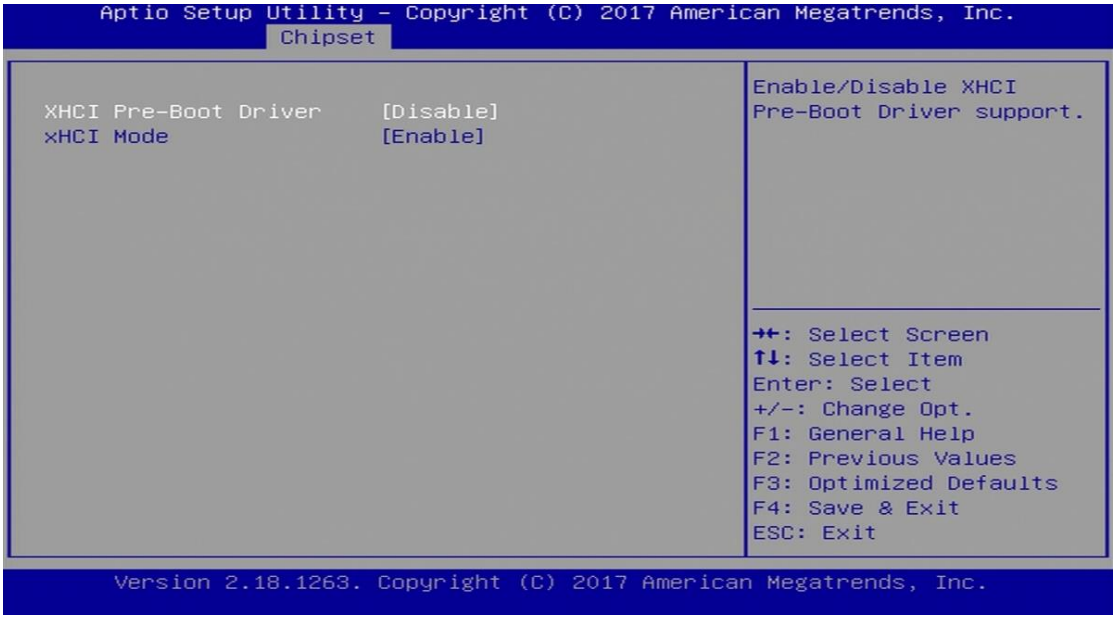
BIOS Setting	Description
ASPM	Sets up PCI Express active state power management. Options: Disable / L0s / L1 / L0sL1 / Auto
L1 Substates	Sets up PCI Express L1 Substates. Options: Disabled / L1.1 / L1.2 / L1.1 & L1.2
PME SCI	Enables / Disables PCIe PME SCI.
PCIe Speed	Configures PCIe speed. Options: Auto, Gen1, Gen2

3.5.2.3. SATA Drivers



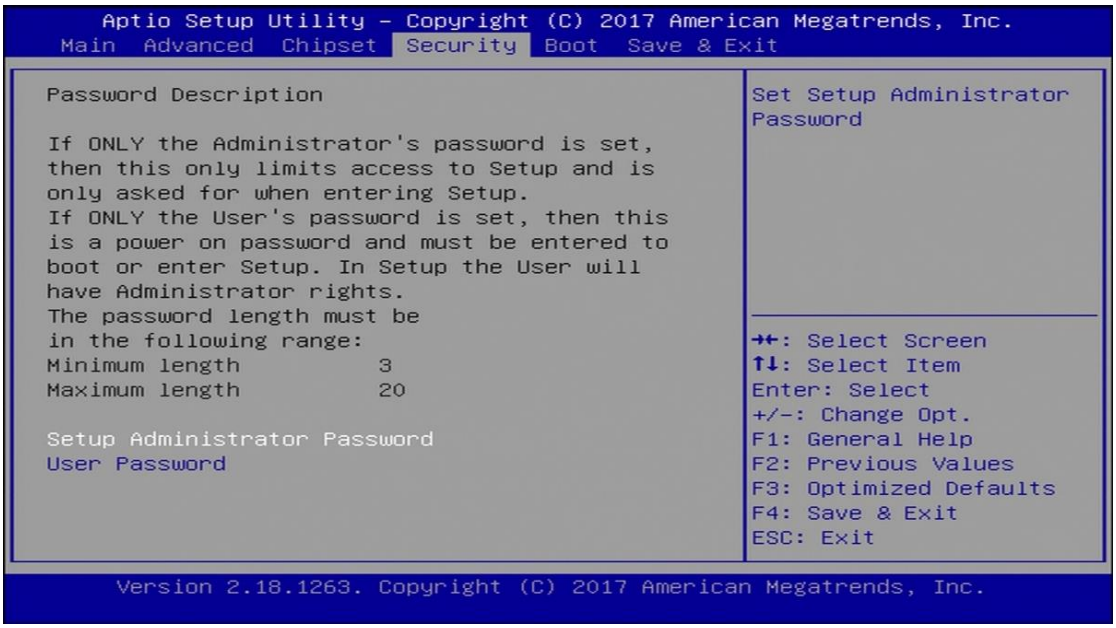
BIOS Setting	Description
Chipset SATA	Enables / Disables the Chipset SATA controller. The chipset SATA controller supports the 2 black internal SATA ports (up to 3 Gb/s supported per port).
SATA Mode Selection	Determines how SATA controller(s) operate. Options: AHCI

3.5.2.4. USB Configuration



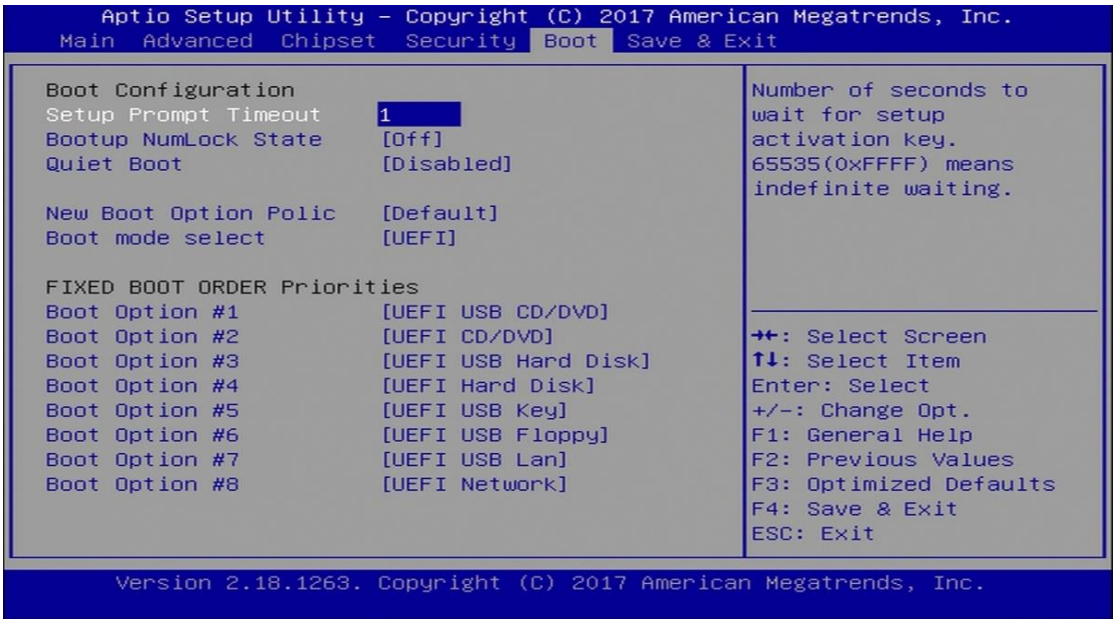
BIOS Setting	Description
HSCI Pre-Boot Driver	Enables / Disables XHCI pre-boot driver support.
XHCI Mode	Once disabled, XHCI controller would be function disabled, none of the USB devices are detectable and usable during boot and in OS. Do not disable it unless for debug.

3.6 Security Settings



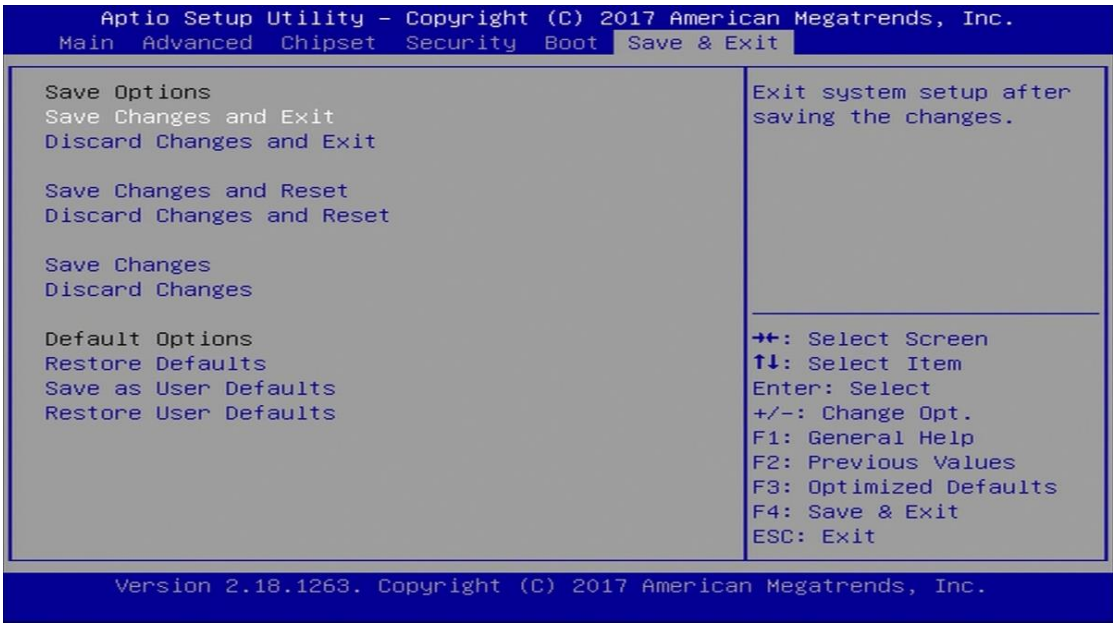
BIOS Setting	Description
Administrator Password	Sets an administrator password for the setup utility.
User Password	Sets a user password.

3.7 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.
New Boot Option Policy	Controls the placement of newly detected UEFI boot options. Options: Default, Place First, Place Last
Boot Mode Select	Selects the boot mode as Legacy or UEFI.
Boot Option Priorities	Sets the system boot order priorities for hard disk, CD/DVD, USB, Network.

3.8 Save & Exit Settings



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

Appendix

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

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

A. I/O Port Address Map

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

Address	Device Description
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
0x0000F040-0x0000F05F	Intel(R) Celeron(R)/Pentium(R) Processor SMBUS - 5AD4
0x0000E000-0x0000EFFF	Intel(R) Celeron(R)/Pentium(R) Processor PCI Express Root Port - 5AD9
0x0000D000-0x0000DFFF	Intel(R) Celeron(R)/Pentium(R) Processor PCI Express Root Port - 5ADA
0x000003F8-0x000003FF	Communications Port (COM1)
0x0000B000-0x0000BFFF	PCI-to-PCI Bridge
0x0000C000-0x0000CFFF	PCI-to-PCI Bridge

Address	Device Description
0x00009000-0x00009FFF	PCI-to-PCI Bridge
0x00009000-0x00009FFF	Intel(R) Celeron(R)/Pentium(R) Processor PCI Express Root Port - 5AD8
0x00009000-0x00009FFF	PCI-to-PCI Bridge
0x0000A000-0x0000AFFF	PCI-to-PCI Bridge
0x00000000-0x0000006F	PCI Express Root Complex
0x00000078-0x00000CF7	PCI Express Root Complex
0x00000D00-0x0000FFFF	PCI Express Root Complex
0x00000020-0x00000021	Programmable interrupt controller
0x00000024-0x00000025	Programmable interrupt controller
0x00000028-0x00000029	Programmable interrupt controller
0x0000002C-0x0000002D	Programmable interrupt controller
0x00000030-0x00000031	Programmable interrupt controller
0x00000034-0x00000035	Programmable interrupt controller
0x00000038-0x00000039	Programmable interrupt controller
0x0000003C-0x0000003D	Programmable interrupt controller
0x000000A0-0x000000A1	Programmable interrupt controller
0x000000A4-0x000000A5	Programmable interrupt controller
0x000000A8-0x000000A9	Programmable interrupt controller
0x000000AC-0x000000AD	Programmable interrupt controller
0x000000B0-0x000000B1	Programmable interrupt controller
0x000000B4-0x000000B5	Programmable interrupt controller
0x000000B8-0x000000B9	Programmable interrupt controller
0x000000BC-0x000000BD	Programmable interrupt controller
0x000004D0-0x000004D1	Programmable interrupt controller
0x0000F000-0x0000F03F	Intel(R) HD Graphics
0x0000F090-0x0000F097	Standard SATA AHCI Controller
0x0000F080-0x0000F083	Standard SATA AHCI Controller
0x0000F060-0x0000F07F	Standard SATA AHCI Controller
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 25	High Definition Audio Controller
IRQ 39	Intel SD Host Controller
IRQ 8	High precision event timer
IRQ 4294967261	Intel(R) I211 Gigabit Network Connection
IRQ 4294967260	Intel(R) I211 Gigabit Network Connection
IRQ 4294967259	Intel(R) I211 Gigabit Network Connection
IRQ 4294967258	Intel(R) I211 Gigabit Network Connection
IRQ 4294967257	Intel(R) I211 Gigabit Network Connection
IRQ 4294967256	Intel(R) I211 Gigabit Network Connection
IRQ 4	Communications Port (COM1)
IRQ 4294967255	Intel(R) USB 3.0 eXtensible Host Controller - 1.0 (Microsoft)
IRQ 4294967279	Intel(R) I211 Gigabit Network Connection #3
IRQ 4294967278	Intel(R) I211 Gigabit Network Connection #3
IRQ 4294967277	Intel(R) I211 Gigabit Network Connection #3
IRQ 4294967276	Intel(R) I211 Gigabit Network Connection #3
IRQ 4294967275	Intel(R) I211 Gigabit Network Connection #3
IRQ 4294967274	Intel(R) I211 Gigabit Network Connection #3
IRQ 4294967273	Intel(R) I211 Gigabit Network Connection #4
IRQ 4294967272	Intel(R) I211 Gigabit Network Connection #4
IRQ 4294967271	Intel(R) I211 Gigabit Network Connection #4
IRQ 4294967270	Intel(R) I211 Gigabit Network Connection #4
IRQ 4294967269	Intel(R) I211 Gigabit Network Connection #4
IRQ 4294967268	Intel(R) I211 Gigabit Network Connection #4
IRQ 54 ~ IRQ 204	Microsoft ACPI-Compliant System
IRQ 256 ~ IRQ 511	Microsoft ACPI-Compliant System
IRQ 4294967292	Intel(R) Trusted Execution Engine Interface
IRQ 4294967293	Intel(R) HD Graphics
IRQ 4294967291	Intel(R) I211 Gigabit Network Connection #5

Level	Function
IRQ 4294967290	Intel(R) I211 Gigabit Network Connection #5
IRQ 4294967289	Intel(R) I211 Gigabit Network Connection #5
IRQ 4294967288	Intel(R) I211 Gigabit Network Connection #5
IRQ 4294967287	Intel(R) I211 Gigabit Network Connection #5
IRQ 4294967286	Intel(R) I211 Gigabit Network Connection #5
IRQ 14	Intel(R) Serial IO GPIO Host Controller - INT3452
IRQ 4294967294	Standard SATA AHCI Controller
IRQ 4294967285	Intel(R) I211 Gigabit Network Connection #6
IRQ 4294967284	Intel(R) I211 Gigabit Network Connection #6
IRQ 4294967283	Intel(R) I211 Gigabit Network Connection #6
IRQ 4294967282	Intel(R) I211 Gigabit Network Connection #6
IRQ 4294967281	Intel(R) I211 Gigabit Network Connection #6
IRQ 4294967280	Intel(R) I211 Gigabit Network Connection #6
IRQ 4294967267	Intel(R) I211 Gigabit Network Connection #2
IRQ 4294967266	Intel(R) I211 Gigabit Network Connection #2
IRQ 4294967265	Intel(R) I211 Gigabit Network Connection #2
IRQ 4294967264	Intel(R) I211 Gigabit Network Connection #2
IRQ 4294967263	Intel(R) I211 Gigabit Network Connection #2
IRQ 4294967262	Intel(R) I211 Gigabit Network Connection #2
IRQ 0	System timer

C. Digital I/O Sample Code

1. DIO Sample Code: The file F81964.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 "F81964.H"
#include <dos.h>
//=============================================================================
=====
unsigned int F81964_BASE;
void Unlock_F81964 (void);
void Lock_F81964 (void);
//=============================================================================
=====
unsigned int Init_F81964(void)
{
        unsigned int result;    //
        unsigned char ucDid;

        F81964_BASE = 0x4E;
        result = F81964_BASE;

        ucDid = Get_F81964_Reg(0x20);
        if ((ucDid == 0x07) || (ucDid == 0x10) || (ucDid == 0x15)) //Fintek
81865/81846/81866/81946/81964
        {       goto Init_Finish;}

        F81964_BASE = 0x2E;
        result = F81964_BASE;

        ucDid = Get_F81964_Reg(0x20);
        if ((ucDid == 0x07) || (ucDid == 0x10) || (ucDid == 0x15))       //Fintek
81865/81846/81866/81946/81964
        {       goto Init_Finish;}

        F81964_BASE = 0x00;
        result = F81964_BASE;

Init_Finish:
        return (result);
}
//=============================================================================
=====
```

```
void Unlock_F81964 (void)
{
	outportb(F81964_INDEX_PORT, F81964_UNLOCK);
	outportb(F81964_INDEX_PORT, F81964_UNLOCK);
}
//============================================================================
=====
void Lock_F81964 (void)
{
	outportb(F81964_INDEX_PORT, F81964_LOCK);
}
//============================================================================
=====
void Set_F81964_LD( unsigned char LD)
{
	Unlock_F81964();
	outportb(F81964_INDEX_PORT, F81964_REG_LD);
	outportb(F81964_DATA_PORT, LD);
	Lock_F81964();
}
//============================================================================
=====
void Set_F81964_Reg( unsigned char REG, unsigned char DATA)
{
	Unlock_F81964();
	outportb(F81964_INDEX_PORT, REG);
	outportb(F81964_DATA_PORT, DATA);
	Lock_F81964();
}
//============================================================================
=====
unsigned char Get_F81964_Reg(unsigned char REG)
{
	unsigned char Result;
	Unlock_F81964();
	outportb(F81964_INDEX_PORT, REG);
	Result = inportb(F81964_DATA_PORT);
	Lock_F81964();
	return Result;
}
//============================================================================
=====
```

2. DIO Sample Code: The file F81964.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 __F81964_H
#define __F81964_H              1
//=============================================================================
#define     F81964_INDEX_PORT           (F81964_BASE)
#define     F81964_DATA_PORT            (F81964_BASE+1)
//=============================================================================
#define     F81964_REG_LD               0x07
//=============================================================================
#define F81964_UNLOCK               0x87
#define     F81964_LOCK                 0xAA
//=============================================================================
unsigned int Init_F81964(void);
void Set_F81964_LD( unsigned char);
void Set_F81964_Reg( unsigned char, unsigned char);
unsigned char Get_F81964_Reg( unsigned char);
//=============================================================================
#endif      //__F81964_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 "F81964.H"
//---------------------------------------------------------------------------
void ClrKbBuf(void);
int main (int argc, char *argv[]);
//---------------------------------------------------------------------------
int main (int argc, char *argv[])
{
        unsigned char result;
        char SIO;

        SIO = Init_F81964();
        if (SIO == 0)
        {
                printf("Can not detect Fintek F81964, program abort.\n");
                return(1);
        }

           Set_F81964_LD(0x06);
//switch to logic device 6

           result = ((Get_F81964_Reg(0xE2)) & 0x04) ? 0x01 : 0x00;          //result = 0x00 GPI is
Low / result = 0x01 GPI is High

        return (result);
}
```

D. Watchdog Timer Configuration

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

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

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 "F81964.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 81964 watch dog program\n");
	SIO = Init_F81964();
	if (SIO == 0)
	{
		printf("Can not detect Fintek 81964, 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_F81964_Reg(0x2B);
        bBuf &= (~0x20);
        Set_F81964_Reg(0x2B, bBuf);             //Enable WDTO

        Set_F81964_LD(0x07);                    //switch to logic device 7
        Set_F81964_Reg(0x30, 0x01);             //enable timer

        bBuf = Get_F81964_Reg(0xF5);
        bBuf &= (~0x0F);
        bBuf |= 0x52;
        Set_F81964_Reg(0xF5, bBuf);             //count mode is second
        Set_F81964_Reg(0xF6, interval);         //set timer
        bBuf = Get_F81964_Reg(0xFA);
        bBuf |= 0x01;
        Set_F81964_Reg(0xFA, bBuf);             //enable WDTO output

        bBuf = Get_F81964_Reg(0xF5);
        bBuf |= 0x20;
        Set_F81964_Reg(0xF5, bBuf);             //start counting
}
//-----------------------------------------------------------------------
void DisableWDT(void)
{
        unsigned char bBuf;
        Set_F81964_LD(0x07);                    //switch to logic device 7
        bBuf = Get_F81964_Reg(0xFA);
        bBuf &= ~0x01;
        Set_F81964_Reg(0xFA, bBuf);             //disable WDTO output

        bBuf = Get_F81964_Reg(0xF5);
        bBuf &= ~0x20;
        bBuf |= 0x40;
        Set_F81964_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 "F81964.H"
#include <dos.h>
//---------------------------------------------------------------------------
unsigned int F81964_BASE; void Unlock_F81964 (void); void Lock_F81964 (void);
//---------------------------------------------------------------------------
unsigned int Init_F81964(void)
{
	unsigned int result;
	unsigned char ucDid;

	F81964_BASE = 0x4E;
	result = F81964_BASE;

	ucDid = Get_F81964_Reg(0x20);
	if (ucDid == 0x07)			//Fintek 81964
	{	goto Init_Finish;}

	F81964_BASE = 0x2E;
	result = F81964_BASE;

	ucDid = Get_F81964_Reg(0x20);
	if (ucDid == 0x07)			//Fintek 81964
	{	goto Init_Finish;}

	F81964_BASE = 0x00;
	result = F81964_BASE;

Init_Finish:
	return (result);
}
//---------------------------------------------------------------------------
void Unlock_F81964 (void)
{
	outportb(F81964_INDEX_PORT, F81964_UNLOCK);
	outportb(F81964_INDEX_PORT, F81964_UNLOCK);
}
//---------------------------------------------------------------------------
void Lock_F81964 (void)
{
	outportb(F81964_INDEX_PORT, F81964_LOCK);
}
//---------------------------------------------------------------------------
void Set_F81964_LD( unsigned char LD)
{
	Unlock_F81964();
```

```
	outportb(F81964_INDEX_PORT, F81964_REG_LD);
	outportb(F81964_DATA_PORT, LD); Lock_F81964();
}
//-----------------------------------------------------------------------------
void Set_F81964_Reg( unsigned char REG, unsigned char DATA)
{
	Unlock_F81964();
	outportb(F81964_INDEX_PORT, REG);
	outportb(F81964_DATA_PORT, DATA);
	Lock_F81964();
}
//-----------------------------------------------------------------------------
unsigned char Get_F81964_Reg(unsigned char REG)
{
	unsigned char Result;
	Unlock_F81964();
	outportb(F81964_INDEX_PORT, REG);
	Result = inportb(F81964_DATA_PORT);
	Lock_F81964();
	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	F81964_H
#define	F81964_H	1
//-----------------------------------------------------------------------------
#define	F81964_INDEX_PORT	(F81964_BASE)
#define	F81964_DATA_PORT	(F81964_BASE+1)
//-----------------------------------------------------------------------------
#define	F81964_REG_LD	0x07
//-----------------------------------------------------------------------------
#define F81964_UNLOCK 0x87
#define	F81964_LOCK	0xAA
//-----------------------------------------------------------------------------
unsigned int Init_F81964(void);
void Set_F81964_LD( unsigned char);
void Set_F81964_Reg( unsigned char, unsigned char); unsigned char
Get_F81964_Reg( unsigned char);
//-----------------------------------------------------------------------------
#endif //	F81964_H
```