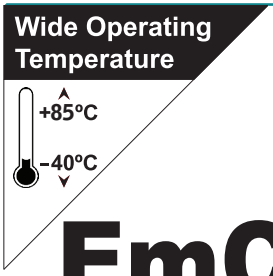


EmCORE-i89M2

3.5” Compact Board

User’s Manual

Version 1.4

2021.02

Revision History

Version	Date	Description
1.0	2018.01	Initial release
1.1	2019.03	1.1. Packing List: Remove driver CD 1.3. Driver Installation: Revise installation instructions 1.4. Specifications: Revise specification descriptions of CPU, Ethernet controller and graphic interface 2.1. Block Diagram: Revise PWR1 description 2.2. Jumpers & Connectors Quick Reference / 2.3. Jumpers & Connectors Location: Revise COM port 1-6/ LAN1&2 pin assignment descriptions Revise DP & HDMI port drawings
1.2	2019.09	Both dual-channel 24-bit LVDS become standard. Jumpers JVLCD2 / JINV2 and connectors LVDS2 / INV2 become standard. Revised sections include: 1.4. Specifications 2.2. Jumpers & Connectors Quick Reference 2.3. Jumpers & Connectors Location
1.3	2020.04	1.4. Specifications: “OEM request” added to “SIM socket”
1.4	2021.02	Remove SIM card socket and revise Realtek ALC 662 to ALC 886 1.4. Specifications 2.1. Block Diagram 2.2. Jumpers & Connectors Quick Reference 2.3. Jumpers & Connectors Location

Contents

Preface

Copyright Notice

The information in this document is subject to change without prior notice in order to improve the reliability, design and function. It does not represent a commitment on the part of the manufacturer.

Under no circumstances will the manufacturer be liable for any direct, indirect, special, incidental, or consequential damages arising from the use or inability to use the product or documentation, even if advised of the possibility of such damages.

Declaration of Conformity

CE

The CE symbol on your product indicates that it is in compliance with the directives of the Union European (EU). A Certificate of Compliance is available by contacting Technical Support.

This product has passed the CE test for environmental specifications when shielded cables are used for external wiring. We recommend the use of shielded cables. This kind of cable is available from ARBOR. Please contact your local supplier for ordering information.

This product has passed the CE test for environmental specifications. Test conditions for passing included the equipment being operated within an industrial enclosure. In order to protect the product from being damaged by ESD (Electrostatic Discharge) and EMI leakage, we strongly recommend the use of CE-compliant industrial enclosure products.

Warning

This is a class A product. In a domestic environment this product may cause radio interference in which case the user may be required to take adequate measures.

FCC Class A

This device complies with Part 15 of the FCC Rules. Operation is subject to the following two conditions:

(1)This device may not cause harmful interference, and

(2)This device must accept any interference received, including interference that may cause undesired operation.

NOTE:

This equipment has been tested and found to comply with the limits for a Class A digital device, pursuant to Part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference when the equipment is operated in a commercial environment. This equipment generates, uses, and can radiate radio frequency energy and, if not installed and used in accordance with the instruction manual, may cause harmful interference to radio communications. Operation of this equipment in a residential area is likely to cause harmful interference in which case the user will be required to correct the interference at his own expense.

RoHS

ARBOR Technology Corp. certifies that all components in its products are in compliance and conform to the European Union's Restriction of Use of Hazardous Substances in Electrical and Electronic Equipment (RoHS) Directive 2002/95/EC.

The above mentioned directive was published on 2/13/2003. The main purpose of the directive is to prohibit the use of lead, mercury, cadmium, hexavalent chromium, polybrominated biphenyls (PBB), and polybrominated diphenyl ethers (PBDE) in electrical and electronic products. Member states of the EU are to enforce by 7/1/2006.

ARBOR Technology Corp. hereby states that the listed products do not contain unintentional additions of lead, mercury, hex chrome, PBB or PBDB that exceed a maximum concentration value of 0.1% by weight or for cadmium exceed 0.01% by weight, per homogenous material. Homogenous material is defined as a substance or mixture of substances with uniform composition (such as solders, resins, plating, etc.). Lead-free solder is used for all terminations (Sn(96-96.5%), Ag(3.0-3.5%) and Cu(0.5%)).

SVHC / REACH

To minimize the environmental impact and take more responsibility to the earth we live, Arbor hereby confirms all products comply with the restriction

of SVHC (Substances of Very High Concern) in (EC) 1907/2006 (REACH --Registration, Evaluation, Authorization, and Restriction of Chemicals) regulated by the European Union.

All substances listed in SVHC < 0.1 % by weight (1000 ppm)

About This User's Manual

This user's manual provides general information and installation instructions about the product. This User's Manual is intended for experienced users and integrators with hardware knowledge of personal computers. If you are not sure about any description in this booklet. Please consult your vendor before further handling.

Warning

Single Board Computers and their components contain very delicate Integrated Circuits (IC). To protect the Single Board Computer and its components against damage from static electricity, you should always follow the following precautions when handling it :

1. Disconnect your Single Board Computer from the power source when you want to work on the inside.
2. Hold the board by the edges and try not to touch the IC chips, leads or circuitry.
3. Use a grounded wrist strap when handling computer components.
4. Place components on a grounded antistatic pad or on the bag that comes with the Single Board Computer, whenever components are separated from the system.

Replacing the Lithium Battery

Incorrect replacement of the lithium battery may lead to a risk of explosion.

The lithium battery must be replaced with an identical battery or a battery type recommended by the manufacturer.

Do not throw lithium batteries into the trash-can. It must be disposed of in accordance with local regulations concerning special waste.

Technical Support

If you have any technical difficulties, please do not hesitate to call or e-mail our customer service.

https://www.arbor-technology.com

E-mail:info@arbor.com.tw

Warranty

This product is warranted to be in good working order for a period of two years from the date of purchase. Should this product fail to be in good working order at any time during this period, we will, at our option, replace or repair it at no additional charge except as set forth in the following terms. This warranty does not apply to products damaged by misuse, modifications, accident or disaster.

Vendor assumes no liability for any damages, lost profits, lost savings or any other incidental or consequential damage resulting from the use, misuse of, or inability to use this product. Vendor will not be liable for any claim made by any other related party.

Vendors disclaim all other warranties, either expressed or implied, including but not limited to implied warranties of merchantability and fitness for a particular purpose, with respect to the hardware, the accompanying product's manual(s) and written materials, and any accompanying hardware. This limited warranty gives you specific legal rights.

Return authorization must be obtained from the vendor before returned merchandise will be accepted. Authorization can be obtained by calling or faxing the vendor and requesting a Return Merchandise Authorization (RMA) number. Returned goods should always be accompanied by a clear problem description.

Chapter 1

Introduction

1.1. Packing List

Before starting with the installation, make sure the following items are shipped. If any item appears damaged or is missing, contact your vendor immediately:

1 x EmCORE-i89M2 3.5” Compact Board with Cooler

1 x Quick Installation Guide

1.2. Ordering Information

EmCORE-i89M2-6442EQ	6th Generation Intel® Core™ i5-6442EQ / QM170 3.5” compact board
EmCORE-i89M2-6822EQ	6th Generation Inte® Core™ i7-6822EQ / QM170 3.5” compact board

Optional Accessories

CBK-15-89M2-00	Cable kit 1 x Audio cable 2 x SATA cables 1 x SATA power cable 2 x USB 2.0 cables 1 x USB 3.0 cable 2 x LAN cables 6 x COM cables

1.3. Driver (6.6A) Installation

To install the drivers, please visit our website at **www.arbor-technology.com** and download the driver pack from the product page.

The driver path is listed as below:

Windows 7 & Windows 10 64-bit

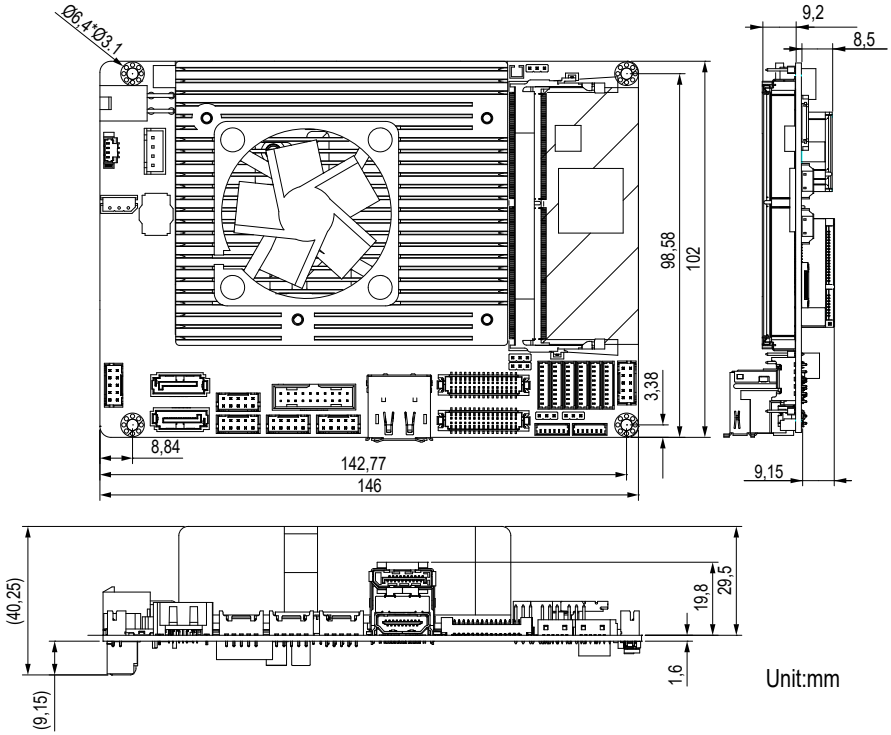
Chipset	\i89X\Chipset\Chipset_10.1.1.13_Public
Audio	\i89X\Audio\7687_PG436_Win10_Win8.1_Win8_Win7_WHQLx64
LAN	\i89X\Ethernet
Graphic	\i89X\Graphic\IntelR Graphics Driver Production Version 15.40.16.64.4364
ME	\i89X\ME\Intel(R)_ME_11.0_Corporate_11.0.0.1202
RAID	\i89X\RAID\Intel Rapid Storage Technology Driver 14.8.0.1042
USB3.0	\i89X\USB3.0\win8.1 64bit\Intel_USB_3.0_xHC_Adaptation_ Driver_MR1_Release_1.0.1.45_PV

1.4. Specifications

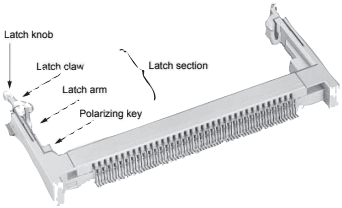
Form Factor	3.5" Compact Board
CPU	Soldered onboard 6th Generation Intel® Quad Core™ Processor i5-6442EQ 1.9GHz (base)/2.7GHz (Turbo); i7-6822EQ 2.0GHz 2.0GHz (Base)/2.8GHz(Turbo)
Chipset	Intel® PCH QM170
Memory	1 x DDR4 SO-DIMM socket, supporting 2133/1866MHz SDRAM up to 16GB
BIOS	AMI UEFI BIOS
Watchdog Timer	1~255 levels reset
Super I/O	Fintek F81866
USB Port	2 x USB 3.0/2.0 ports 4 x USB 2.0 ports
Serial Port	6 x COM ports - 4 x RS-232 - 2 x RS-232/422/485 selectable
Expansion	1 x Mini-Card Socket
Storage	2 x Serial ATA ports with 600MB/s HDD transfer rate
	1 x NGFF M.2 socket for M-Key to support SATA/ PCIe x4 depending on SSD module
Ethernet Chipset	1 x Intel® i219LM PCIe PHY 1 x Intel® i210IT PCIe GbE controller
Digital I/O	8-bit Programmable
Audio	Realtek® ALC886 HD Audio CODEC, Mic-in/ Line-in/ Line-out
Graphic Chipset	Integrated Intel® HD Graphics 5x0
Graphic Interface	2 x Dual Channel 24-bit LVDS
	1 x HDMI
	1 x DisplayPort
OS Support	
Windows 7, Windows 10 64-bit, Linux Ubuntu	
Power Requirement	+12V DC-In

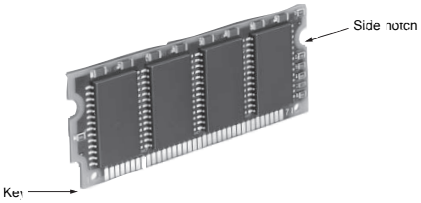
Power Consumption	1.77A @+12V with i5-6442EQ (Typical) 1.82A @+12V with i7-6822EQ (Typical)
Operating Temp.	-40 ~ 85°C (-40 ~ 185ºF)
Operating Humidity	10 ~ 95% @ 85°C (non-condensing)
Dimensions (L x W)	146 x 102 mm (5.7" x 4.0")

1.5. Board Dimensions



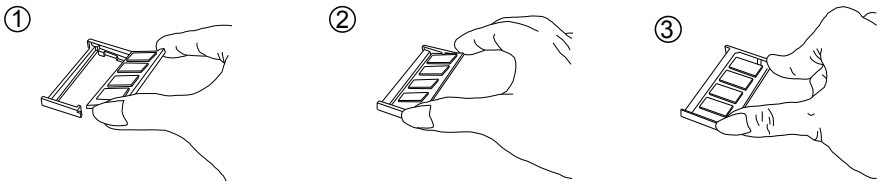
1.6 Installing the Memory





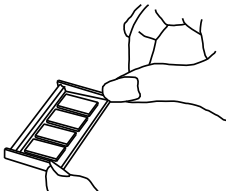
To install the Memory module, locate the Memory SO-DIMM slot on the board and perform as below:

1. Adjust the socket polarizing key and the board key to the same direction.
2. Insert the board obliquely. Moreover, lay the board in parallel to the opening at angle of 20^{o} to 30^{o}, and softly insert the board so as to hit the socket bottom. Stopping insertion halfway will result in improper insertion.
3. Applying the board side notch in parallel to the socket bottom so that the board position cannot be displaced, press the board side notch up, and fix it to the latch portion at both socket edges. Press the board side notch, and release the notch with a snap “click” tone, if the printed board exceeds the latch claw head.



Procedures for board extraction

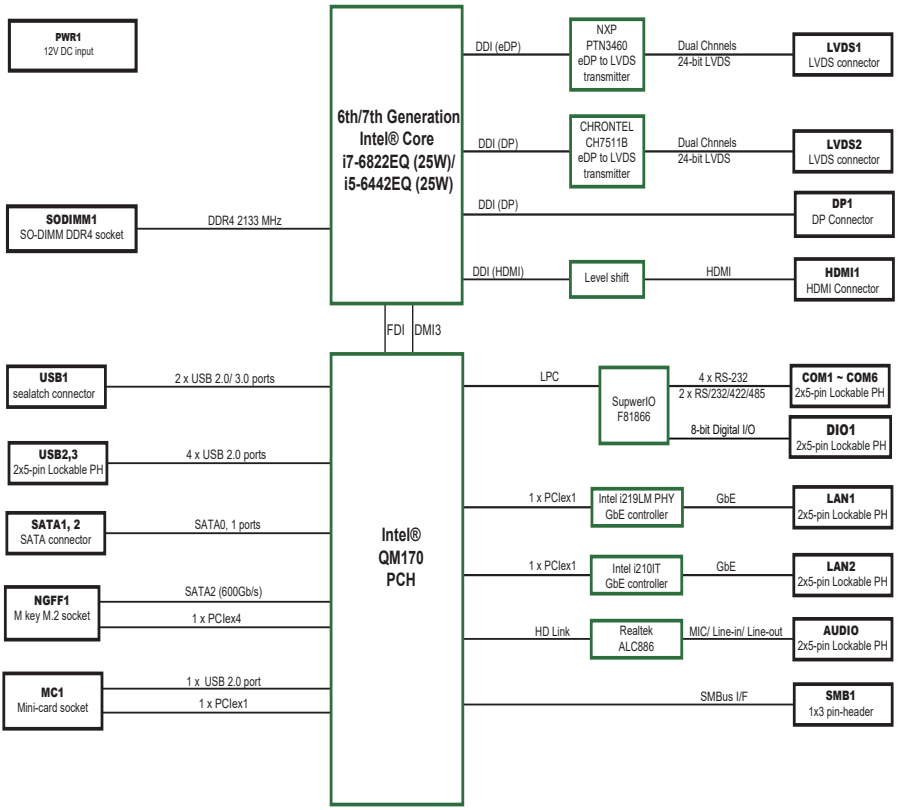
Apply the thumb nail to the latch knob at both socket edges. Forcibly widen the latch knobs to right and left ways, and release the latch. Then draw the board out along an angle where the board is raised.



This page is intentionally left blank.

Chapter 2
Installation

2.1. Block Diagram



2.2. Jumpers & Connectors Quick Reference

Jumpers

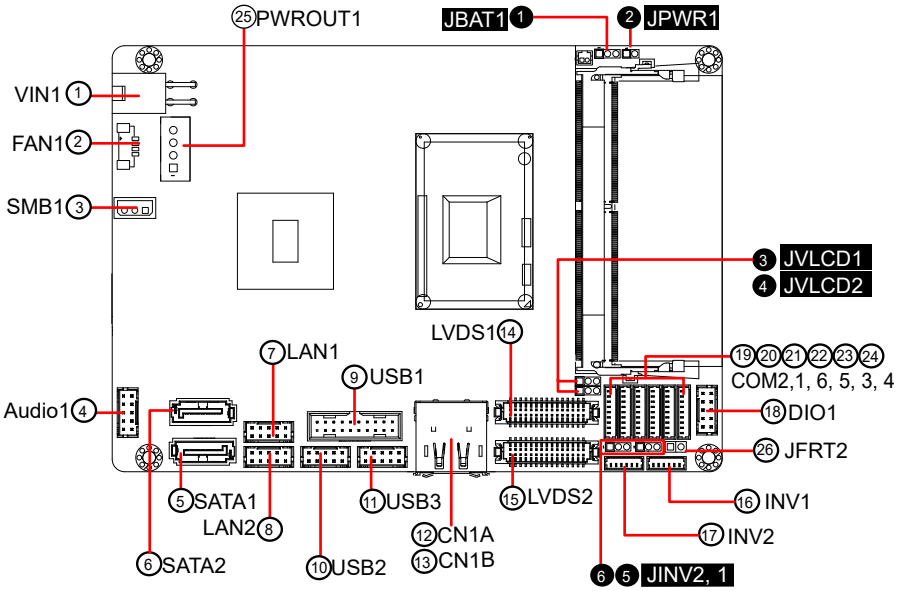
Jumper	Description
❶JBAT1	Clears/keeps CMOS
❷JPWR1	Sets the AT/ATX mode
❸❹JVLCD1, 2	Sets the power voltage for LVDS1, 2
❺❻JINV1, 2	Sets the inverter voltage for LVDS1, 2

Connectors

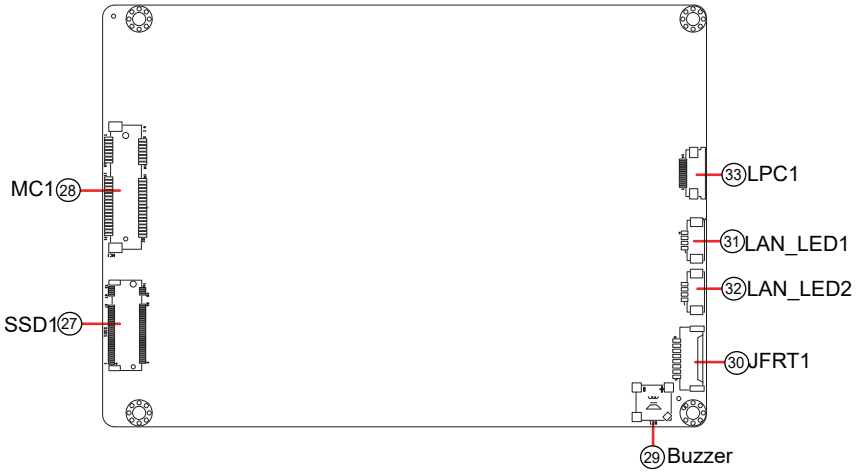
Connector	Description
①VIN1	12V DC IN Connector
②FAN1	CPU fan connector
③SMB1	SMBus Connector
④Audio1	Audio connector
⑤⑥SATA1, 2	Serial ATA Connectors
⑦⑧LAN1, 2	Ethernet connectors
⑨USB1	USB 3.0 connector
⑩⑪USB2, 3	USB 2.0 connectors
⑫CN1A	DisplayPort Connector
⑬CN1B	HDMI connector
⑭⑮LVDS1, 2	LVDS1, 2 LCD panel connectors
⑯⑰INV1, 2	LCD1, 2 inverter connectors
⑱DIO1	Digital I/O connector
⑲~㉔ COM1~6	Serial port connectors COM1-2: RS-232/422/485 selectable, COM3-6: RS-232
㉕PWROUT1	SATA power connector
㉖JFRT2	Power Button
㉗SSD1	NGFF M.2 M-Key socket
㉘MC1	Mini-card socket
㉙BUZZER	Buzzer
㉚JFRT1	Front-panel connector
㉛㉜LAN_LED1, 2	LAN1, 2 LEDs
㉝LPC1	Low pin count connector

2.3. Jumpers & Connectors Location

Board Top



Board Bottom

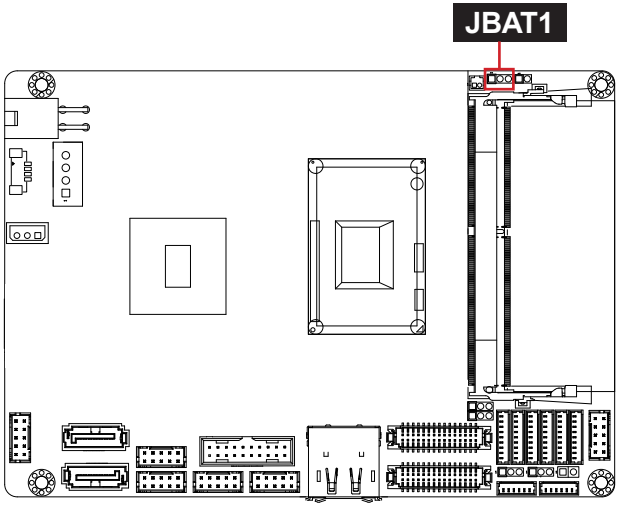


2.3.1. Jumpers

❶ JBAT1

Function: Clears/keeps CMOS
Jumper type: 2.00 mm pitch 1x3-pin header

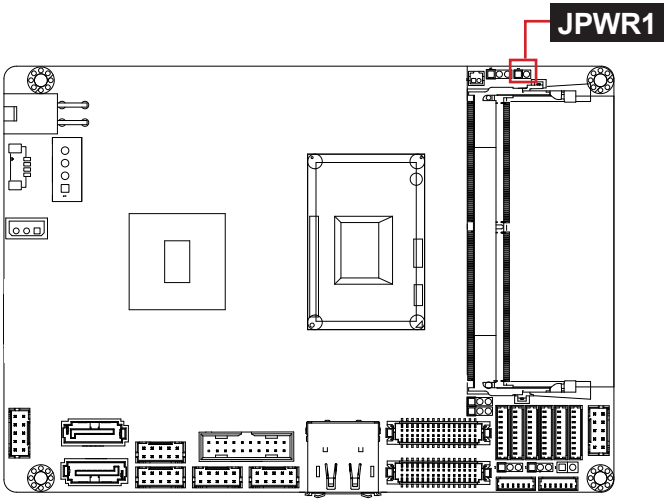
Pin	Description	
1-2	Keeps CMOS (default)	
2-3	Clears CMOS	



❷ JPWR1

Function: Sets the AT/ATX mode
Jumper type: 2.00mm pitch 2x3-pin header

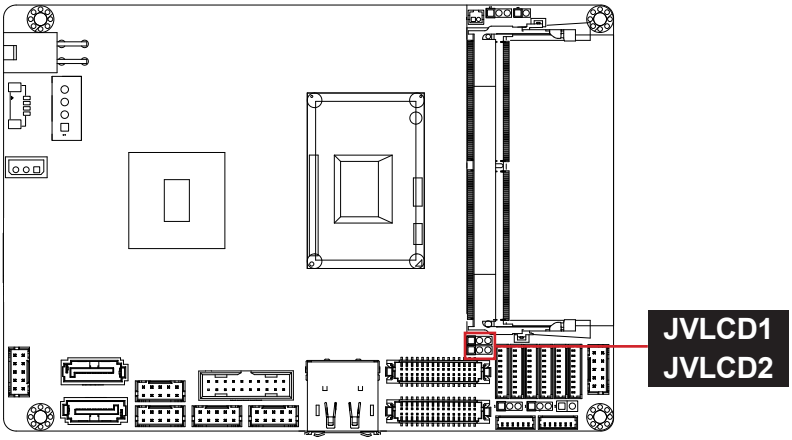
Pin	Description	
Short	ATX (default)	
Open	AT	



❸❹ JVLCD1, 2

Function: Sets the power voltage for LVDS1, 2
Jumper type: 2.00mm pitch 1x3-pin header

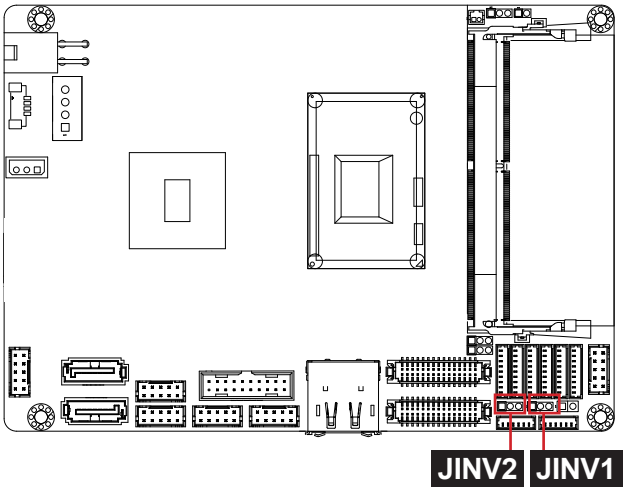
Pin	Description	
1-2	+5V	
2-3	+3.3V (default)	



❺❻ JINV1, 2

Function: Sets the inverter voltage for LVDS1, 2
Jumper type: 2.00mm pitch 1x3-pin header

Pin	Description	
1-2	+12V (default)	
2-3	+5V	

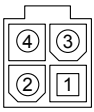


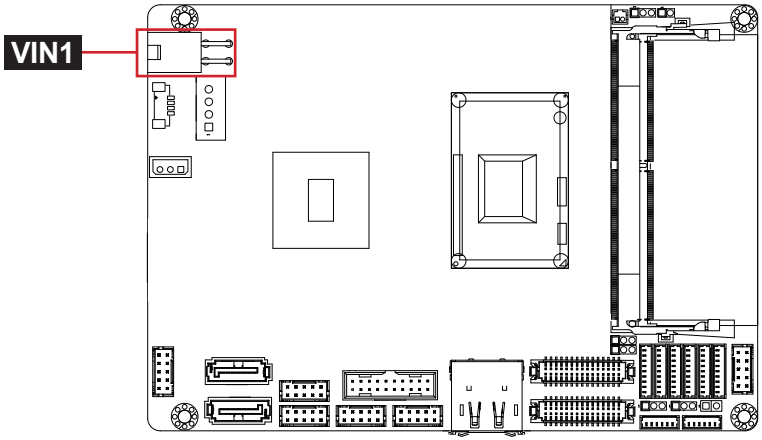
2.3.2. Connectors

①VIN1

Function: 12V DC IN Connector
Connector Type: 4-pin power connector

Pin	Desc.	Pin	Desc.
4	+12V	3	+12V
2	GND	1	GND

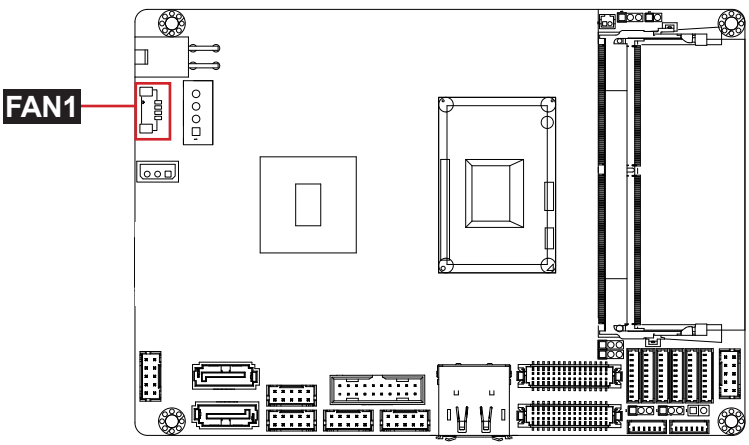




② FAN1

Function: CPU Fan Connector
Connector type: 2.54mm pitch 1x4-pin wafer connector.

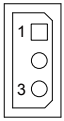
Pin	Description
1	GND
2	+12V
3	RPM
4	CTRL

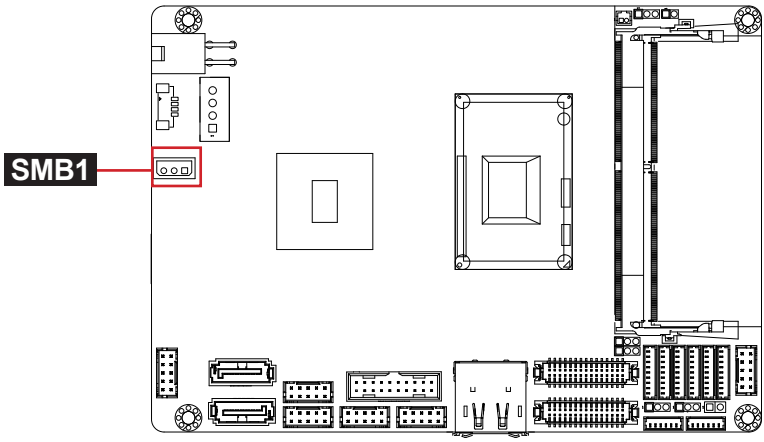


③ SMB1

Function: SMBus Connector
Connector type: 2.54mm pitch 1x3-pin header

Pin	Description
1	SM_DATA
2	SM_CLK
3	GND

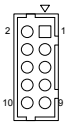


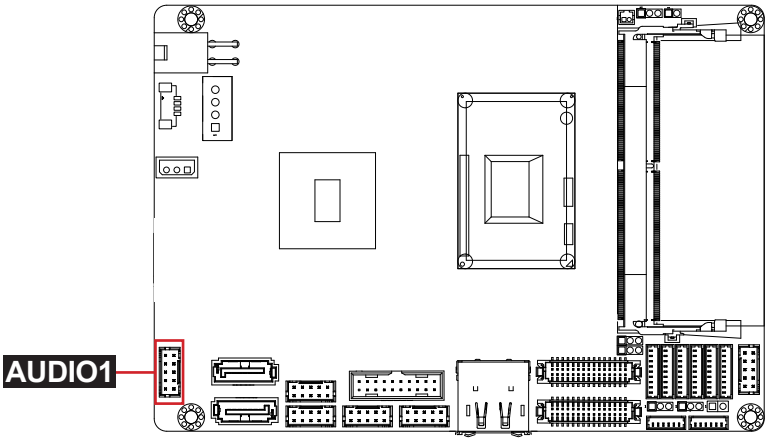


④ AUDIO1

Function: Audio connector
Connector type: 2.00mm pitch 2x5-pin box wafer connector

Pin	Description	Pin	Description
2	LINE_R	1	LINE_L
4	GND3	3	GND1
6	NC/MIC2	5	MIC1
8	GND4	7	GND2
10	LOUT_R	9	LOUT_L

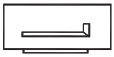


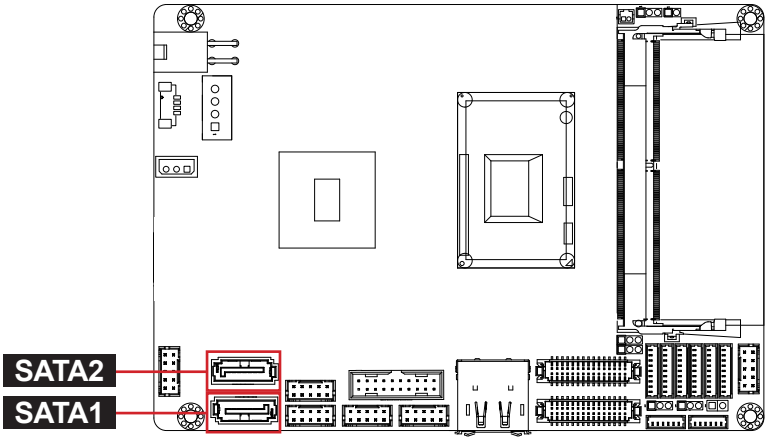


⑤⑥ SATA1, 2

Function: Serial ATA connector
Connector type: SATA connector

The pin assignments conform to the industry standard.

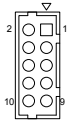


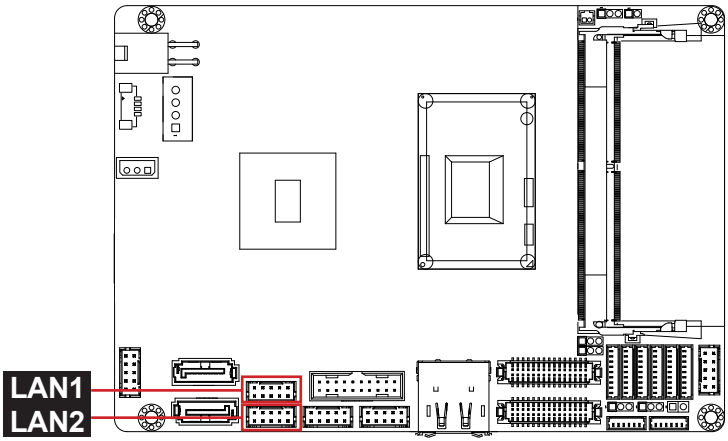


⑦⑧ LAN1, 2

Function: Ethernet connectors
Connector type: 2.00mm pitch 2x5-pin wafer connector that supports 10/100/1000Mbps fast Ethernet

Pin	Description	Pin	Description
2	MDI0-	1	MDI0+
4	MDI2+	3	MDI1+
6	MDI1-	5	MDI2-
8	MDI3-	7	MDI3+
9	N/C	10	N/C

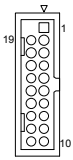


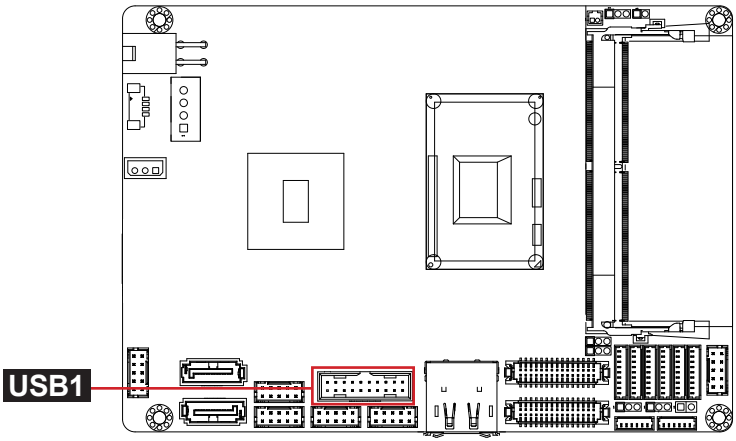


⑨USB1

Function: USB 3.0 connector
Connector type: 2.00mm pitch 2x10-pin box header

Pin	Description	Pin	Description
		1	+V5S
19	+V5S	2	USB3_RXN1
18	USB3_RXN2	3	USB3_RXP1
17	USB3_RXP2	4	GND
16	GND	5	USB3_TXN1
15	USB3_TXN2	6	USB3_TXP1
14	USB3_TXP2	7	GND
13	GND	8	USBP1N
12	USBP2N	9	USBP1P
11	USBP2P	10	N/C

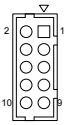


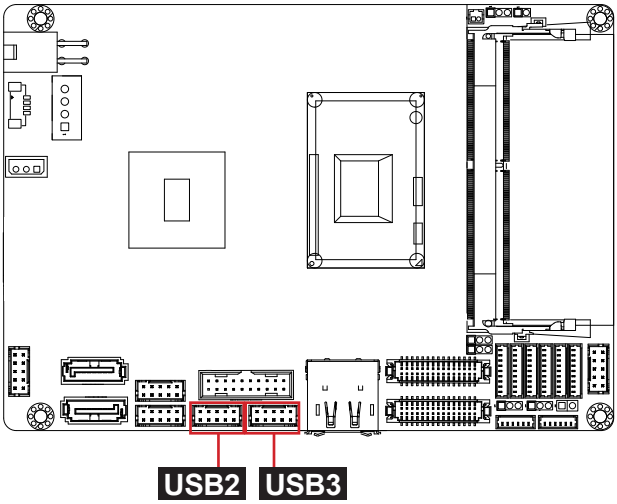


⑩⑪ USB2, 3

Function: USB 2.0 connector
Connector type: 2.00mm pitch 2x5-pin wafer connector

Pin	Description	Pin	Description
2	+5VS	1	+5VS
4	USBP4N	3	USBP3N
6	USBP4P	5	USBP3P
8	GND	7	GND
10	GND	9	GND

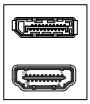




⑫CN1A

Function: DisplayPort Connector
Connect the display device to the DisplayPort Connector

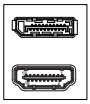
The pin assignments conform to the industry standard.

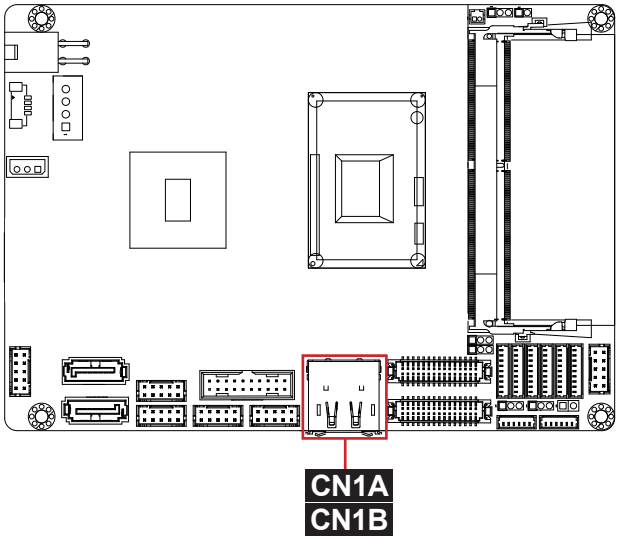


⑬ CN1B

Function: HDMI connector
Connector Type: 19-pin HDMI connector with flange

The pin assignments conform to the industry standard.

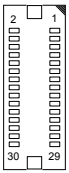


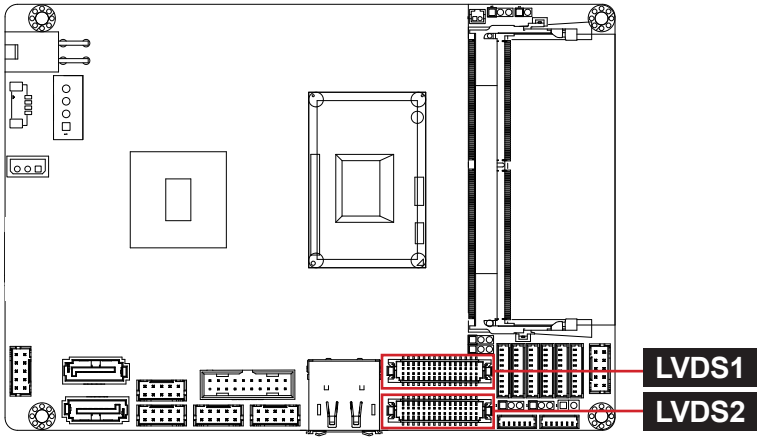


⑭⑮ LVDS1, 2

Function: LVDS1, 2 LCD panel connectors
Connector type: ACES 1.25mm 87209-3040-06 connector that supports 24-bit dual channels.

Pin	Description	Pin	Description
2	VDD2	1	VDD1
4	TX2_CLK+	3	TX1_CLK+
6	TX2_CLK-	5	TX1_CLK-
8	GND5	7	GND1
10	TX2_D0+	9	TX1_D0+
12	TX2_D0-	11	TX1_D0-
14	GND6	13	GND2
16	TX2_D1+	15	TX1_D1+
18	TX2_D1-	17	TX1_D1-
20	GND7	19	GND3
22	TX2_D2+	21	TX1_D2+
24	TX2_D2-	23	TX1_D2-
26	GND8	25	GND4
28	TX2_D3+	27	TX1_D3+
30	TX2_D3-	29	TX1_D3-

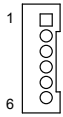


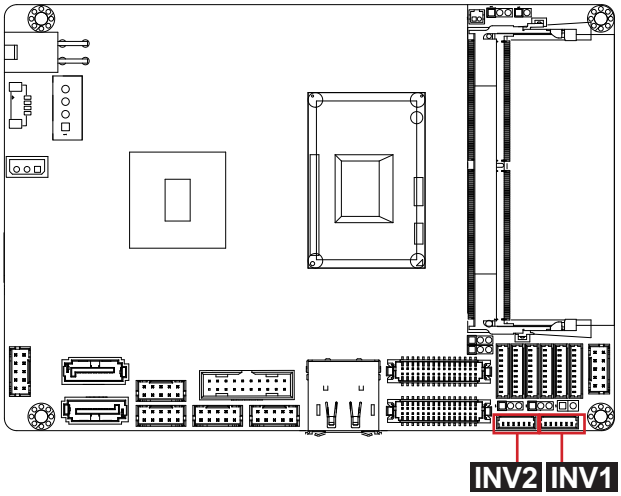


⑯⑰ INV1, 2

Function: LCD1, 2 inverter connectors
Connector type: 1.25mm pitch 1x6-pin box wafer connector

Pin	Description
1	LVDS_INV_VDD
2	LVDS_INV_VDD
3	LVDS_BKLT_EN
4	LVDS_BKLT_CTRL
5	GND
6	GND

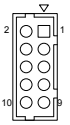


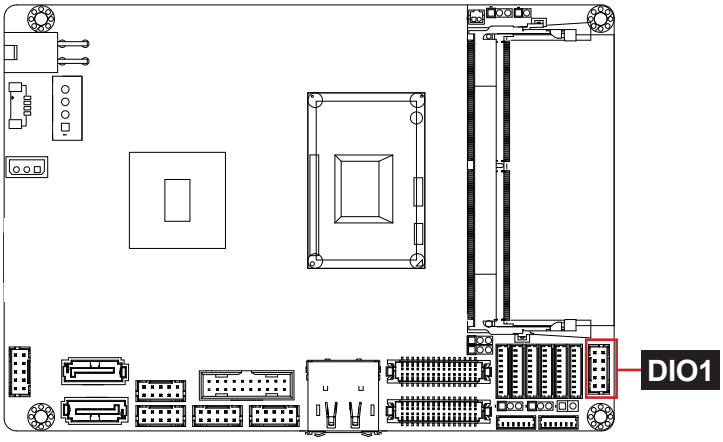


⑱ DIO1

Function: Digital I/O connector
Connector type: 2.00mm pitch 2x5-pin box headers

Pin	Desc.	Pin	Desc.
2	DIO1	1	DIO0
4	DIO3	3	DIO2
6	DIO5	5	DIO4
8	DIO7	7	DIO6
10	GND	9	+V5S

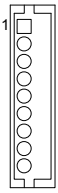


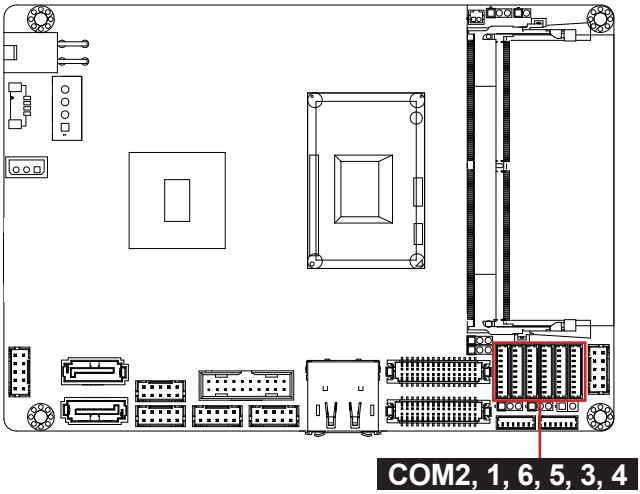


⑲~㉔ COM1~6

Function: Serial port connector
COM1-2: RS-232/422/485 selectable
COM3-6: RS-232
Connector type: 1.25mm pitch 1x9-pin wafer connector

	COM1-2			COM3-6
Pin	**RS-232**	**RS-422**	**RS-485**	**RS-232**
1	XDCD1#	T-	D-	XDCD#
2	XDSR1#			XDSR#
3	XRXD1	T+	D+	XRXD
4	XRTS1#			XRTS#
5	XTXD1	R+		XTXD
6	XCTS1#			XCTS#
7	XDTR1#	R-		XDTR#
8	XRI1#			XRI#
9	GND			GND

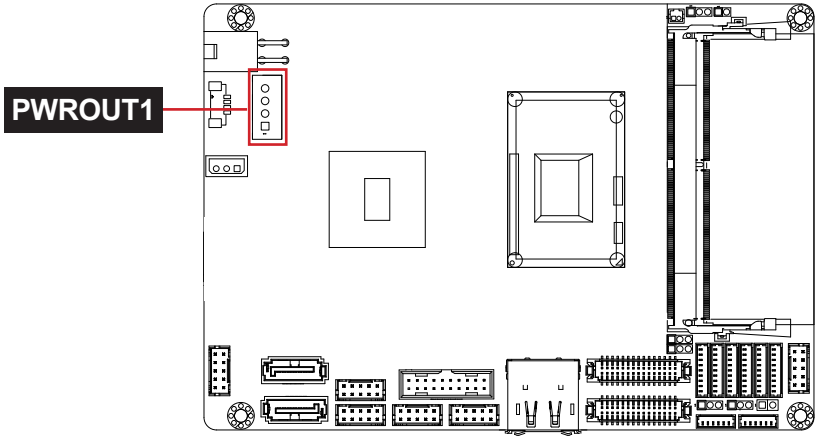




㉕ PWROUT1

Function: SATA power connector
Connector type: 2.50mm pitch 1x4-pin wafer connector

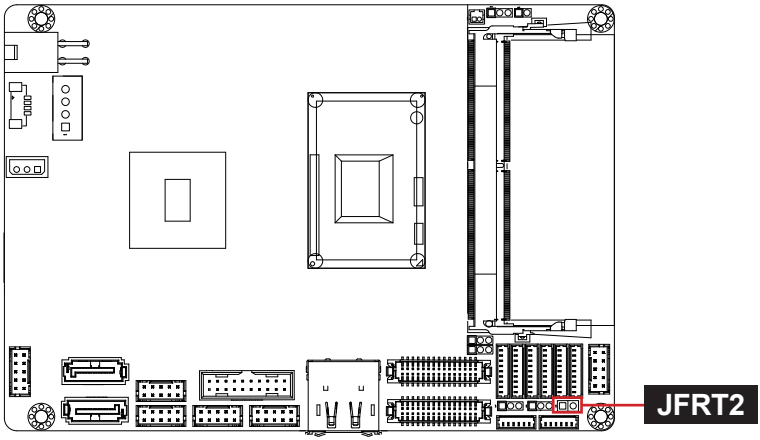
Pin	Description
1	+5VS
2	GND
3	GND
4	+12VS



㉖ JFRT2

Function: Power Button
Jumper type: 2.00mm pitch 1x2-pin header

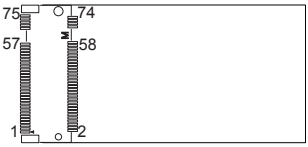
Pin	Description
1	PSON+
2	PSON-

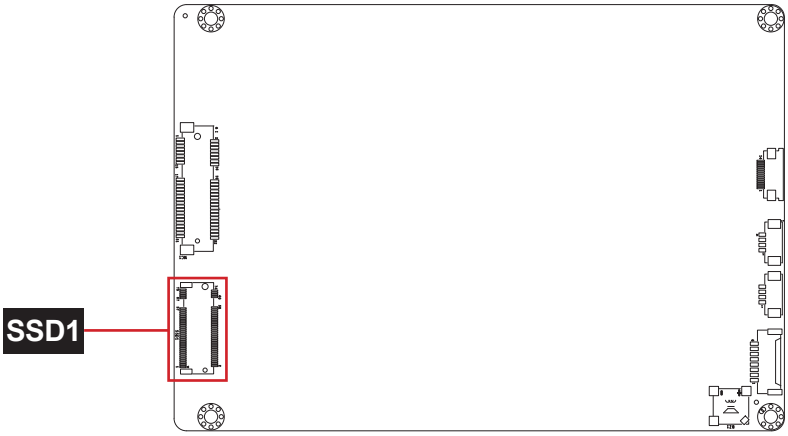


㉗ SSD1

Function: NGFF M.2 M-Key Socket
Connector Type: NGFF M.2 socket for M-Key 22x42 type to support SATA/ PCIe x4 depending on SSD module

The pin assignments conform to the industry standard.

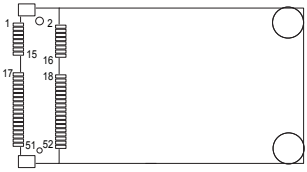


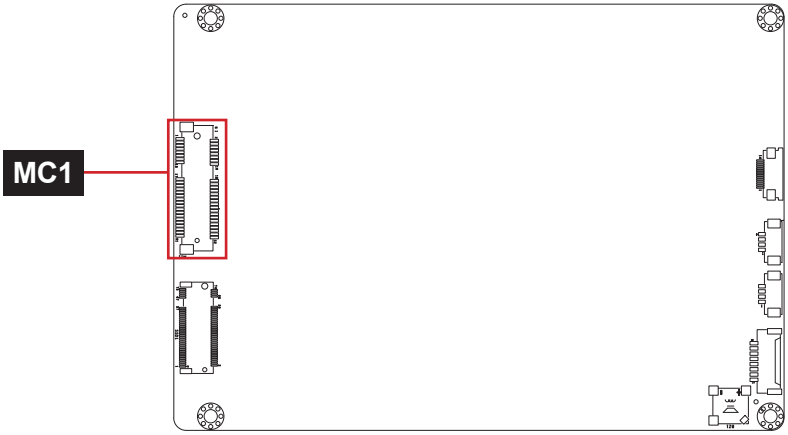


㉘ MC1

Function: Mini-card socket
Connector type: Onboard 0.8mm-pitch 52-pin edge card connector

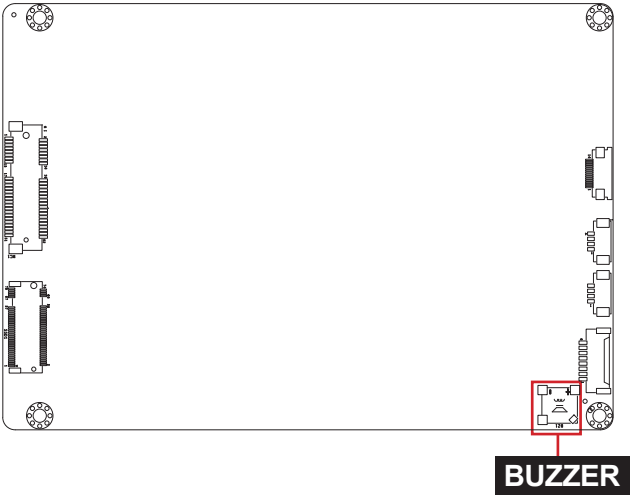
The pin assignments conform to the industry standard.





㉙ BUZZER

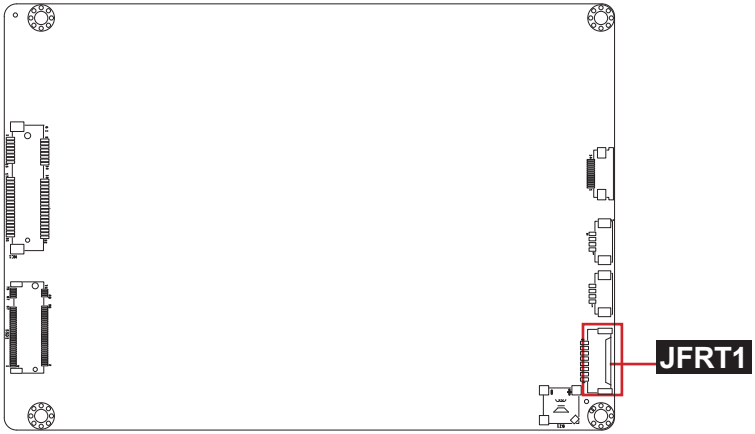
Function: Buzzer
Type: Onboard buzzer



㉚ JFRT1

Function: Front-panel connector
Connector type: Onboard 1.25mm pitch 1x8-pin wafer connector

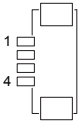
Pin	Description
1	RSTBTN#
2	GND
3	PW_LED
4	GND
5	HDD_LED
6	-HDDLED
7	SPKOUT+
8	SPKOUT#

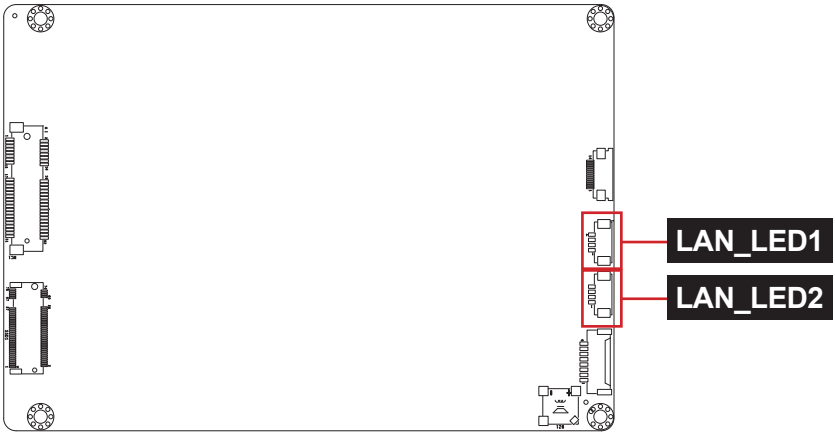


㉛㉜ LAN_LED1, 2

Function: LAN1, 2 LEDs
Connector type: Onboard 1.25mm pitch 1x4 pin wafter connector

Pin	Description
1	LAN_LED_LNK#_ACT
2	+V3.3M_LAN
3	LAN_LED_100#
4	LAN_LED_1000#

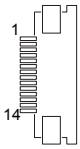


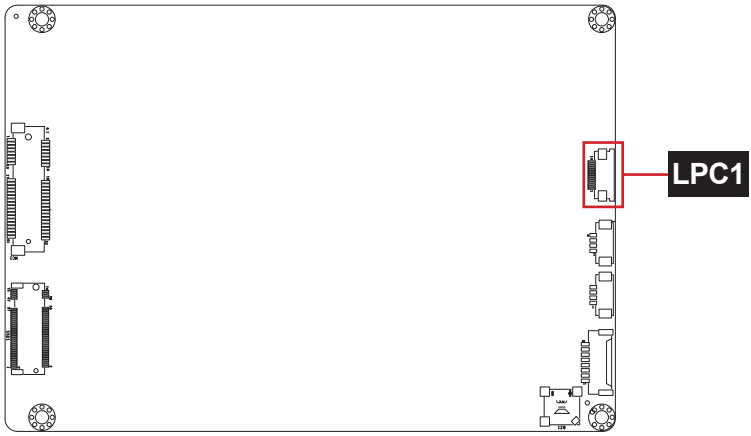


㉝LPC1

Function: Low Pin Count Connector
Connector type: Onboard 1.25mm pitch 1x14 pin wafter connector

Pin	Description
1	LD0
2	LD1
3	LD2
4	LD3
5	GND1
6	L_FRAME#
7	SER_IRQ
8	LPC_RST#
9	GND3
10	LPC_CLK(33M)
11	GND4
12	GND5
13	VCC3(1)
14	VCC3(2)





This page is intentionally left blank.

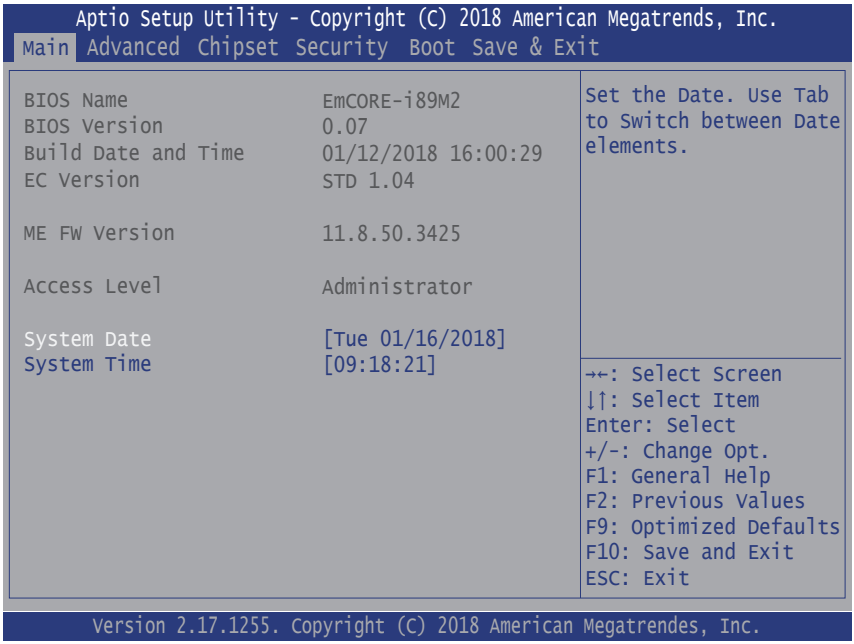
Chapter 3

BIOS

3.1 Main

The AMI BIOS provides a Setup utility program for specifying the system configurations and settings. The BIOS RAM of the system stores the Setup utility and configurations. When you turn on the computer, the AMI BIOS is immediately activated. To enter the BIOS SETUP UTILITY, press **"Delete"** once the power is turned on.

The **Main Setup** screen lists the following information:



Setting	Description
System Date	Set the system date. Use Tab to switch between Data elements. Note that the 'Day' automatically changes when you set the date. ► The date format is: **Day:** Sun to Sat **Month:** 1 to 12 **Date:** 1 to 31 **Year:** 1998 to 2099

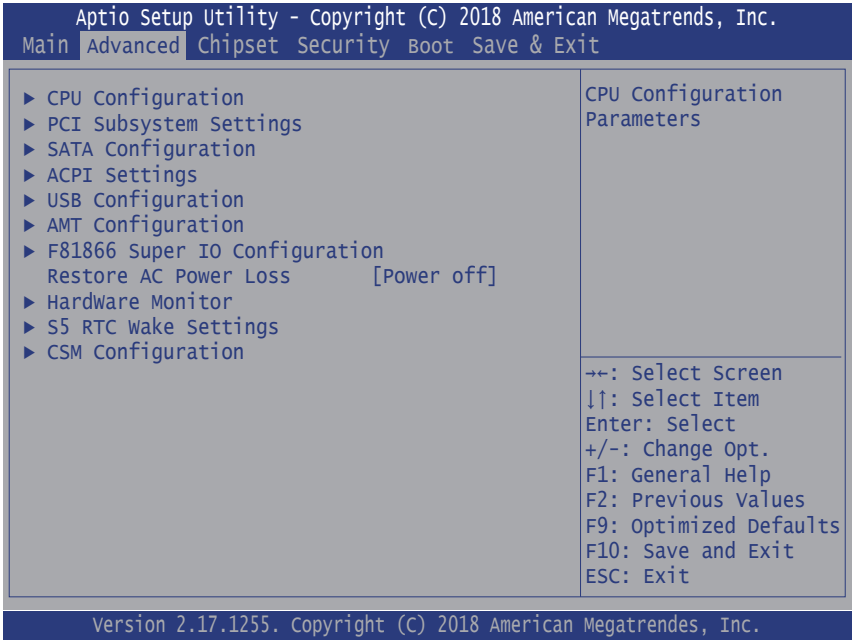
System Time	Set the system time. Use Tab to switch between Time elements. ► The time format is: **Hour:** 00 to 23 **Minute:** 00 to 59 **Second:** 00 to 59

Key Commands

BIOS Setup Utility is mainly a key-based navigation interface. Please refer to the following key command instructions for navigation process.

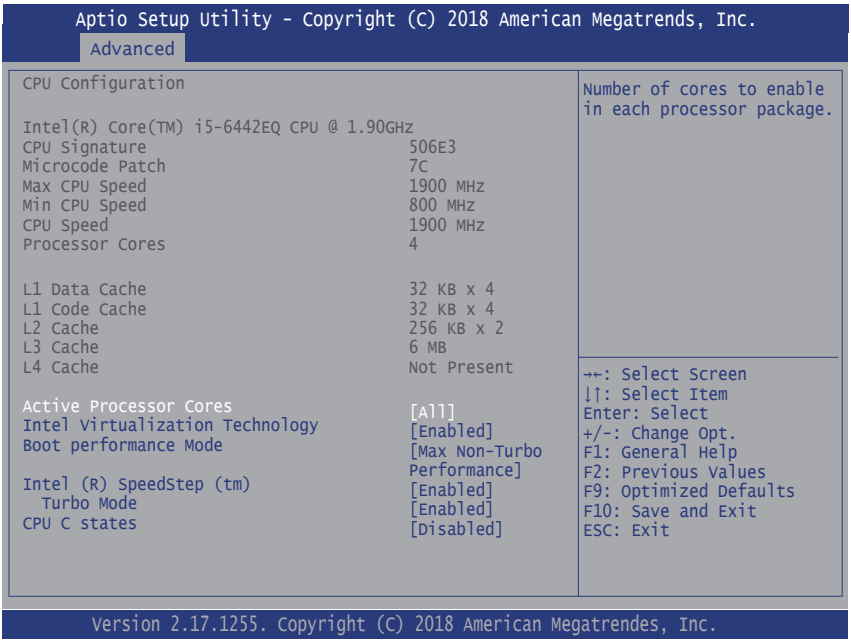
Keystroke	Function
◄ ►	Move to highlight a particular configuration screen from the top menu bar / Move to highlight items on the screen
▼ ▲	Move to highlight previous/next item
Enter	Select and access a setup item/field
Esc	On the Main Menu – Quit the setup and not save changes into CMOS (a message screen will display and ask you to select “OK” or “Cancel” for exiting and discarding changes. Use “←” and “→” to select and press “Enter” to confirm) On the Sub Menu – Exit current page and return to main menu
Page Up / +	Increase the numeric value on a selected setup item / make change
Page Down / -	Decrease the numeric value on a selected setup item / make change
F1	Activate “General Help” screen
F10	Save the changes that have been made in the setup and exit. (a message screen will display and ask you to select “OK” or “Cancel” for exiting and saving changes. Use “←” and “→” to select and press “Enter” to confirm)

3.2 Advanced



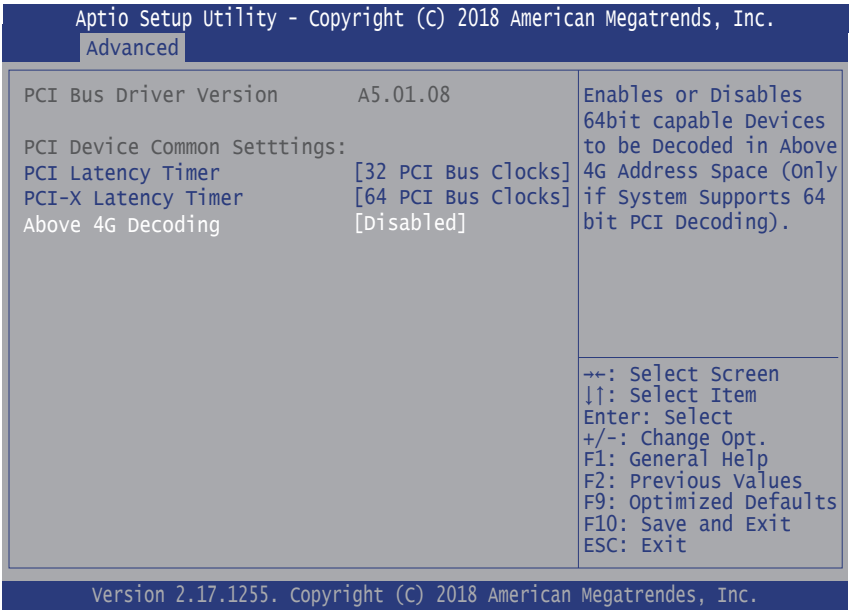
Setting	Description
CPU Configuration	See section 3.2.1 CPU Configuration on page 43
PCI Subsystem Settings	See section 3.2.2 PCI Subsystem Settings on page 44
SATA Configuration	See section 3.2.3 SATA Configuration on page 45
ACPI Settings	See section 3.2.4 ACPI Settings on page 46
USB Configuration	See section 3.2.5 USB Configuration on page 47
AMT Configuration	See section 3.2.6 AMT Configuration on page 49
F81866 Super IO Configuration	See section 3.2.7 F81866 Super IO Configuration on page 50
Hardware Monitor	See section 3.2.8 Hardware Monitor on page 52
S5 RTC Wake Settings	See section 3.2.9 S5 RTC Wake Settings on page 53
CSM Configuration	See section 3.2.10 CSM Configuration on page 54

3.2.1 CPU Configuration



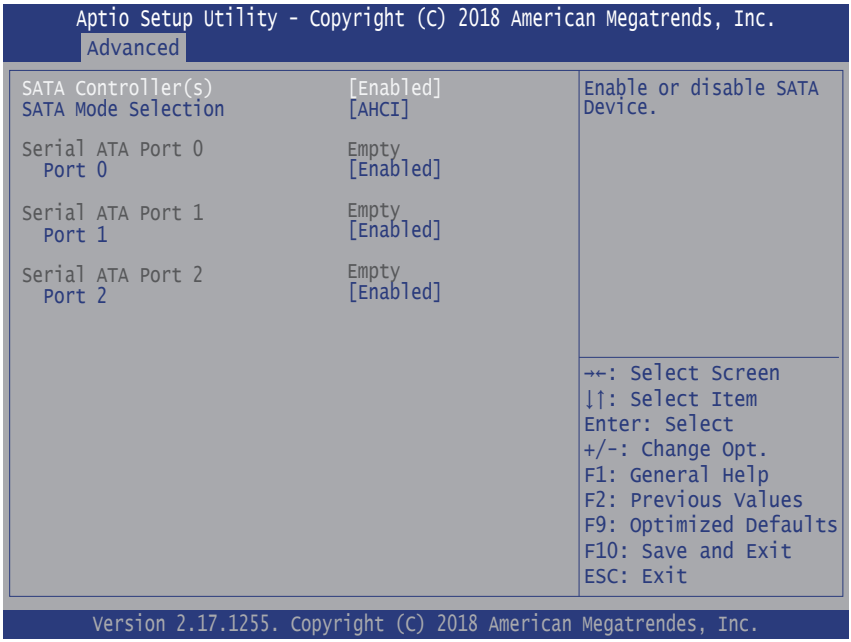
Setting	Description
Active Processor Cores	Number of cores to enable in each processor package. ► Options: **All** (default), **1**, **2** and **3.**
Intel Virtualization Technology	When enabled, a VMM can utilize the additional hardware capabilities provided by Vanderpool Technology ► Options: **Enabled** (default) or **Disabled**
Boot performance Mode	Set the performance state that the BIOS will set before the OS handoff. ► Options: **Max Battery, Max Non-Turbo Performance** (default) and **Turbo Performance**.
Intel (R) Speed Step (tm)	**Enable** (default)/**Disable** Intel SpeedStep
Turbo Mode	Only available when Intel Speed Step is **Enabled**. **Enable** (default)/**Disable** Turbo Mode
CPU C States	**Enable** /**Disable** (default) CPU C States

3.2.2 PCI Subsystem Settings



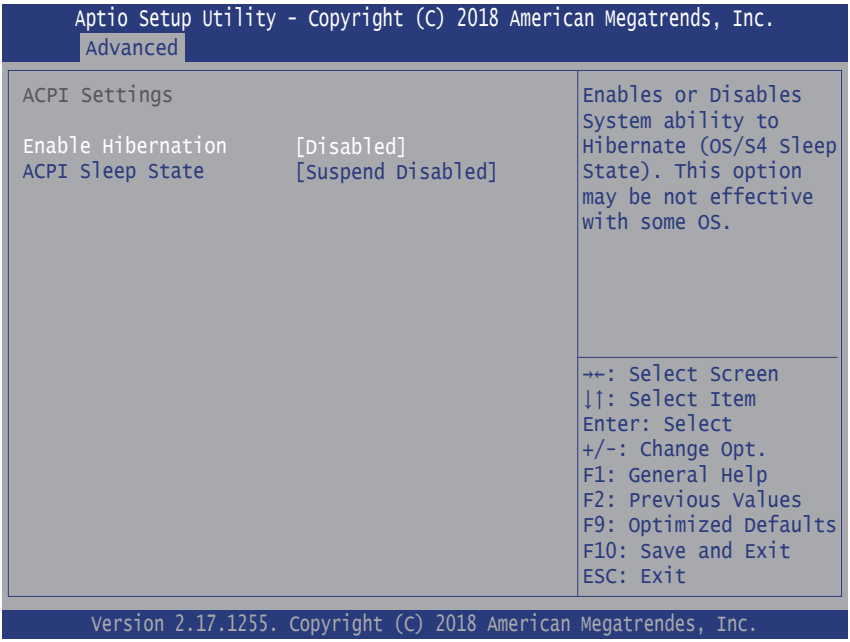
Setting	Description
PCI Latency Timer	Value to be programmed into PCI Latency Timer Register. ► **32** (default), **64**, **96**, **128**, **160**, **192**, **224** and **248 PCI Bus Clocks**.
PCI-X Latency Timer	Value to be programmed into PCI-X Latency Timer Register. ► **32**, **64** (default), **96**, **128**, **160**, **192**, **224** and **248 PCI Bus Clocks**.
Above 4G Decoding	**Enable**/**Disable** (default) 64bit capable Devices to be Decoded in Above 4G Address Space (Only if System Supports 64 bit PCI Decoding).

3.2.3 SATA Configuration



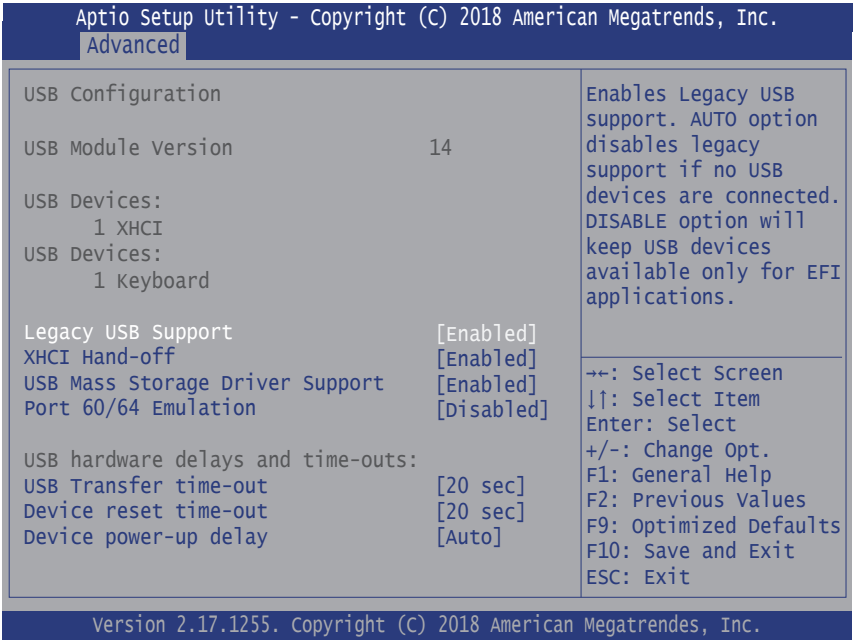
Setting	Description
SATA Controller(s)	**Enable** (default) or **disable** SATA Device.
SATA Mode Selection	Determines how SATA controller(s) operate. ► Options: **AHCI** (default) and **RAID**
Port 0/1/2	**Enable** (default) or **disable** SATA Port.

3.2.4 ACPI Settings



Setting	Description
Enable Hibernation	**Enable** (default) or **Disable** System ability to Hibernate (OS/S4 Sleep State). This option may be not effective with some OS.
ACPI Sleep State	Select the highest ACPI sleep state the system will enter when the SUSPEND button is pressed. ► Options: **Suspend Disabled** and **S3 (Suspend to RAM)** (default).

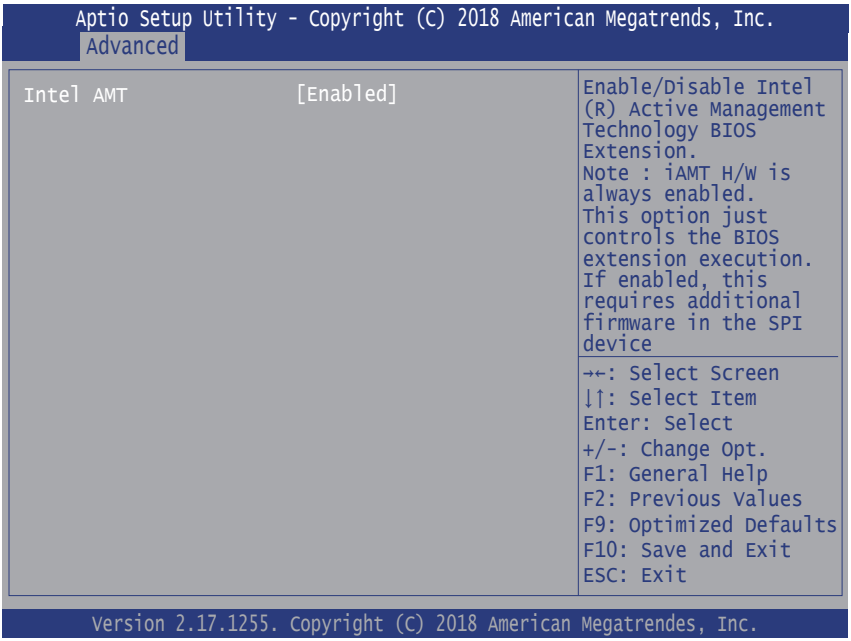
3.2.5 USB Configuration



Setting	Description
Legacy USB Support	Sets legacy USB support. ► Options: **Enabled** (default), **Disabled** and **Auto**. **AUTO** option disables legacy support if no USB devices are connected. **Disable** option will keep USB devices available only for EFI applications.
XHCI Hand-off	**Enable** (default) or **Disable** 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	**Enable** (default) or **Disable** USB Mass Storage Driver Support.

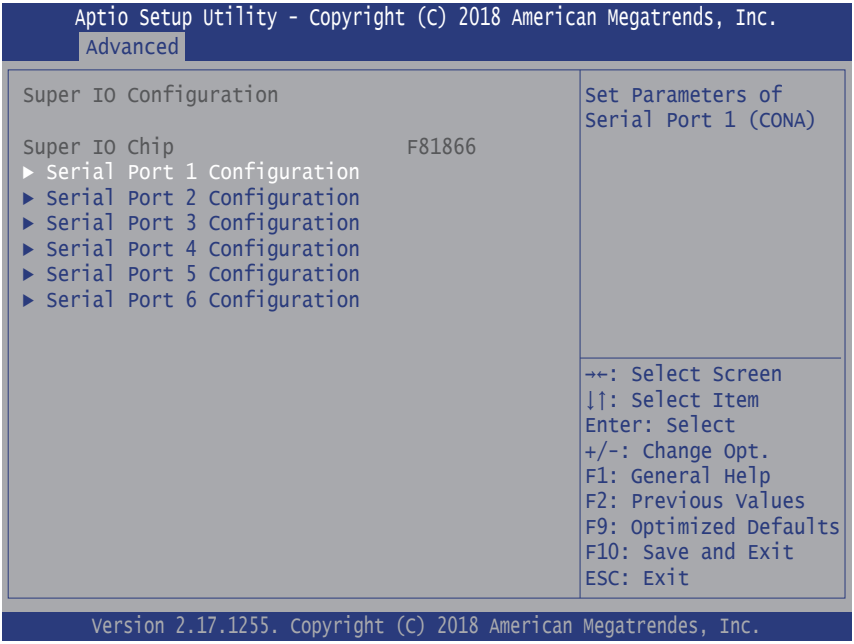
Port 60/64 Emulation	**Enable** or **Disable** (default) I/O port 60h/64h emulation support. This should be enabled for the complete USB keyboard legacy suppot for non-USB aware OSes.
USB hardware delay and time-out	
USB Transfer time-out	Use this item to set the time-out value for control, bulk, and interrupt transfers. ▸ Options available are: **1 sec, 5 sec, 10 sec, 20 sec** (default)
Device reset time-out	Use this item to set USB mass storage device start unit command time-out. ▸ Options available are: **10 sec, 20 sec** (default)**, 30 sec, 40 sec**
Device power-up delay	Use this item to set maximum time the device will take before it properly reports itself to the host controller. ▸ Options available are: **Auto** (Default): 'Auto' uses default value: for a root port it is 100 ms, for a hub port the delay is taken from hub descriptor. **Manual**: Select **Manual** you can set value for the following sub-item: '**Device Power-up delay in seconds**', the delay range in from 1 to 40 seconds, in one second increments.

3.2.6 AMT Configuration



Setting	Description
Intel AMT	**Enable** (default)/**Disable** Intel(R) Active Management Technology BIOS Extension. Note : iAMT H/W is always enabled. This option just controls the BIOS extension execution. If enabled, this requires additional firmware in the SPI device.

3.2.7 F81866 Super IO Configuration

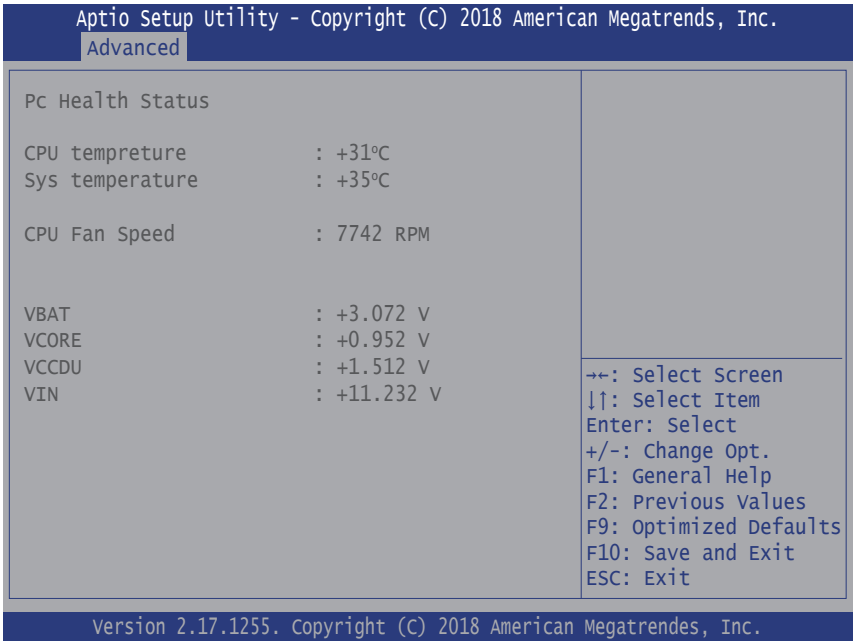


Setting	Description
Serial Port 1~6 Configuration	See next page.
Restore AC Power Loss	Specify what state to go to when power is re-applied after a power failure. ► Options: **Power On** and **Power Off** (default)

Serial Port 1~6 Configuration

Setting	Description
Serial Port	**Enable** (default) or **Disable** Serial Port (COM).
Change Settings	Select an optimal setting for Super IO device. ► Options for Serial Port 1: **Auto**; **IO=3F8h; IRQ=4** (default) ; **IO=3F8h; IRQ=3, 4, 5, 6, 7, 9, 10, 11, 12**; **IO=2F8h; IRQ=3, 4, 5, 6, 7, 9, 10, 11, 12**; **IO=3E8h; IRQ=3, 4, 5, 6, 7, 9, 10, 11, 12**; **IO=2E8h; IRQ=3, 4, 5, 6, 7, 9, 10, 11, 12**; ► Options for Serial Port 2: **Auto** **IO=2F8h; IRQ=3** (default) **IO=3F8h; IRQ=3, 4, 5, 6, 7, 9, 10, 11, 12**; **IO=2F8h; IRQ=3, 4, 5, 6, 7, 9, 10, 11, 12**; **IO=3E8h; IRQ=3, 4, 5, 6, 7, 9, 10, 11, 12**; **IO=2E8h; IRQ=3, 4, 5, 6, 7, 9, 10, 11, 12**; ► Options for Serial Port 3: **Auto** **IO=3F8h; IRQ=7** (default) **IO=3E8h; IRQ=3, 4, 5, 6, 7, 9, 10, 11, 12**; **IO=2E8h; IRQ=3, 4, 5, 6, 7, 9, 10, 11, 12**; **IO=2F8h; IRQ=3, 4, 5, 6, 7, 9, 10, 11, 12**; **IO=2E8h; IRQ=3, 4, 5, 6, 7, 9, 10, 11, 12**; ► Options for Serial Port 4: **Auto** **IO=2E8h; IRQ=7** (default) **IO=3E8h; IRQ=3, 4, 5, 6, 7, 9, 10, 11, 12**; **IO=2E8h; IRQ=3, 4, 5, 6, 7, 9, 10, 11, 12**; **IO=2F8h; IRQ=3, 4, 5, 6, 7, 9, 10, 11, 12**; **IO=2E8h; IRQ=3, 4, 5, 6, 7, 9, 10, 11, 12**; ► Options for Serial Port 5: **Auto** **IO=2E0h; IRQ=7** (default) **IO=3E8h; IRQ=3, 4, 5, 6, 7, 9, 10, 11, 12**; **IO=2E8h; IRQ=3, 4, 5, 6, 7, 9, 10, 11, 12**; **IO=2F0h; IRQ=3, 4, 5, 6, 7, 9, 10, 11, 12**; **IO=2E0h; IRQ=3, 4, 5, 6, 7, 9, 10, 11, 12**;

3.2.8 Hardware Monitor

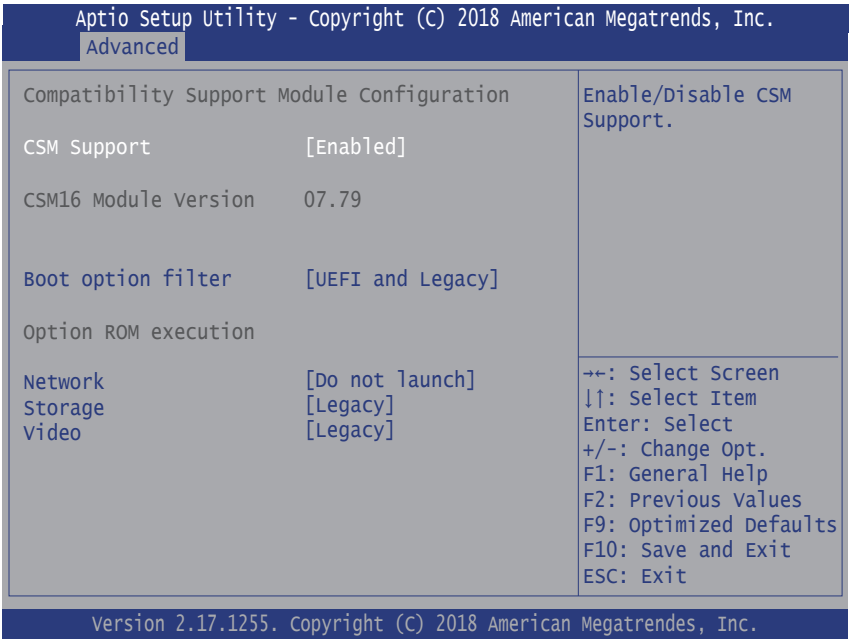


Access this submenu to monitor the hardware status.

3.2.9 S5 RTC Wake Settings

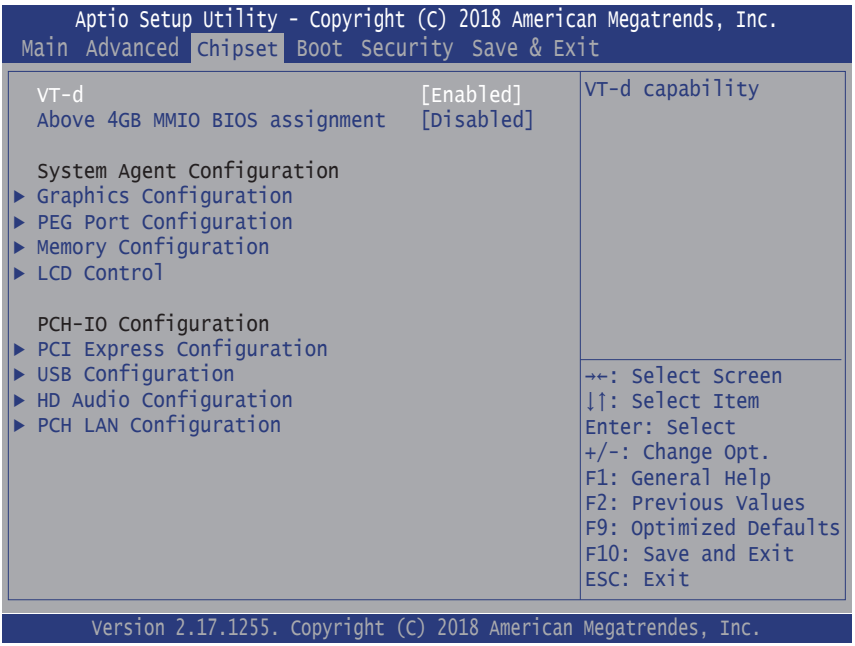
Setting	Description
Wake System from S5	**Enable** or **Disable** (default) system wake on alarm event. ► Options available are: **Disabled** (default): **Fixed Time:** System will wake on the hr::min::sec specifiedc. **DynamicTime:** If selected, you need to set **Wake up minute increase** from 1 - 5. System will wake on the current time + increase minute(s).

3.2.10 CSM Configuration



Setting	Description
CSM Support	**Enable** (default) or **Disable** CSM Support.
Boot option filter	Control the Legacy/UEFI ROMs priority. ► Options: **UEFI and Legacy** (default), **Legacy only**, **UEFI only**
Network	Control the execution of UEFI and Legacy PXE OpROM ► Options: **Do not launch** (default) and **Legacy**
Storage	Control the execution of UEFI and Legacy Storage OpROM ► Options: **Do not launch** and **Legacy** (default)
Video	Control the execution of UEFI and Legacy Video OpROM ► Options: **UEFI** and **Legacy** (default)

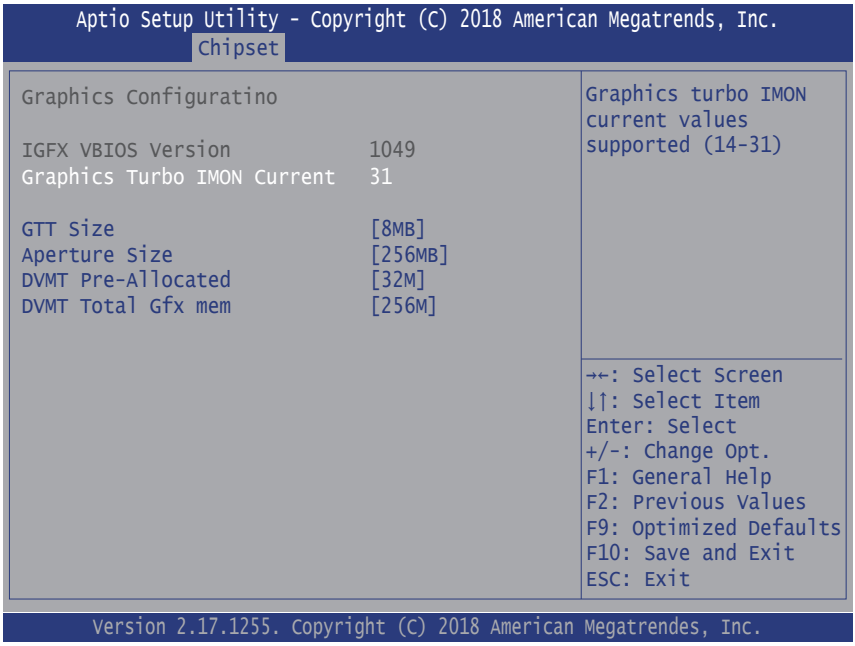
3.3 Chipset



Setting	Description
VT-d	**Enable** (default) or **Disable** VT-d function
Above 4GB MMIO BIOS assignment	**Enable** or **Disable** (default) Above 4GB MMIO BIOS assignment
System Agent (SA) Configuration	
Graphics Configuration	See section 3.3.1 Graphics Configuration on page 57
Memory Configuration	See section 3.3.2 Memory Configuration on page 59
LCD Control	See section 3.3.3 LCD Control on page 60
PCH-IO Configuration	
PCI Express Configuration	See section 3.3.4 PCI Express Configuration on page 61

USB Configuration	See section 3.3.5 USB Configuration on page 62
HD Audio Configuration	Control Detection of the HD-Audio device. ► Options available are: **Disabled**: HDA will be unconditionally disabled **Enabled** (default) : HDA will be unconditionally Enabled **Auto** = HDA will be enabled if present, disabled otherwise.
PCH LAN Controller	Enables/Disables onboard NIC. ► Options: **Enabled** (default) and **Disabled** If enabled, “Wake on LAN” option will be available to **Enable** (default) / **Disable** integrated LAN to wake the system. (the Wake On LAN cannot be disabled if ME is on at Sx state.)

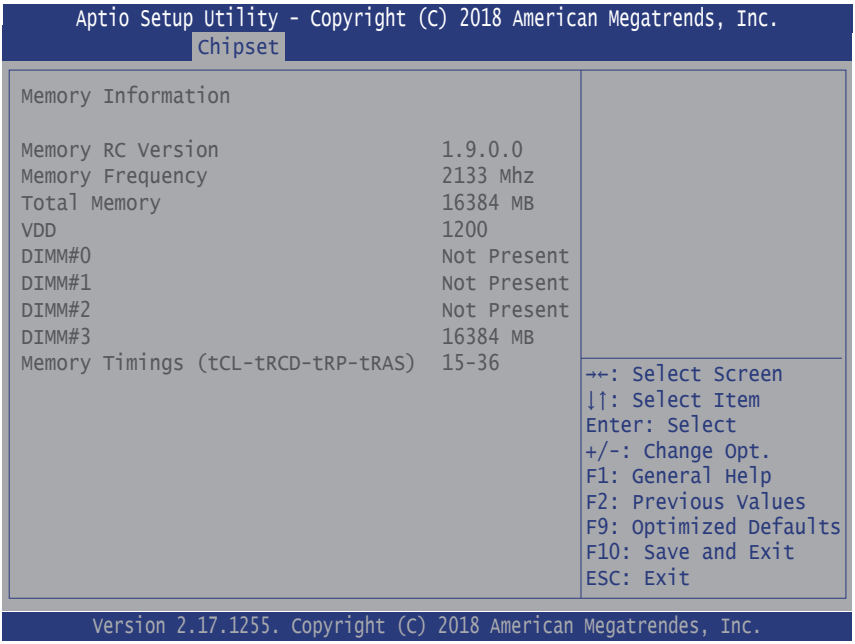
3.3.1 Graphics Configuration



Setting	Description
Graphics Turbo IMON Current	Sets the graphics turbo IMON current values. ► Options available are **14** to **31** (default).
GTT Size	Select the GTT Size. ► Options: **4MB, 2MB** and **8MB** (default)**.**
Apeture Size	Select the Apeture Size. Note that above 4GB MMIO BIOS assignment is automatically enabled when selecting 2048MB aperture. To use this feature, please disable CSM support. ► Options: **128MB**, **256MB** (default)**, 512MB**, **1024MB**, **2048MB** and **4096MB**.

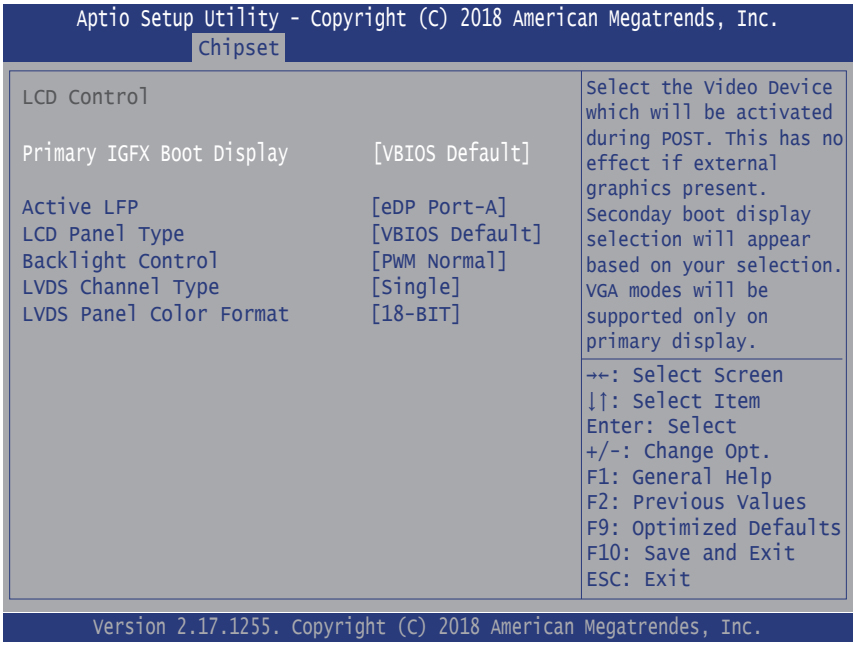
DVMT Pre-Allocated	Select the DVMT 5.0 Pre-allocated (Fixed) Graphic Memory size used by the Internal Graphic Device. ► Options: **32M** is the default.
DVMT Total Gfx Mem	Select the DVMT 5.0 Total Graphic Memory size used by the Internal Graphic Device. ► Options: **128MB, 256MB** (default) and **Max.**

3.3.2 Memory Configuration



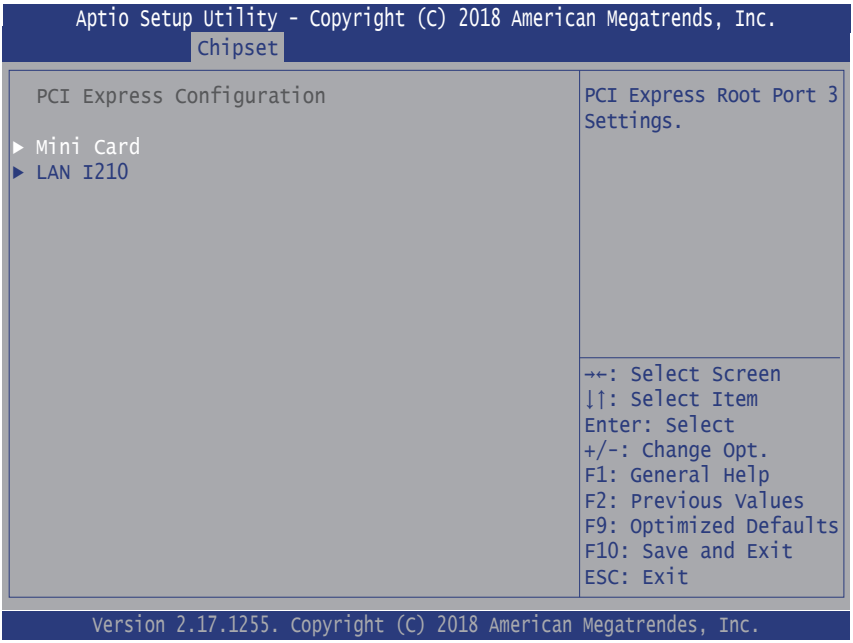
Access this submenu to view the memory configuration.

3.3.3 LCD Control



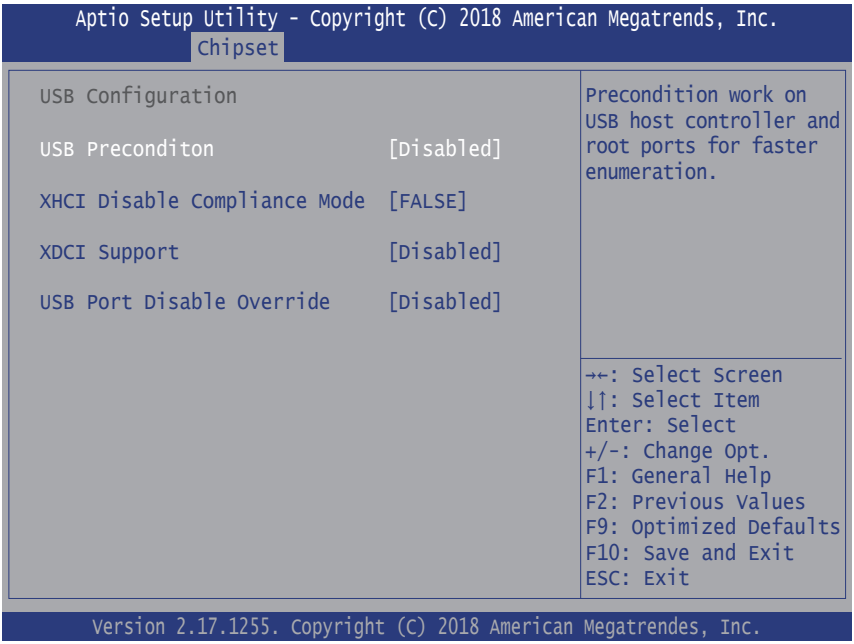
Setting	Description
Primary IGFX Boot Display	Select the Video Device which will be activated during POST. This has no effect if external graphics present. Secondary boot display selection will appear based on your selection. VGA modes will be supported only on primary display. ► Options: **VBIOS Default** (default), **DVI**, **DP** and **EFP3**.
Active LFP	Configuring LFP usage ► Options: **No LVDS** and **eDP Port-A** (default)
LCD Panel Type	Select LCD panel used by Internal Graphics Device by selecting the appropriate setup item. Default: : **VBIOS Default**
Backlight Control	Set the Back Light Control. ► Options: **PWM Inverted** and **PWM Normal** (default)
LVDS Channel Type	Select single or dual channel ► Options: **Dual** and **Single** (default)
LVDS Panel Color Format	Select LVDS color display mode ► Options: **24-BIT** and **18-BIT** (default)

3.3.4 PCI Express Configuration



Setting		Description
Mini Card/ LAN I210	Mini Card/LAN I210	**Enable** (default) or **Disable** the PCIe Express Root Port.
	ASPM Support	Disable or set the ASPM level. Force L0s will force all inks to L0s state. “Auto” will allow BIOS to auto configure.”Disable” will disable ASPM. ► Options: **Disabled** (default), **L0s**, **L1**, **L0sL1** and **Auto**.
	PCIe Speed	Select PCI Express port speed. ► Options: **Auto** (default), **Gen1**, **Gen2** and **Gen3**

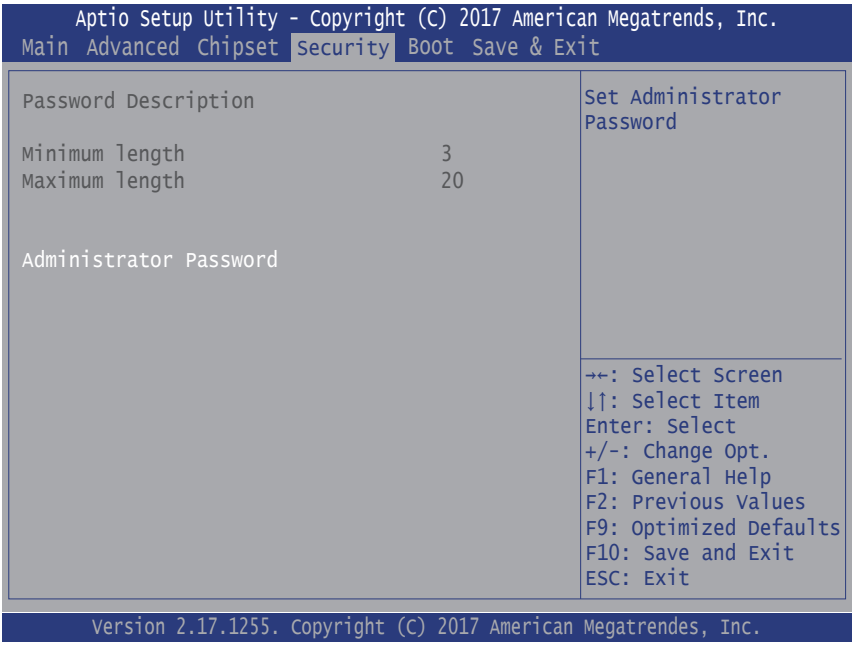
3.3.5 USB Configuration



Setting	Description
USB Precondition	Precondition work on USB host controller and root ports for faster enumeration. ► Options: **Enable**/**Disable** (default).
XHCI Disable Compliance Mode	Options to disable Compliance Mode. Default is FALSE to not disable Compliance Mode. Set TRUE to disable Compliance Mode. ► Options: **False** (default)/**True**.
xDCI Support	**Enable** or **Disable** (default) xDCI (USB OTG Device.
USB Port Disable Override	Selectively enable/disable the corresponding USB port from reporting a Device Connection to the controller. ► Options: **Disabled** (default) / **Select Per-Pin**.

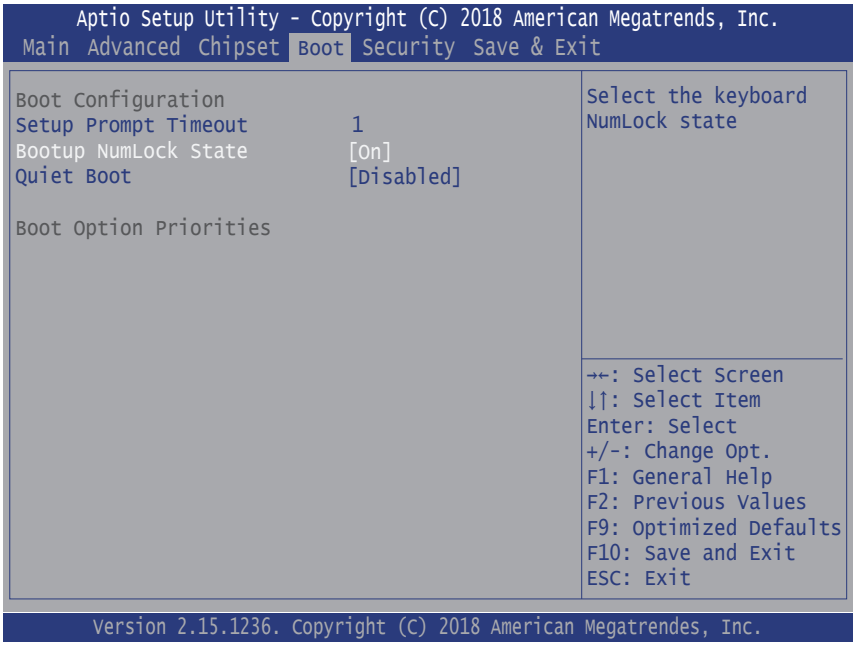
3.4 Security

The **Security** menu sets up the administrator password.



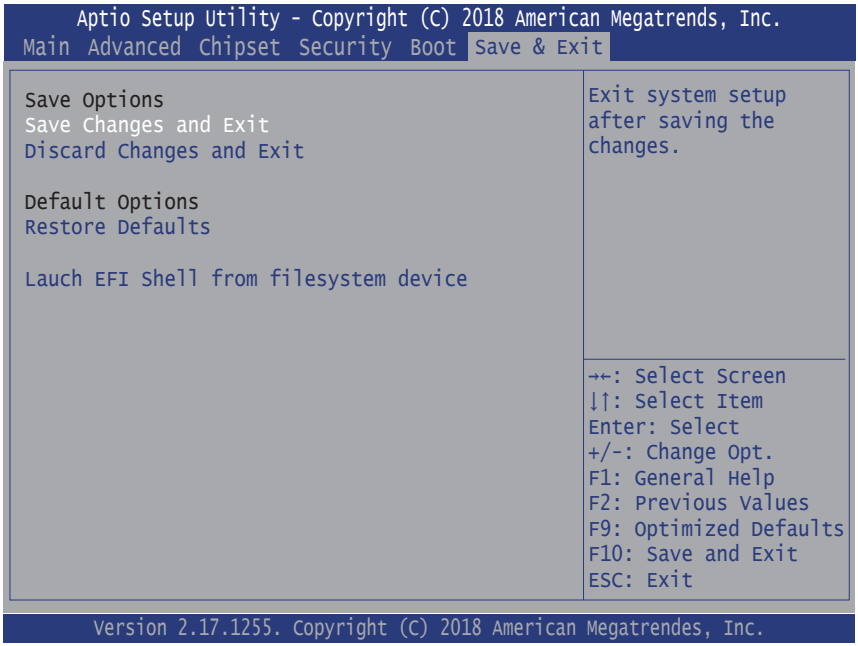
Setting	Description
Administrator Password	To set up an administrator password: 1. Select **Administrator Password**. The screen then pops up an **Create New Password** dialog. 2. Enter your desired password that is no less than 3 characters and no more than 20 characters. 3. Hit [Enter] key to submit.

3.5 Boot



Setting	Description
Setup Prompt Timeout	Number of seconds to wati for setup activation key. 65535 (0XFFFF) means indefinite waiting.
Boot NumLock State	Select the keyboard NumLock state. ► Options: **On** (default) and **Off**.
Quiet Boot	**Enable** or **Disable** (default) Quiet Boot option.

3.6 Save & Exit



Setting	Description
Save Changes and Exit	Exit system setup after saving the changes. ► Enter the item and then a dialog box pops up: **Save configuration and exit? (Yes/ No)**
Discard Changes and Exit	Exit system setup without saving the changes. ► Enter the item and then a dialog box pops up: **Quit without saving? (Yes/ No)**
Restore Defaults	Restore/Load Default values for all the setup options. ► Enter the item and then a dialog box pops up: **Load Optimized Defaults? (Yes/ No)**
Launch EFI Shell from filesystem device	Attempts to launch EFI shell application (Shell.efi) from one of the available filesystem devices.

3.7 Beep Sound codes list

3.7.1 Boot Block Beep codes

Number of Beeps	Description
1	Insert diskette in floppy drive A:
2	'AMIBOOT.ROM' file not found in root directory of diskette in A:
4	Flash Programming successful
5	Floppy read error
6	Keyboard controller BAT command failed
7	No Flash EPROM detected
8	Floppy controller failure
9	Boot Block BIOS checksum error
10	Flash Erase error
11	Flash Program error
12	'AMIBOOT.ROM' file size error
13	BIOS ROM image mismatch (file layout does not match image present in flash device)

3.7.2 POST BIOS Beep codes

Number of Beeps	Description
1	Memory refresh timer error.
2	Parity error in base memory (first 64KB block)
4	Motherboard timer not operational
5	Processor error
6	8042 Gate A20 test error (cannot switch to protected mode)
7	General exception error (processor exception interrupt error)
8	Display memory error (system video adapter)
9	AMIBIOS ROM checksum error
10	CMOS shutdown register read/write error
11	Cache memory test failed

3.7.3 Troubleshooting POST BIOS Beep codes

Number of Beeps	Description
1, 2 or 3	Reseat the memory, or replace known good modules.
4-7, 9-11	Fatal error indicating a serious problem with the system. Consult your system manufacturer. Before declaring the motherboard beyond all hope, eliminate the possibility of interference by a malfunctioning add-in card. Remove all expansion cards except the video adapter. • If beep codes are generated when all other expansion cards are absent, consult your system manufacturer's technical support. • If beep codes are not generated when all other expansion cards are absent, one of the add-in cards is causing the malfunction. Insert the cards back into the system one at a time until the problem is solved.
8	If the system video adapter is an add-in card, replace or reseat the video adapter. If the video adapter is an integrated part of the system board, the board may be faulty.

3.8 AMI BIOS Checkpoints

3.8.1 Bootblock Initialization Code Checkpoints

The Bootblock initialization code sets up the chipset, memory and other components before system memory is available. The following table describes the type of checkpoints that may occur during the bootblock initialization portion of the BIOS *(Note)*:

Checkpoint	Description
Before D0	If boot block debugger is enabled, CPU cache-as-RAM functionality is enabled at this point. Stack will be enabled from this point.
D0	Early Boot Strap Processor (BSP) initialization like microcode update, frequency and other CPU critical initialization. Early chipset initialization is done.
D1	Early super I/O initialization is done, including RTC and keyboard controller. Serial port is enabled at this point if needed for debugging. NMI is disabled. Perform keyboard controller BAT test. Save power-on CPUID value in scratch CMOS. Go to flat mode with 4GB limit and GA20 enabled.
D2	Verify the boot block checksum. System will hang here if checksum is bad.
D3	Disable CACHE before memory detection. Execute full memory sizing module. If memory sizing module is not executed, start memory refresh and do memory sizing in Boot block code. Do additional chipset initialization. Re-enabled CACHE. Verify that flat mode is enabled.
D4	Test base 512KB memory. Adjust policies and cache first 8MB. Set stack.
D5	Bootblock code is copied from ROM to lower system memory and control is given to it. BIOS now executes out of RAM. Copies compressed boot block code to memory in right segments. Copies BIOS from ROM to RAM for faster access. Perform main BIOS checksum and update recovery status accordingly.

D6	Both key sequence and OEM specific method is checked to determine if BIOS recovery is forced. If BIOS recovery is necessary, control flows to checkpoint E0. See *Bootblock Recovery Code Checkpoints* section of document for more information.
D7	Restore CPUID value back into register. The Bootblock-Runtime interface module is moved to system memory and given control to it. Determine whether in memory.
D8	The Runtime module is uncompressed into memory. CPUID information is stored in memory.
D9	Store the Uncompressed pointer for future use in PMM. Copying Main BIOS into memory. Leave all RAM below 1MB Read-Write, including E000 and F000 shadow areas, but close SMRAM.
DA	Restore CPUID value back into register. Give control to BIOS POS (ExecutePOSTKernel). See *POST Code Checkpoints* section of document for more information.
DC	System is waking from ACPI S3 state.
E1 - E8 EC - EE	OEM memory detection / configuration error. This range is reserved for chipset vendors & system manufacturers. The error associated with this value may be different from platform next to it.

3.8.2 Bootblock Recovery Code Checkpoints

The Bootblock recovery code gets control when the BIOS determines that a BIOS recovery needs to occur because the user has forced the update or the BIOS checksum is corrupt. The following table describes the type of checkpoints that may occur during the Bootblock recovery portion of the BIOS *(Note)*:

Checkpoint	Description
E0	Initialize the floppy controller in the super I/O. Some interrupt vectors are initialized. DMA controller is initialized. 8259 interrupt controller is initialized. L2 cache is enabled.
E9	Set up floppy controller and data. Attempt to read from floppy.
EA	Enable ATAPI hardware. Attempt to read from ARMD and ATAPI CDROM.
EB	Disable ATAPI hardware. Jump back to checkpoint E9.
EF	Read error occurred on media. Jump back to checkpoint EB.
F0	Search for pre-defined recovery file name in root directory.
F1	Recovery file not found.
F2	Start reading FAT table and analyze FAT to find the clusters occupied by the recovery file.
F3	Start reading the recovery file cluster by cluster.
F5	Disable L1 cache.
FA	Check the validity of the recovery file configuration in line with the current configuration of the flash part.
FB	Set flash write enabled through chipset and OEM specific method. Detect proper flash part. Verify that the found flash part size equals to the recovery file size.
F4	The recovery file size does not equal to the found flash part size.

FC	Erase the flash part.
FD	Program the flash part.
FF	The flash has been updated successfully. Set flash write disabled. Disable ATAPI hardware. Restore CPUID value back into register. Give control to F000 ROM at F000:FFF0h.

3.8.3 POST Code Checkpoints

The POST code checkpoints are the largest set of checkpoints during the BIOS pre-boot process. The following table describes the type of checkpoints that may occur during the POST portion of the BIOS *(Note)*:

Checkpoint	Description
03	Disable NMI, Parity, video for EGA, and DMA controllers. Initialize BIOS, POST, Runtime data area. Also, initialize BIOS modules on POST entry and GPNV area. Initialize CMOS as mentioned in the Kernel Variable "wCMOSFlags."
04	Check CMOS diagnostic byte to determine if battery power is OK and CMOS checksum is OK. Verify CMOS checksum manually by reading storage area. If the CMOS checksum is bad, update CMOS with power-on default values and clear passwords. Initialize status register A. Initialize data variables based on CMOS setup questions. Initialize both 8259 compatible PICs in the system.
05	Initialize the interrupt controlling hardware (generally, PIC) and interrupt vector table.
06	Do R/W test for CH-2 count reg. Initialize CH-0 as system timer. Install the POSTINT1Ch handler. Enable IRQ-0 in PIC for system timer interrupt. Trap INT1Ch vector in "POSTINT1ChHandlerBlock."
07	Fix CPU POST interface calling pointer.
08	Initialize the CPU. The BAT test is being done on KBC. Program the keyboard controller command byte after Auto detection of KB/MS uses AMI KB-5.
C0	Early CPU Init Start -- Disable Cache - Init Local APIC
C1	Set up boot strap processor Information
C2	Set up boot strap processor for POST
C5	Enumerate and set up application processors
C6	Re-enable cache for boot strap processor

C7	Early CPU Init Exit
0A	Initialize the 8042 compatible Key Board Controller.
0B	Detect the presence of PS/2 mouse.
0C	Detect the presence of Keyboard in KBC port.
0E	Test and initialize different input devices. Also, update the Kernel Variables. Trap the INT09h vector, so that the POST INT09h handler gets control over IRQ1. Uncompress all available language, BIOS logo, and Silent logo modules.
13	Early POST initialization of chipset registers.
20	Relocate System Management Interrupt vector for all CPU in the system.
24	Uncompress and initialize any platform of specific BIOS modules. GPNV is initialized at this checkpoint.
2A	Initialize different devices through DIM. See DIM Code Checkpoints section in document for more information.
2C	Initialize different devices. Detect and initialize the video adapter installed in the system that has optional ROMs.
2E	Initialize all the output devices.
31	Allocate memory for ADM module and uncompress it. Give control to ADM module for initialization. Initialize language and font modules for ADM. Activate ADM module.
33	Initialize the silent boot module. Set the window for displaying text information.
37	Displaying sign-on message, CPU information, setup key message, and any specific OEM information.
38	Initialize different devices through DIM. See DIM Code Checkpoints section in document for more information. USB controllers are initialized at this point.

39	Initialize DMAC-1 & DMAC-2.
3A	Initialize RTC date/time.
3B	Test for total memory installed in the system. Also, check for DEL keys to limit memory test. Display total memory in the system.
3C	Mid POST initialization of chipset registers.
40	Detect different devices (Parallel ports, serial ports, and coprocessor in CPU, … etc.) successfully installed in the system and update the BDA, EBDA…, etc.
52	Update CMOS memory size from memory found in memory test. Allocate memory for Extended BIOS Data Area from base memory. Program the memory hole or any kind of implementation that needs adjustment in system RAM size if needed.
60	Initialize NUM-LOCK status and program the KBD typematic rate.
75	Initialize Int-13 and prepare for IPL detection.
78	Initialize IPL devices controlled by BIOS and optional ROMs.
7C	Generate and write contents for ESCD in NVRam.
84	Log errors encountered during POST.
85	Display errors to user and get user's error response.
87	Execute BIOS setup if needed/requested. Check boot password if installed.
8C	Late POST initialization of chipset registers.
8D	Build ACPI tables (if ACPI is supported)
8E	Program the peripheral parameters. Enable/Disalbe NMI as selected.
90	Initialization of system management interrupted by invoking all handlers.
A1	Line-up work needed before booting to OS.
A2	Take care of runtime image preparation for different BIOS modules. Fill the free area in F000h segment with 0FFh. Initialize the Microsoft IRQ Routing Table. Prepare the runtime language module. Disable the system configuration display if needed.

A4	Initialize runtime language module. Display boot option's popup menu.
A7	Display the system configuration screen if enabled. Initialize the CPU's before boot, which includes the programming of the MTRR's.
A9	Wait for user input at config display if needed.
AA	Uninstall POST INT1Ch vector and INT09h vector.
AB	Prepare BBS for Int 19 boot. Init MP tables.
AC	End of POST initialization of chipset registers. De-initialize the ADM module.
B1	Save system context for ACPI. Prepare CPU for OS boot, including final MTRR values.
00	Pass control to OS Loader (typically INT19h).

3.8.4 DIM Code Checkpoints

The Device Initialization Manager (DIM) gets control at various times during BIOS POST tries to initialize different system buses. The following table describes the main checkpoints where the DIM module is accessed *(Note)*:

Checkpoint	Description
2A	Initialize different buses and perform the following functions: Reset, Detect, and Disable (function 0); Static Device Initialization (function); Boot Output Device Initialization (function 2). Function 0 disables all device nodes, PCI devices, and PnP ISA cards. It also assigns PCI bus numbers. Function 1 initializes all static devices that include manual configured onboard peripherals, memory and I/O decode windows in PCI-PCI bridges, and non-compliant PCI devices. Static resources are also reserved. Function 2 searches for and initializes any PnP, PCI, or AGP video devices.
38	Initialize different buses and perform the following functions: Boot Input Device Initialization (function 3); IPL Device Initialization (function 4); General Device Initialization (function 5). Function 3 searches for and configures PCI input devices and detects if system has standard keyboard controller. Function 4 searches for and configures all PnP and PCI boot devices. Function 5 configures all onboard peripherals that are set as automatic configuration and configures all remaining PnP and PCI devices.

While controlling in the different functions, additional checkpoints are output to port 80h as a word value to identify the routines under execution. The low byte value indicates the main POST Code Checkpoint. The high byte is divided into two nibbles and contains two fields. The details of the high byte of these checkpoints are as follows:

HIGH BYTE XY

The upper nibble “X” indicates the function number that is being executed. “X” can be from 0 to 7.

0 = func#0, disable all devices on the BUS concerned.
2 = func#2, output device initialization on the BUS concerned.

3 = func#3, input device initialization on the BUS concerned.
4 = func#4, IPL device initialization on the BUS concerned.
5 = func#5, general device initialization on the BUS concerned.
6 = func#6, error reporting for the BUS concerned.
7 = func#7, add-on ROM initialization for all BUSes.
8 = func#8, BBS ROM initialization for all BUSes.

The lower nibble 'Y' indicates the BUS on which the different routines are being executed. 'Y' can be from 0 to 5.

0 = Generic DIM (Device Initialization Manager).
1 = On-board System devices.
2 = ISA devices.
3 = EISA devices.
4 = ISA PnP devices.
5 = PCI devices.

3.8.5 ACPI Runtime Checkpoints

ACPI checkpoints are displayed when an ACPI capable operating system either enters or leaves a sleep state. The following table describes the type of checkpoints that may occur during ACPI sleep or wake events *(Note)*:

Checkpoint	Description
AC	First ASL check point. Indicates the system is running in ACPI mode.
AA	System is running in APIC mode.
01, 02, 03, 04, 05	Enter sleep state S1, S2, S3, S4, or S5.
10, 20, 30, 40, 50	Wake from sleep state S1, S2, S3, S4, or S5.

Note:
Please note that checkpoints may differ between different platforms based on system configuration. Checkpoints may change due to vendor requirements, system chipset or optional ROMs from add-in PCI devices.

This page is intentionally left blank.

Appendix

Appendix A. Watchdog Timer (WDT) Setting

WDT is widely used for industrial application to monitor CPU activities. The application software depends on its requirement to trigger WDT with adequate timer setting. Before WDT timeout, the functional normal system will reload the WDT. The WDT never time-out for a normal system. The WDT will not be reloaded by an abnormal system, then WDT will time-out and auto-reset the system to avoid abnormal operation.

This computer supports 255 levels watchdog timer by software programming I/O ports.

Below is an program example to disable and load WDT.

Sample Codes:

```
/*---------------------------------------------------------------------------*/
#include <math.h>
#include <stdio.h>
#include <dos.h>

int WDTCount;

int main(void)
{
          unsigned char       iCount;

          printf("WDT Times ( 1 ~ 255 ) : ");
          scanf("%d",&iCount);
          printf("\n");

          WDT_Start(iCount);

          return 0;
}

void WDT_Start(int iCount)
{
          outportb(0x66,0xBA);                    /* Enable Watch Dog */
          delay(1000);

          WDTCount = iCount;
          outportb(0x62, WDTCount);               /* Number is Watch Dog Down count number */
          delay(1000);

          outportb(0x62, 0x00);                   /* Minute is 1 count unit by minute */
                                                  /* Minute is 0 count unit by second */
}

void WDT_Stop(void)
{
          outportb(0x66,0xBB);                    /* Disable Watch Dog */
}

void WDT_Clear(void)
{
          outportb(0x66,0xBA);                    /* Enable Watch Dog */
          delay(1000);

          outportb(0x62, WDTCount);               /* Number is Watch Dog Down count number */
          delay(1000);
```

```
        outportb(0x62, 0x00);                          /* Minute is 1 count unit by minute
*/
                                                                                    /   *
Minute is 0 count unit by second */
}
```

Appendix B. Digital I/O Setting

Digital I/O can read from or write to a line or an entire digital port, which is a collection of lines. This mechanism helps users achieve various applications such as industrial automation, customized circuit, and laboratory testing. Take the source code below that is written in C for the digital I/O application example.

Sample Codes:

```
/*-----   Include Header Area -----*/
#include "math.h"
#include "stdio.h"
#include "dos.h"

#define    sioIndex  0x2E
#define    sioData   0x2F

/*-----   routing, sub-routing -----*/
void main()
{
    int iData;

    SioGPIOMode(0x0F);
    delay(2000);

    SioGPIOData(0x05);
    delay(2000);

    iData = SioGPIOStatus();
    printf(" Input : %2x \n",iData);
    delay(2000);

    SioGPIOData(0x0A);
    delay(2000);

    iData = SioGPIOStatus();
    printf(" Input : %2x \n",iData);
    delay(2000);
}

void SioGPIOMode(int iMode)
{
    outportb(sioIndex,0x87);                          /* Enable Super I/O */
    outportb(sioIndex,0x87);

    outportb(sioIndex,0x07);                          /* Select logic device - GPIO */
    outportb(sioData, 0x06);

    outportb(sioIndex,0x30);                          /* Enable GPIO */
    outportb(sioData, 0x01);

    outportb(sioIndex,0x88);                          /* GPIO 80~87 - Output Enable */
    outportb(sioData,iMode);

    outportb(sioIndex,0xAA);                          /* Disable Super I/O */
}

void SioGPIOData(int iData)
{
    outportb(sioIndex,0x87);                          /* Enable Super I/O */
    outportb(sioIndex,0x87);
```

```
    outportb(sioIndex,0x07);                        /* Select logic device - GPIO */
    outportb(sioData, 0x06);

    outportb(sioIndex,0x89);                        /* GPIO 80~87 - Output Data */
    outportb(sioData,iData);

    outportb(sioIndex,0xAA);                        /* Disable Super I/O */
}

int SioGPIOStatus()
{
    int iStatus = 0x00;

    outportb(sioIndex,0x87);                        /* Enable Super I/O */
    outportb(sioIndex,0x87);

    outportb(sioIndex,0x07);                        /* Select logic device - GPIO */
    outportb(sioData, 0x06);


    outportb(sioIndex,0x8A);                        /* GPIO 80~87 - Status */
    iStatus  = inportb(sioData);

    outportb(sioIndex,0xAA);                        /* Disable Super I/O */

    return iStatus;
}
```