Custom Embedded Solutions

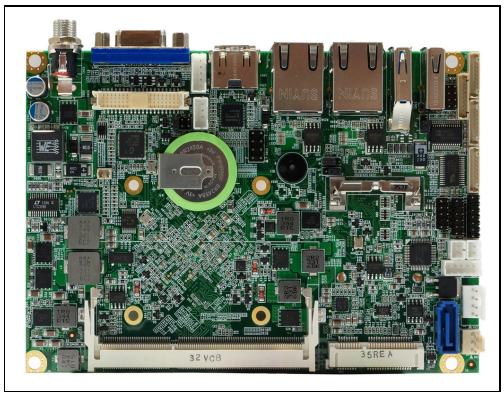
User's Manual

MB-80580

Wide temperature range 3.5" SBC supports Intel® Atom™ Processor E3800 series; operates within -40°C ~ 85°C, full-range DC input.

PL-80580

Fanless system support Intel® Atom™ Processor E3800 series operates within -10°C ~ 60°C , full-range DC input.



Ver.	Release Date	Update
1.0	2014.06	Release

Custom Embedded Solutions

Copyright

Trademark

Limitation of liability

For more information or other WIN products, please visit our website http://www.win-ent.com.

For technical supports, please send your inquiry to sales@win-ent.com.

Packing list

Before using this product make sure that the following materials have been shipped.

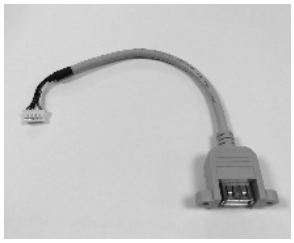
- 1 x MB-80580 3.5" SBC
- 1 x CPU cooling fan (P/N: CB-F0056-00) (Note)
- 1 x SATA cable, L/ 200mm (P/N: CB-SATA11-00)
- 1 x 12V/5V SATA power cable , L/ 150mm (P/N: CB-IPOW41-00)
- 1 x COM port, DB9 type, L/ 180mm, without bracket (P/N: CB-ICOM34-01)
- 1 x CD Utility

Note: CPU cooling fan only comes with verification purpose to ensure motherboard can be operated in a wide range temperature environment. Customers should use proper cooling solution for their application.

CB-F0056-00	CB-SATA11-00	CB-IPOW41-00	CB-ICOM34-01
			

Model Name	Description
MB-8058A	Onboard Intel® Atom™ E3845, DDR3 up to 4GB, HDMI/VGA/LVDS, 2 x Mini-PCIe, SATA, Audio, 4 x COM, USB, DC 8V~32V in
MB-8058B	Onboard Intel® Atom™ E3826, DDR3 up to 4GB, HDMI/VGA/LVDS, 2 x Mini-PCIe, SATA, Audio, 4 x COM, USB, DC 8V~32V in
MB-8058C	Onboard Intel® Atom™ E3815, DDR3 up to 4GB, HDMI/VGA/LVDS, 2 x Mini-PCIe, SATA, Audio, 4 x COM, USB, DC 8V~32V in
* If you find any items to be missing or damaged, contact your sales representative or WIN sales manager.	

Optional Accessories

Photo	Model Name	
	P/N:	CB-ICOM34-01
	Single COM port, DB9 type, L/ 180mm, without bracket	
	P/N:	CB-IUSB2B-01
	USB cable, L/ 155mm, without bracket	
	P/N:	CB-IAUD15-01
	Line-out , Line-In , Mic-In audio cable, L/ 180mm, without bracket	

Safety Information

- To prevent electrical shock hazard, disconnect the power cable from the electrical outlet before relocating the system.
- When adding or removing devices to or from the system, ensure that the power cables for the devices are unplugged before the signal cables are connected. If possible, disconnect all power cables from the existing system before adding a device.
- Before connecting or removing signal cables from the motherboard, ensure that all power cables are unplugged.
- Seek professional assistance before using an adapter or extension cord. These devices could interrupt the grounding circuit.
- Make sure that your power supply is set to the correct voltage in your area.
- If you are not sure about the voltage of the electrical outlet you are using, contact your local power company.
- If the power supply is broken, do not try to fix it by yourself. Contact a qualified service technician or your retailer.

Operational Safety

- Before installing the motherboard and adding devices, carefully read all the manuals that came with the package(s).
- Before using the product, make sure all cables are correctly connected and the power cables are not damaged. If you detect any damage, contact your dealer immediately.
- To avoid short circuits, keep paper clips, screws, and staples away from connectors, slots, sockets and circuitry.
- Avoid dust, humidity, and temperature extremes. Do not place the product in any area where it may become wet.
- Place the product on a stable surface.
- If you encounter technical problems with the product, contact a qualified service technician or your retailer.

CONTENTS

Custom Embedded Solutions

1.1 Introduction

MB-80580 is a 3.5" SBC with onboard Intel® Atom™ Processor E3800 series SOC with Intel® 2D/3D Graphics Gen7. Integrated graphics for three display options include HDMI, VGA and 24-bit dual-channel LVDS. One DDR3 SO-DIMM supports a maximum of 4GB DDR3 1333 of system memory.

There're two flexible Mini-PCIe sockets for expansion, one is Full-size type and the other one is half-size type. Both of them are support standard Mini-PCIe card for PCIe & USB (Full-size only) signal-base, such as WiFi. Besides that, A SIM card holder that could install SIM card when install a Full-size Mini-PCIe 3G module for wireless connection. For the Half-size Mini-PCIe socket, it's also support mSATA SSD as storage device.

Regarding I/O ports , MB-80580 provides plenty of connectivity , such as 1 x Intel® gigabit Ethernet provided by Intel® i210IT controller, 1 x RS232/422/485 & 3 x RS232, 3 x USB, Audio, 1 x SATA 6Gb/s, 1 x LPC pin-header that could support WIN's TPM module if customer needs to protect information.

MB-80580 that could accept wide range DC 8V ~ 32V input, a external DC locking power jack or a 4-pin internal power connector for easier power integration.

MB-80580 is a small form-factor embedded platform equip with Intel® 2D/3D Graphics Gen7 and also rich IO ports, make MB-80580suitable for a wide range of applications in digital signage, POS, kiosks, and factory automation. Fanless design are allowed for high temperature and dusty environments.

WIN offers reliable and solid products which are produced under Management System Standards: ISO9001-2000 Certificate. The certificate keeps us focused on our quality objectives of management and environmental production. Its willingness to customize standard products for meet unique customer needs makes WIN different. All ODM projects are welcome. Years of experiences enables WIN to fulfill the customer's vision, by delivering products to exact specifications. WIN's R&D team is proud of its strong engineering background. R&D professionals account for 25% of the WIN workforce. We focus on developing new products for both emerging and established markets.

For more information about OEM/ODM, please contact us :
Email: sales@win-ent.com, TEL: +1 (978) 688-2000.

1.2 Specifications

System	
From Factor	3.5" SBC
CPU	Onboard Intel® Atom Bay Trail-I, Type 3 Intel® Atom E3845 (Q/C , 1.91 GHz), 10W TDP. **(Model MB-8058A)** Intel® Atom E3826 (D/C , 1.46 GHz), 7W TDP. (**Model : MB-8058B)** Intel® Atom E3815 (S/C , 1.46 GHz), 5W TDP. (**Model MB-8058C)**
Chipset	Integrated
Memory	1 x DDR3L SO-DIMM 1066/1333 MHz up to 4GB, w/o ECC support
BIOS	AMI / Insyde SPI BIOS
SSD	1 x Half-Size Mini-PCIe support mSATA
Watchdog timer	255 levels, 1 ~ 255 sec
Expansion	1 x Full-size Mini-PCIe socket w/SIM slot. With PCIe X1 & USB signal. 1 x Half-Size Mini-PCIe socket. With PCIe X1 / SATA
Board Size	146mm x 101mm
Operating Temp.	**-40℃~85℃**
Storage Temp.	-20℃~80℃
Operating Hum.	10%~90% (non-condensing)

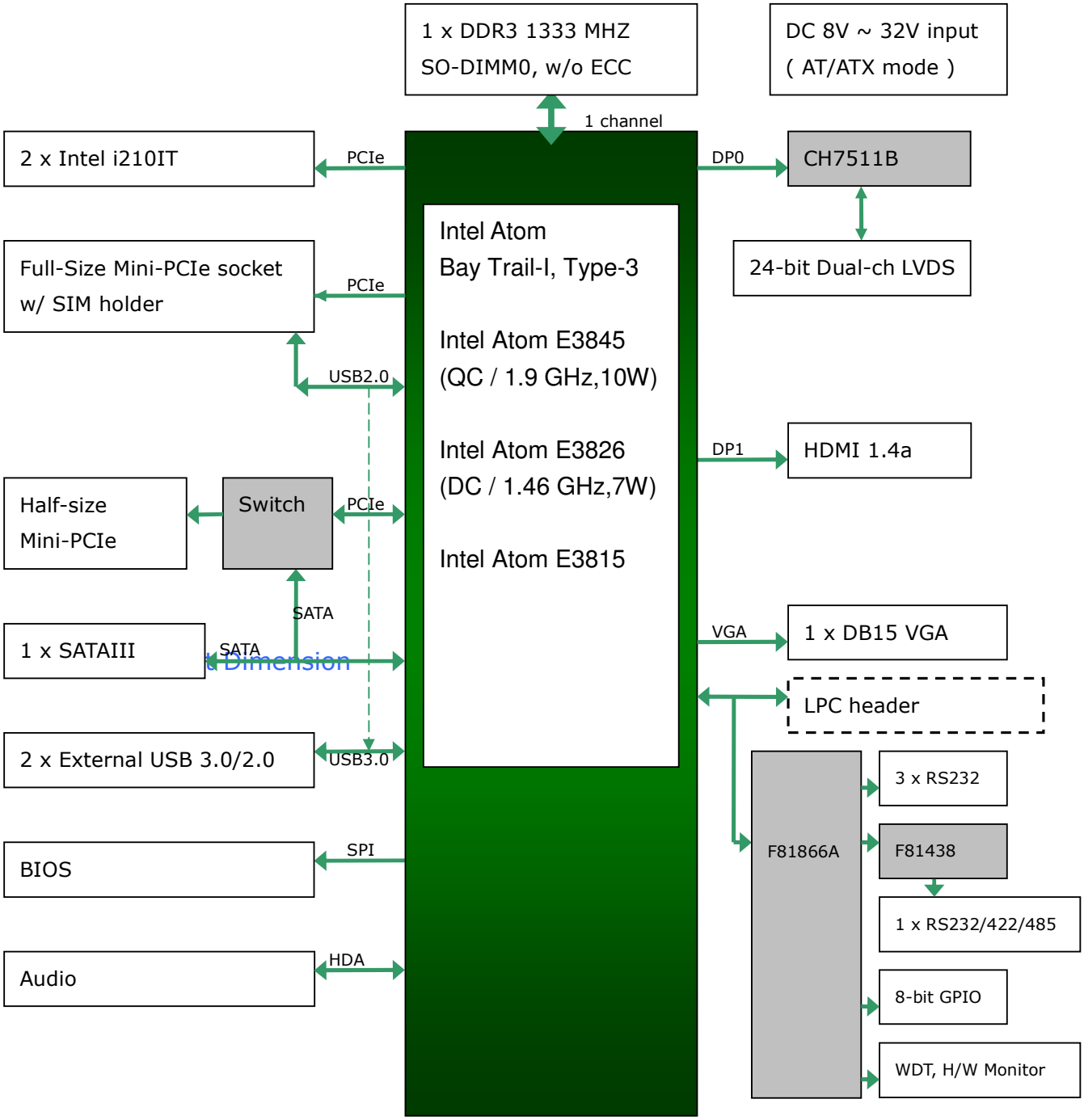
Display	
Chipset	Intel® Atom SOC Integrated
Display interface	1 x external HDMI 1.4a
	1 x external VGA
	1 x internal 24-bit Dual Channel LVDS. (12V/5V/3.3V power select by jumper)

I/O	
Series Port	1 x RS232/422/485, 3 x RS232 (Pin-header)
SATA	1 x SATAII-300
USB	1 x USB2.0 (External on rear IO), 1 x USB3.0 (External on rear IO), 1 x USB2.0 (internal pin header)
Ethernet	2 x Intel® I210IT Gigabit Ethernet controller
Audio	Onboard Pin-Header (Line-In/Out , Mic-In)
Digital I/O	8-bit GPIO interface (pitch : 2.0)
LPC	1 x LPC header for Optional TPM module
Others	1 x 4-pin 12V/5V SATA power connector 1 x LVDS backlight header

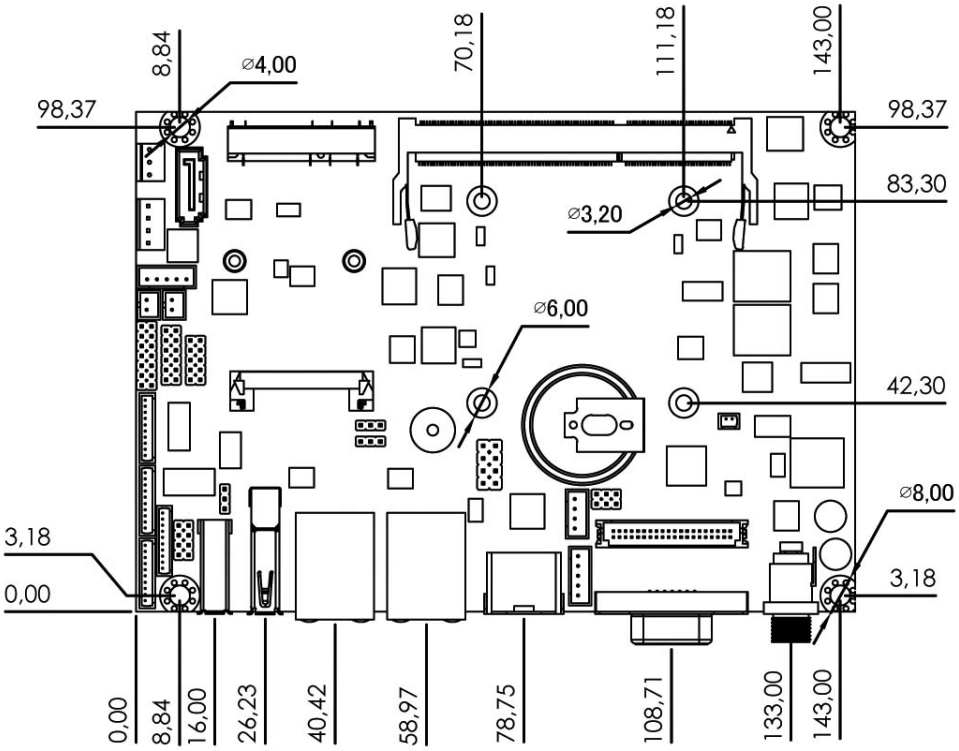
Power		
Power in	Wide range DC 8V ~ 32V input (AT/ATX mode by jumper select). Jack connector Co-lay P4 4-pin . (Default : DC Jack connector)	DC

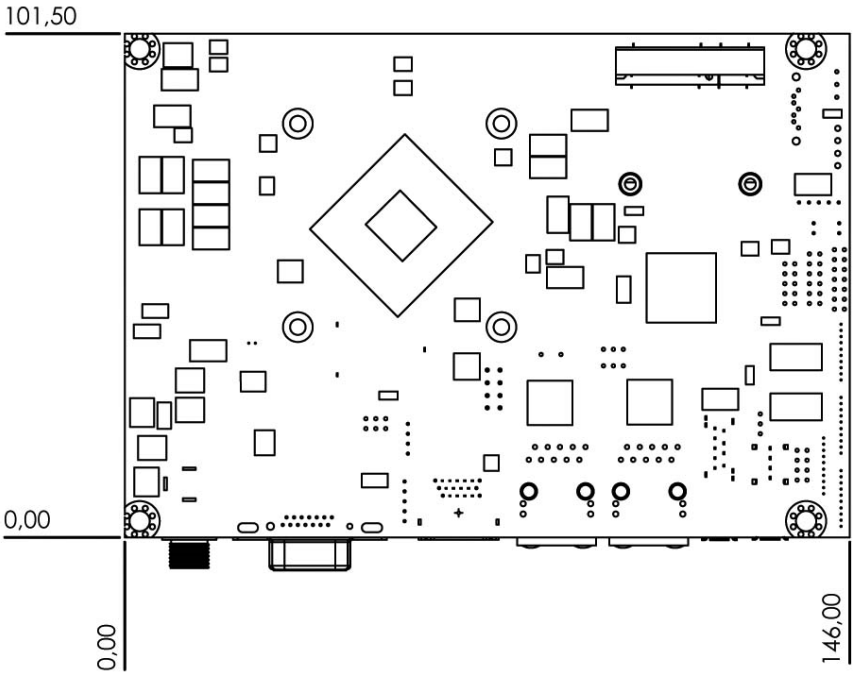
Note: Specifications and photos subject to change without prior notice.

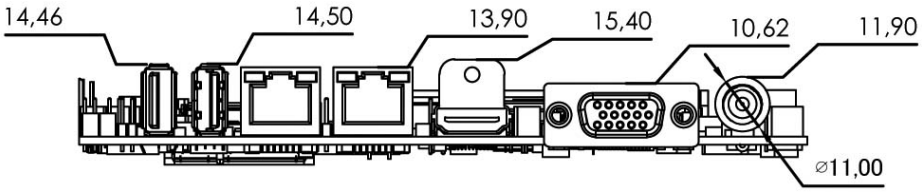
1.3 Block Diagram


1 x DDR3 1333 MHZ
SO-DIMM0, w/o ECC
DC 8V ~ 32V input
(AT/ATX mode)
1 channel
2 x Intel i210IT
PCIe
DP0
CH7511B
Intel Atom
Bay Trail-I, Type-3
Intel Atom E3845
(QC / 1.9 GHz,10W)
Intel Atom E3826
(DC / 1.46 GHz,7W)
Intel Atom E3815
24-bit Dual-ch LVDS
Full-Size Mini-PCIe socket
w/ SIM holder
PCIe
USB2.0
DP1
HDMI 1.4a
Half-size
Mini-PCIe
Switch
PCIe
SATA
VGA
1 x DB15 VGA
1 x SATAIII
SATA
LPC header
2 x External USB 3.0/2.0
USB3.0
F81866A
3 x RS232
F81438
1 x RS232/422/485
BIOS
SPI
8-bit GPIO
Audio
HDA
WDT, H/W Monitor

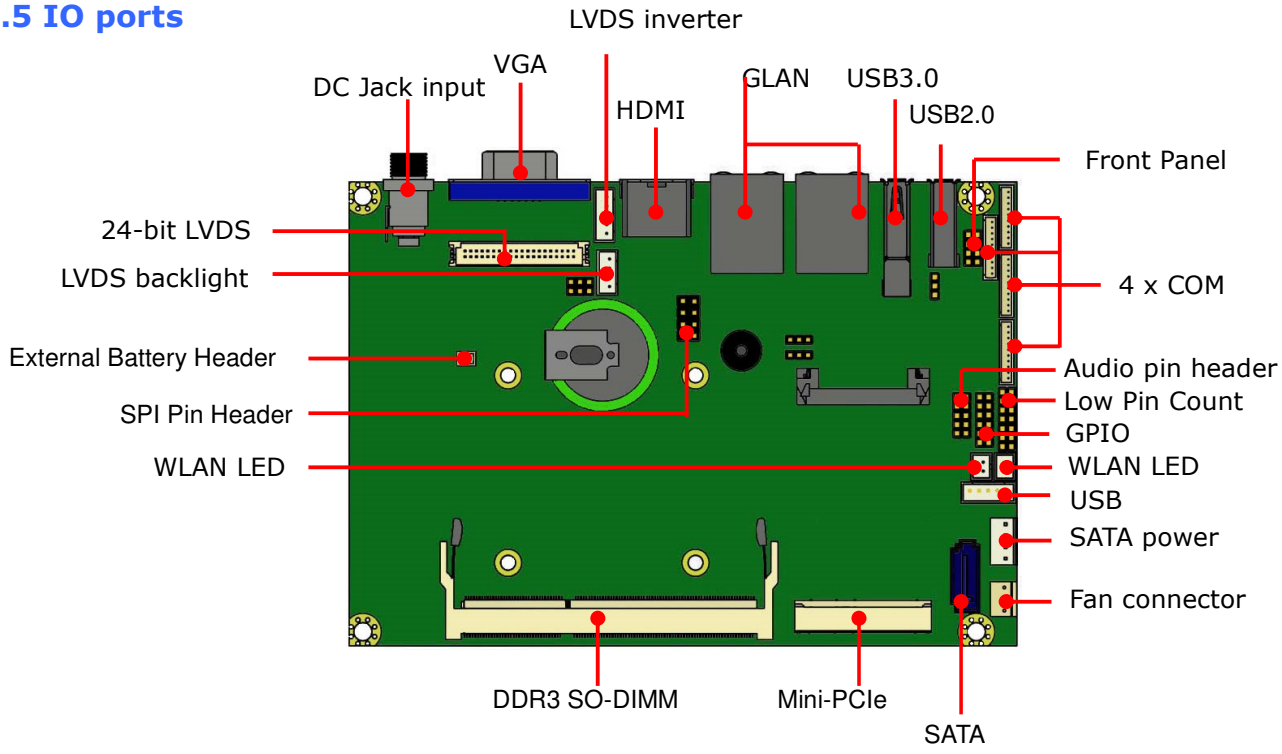
1.4 Board Layout / Dimensions

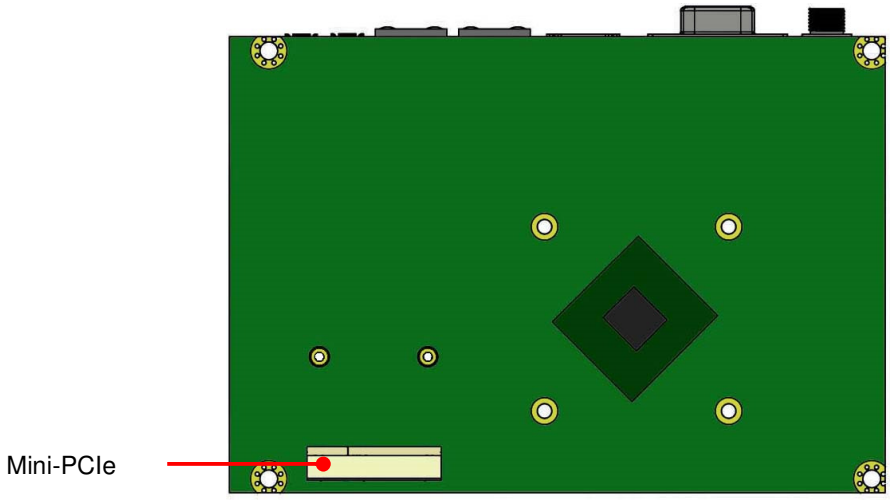


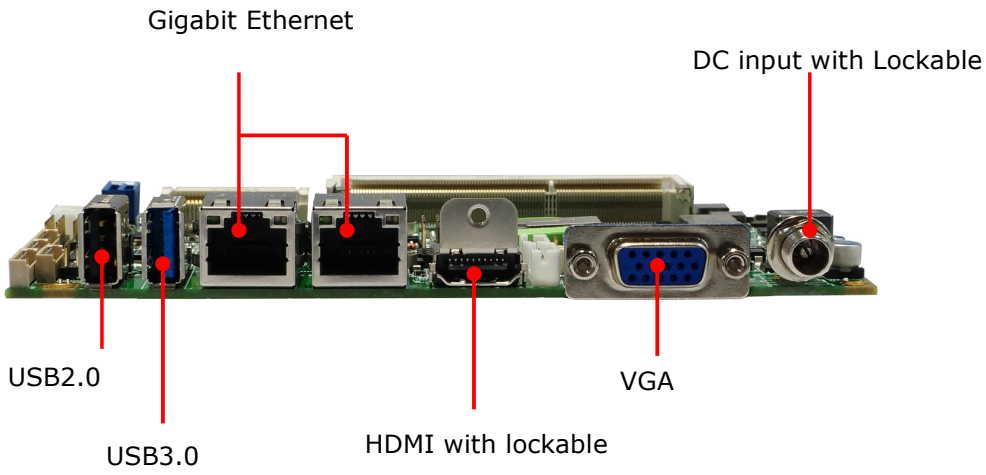



14,46
14,50
13,90
15,40
10,62
11,90
⌀11,00

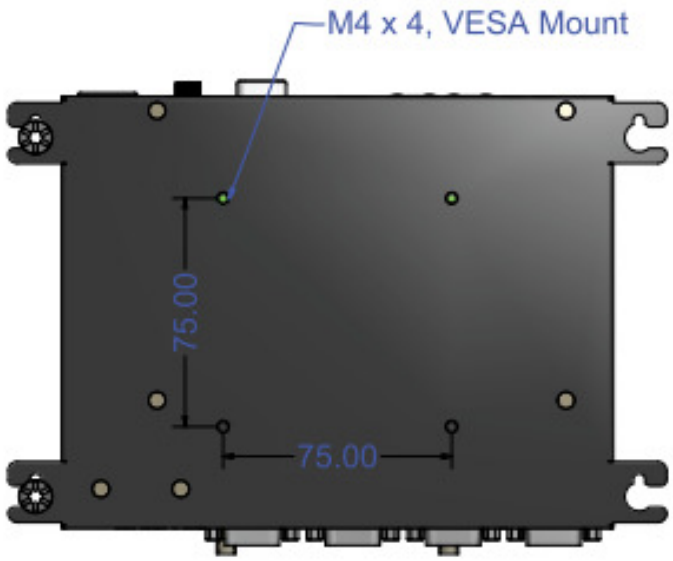
1.5 IO ports

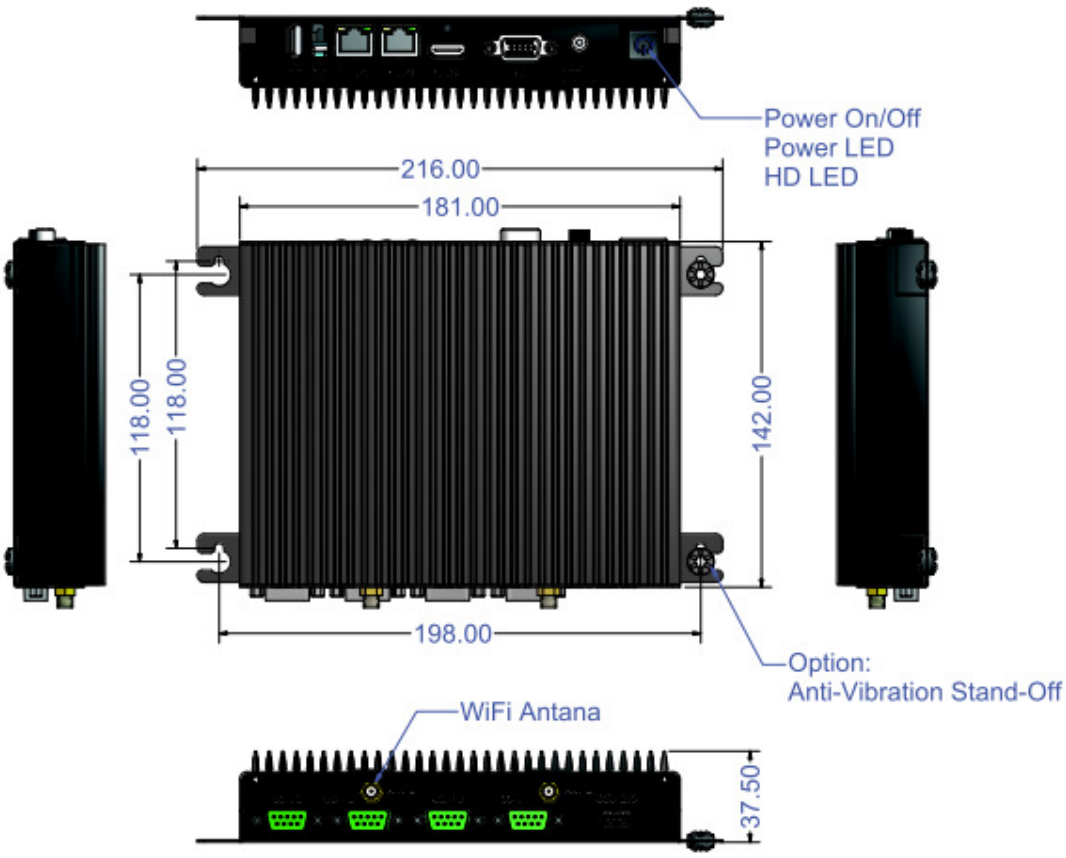

LVDS inverter
VGA
DC Jack input
HDMI
GLAN
USB3.0
USB2.0
Front Panel
24-bit LVDS
LVDS backlight
4 x COM
External Battery Header
Audio pin header
SPI Pin Header
Low Pin Count
GPIO
WLAN LED
WLAN LED
USB
SATA power
Fan connector
DDR3 SO-DIMM
Mini-PCIe
SATA

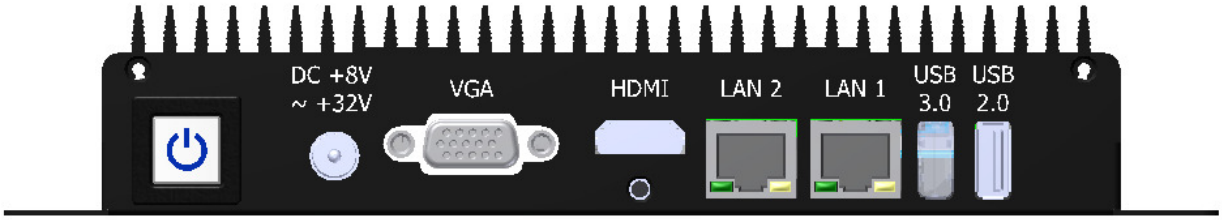

Mini-PCIe

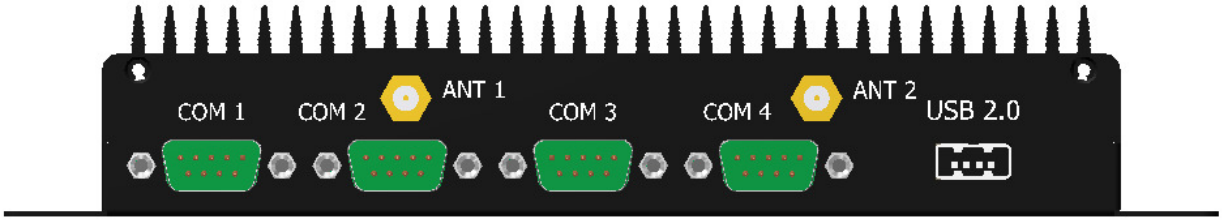

Gigabit Ethernet
DC input with Lockable
USB2.0
USB3.0
HDMI with lockable
VGA

1.6 System Layout and Dimensions (units in mm)

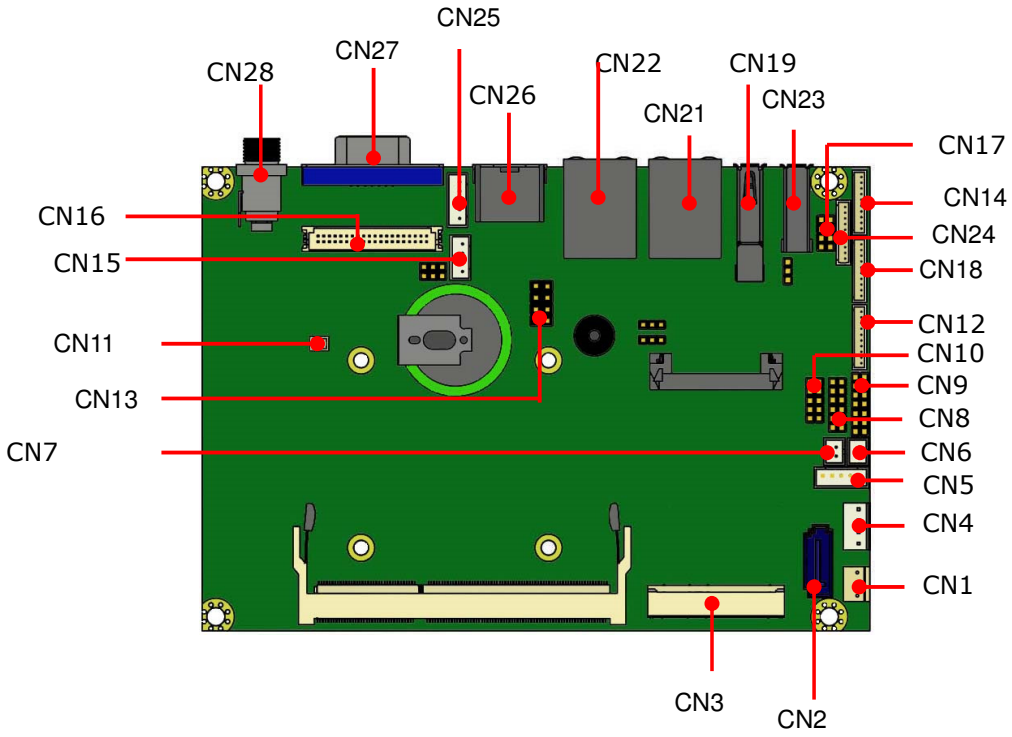


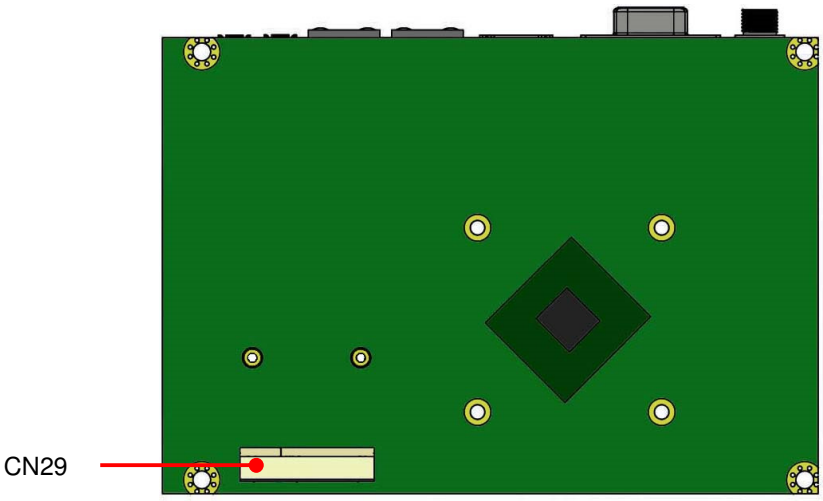

Power On/Off
Power LED
HD LED
216.00
181.00
118.00
118.00
142.00
198.00
Option:
Anti-Vibration Stand-Off
WiFi Antana
37.50


DC +8V
~ +32V
VGA
HDMI
LAN 2
LAN 1
USB 3.0
USB 2.0


COM 1
COM 2
ANT 1
COM 3
COM 4
ANT 2
USB 2.0

2.1 Location of onboard connectors

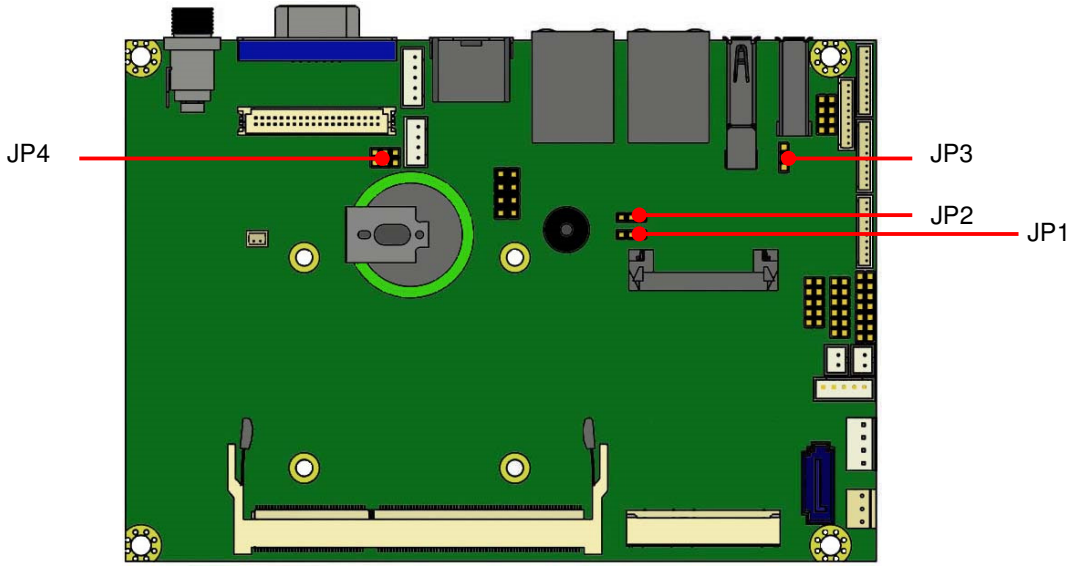




Label	Function
CN1	FAN Connector
CN2	SATA Connector
CN3	MINI-PCIE Socket.(Full Size)
CN4	SATA Power Connector
CN5	USB Port(Internal)
CN6	WLAN LED Connector (For CN3)
CN7	WLAN LED Connector (For CN29)
CN8	GPIO Pin Header
CN9	LPC Pin Header
CN10	Audio Pin Header
CN11	External Battery Connector
CN12	COM2 Connector
CN13	SPI Pin Header
CN14	COM4 Connector
CN15	LVDS Pannel Backlight Adjustment Connector
CN16	LVDS Panel Connector

Label	Function
CN17	Front Panel Pin Header
CN18	COM3 Connector
CN19	USB 3.0 Connector
CN20	DC-IN Power Connector(Option)
CN21	LAN1 Connector
CN22	LAN2 Connector
CN23	USB 2.0 Connector
CN24	COM1 Connector
CN25	LVDS Pannel Inverter Power Connector
CN26	HDMI Connector
CN27	VGA Connector
CN28	DC-IN Power Jack
CN29	MINI-PCIE Socket.(Half Size)

2.2 Location of onboard jumpers



Label	Function
JP1	LVDS Panel Backlight Control Mode Select (1-2 short : PWM Mode ; 2-3 short : DC Mode)
JP2	Clear CMOS (1-2 short : Hold CMOS ; 2-3 short : Clear CMOS)
JP3	AT/ATX MODE SELECT (1-2 short : AT ; 2-3 short : ATX)
JP4	LVDS Panel Power Level Select (1-2 short : +3.3V ; 3-4 short : +5V ; 5-6 short : +12V)

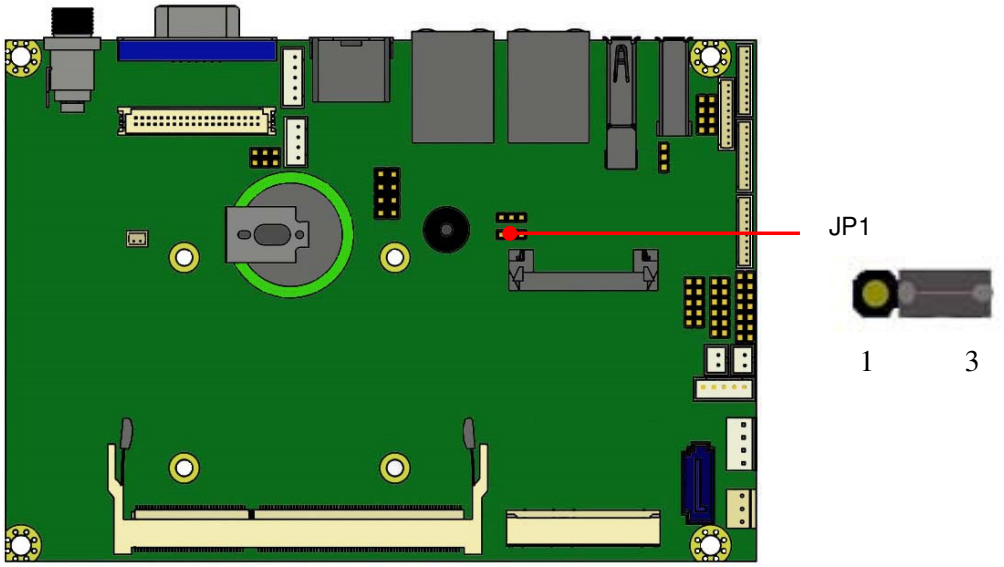
* Default setting

2.3 The function list of onboard jumpers settings

- 2.3.1: JP1 for LVDS panel backlight control mode select

JP1 : 1 x 3 header , pitch 2.0 mm	
Closed Pin	Result
1-2	PWM mode
2-3 *	DC mode

* Default setting

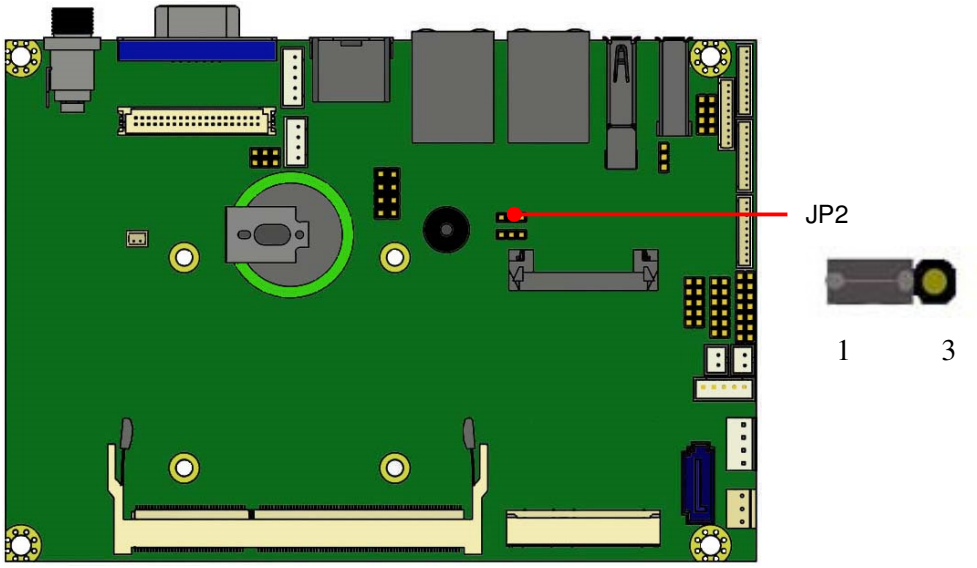


- 2.3.2: JP2 for Clear CMOS

If you want to clean the CMOS data, set jumper to 2-3 just for few seconds then move jumper back to 1-2 pin.

JP2 : 1 x 3 header , pitch 2.0 mm	
Closed Pin	Result
1-2 *	Normal
2-3	Clear CMOS

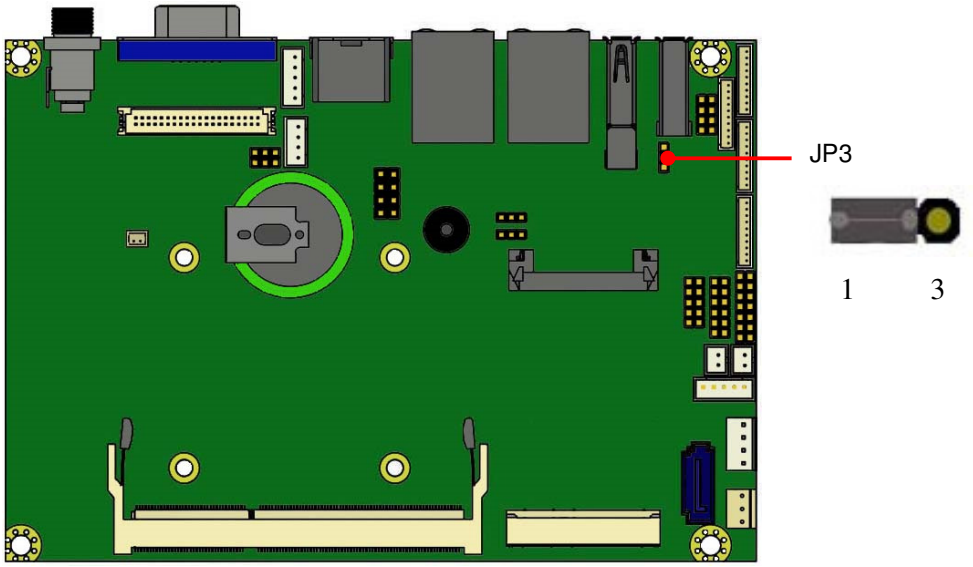
* Default setting



- 2.3.3 : JP3 for AT/ATX mode select

JP3 : 1 x 3 header , pitch 2.0 mm	
Closed Pin	Result
1-2 *	AT mode
2-3	ATX mode

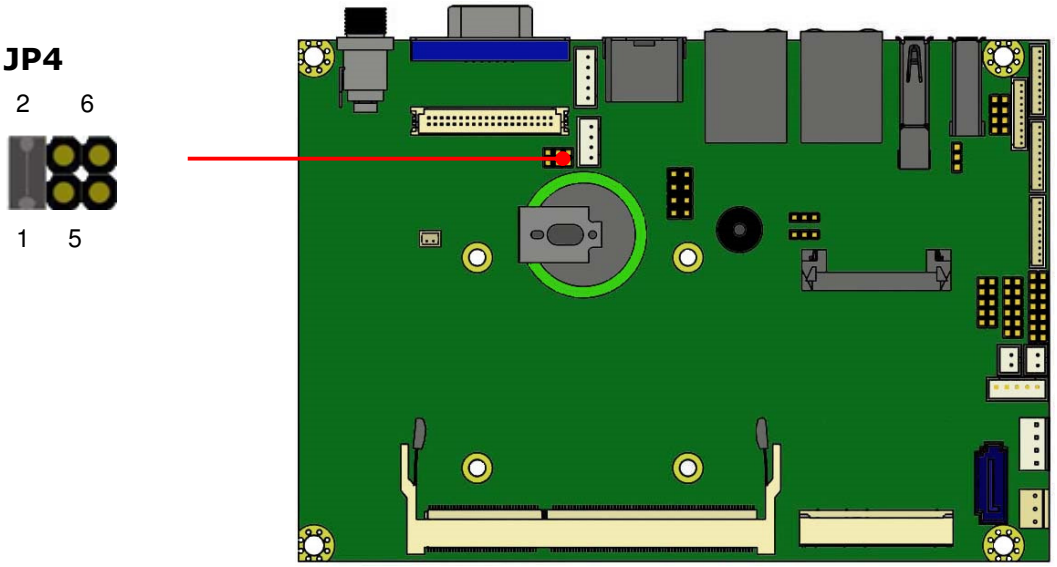
* Default setting



- 2.3.4 : JP4 for LVDS panel power level select

JP4 : 2 x 3 header , pitch 2.0 mm	
Closed Pin	Result
1-2 *	+3.3V for LVDS_VCC
3-4	+5V for LVDS_VCC
5-6	+12V for LVDS_VCC

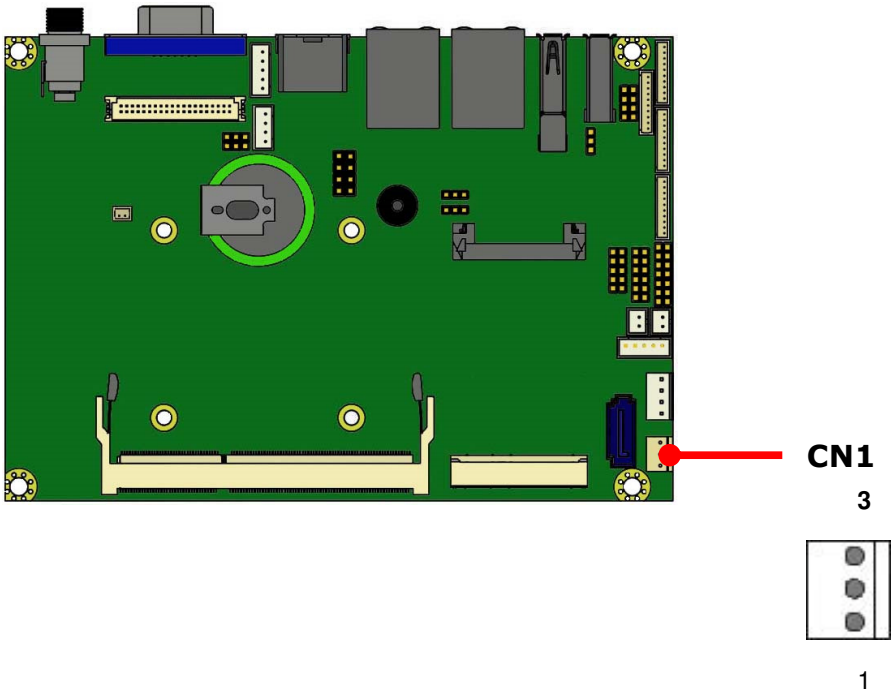
* Default setting



2.4 The pin define of onboard pin header

- 2.4.1 : CN1 for 3-pin Fan connector

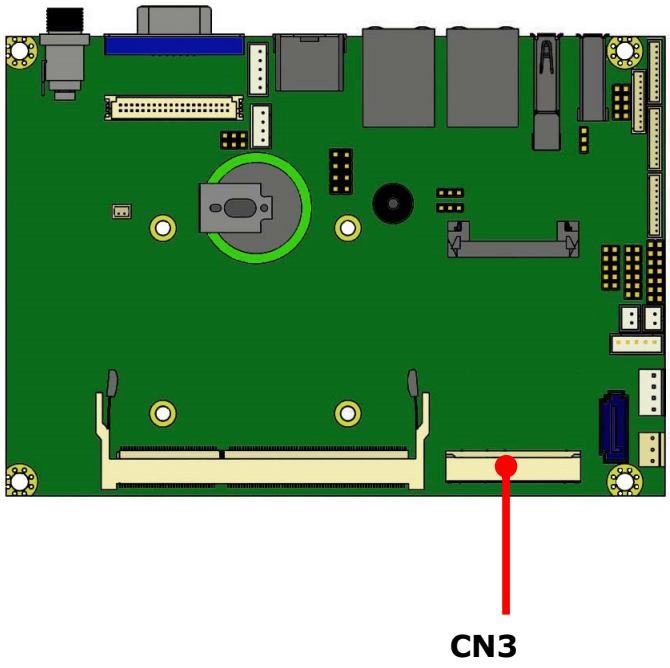
CN1: 1 x 3 wafer			
Pin	**Signal**	**Pin**	**Signal**
1	GND	2	+12V
3	Sensor		



- 2.4.2 : CN3 for Full-size Mini-PCIe socket

Note: Full-size Mini-PCIe only supports PCIe + USB signal,
Does not support SATA signal, such as mSATA SSD, etc.

Pin	Signal	Pin	Signal
1	WAKE	27	GND
2	+3.3V AUX	28	+1.5V
3	N/C	29	GND
4	GND	30	SMBCLK
5	N/C	31	PETN0
6	+1.5V	32	SMBDATA
7	CLKREQ	33	PETP0
8	UIM_PWR	34	GND
9	GND	35	GND
10	UIM_DATA	36	USB_D-
11	REFCLK-	37	GND
12	UIM_CLK	38	USB_D+
13	REFCLK+	39	+3.3V AUX
14	UIM_RESET	40	GND
15	GND	41	+3.3V AUX
16	UIM_VPP	42	LED_WWAN
17	UIM_C8	43	GND
18	GND	44	LED_WLAN
19	UIM_C4	45	N/C
20	W_Disable	46	N/C
21	GND	47	N/C
22	PERST	48	+1.5V
23	PERN0	49	N/C
24	+3.3V AUX	50	GND
25	PERP0	51	N/C
26	GND	52	+3.3V AUX

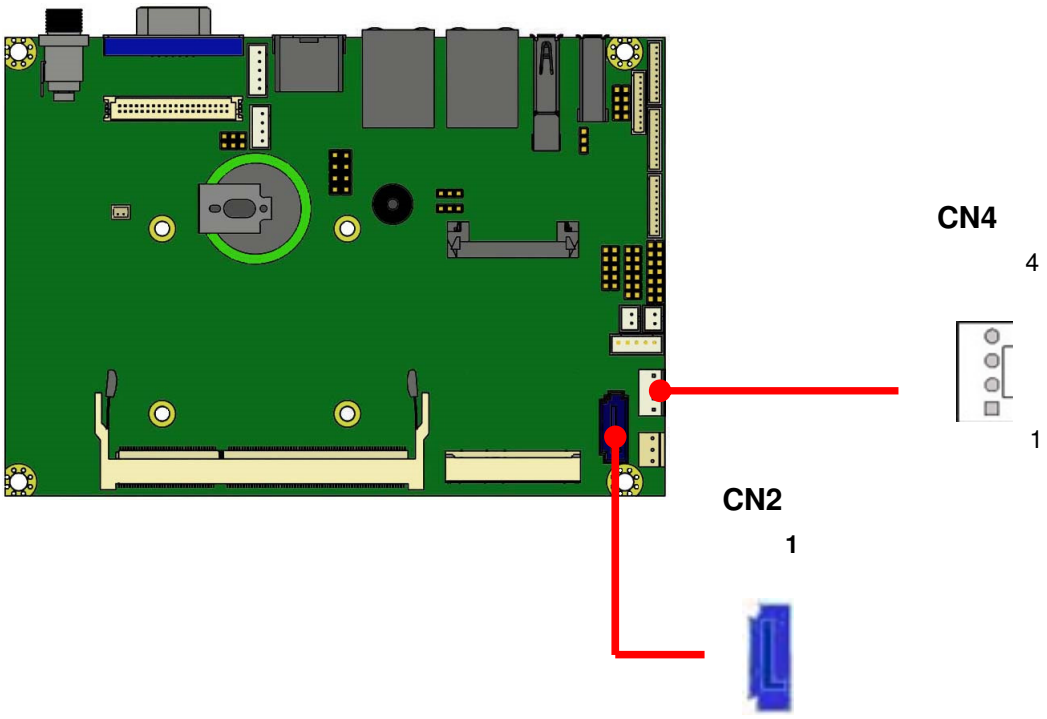


- 2.4.3 : CN2 & CN4 for SATA 3.0 connector and SATA power connector

CN2 : SATA 3.0 connector			
Pin	**Signal**	**Pin**	**Signal**
1	GND	2	SATA_TX_P0
3	SATA_TX_N0	4	GND
5	SATA_RX_N0	6	SATA_RX_P0
7	GND		

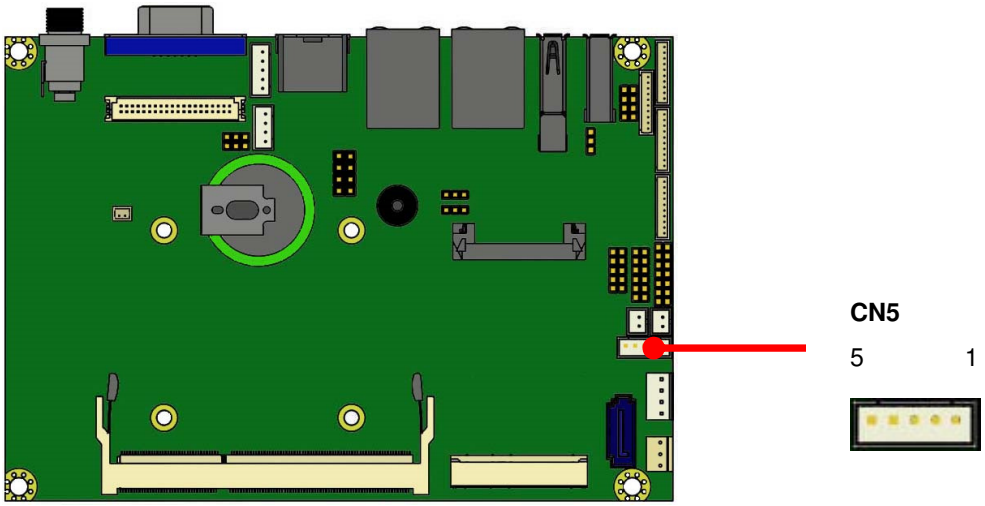
CN4 : 4-pin wafer for SATA power connector			
Pin	**Signal**	**Pin**	**Signal**
1	+12V	2	GND
3	GND	4	+5V

Note: Maximum output current 12V/1A, 5V/1A.



- 2.4.4 : CN5 for USB2.0 internal Pin Header

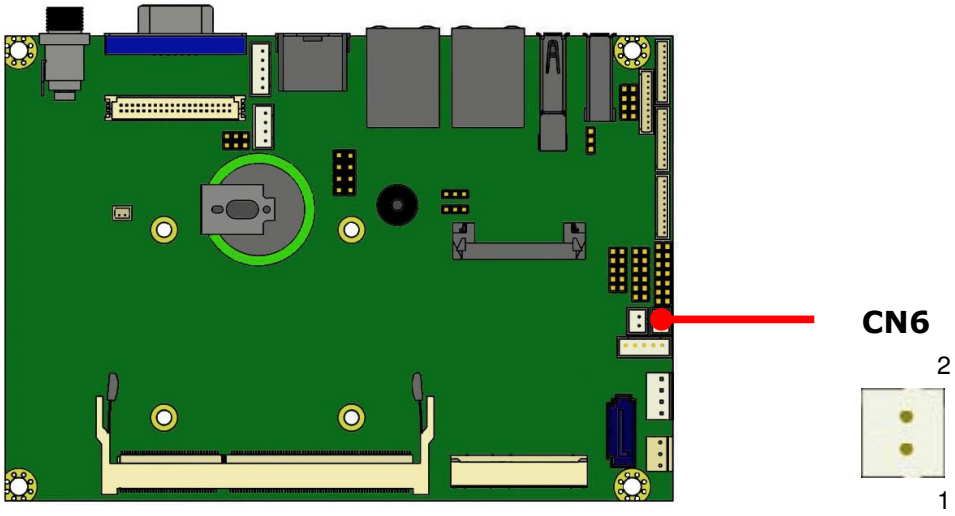
CN5 : USB2.0 internal Pin Header	
Pin	**Signal**
1	+5V
2	USB D-
3	USB D+
4	GND
5	GND



- 2.4.5 : CN6 for WLAN LED connector for Full-size Mini-PCIe socket

CN6 : 1 x 2 wafer			
Pin	**Signal**	**Pin**	**Signal**
1	LED-	2	+3.3V

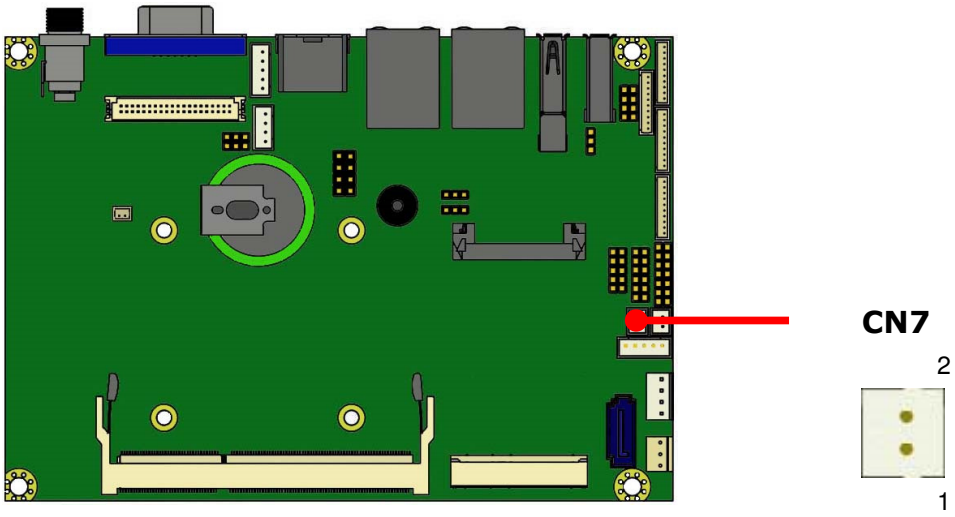
Note: Supports WWAN / wireless LAN signal



- 2.4.6 : CN7 for WLAN LED connector for Half-size Mini-PCIe socket

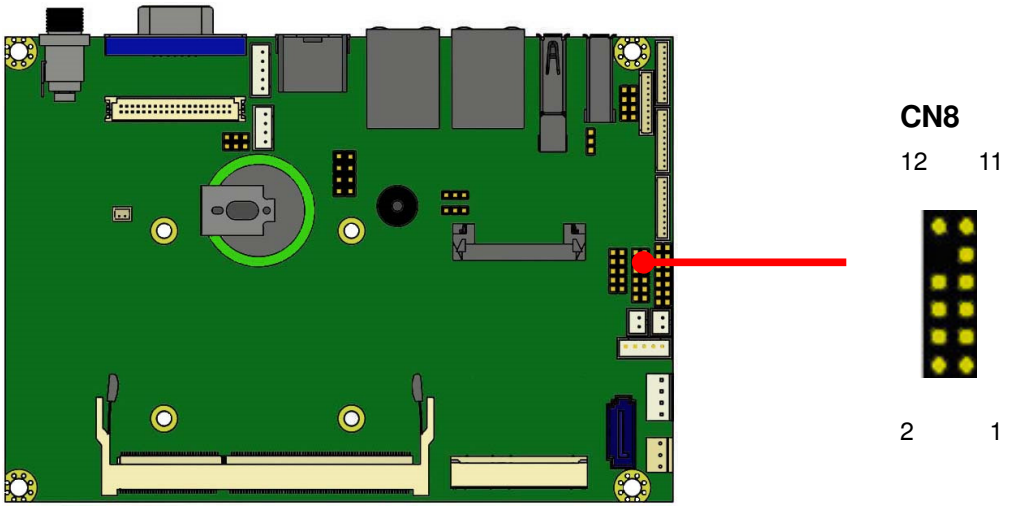
CN7 : 1 x 2 wafer			
Pin	**Signal**	**Pin**	**Signal**
1	+3.3V	2	LED-

Note: Supports wireless LAN signal only



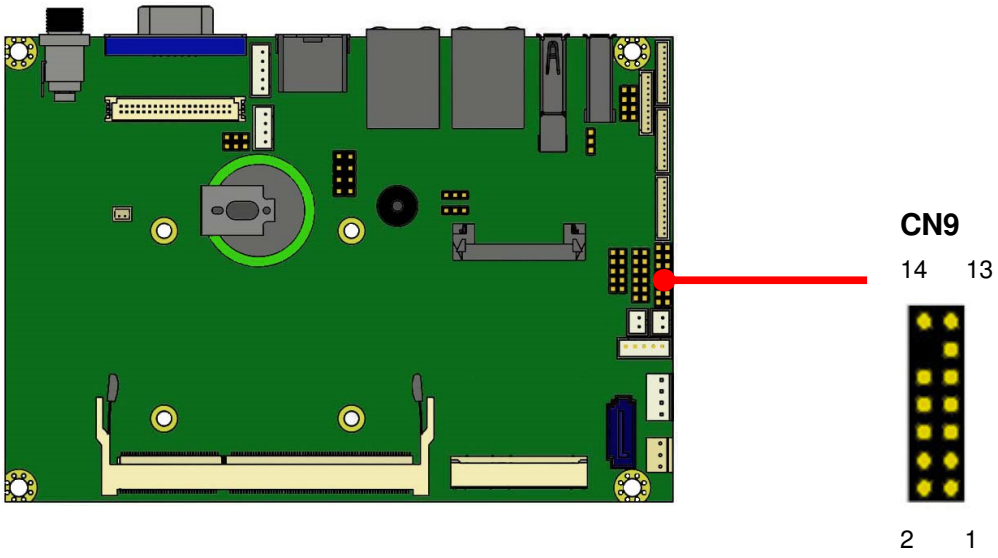
- 2.4.7 : CN8 for GPIO pin header

CN8 : 2 x 6 header , pitch 2.0 mm			
Pin	**Signal**	**Pin**	**Signal**
1	+3.3V	2	GPIO0
3	GPIO1	4	GPIO2
5	GPIO3	6	GPIO4
7	GPIO5	8	GPIO6
9	GPIO7	10	Key
11	+5V	12	GND



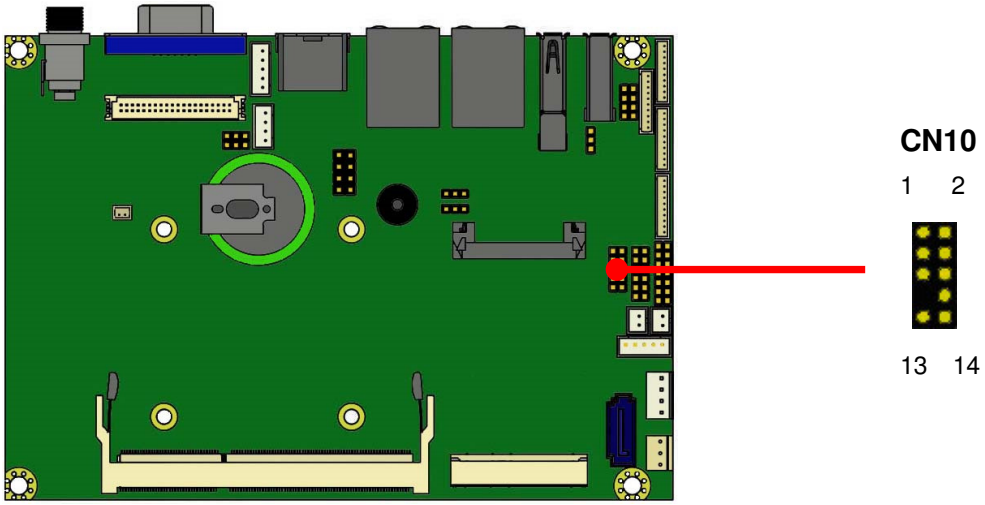
- 2.4.8 : CN9 for Low Pin Count pin header

CN9 : 2 x 7 header , pitch 2.0 mm			
Pin	**Signal**	**Pin**	**Signal**
1	+3.3V	2	LAD0
3	LAD1	4	LAD2
5	LAD3	6	LFRAME_N
7	PLTRST_N	8	+5V
9	LPC_Clock	10	GND
11	GND	12	Key
13	SERIRQ	14	LPC_DREQ



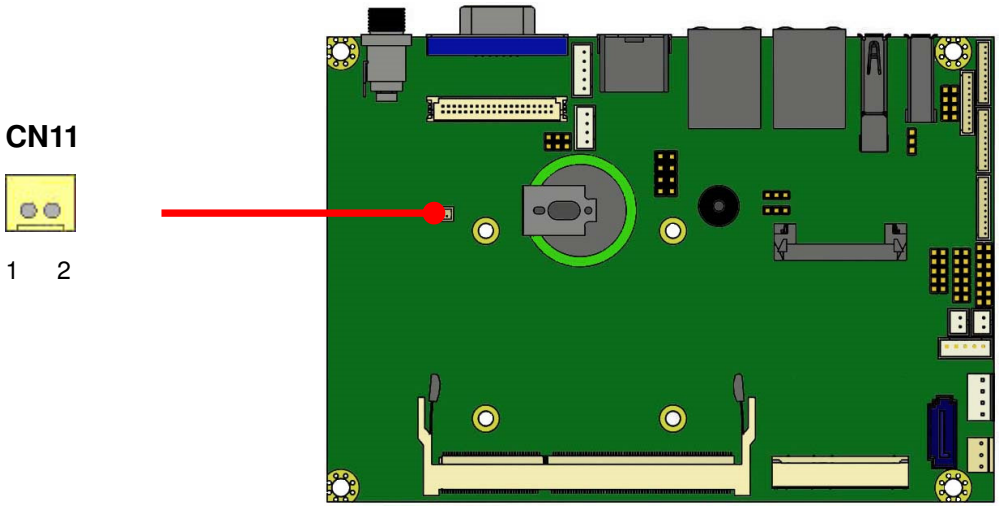
- 2.4.9 : CN10 for Audio pin header

CN10: 2 x 5 header , Pitch 2.0mm			
Pin	**Signal**	**Pin**	**Signal**
1	LINE1-L	2	LINE1-R
3	GND	4	GND
5	MIC1_L	6	MIC1_R
7	key	8	GND
9	FRONT_L	10	FRONT_R



- 2.4.10 : CN11 for Battery connector

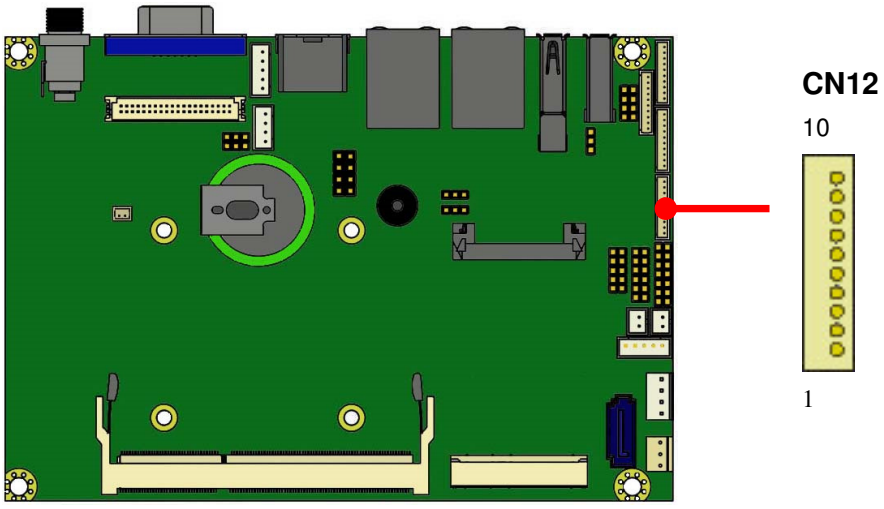
CN11 : 1 x 2 header , pitch 2.0 mm			
Pin	**Signal**	**Pin**	**Signal**
1	+3V	2	GND



- 2.4.11 : CN12 for COM 2 , RS232/422/485

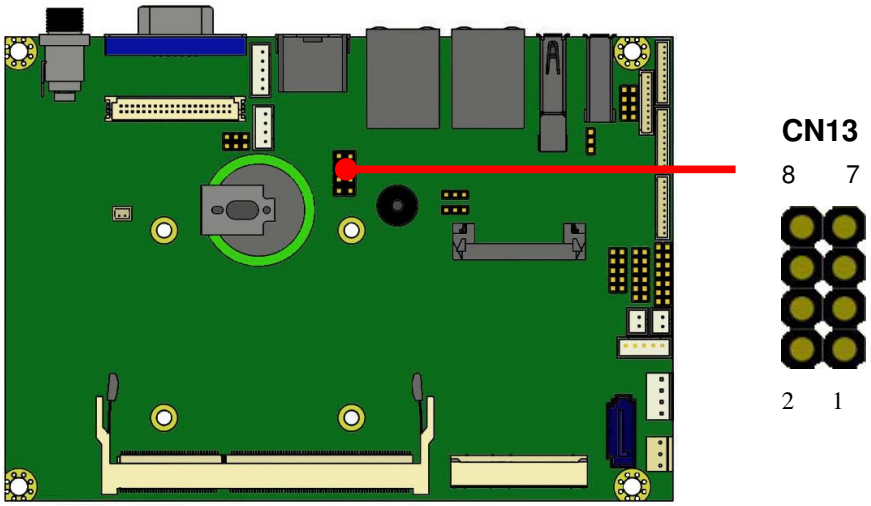
Note: COM2 RS-232/422/485 can be set by BIOS setting. Default is RS-232.

CN12 : Wafer 1 x 10 header, pitch 1.25 mm, connector type : YIMTEX 501MW1*10STR			
Pin	**RS232 mode**	**RS422 mode**	**RS485 mode**
1	DCD, Data carrier detect	TXD-	TXD-
2	DSR, Data set ready		
3	RXD, Received Data	TXD+	TXD+
4	RTS, Request to send		
5	TXD, Transmitted Data	RXD+	
6	CTS, Clear to sent		
7	DTR, Data terminal ready	RXD-	
8	RI, Ring indicator		
9	GND		
10	N/C		



- 2.4.12 : CN13 for SPI pin header

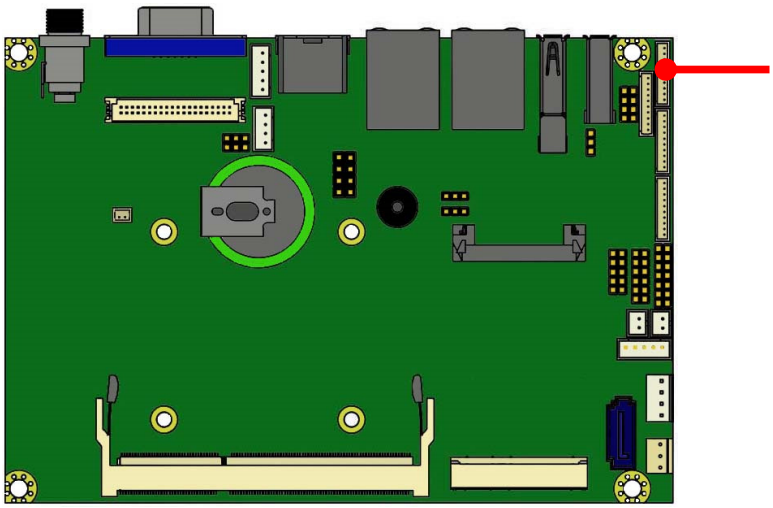
CN13 : 2 x 4 header , pitch 2.00 mm			
Pin	**Signal**	**Pin**	**Signal**
1	+1.8V	2	GND
3	SPI_CS	4	SPI_CLK
5	SPI_MISO	6	SPI_MOSI
7	N/A	8	Flash IO



- 2.4.13 : CN14 for COM 4 , RS232/422/485

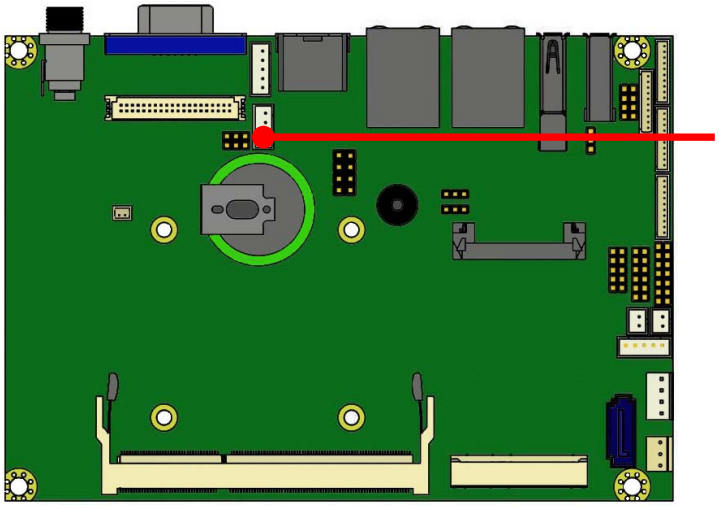
Note: COM2 RS-232/422/485 can be set by BIOS setting. Default is RS-232.

CN14 : Wafer 1 x 10 header, pitch 1.25 mm, connector type : YIMTEX 501MW1*10STR			
Pin	**RS232 mode**	**RS422 mode**	**RS485 mode**
1	DCD, Data carrier detect	TXD-	TXD-
2	DSR, Data set ready		
3	RXD, Received Data	TXD+	TXD+
4	RTS, Request to send		
5	TXD, Transmitted Data	RXD+	
6	CTS, Clear to sent		
7	DTR, Data terminal ready	RXD-	
8	RI, Ring indicator		
9	GND		
10	N/C		



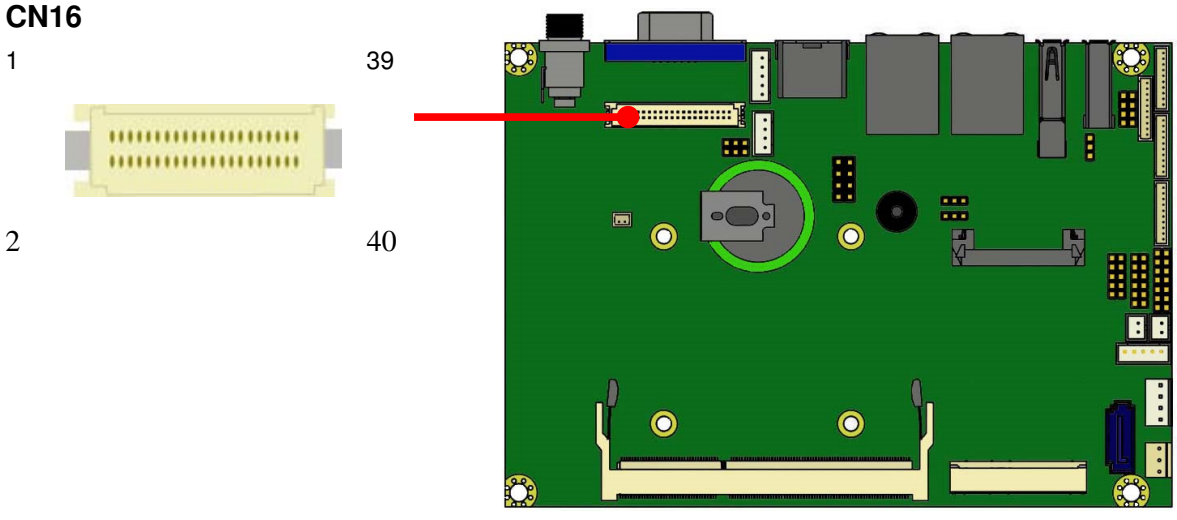
- 2.4.14 : CN15 for LVDS panel backlight adjustment connector

CN15 : 1 x 4 wafer			
Pin	**Signal**	**Pin**	**Signal**
1	LVDS_BLUP	2	GND
3	GND	4	LVDS_BLDN

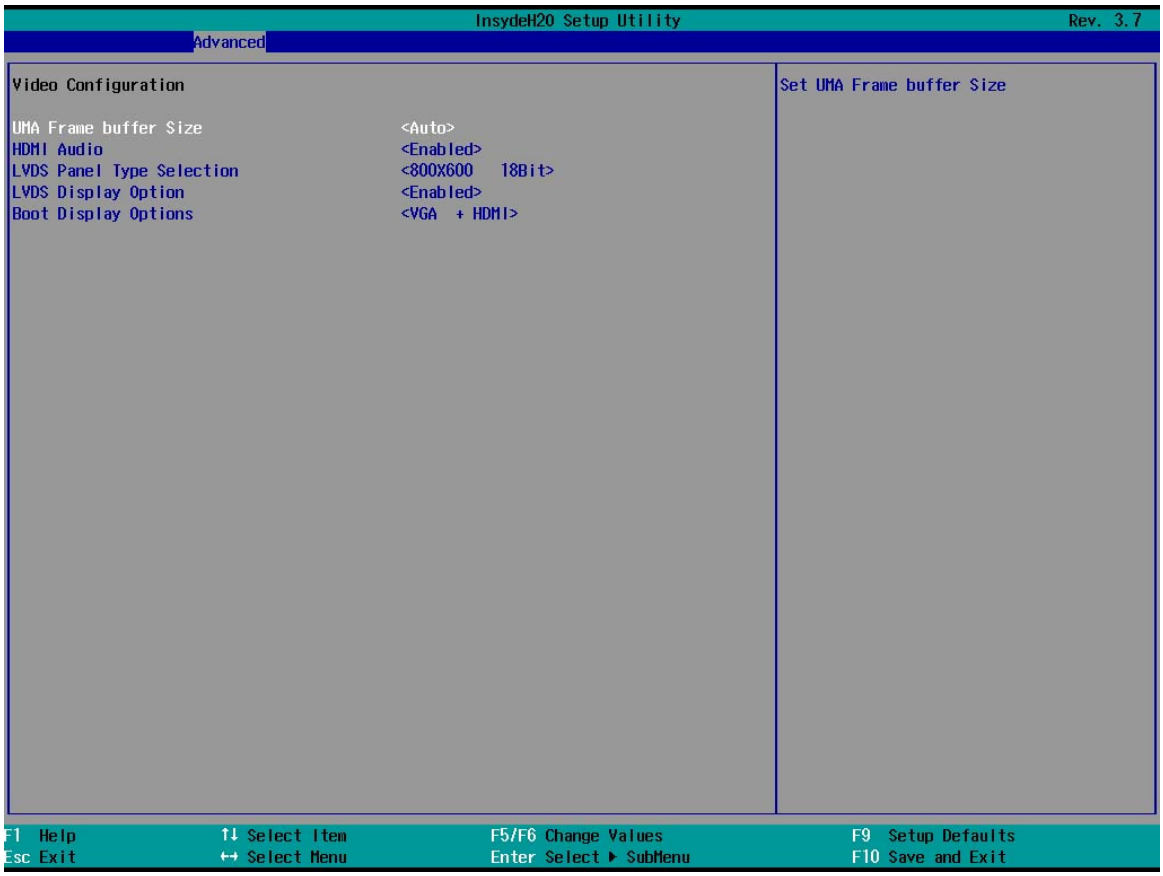


- 2.4.15 : CN16 for LVDS connector

CN16 : connector type : DF13A-40DP-1.25V					
Pin	**Signal**	**Pin**	**Signal**	**Pin**	**Signal**
1	LVDS_VCC	15	LVDSA_1+	29	GND
2	LVDS_VCC	16	LVDSB_1+	30	GND
3	LVDS_VCC	17	GND	31	DDC_Clock
4	LVDS_VCC	18	GND	32	DDC_Data
5	GND	19	LVDSA_2-	33	GND
6	GND	20	LVDSB_2-	34	GND
7	LVDSA_0-	21	LVDSA_2+	35	LVDSA_3-
8	LVDSB_0-	22	LVDSB_2+	36	LVDSB_3-
9	LVDSA_0+	23	GND	37	LVDSA_3+
10	LVDSB_0+	24	GND	38	LVDSB_3+
11	GND	25	LVDSA_Clock-	39	SMB_Clock
12	GND	26	LVDSB_Clock-	40	SMB_Data
13	LVDSA_1-	27	LVDSA_Clock+		
14	LVDSB_1-	28	LVDSB_Clock+		

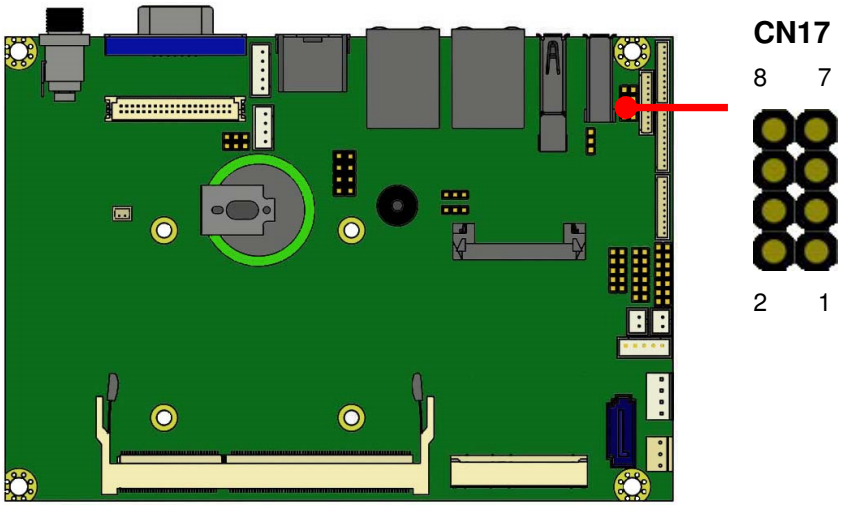


BIOS setting for LVDS



- 2.4.16 : CN17 for front Panel pin header

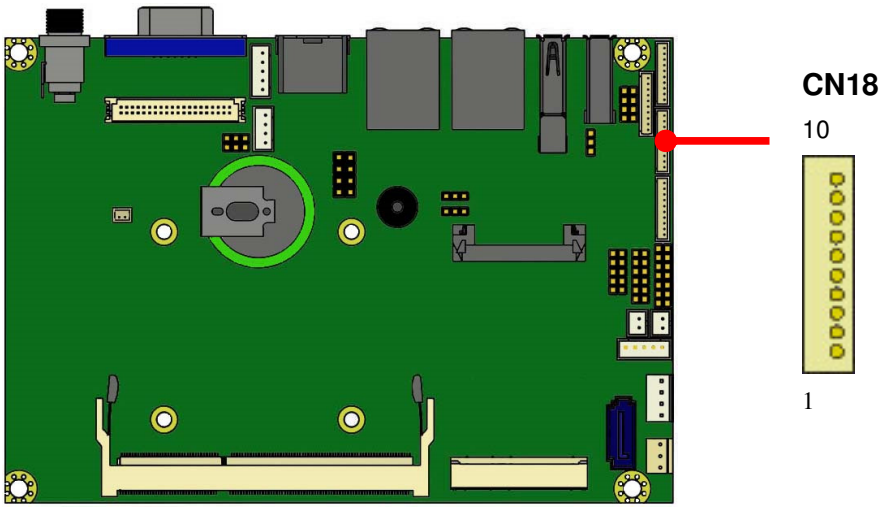
CN17 : 2 x 4 header , pitch 2.00 mm			
Pin	**Signal**	**Pin**	**Signal**
1	Power_LED+	2	Power_LED-
3	HDD_LED+	4	HDD_LED-
5	GND	6	Power on/off
7	Reset	8	GND



- 2.4.17: CN18 for COM 3 , RS232/422/485

Note: COM2 RS-232/422/485 can be set by BIOS setting. Default is RS-232.

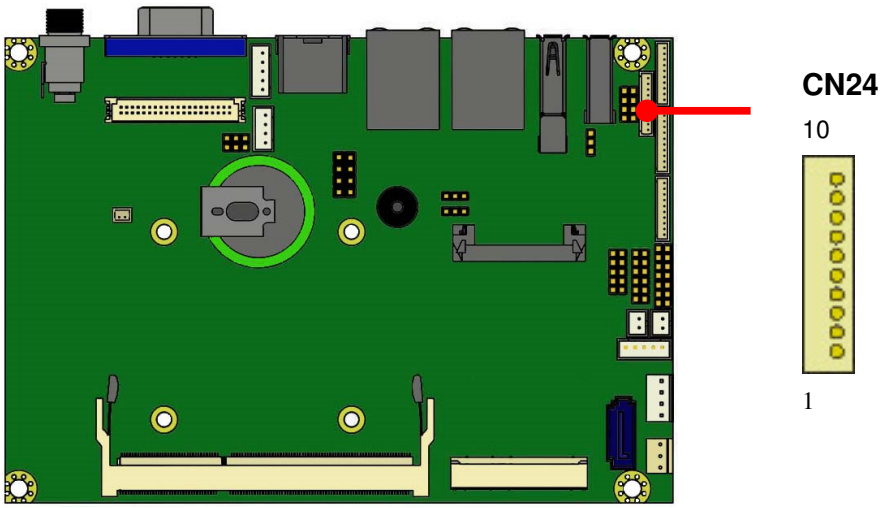
CN18 : Wafer 1 x 10 header, pitch 1.25 mm, connector type : YIMTEX 501MW1*10STR			
Pin	**RS232 mode**	**RS422 mode**	**RS485 mode**
1	DCD, Data carrier detect	TXD-	TXD-
2	DSR, Data set ready		
3	RXD, Received Data	TXD+	TXD+
4	RTS, Request to send		
5	TXD, Transmitted Data	RXD+	
6	CTS, Clear to sent		
7	DTR, Data terminal ready	RXD-	
8	RI, Ring indicator		
9	GND		
10	N/C		



- 2.4.18 : CN24 for COM 1 , RS232/422/485

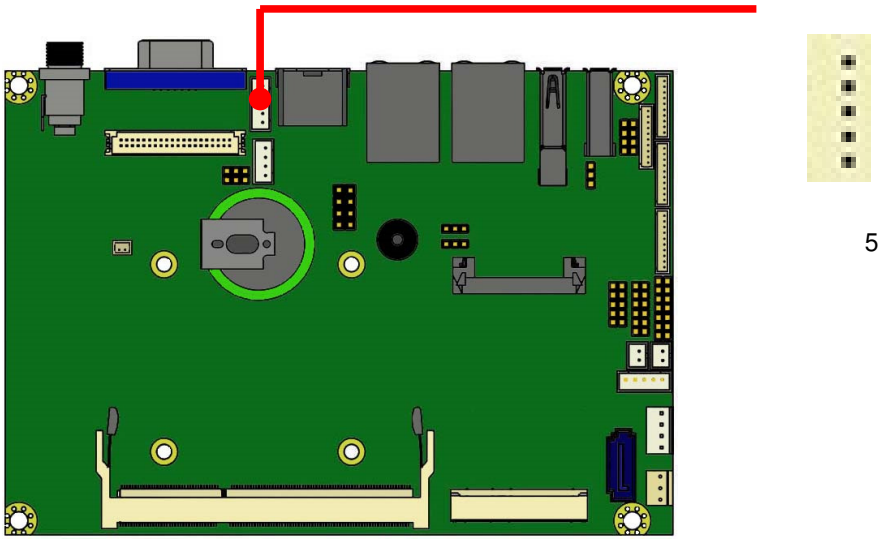
Note: COM2 RS-232/422/485 can be set by BIOS setting. Default is RS-232.

CN18 : Wafer 1 x 10 header, pitch 1.25 mm, connector type : YIMTEX 501MW1*10STR			
Pin	**RS232 mode**	**RS422 mode**	**RS485 mode**
1	DCD, Data carrier detect	TXD-	TXD-
2	DSR, Data set ready		
3	RXD, Received Data	TXD+	TXD+
4	RTS, Request to send		
5	TXD, Transmitted Data	RXD+	
6	CTS, Clear to sent		
7	DTR, Data terminal ready	RXD-	
8	RI, Ring indicator		
9	GND		
10	N/C		



- 2.4.19 : CN25 for LVDS panel inverter power connector

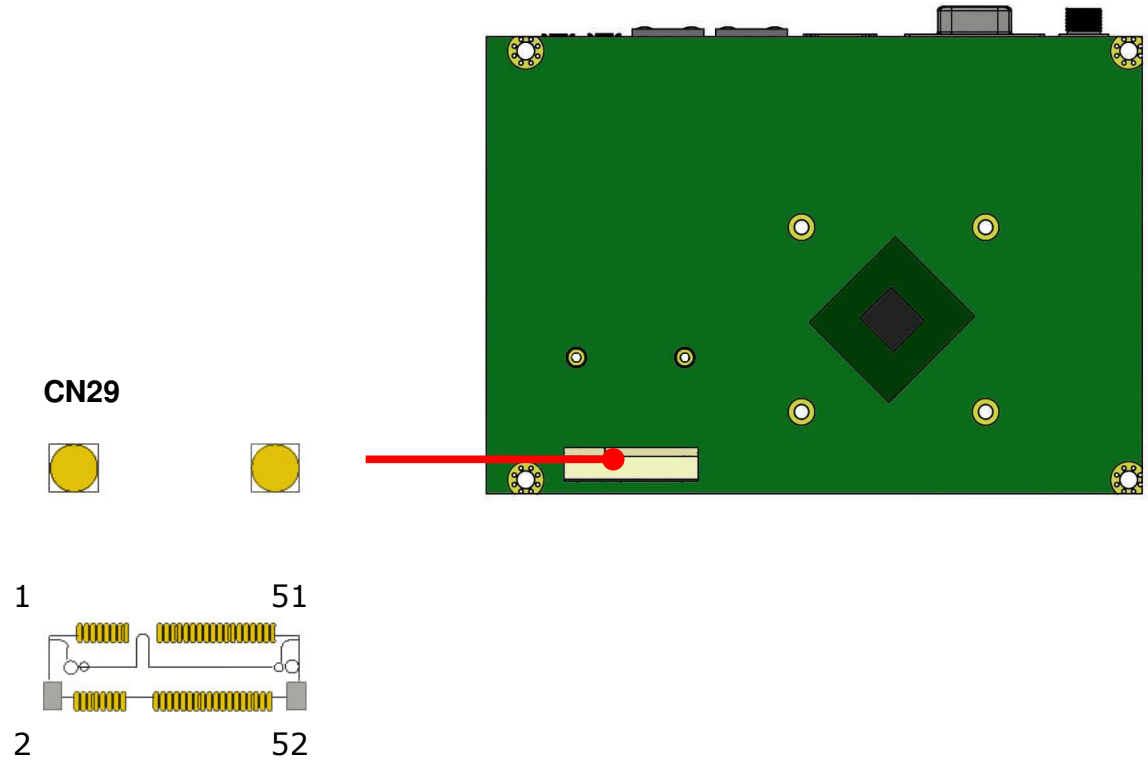
CN25 : 1 x 5 wafer , Pitch : Pitch 2.0 mm			
Pin	**Signal**	**Pin**	**Signal**
1	+12V	2	GND
3	Backlight Enable	4	Backlight control
5	+ 5V		



5

- 2.4.20 : CN29 for half-size Mini-PCIe socket

CN29 : Half-size Mini-PCIe socket					
Pin	**Signal**	**Pin**	**Signal**	**Pin**	**Signal**
1	MSATA_WAKE	19	N/C	37	N/A
2	+3.3V	20	N/C	38	USB_D+
3	N/C	21	GND	39	V3P3_MSATA
4	GND	22	PLTRST_BUF1_N	40	GND
5	N/C	23	MSATA_RXN4	41	V3P3_MSATA
6	+1.5V	24	MSATA_AUX33	42	LED_WWAN
7	MCLKREQ	25	MSATA_RXP4	43	N/A
8	N/C	26	GND	44	LED_WLAN
9	GND	27	GND	45	N/A
10	N/C	28	+1.5V	46	LED_WPAN
11	MSATA_PE_CLKN	29	GND	47	N/A
12	N/C	30	ICH_SMBCLK	48	+1.5V
13	MSATA_PE_CLKP	31	MSATA_TXN4	49	N/A
14	N/C	32	ICH_SMBDATA	50	GND
15	GND	33	MSATA_TXP4	51	N/A
16	N/C	34	GND	52	+3.3V
17	N/C	35	GND	-	
18	N/C	36	USB_D-	-	



Note: (Preliminary definition)

Half-size Mini-PCIe card could support SATA or PCIE signal.

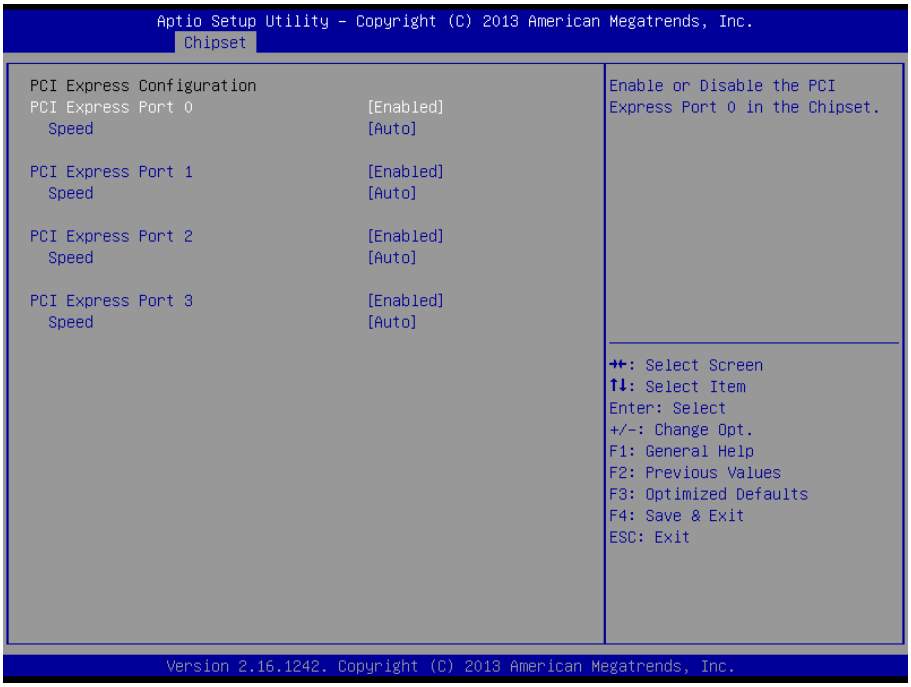
The default setting is set as PCIE signal.

Please into BIOS and select SATA or PCIE signal to match your device.

For example :

Please into BIOS and set ” SATA ” when you install a half-size add-on card, such as mSATA SSD.

Please into BIOS and set ” PCIE ” when you install a half-size add-on card, such as WIFI card.


Aptio Setup Utility – Copyright (C) 2013 American Megatrends, Inc.
Chipset
PCI Express Configuration
PCI Express Port 0 [Enabled]
Speed [Auto]
PCI Express Port 1 [Enabled]
Speed [Auto]
PCI Express Port 2 [Enabled]
Speed [Auto]
PCI Express Port 3 [Enabled]
Speed [Auto]
Enable or Disable the PCI Express Port 0 in the Chipset.
→←: Select Screen
↑↓: Select Item
Enter: Select
+/-: Change Opt.
F1: General Help
F2: Previous Values
F3: Optimized Defaults
F4: Save & Exit
ESC: Exit
Version 2.16.1242. Copyright (C) 2013 American Megatrends, Inc.

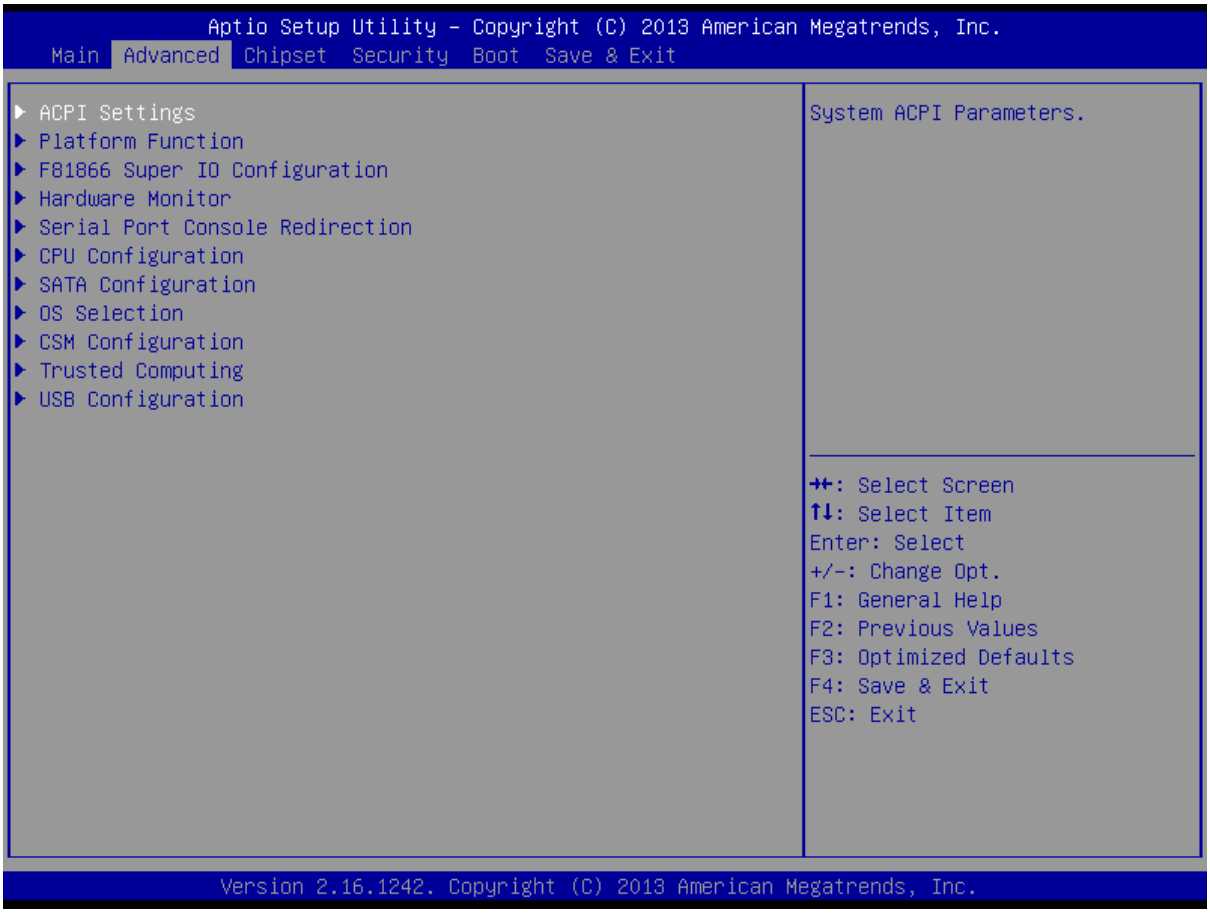
3. BIOS setting Menu

- 3.1 Main Menu

The Main Menu of BIOS Setup Utility provide a quick overview of basic system information and the ability to change the system Date, time ….

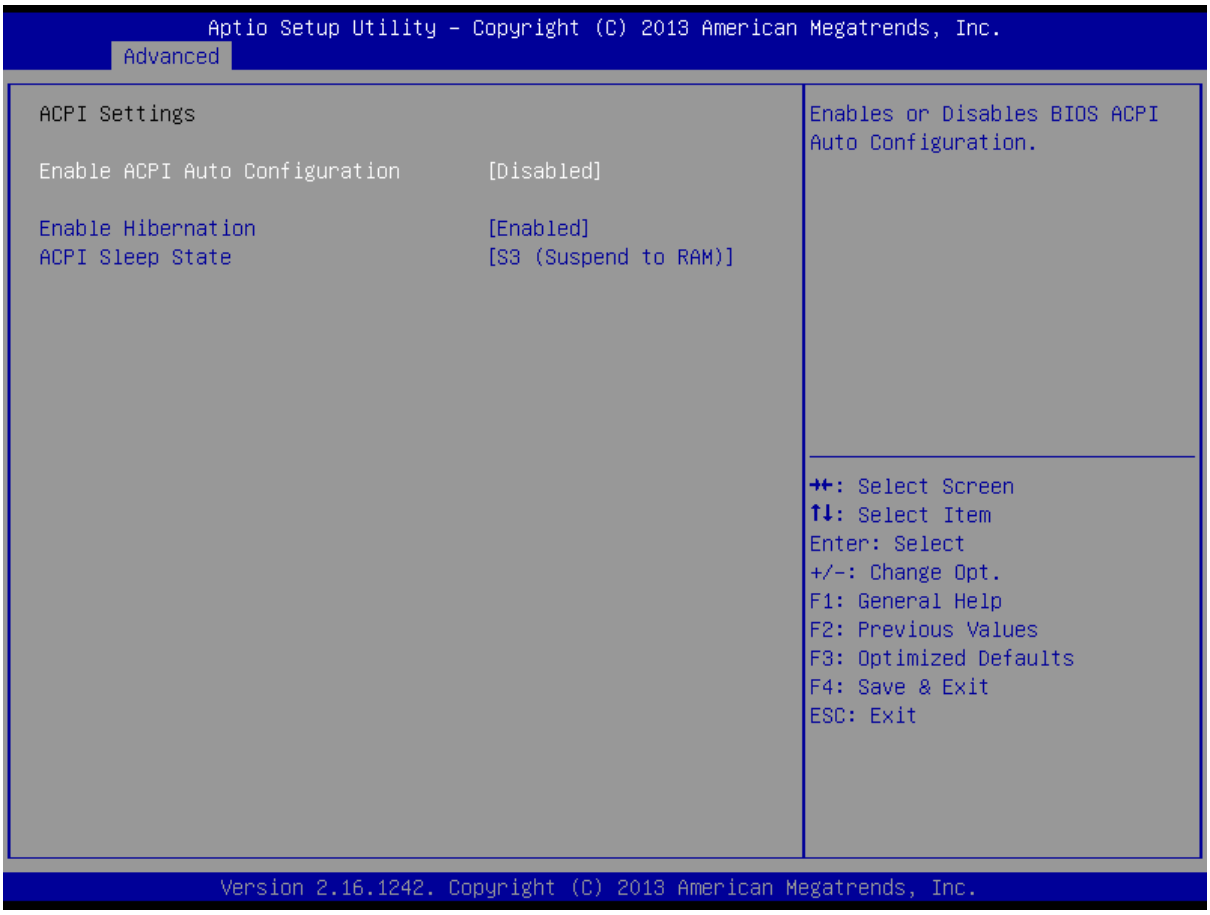
- 3.2 Advanced Menu

The Advanced Menu of BIOS Setup Utility allows users to configure advanced system settings.



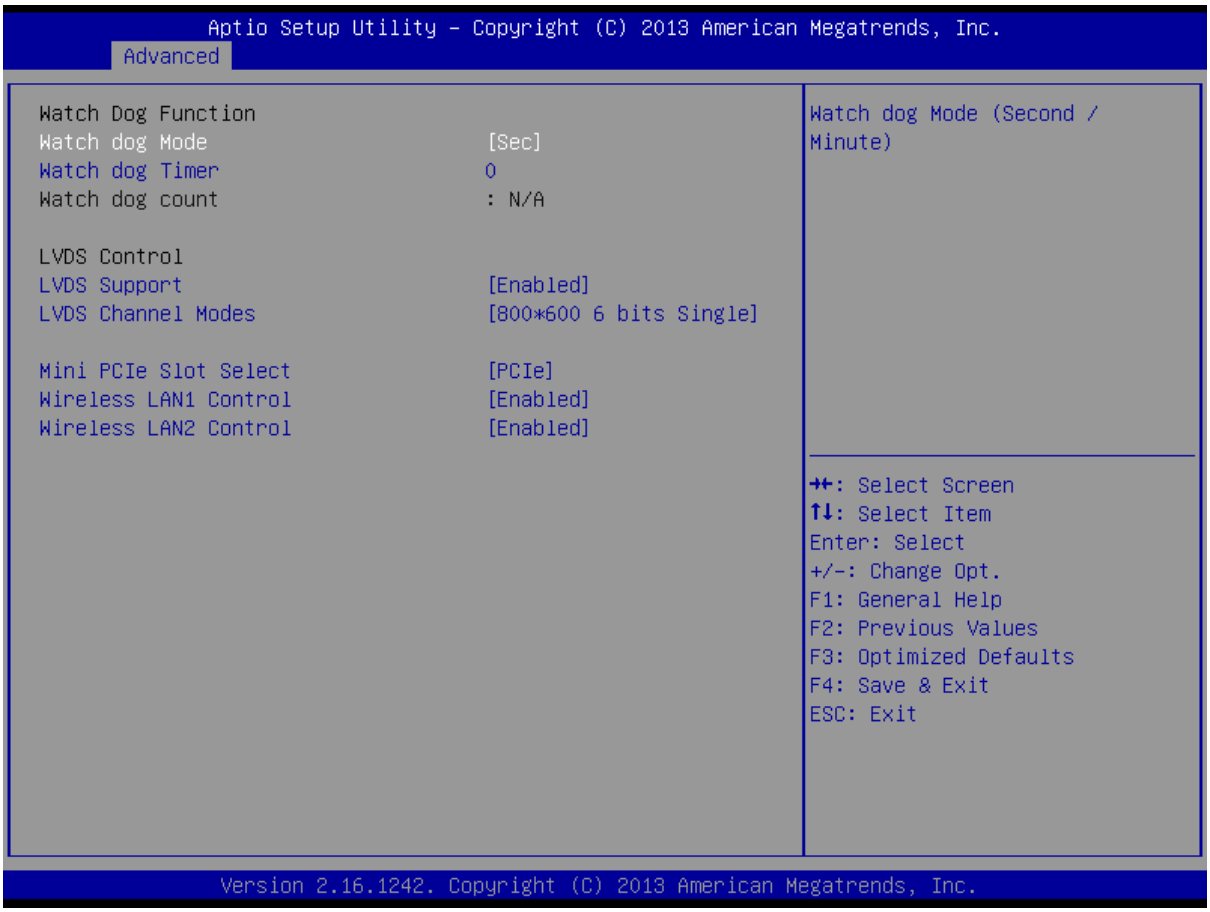
- 3.2.1 ACPI Settings

Description: This item is used to configure ACPI like auto configuration, hibernation, sleep state.



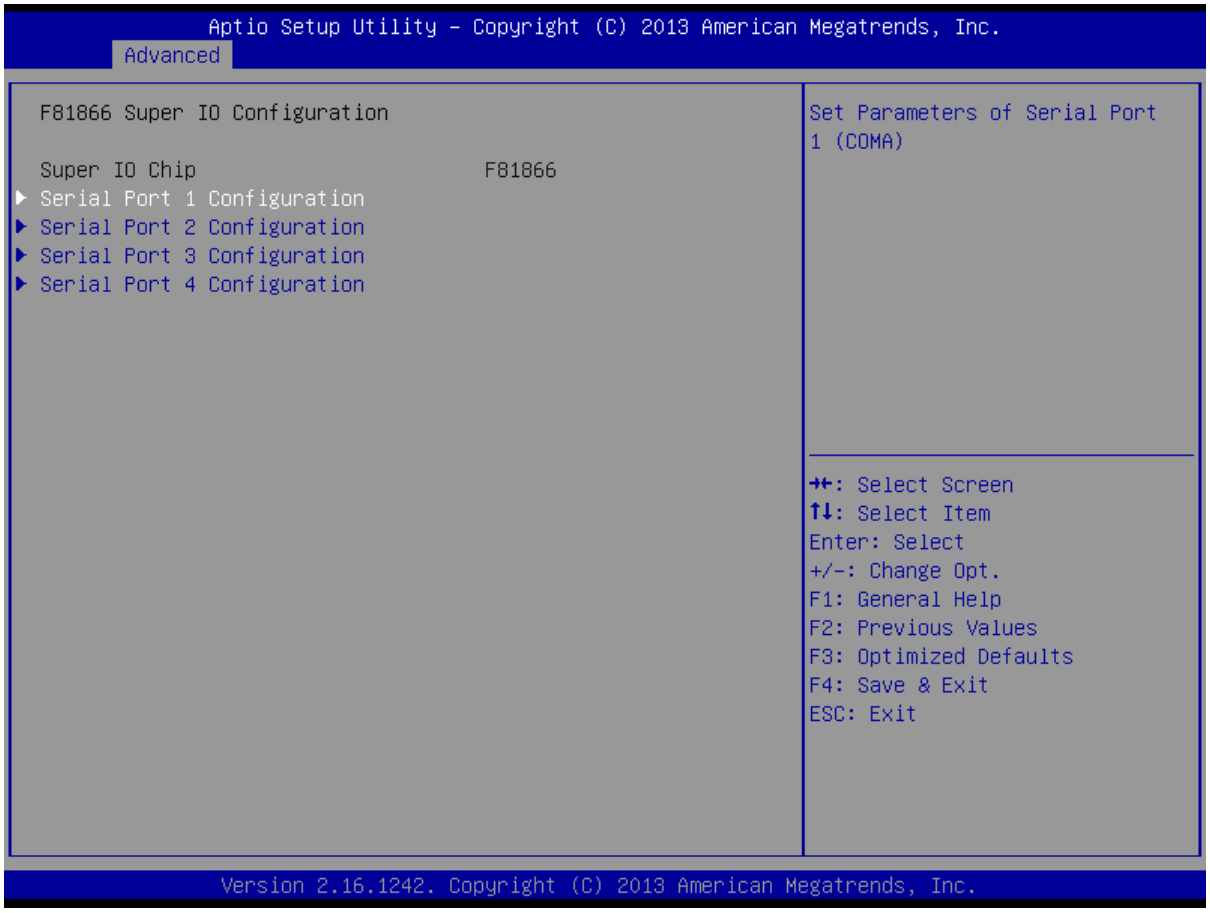
- 3.2.2 Platform Function

Description: Watch Dog can be configured with Mode/Timer/Count parameters. LVDS can be configured with Enable/Channel Model. Mini PCIe can be configured with type/enable options.



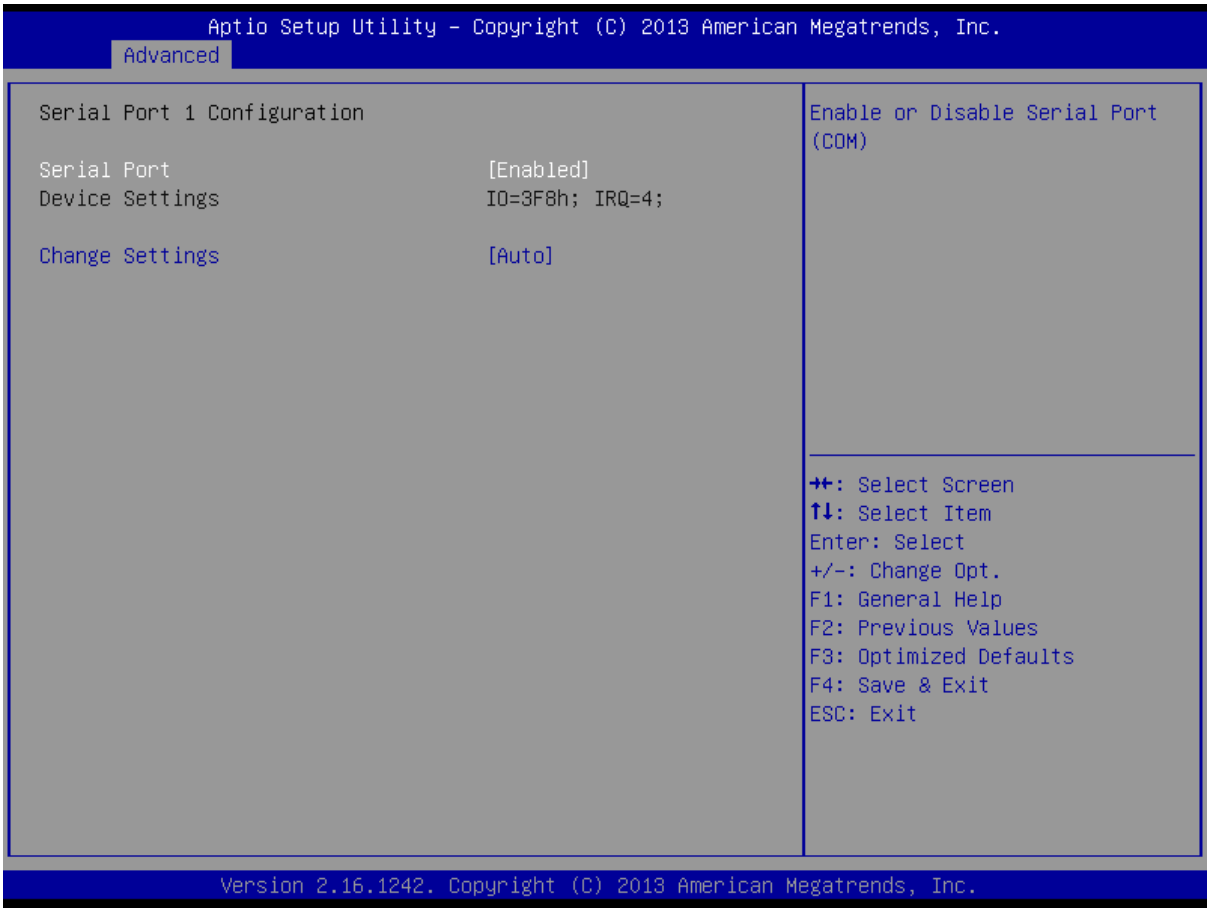
- 3.2.3 F81866 Super IO Configuration

Description: Select 'Enabled' to enable the onboard serial port. The options are Enabled and Disabled.



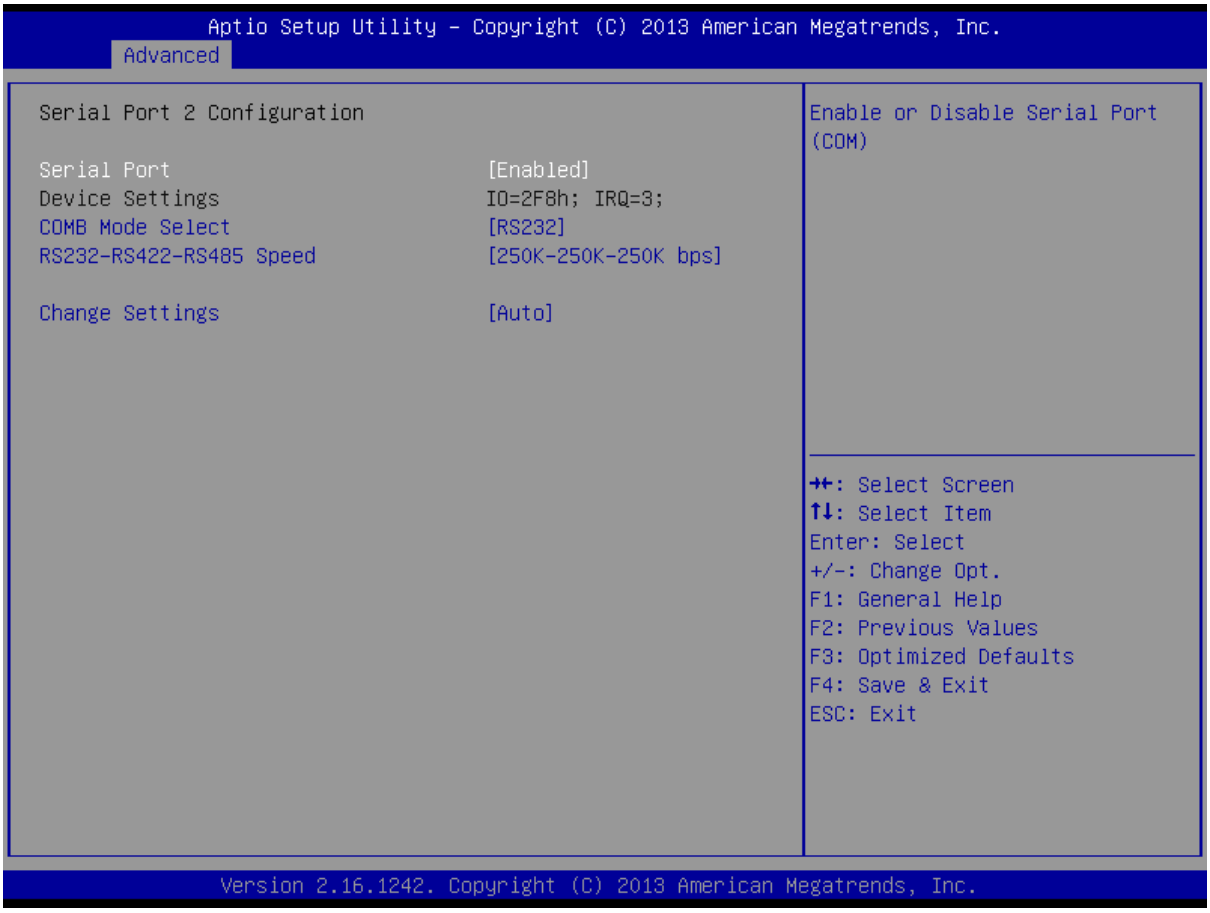
- 3.2.3.1 Serial Port 1 Configuration

Description: This menu can be used to configure serial port 1. The options are Enabled and Disabled.



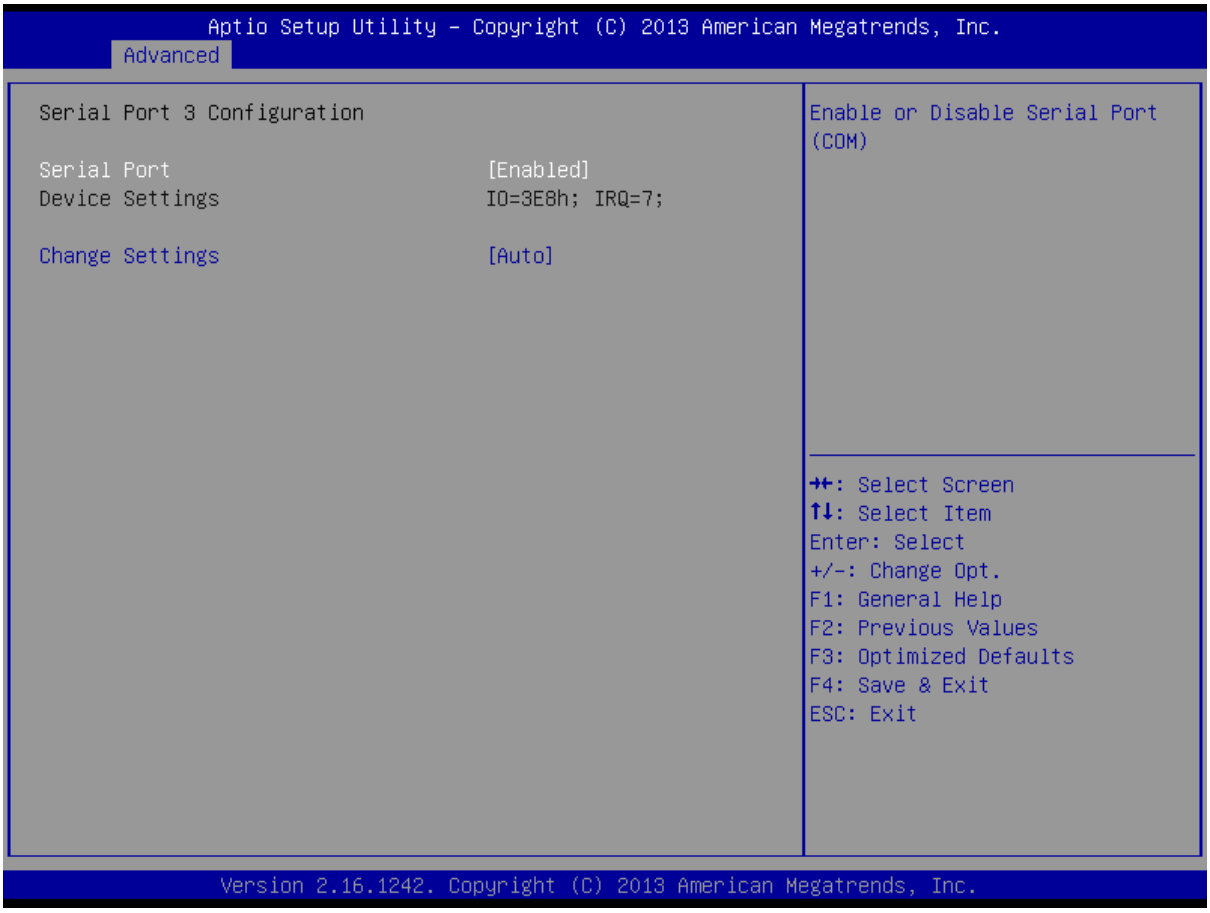
- 3.2.3.2 Serial Port 2 Configuration

Description: This menu can be used to configure serial port 2. Serial port 2 also supports RS-422 and RS-485.



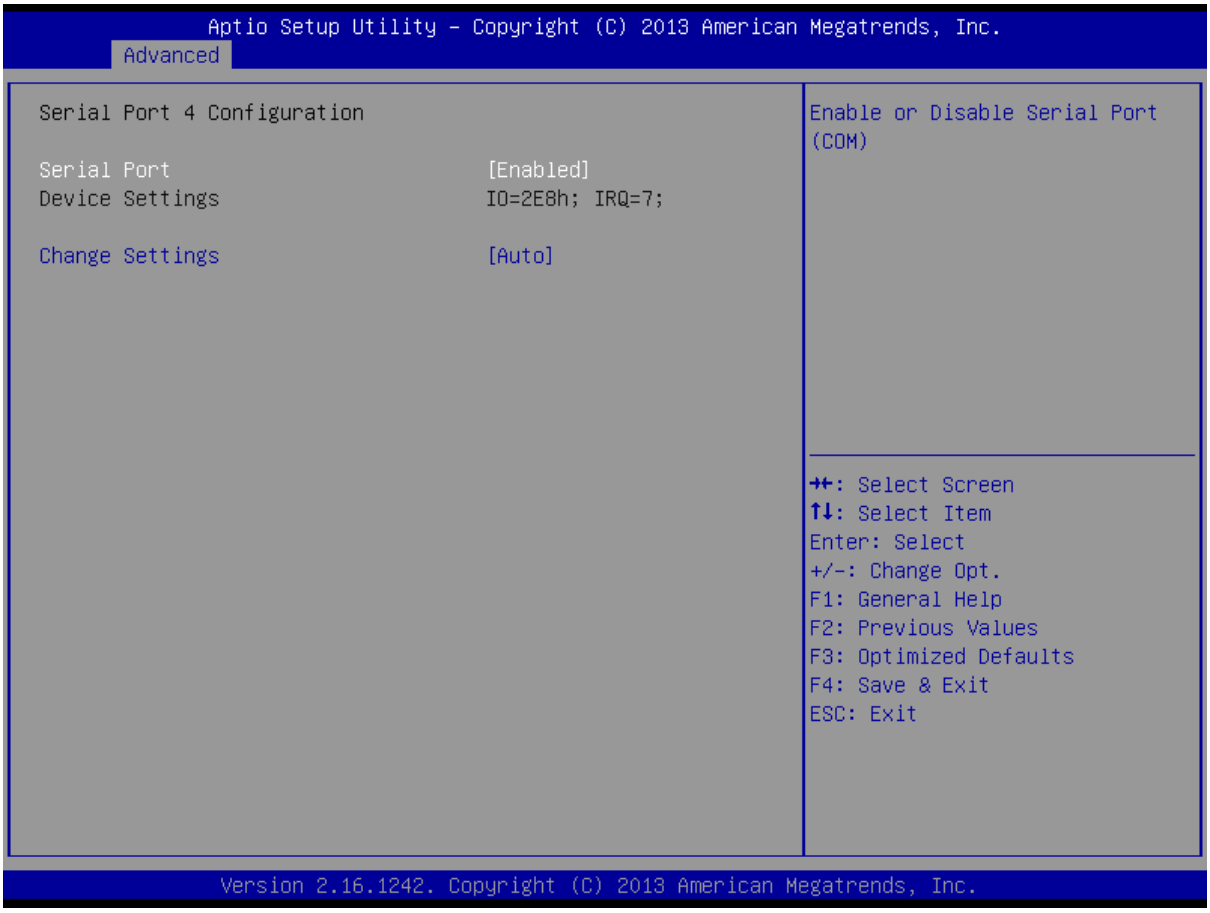
- 3.2.3.3 Serial Port 3 Configuration

Description: This menu can be used to configure serial port 1. The options are Enabled and Disabled.



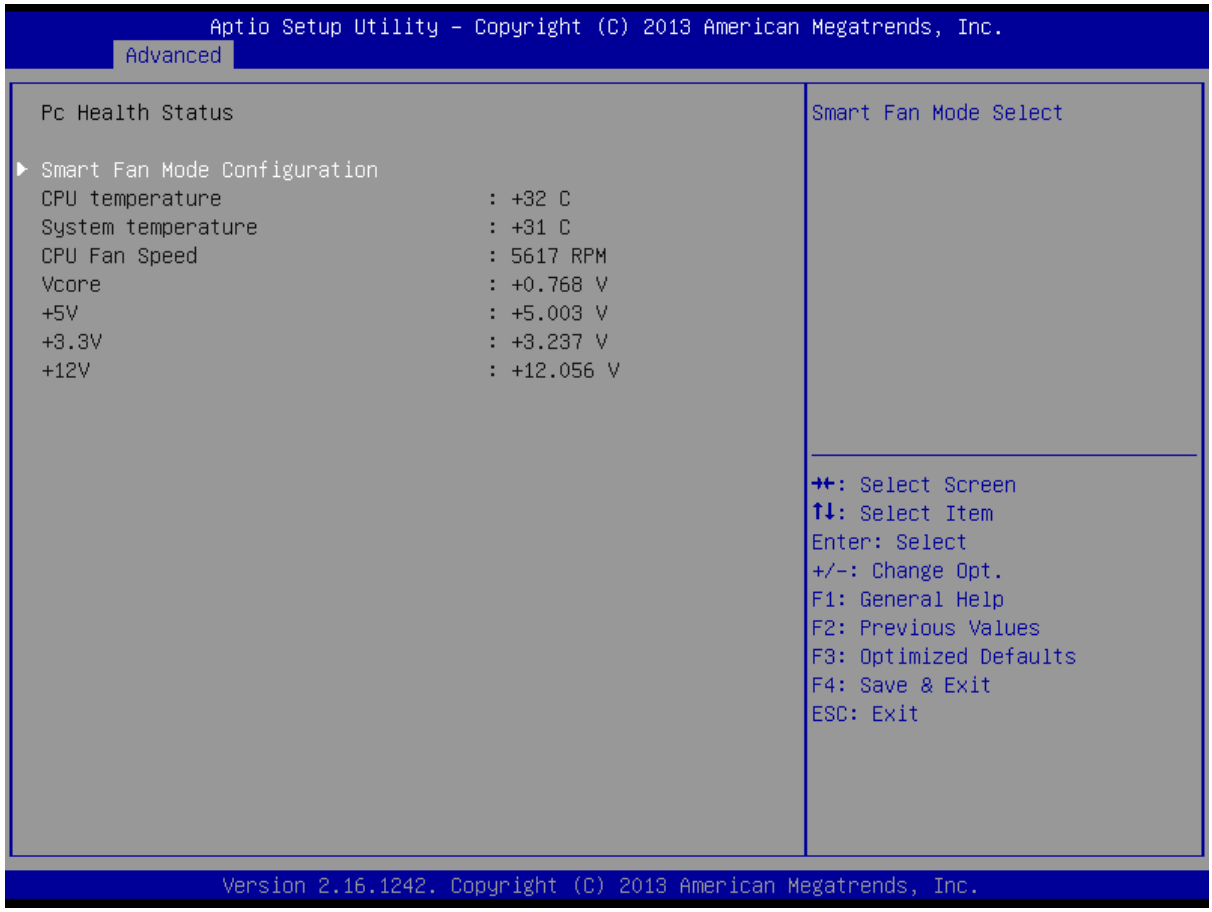
- 3.2.3.4 Serial Port 4 Configuration

Description: This menu can be used to configure serial port 4. The options are Enabled and Disabled.



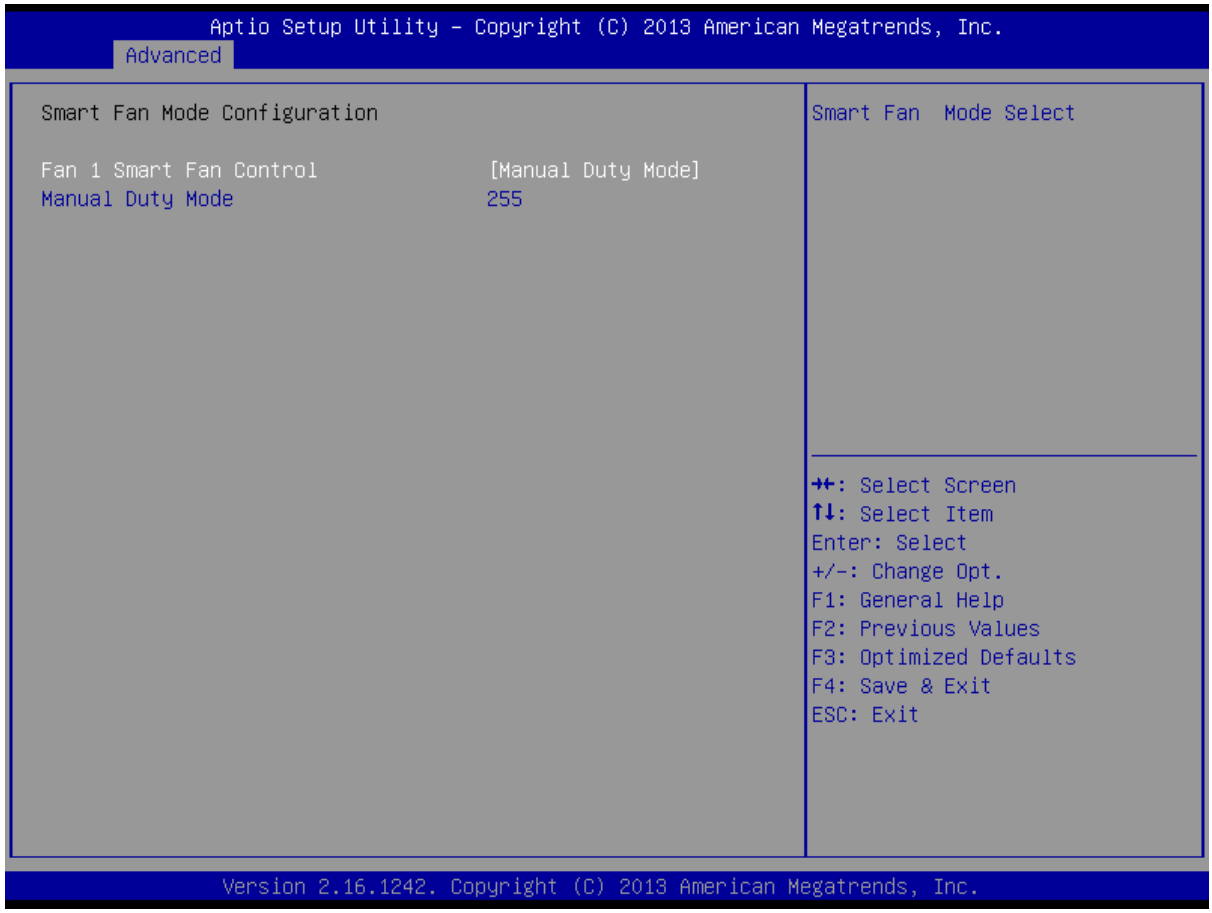
- 3.2.4 Hardware Monitor

Description: This feature allows the user to check CPU temperature and the fan speed; and also displays the current voltages of the above voltage monitors



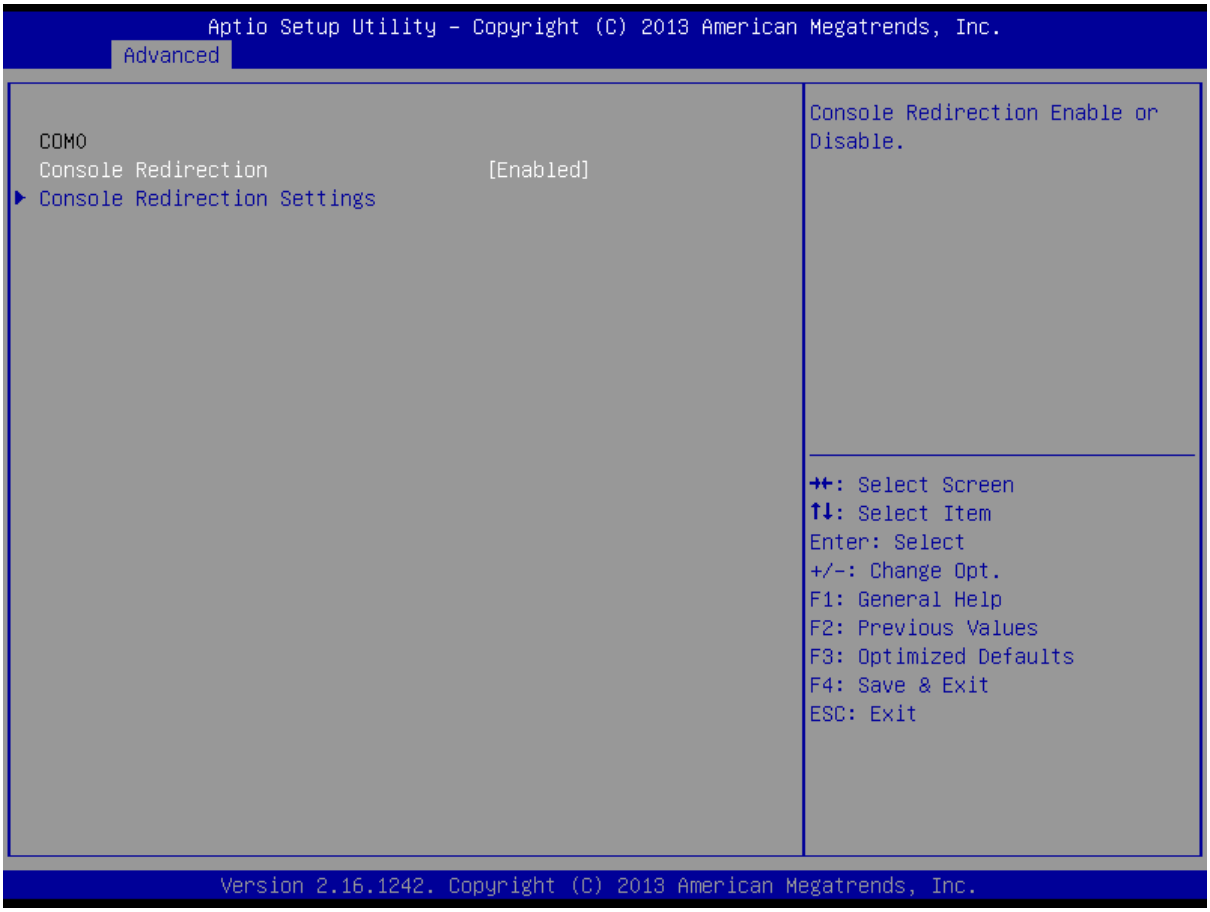
- 3.2.4.1 Smart Fan Mode Configuration

Description: This feature allows the user to configure smart fan by auto or manual mode.



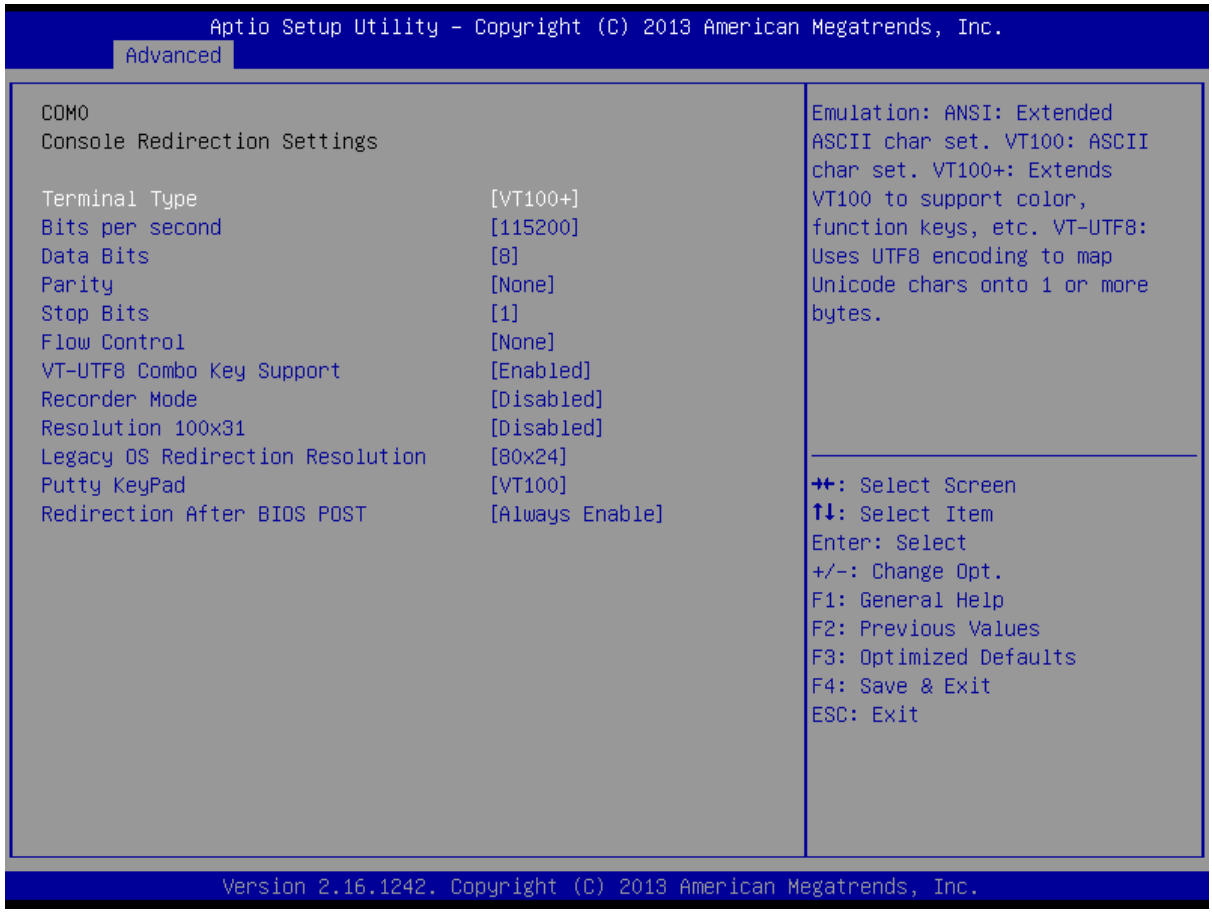
- 3.2.5 Serial Port Console Redirection

Description: This feature allows the user to remotely access the entire boot sequence via a serial console.



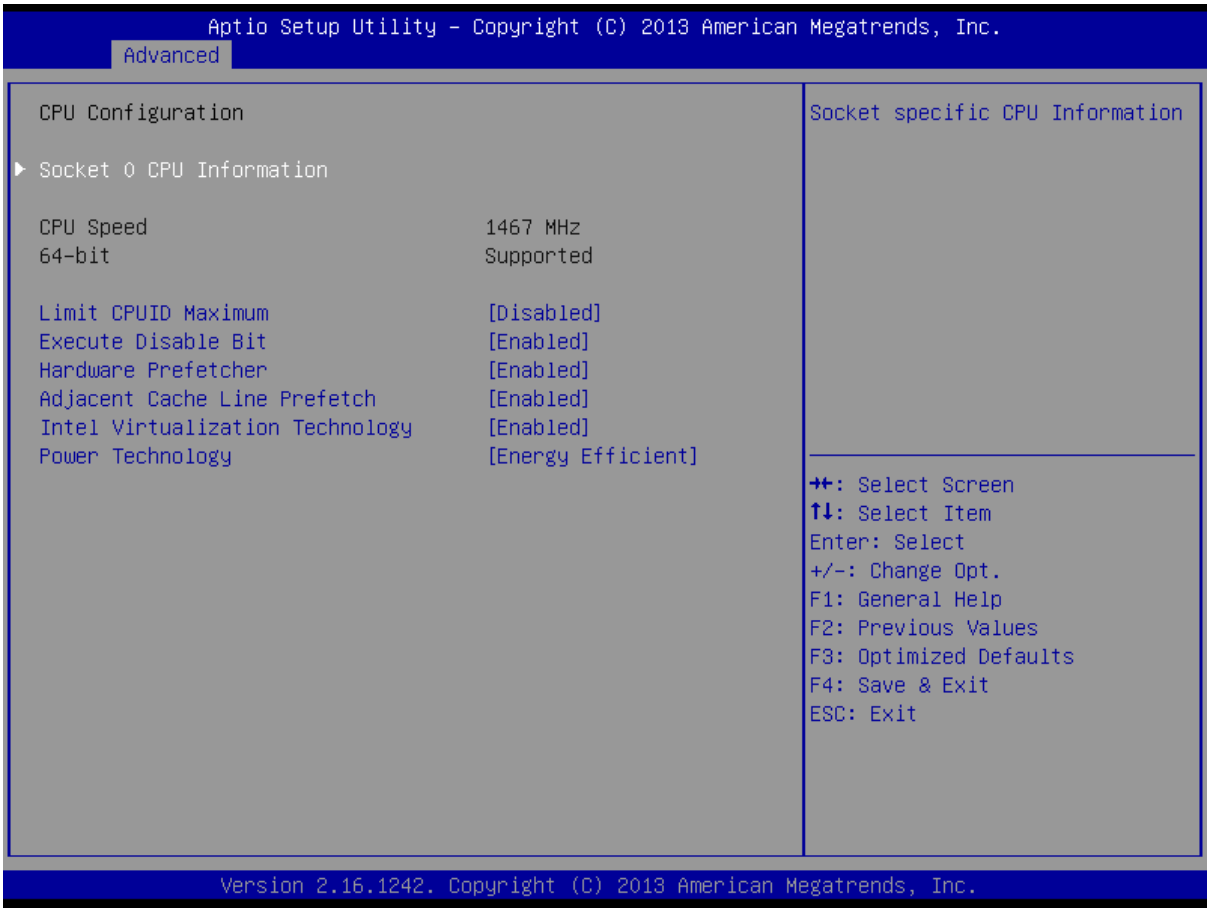
- 3.2.5.1 COM0

Description: This feature allows the user to configure the attribute for serial console.



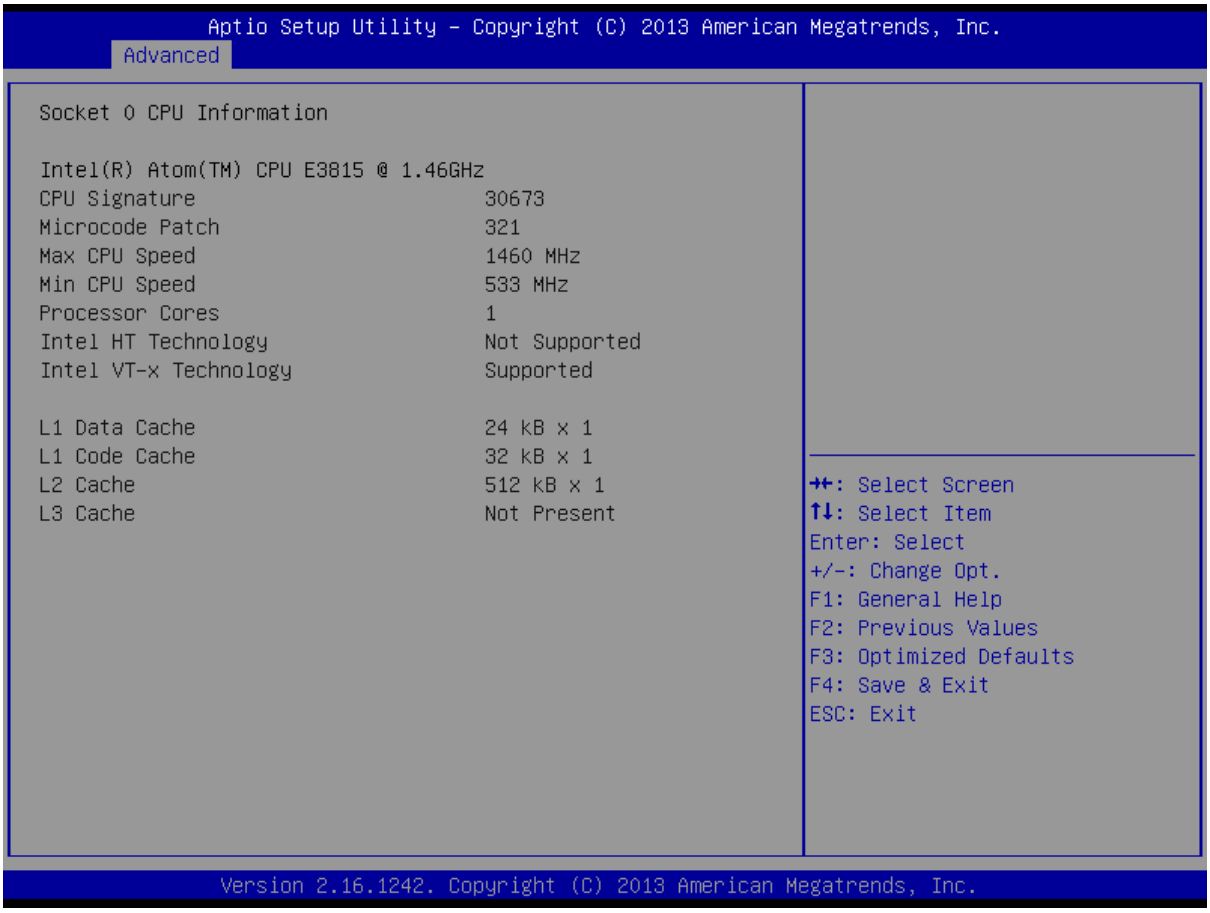
- 3.2.6 CPU Configuration

Description: This menu provide specific CPU information. Options also provided by this menu allow user to configure settings of CPU like Prefetcher, Virtualization, etc.



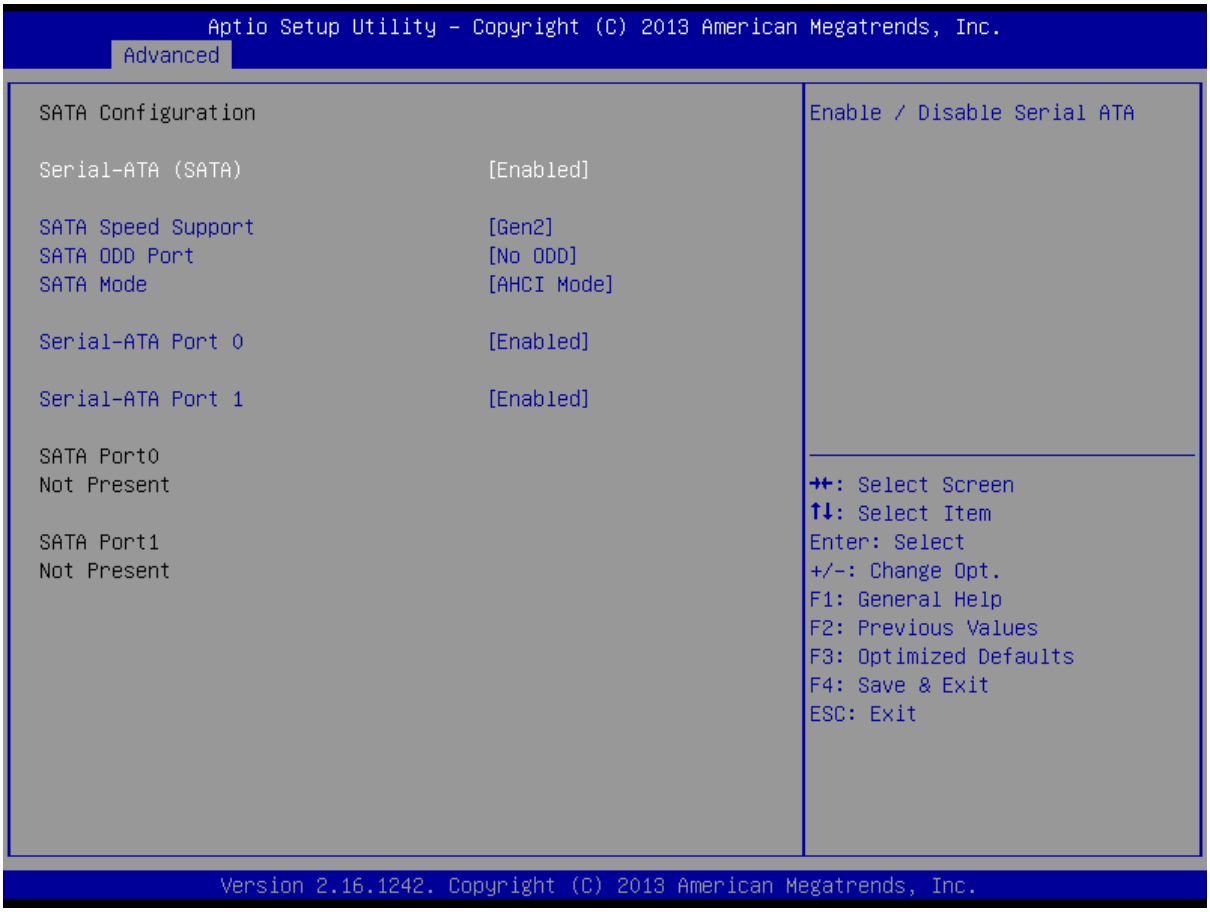
- 3.2.6.1 Socket 0 CPU Information

Description: This menu provides specific CPU information. Options also provided by this menu allow user to configure settings of CPU, like Prefetcher, Virtualization, etc.



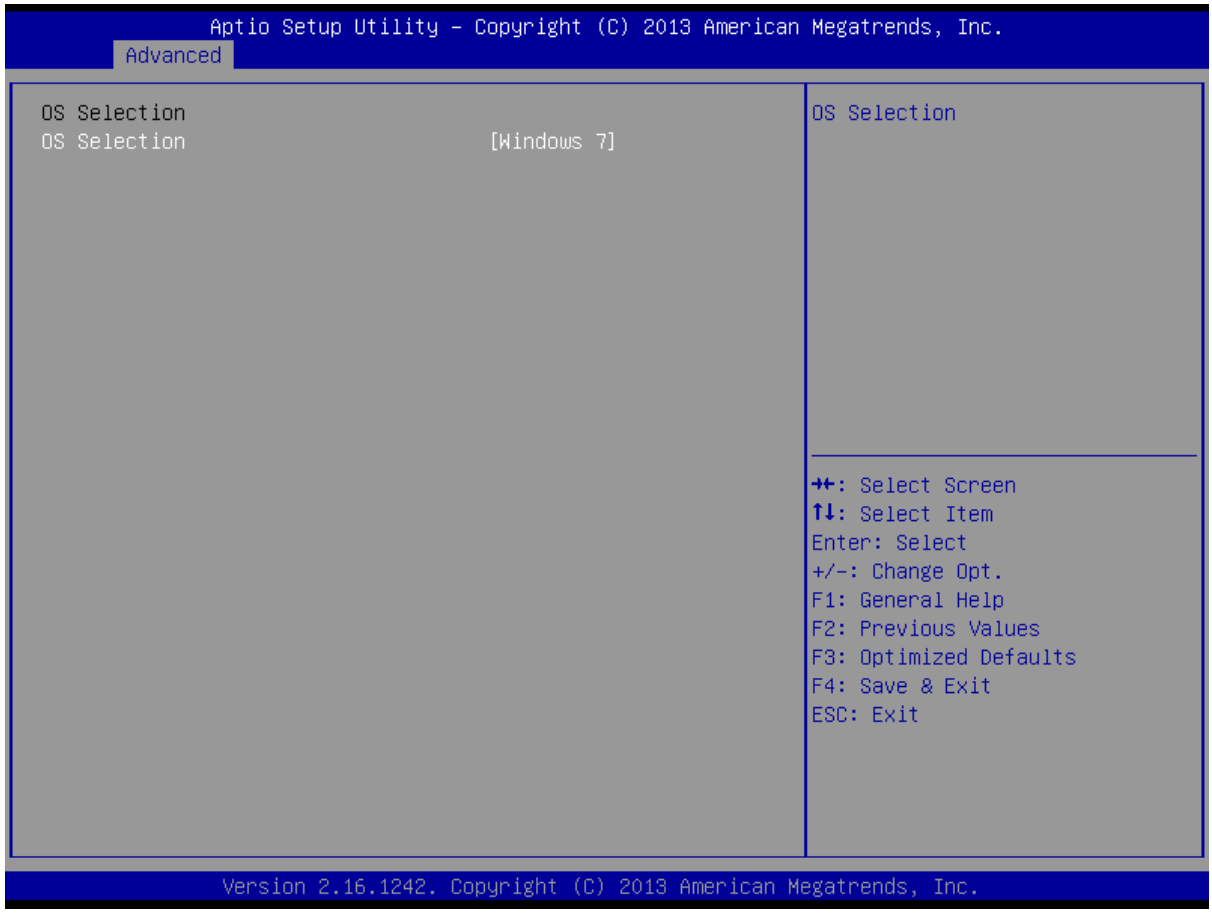
- 3.2.7 SATA Configuration

Description: This option enables user to configure SATA related setting, like function enablement, speed, mode, etc.



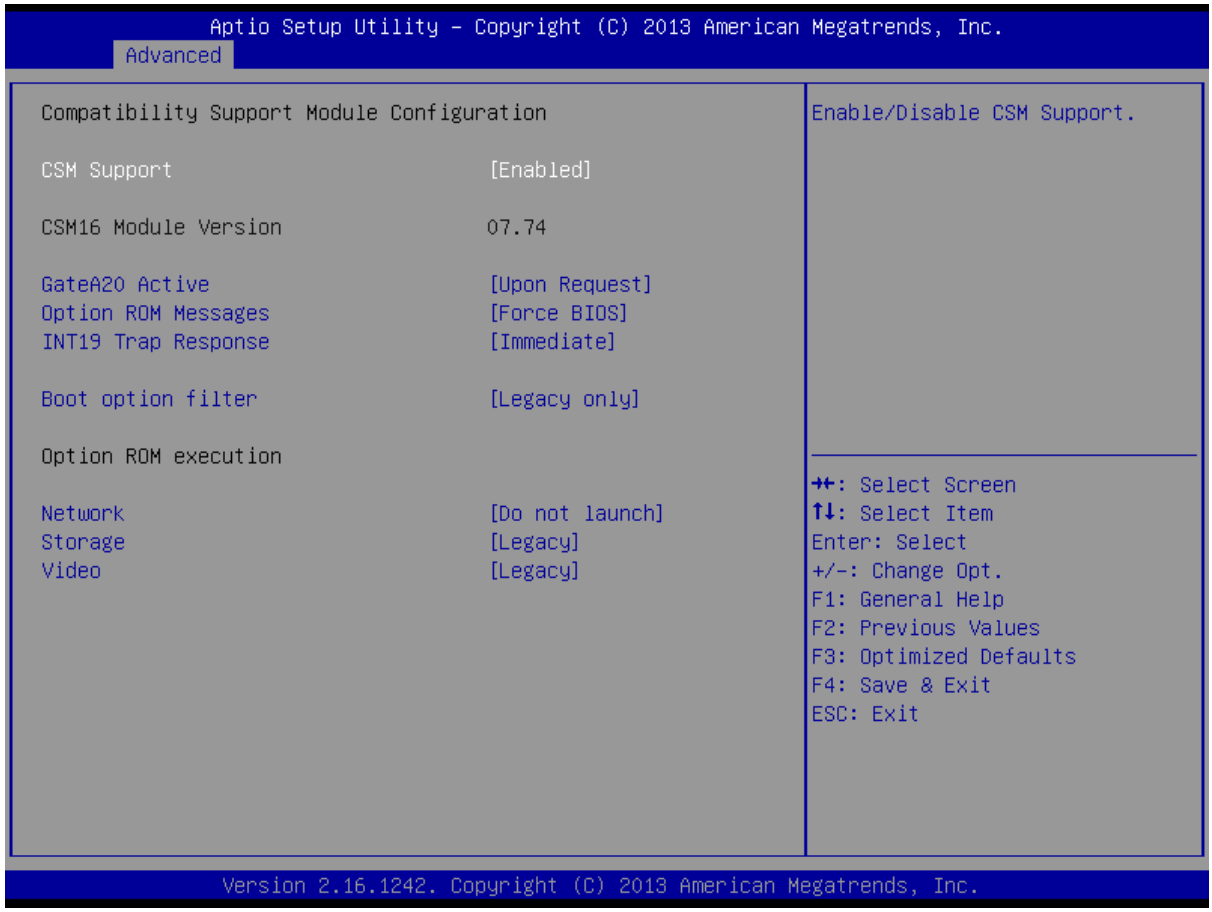
- 3.2.8 OS Selection

Description: This option enables user to select supported OS from menu.



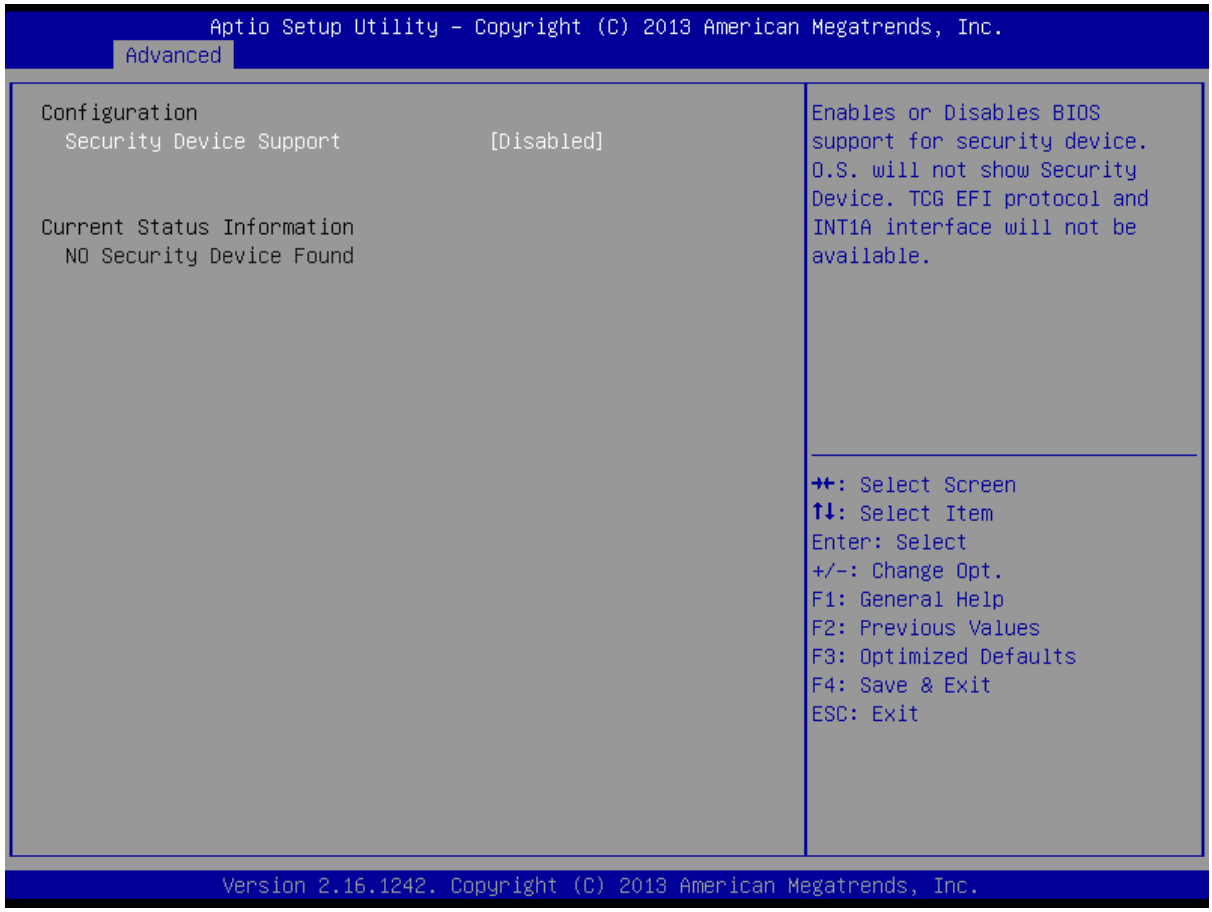
- 3.2.9 CSM Configuration

Description: Once CSM is enabled, further options become available. Further options allow user to configure Gate A20, Option ROM message, INT 19 Trap, etc.



- 3.2.10 Trusted Computing

Description: An optional TPM hardware module can be added and enabled by this option.

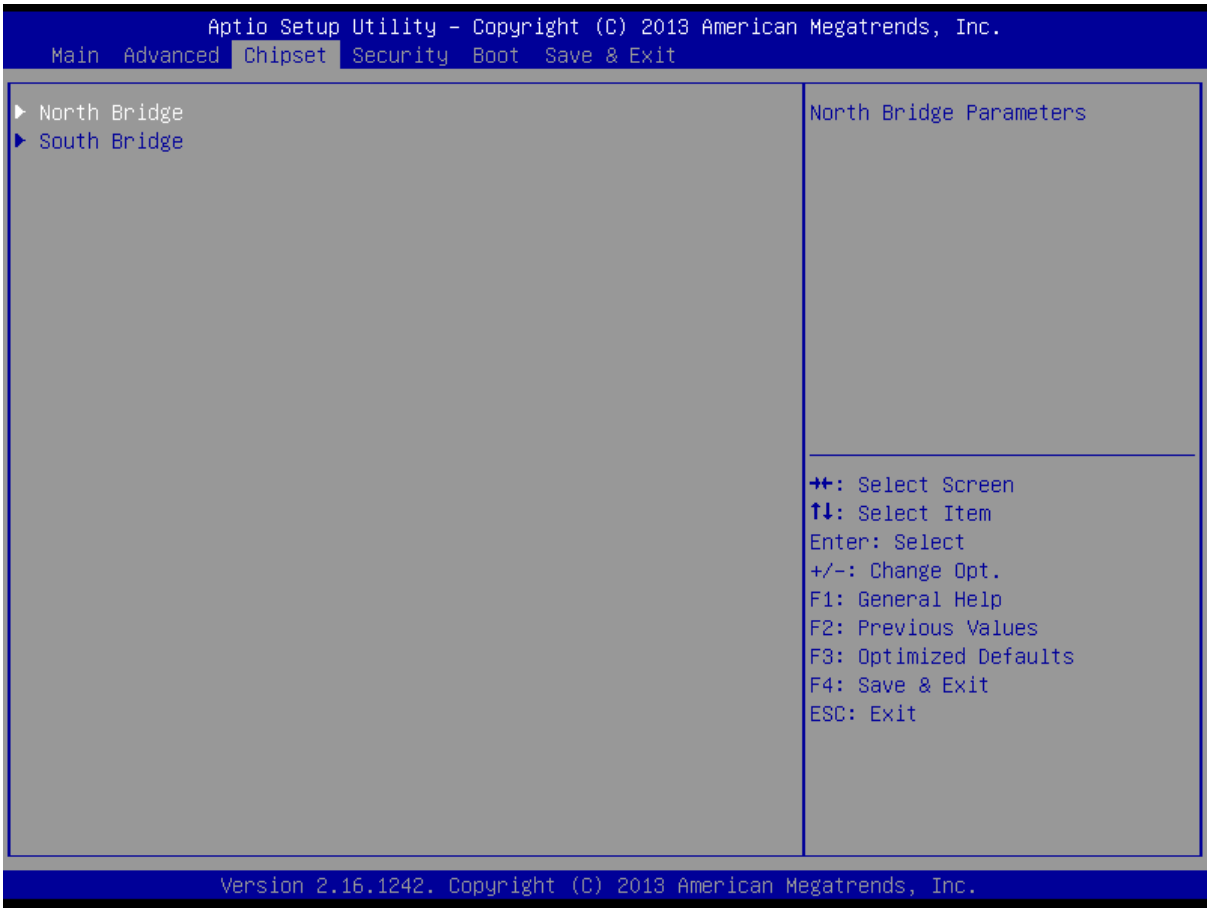


- 3.2.11 USB Configuration

Description: This option allow user to configure USB legacy support, delay and time-out timing setting can be configured in this menu.

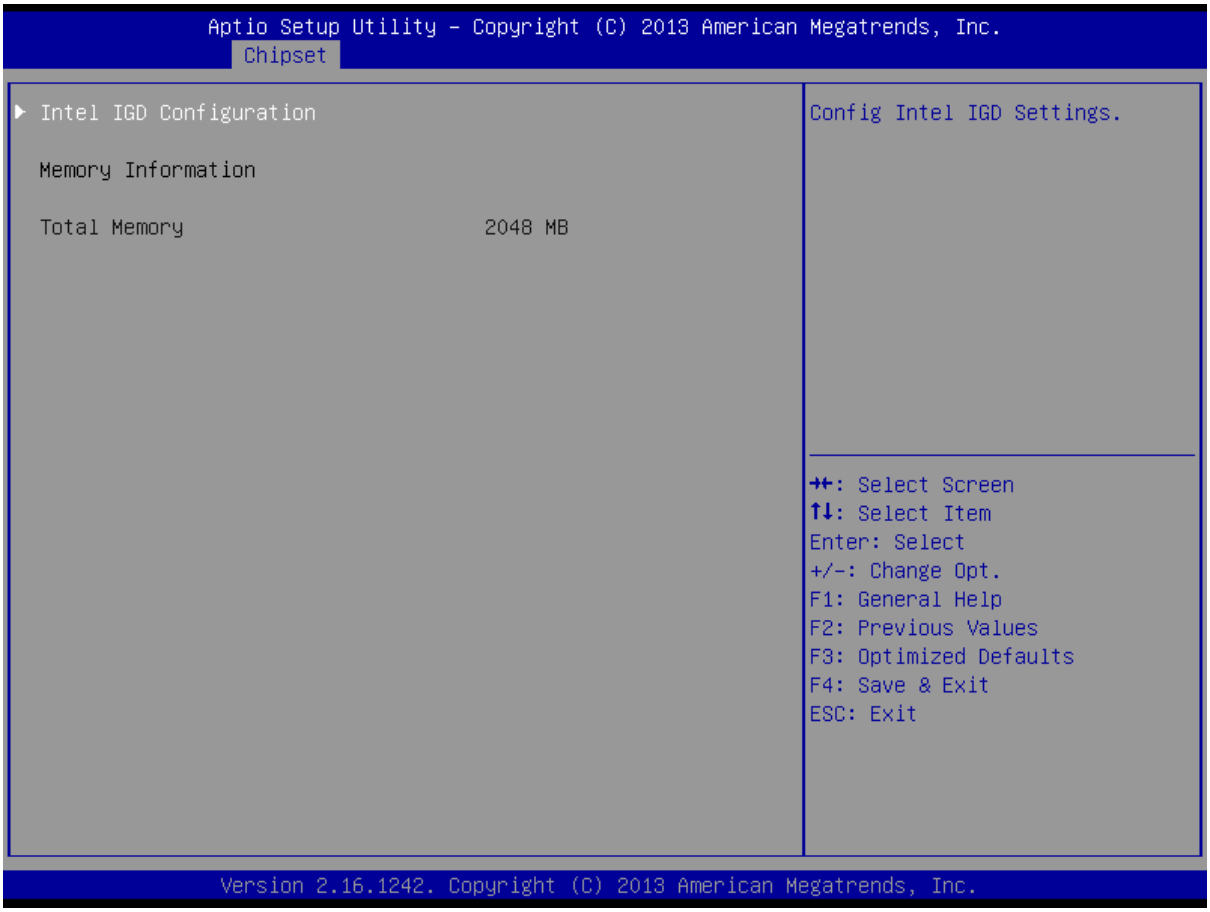
- 3.3 Chipset Menu

Description: Chipset menu divided into 2 group, they are north and south bridge, detail setting of both can be found at following options.



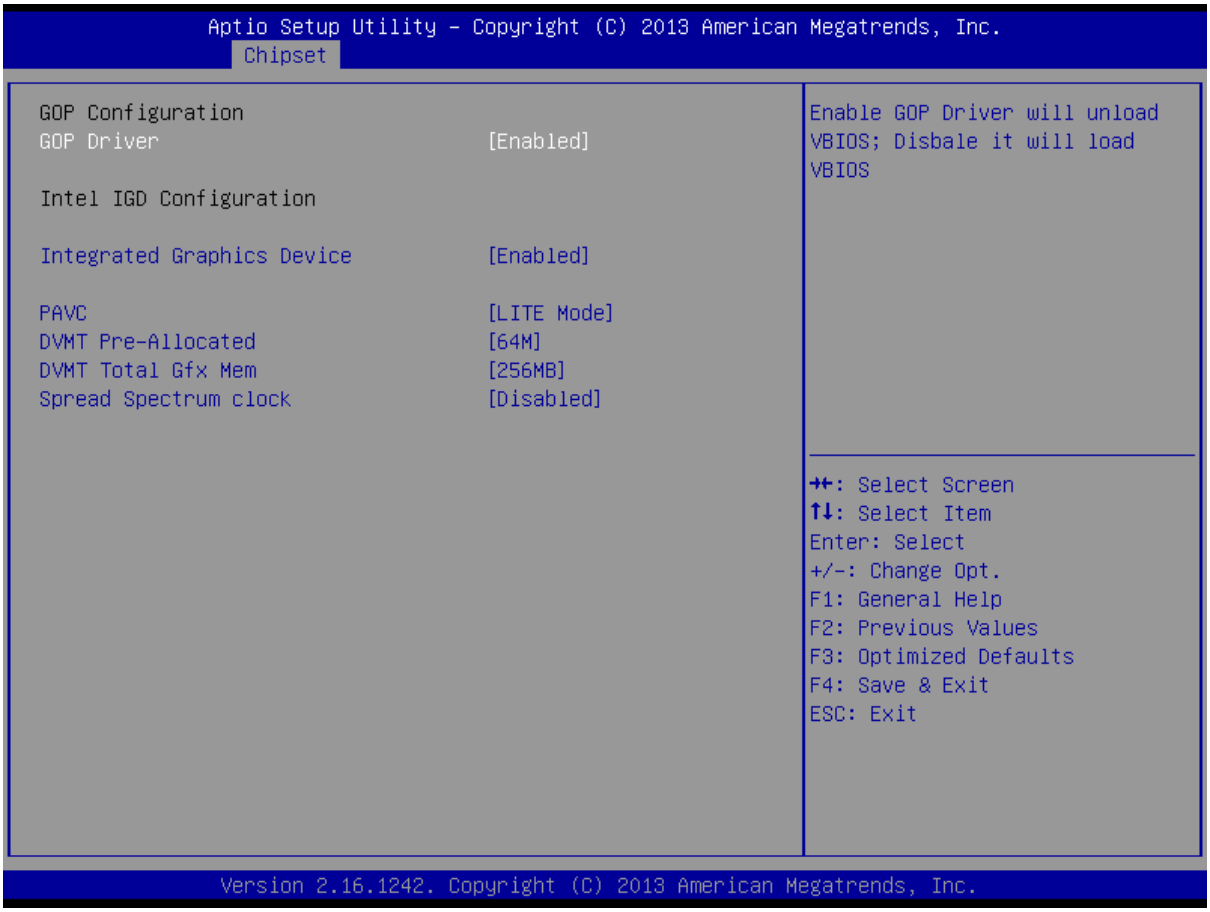
- 3.3.1 North Bridge

Description: This option provides IGD (Integrated Graphic Device) setting and memory information.



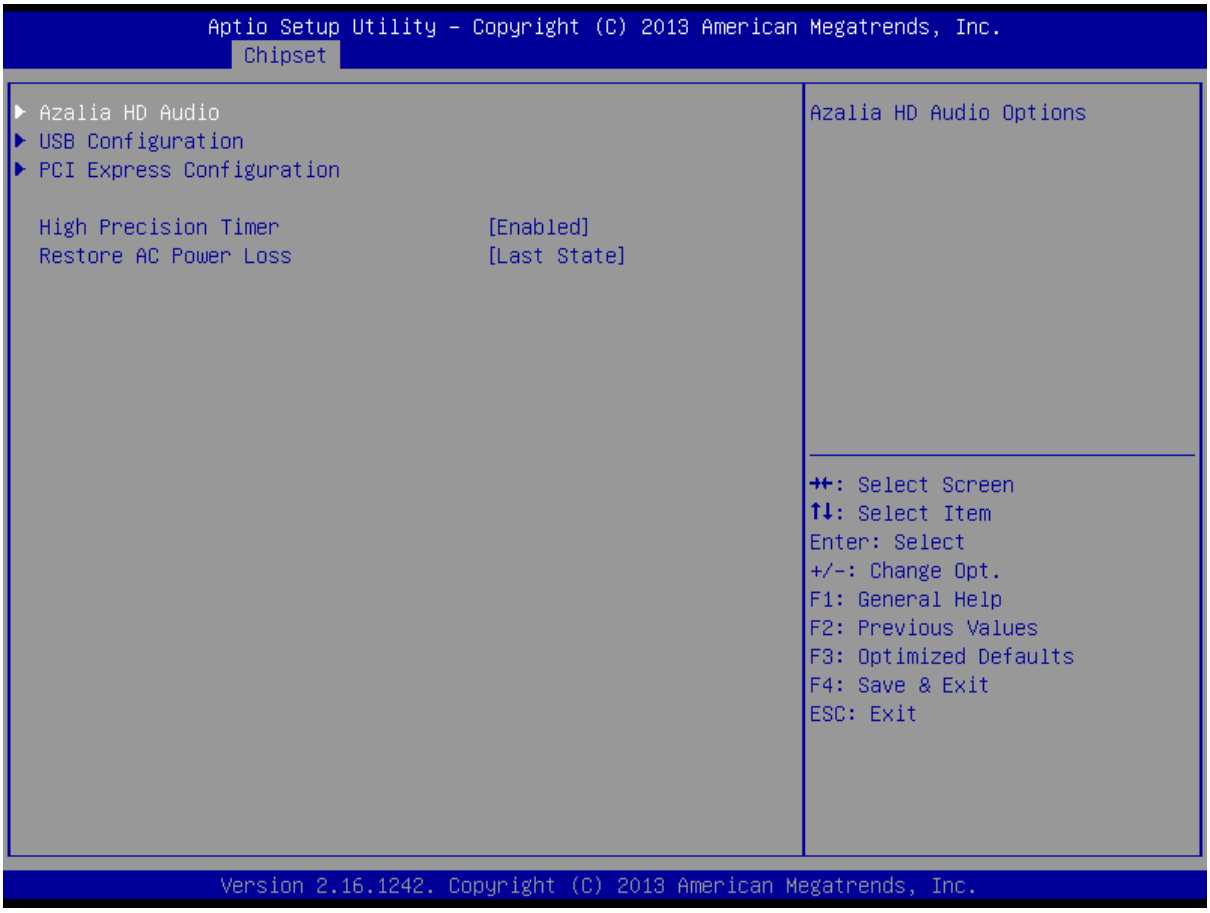
- 3.3.1.1 Intel IGD Configuration

Description: GOP can be enabled by this option, user also allow to configure IGD related setting here.



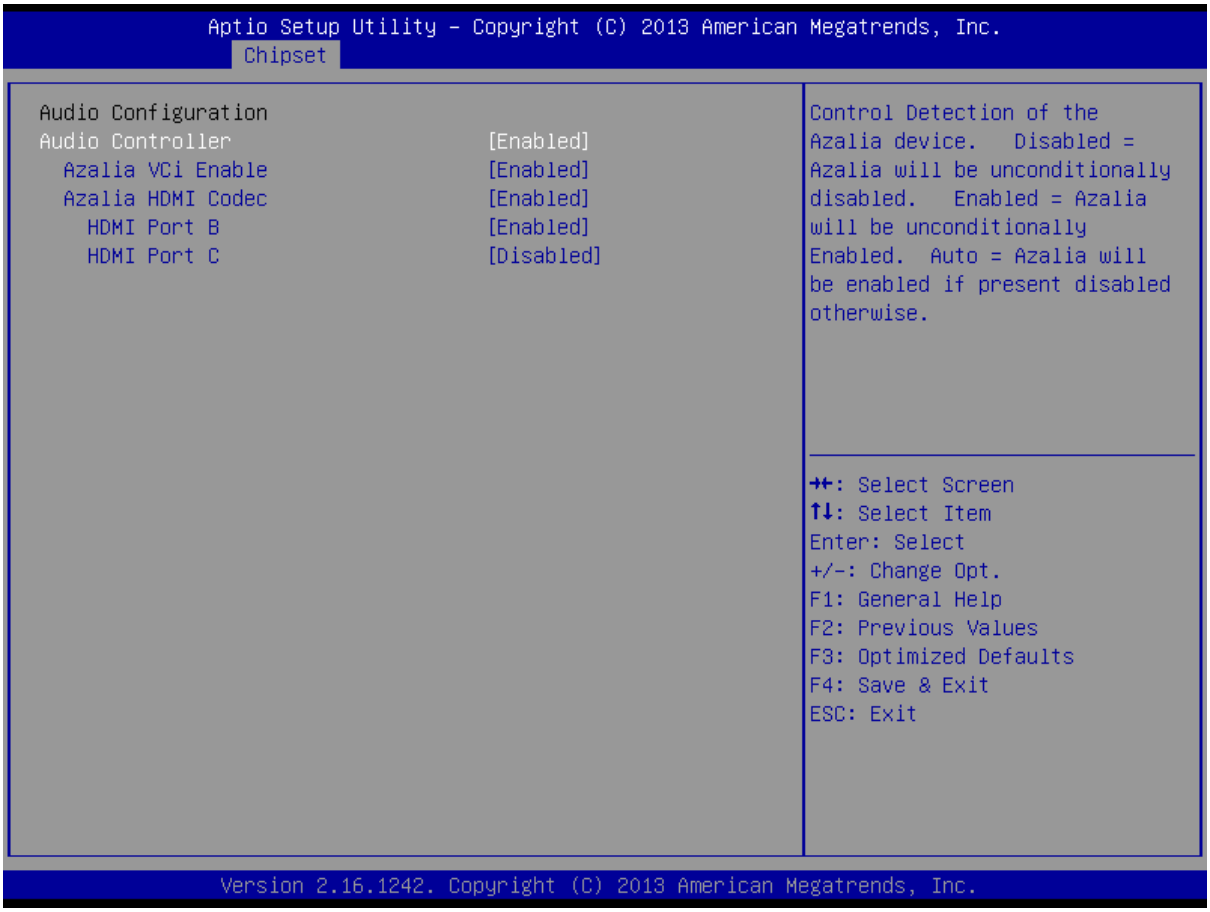
- 3.3.2 South Bridge

Description: In south bridge menu, Audio, USB, PCIe, Timer and Restore can be configured here or further options.



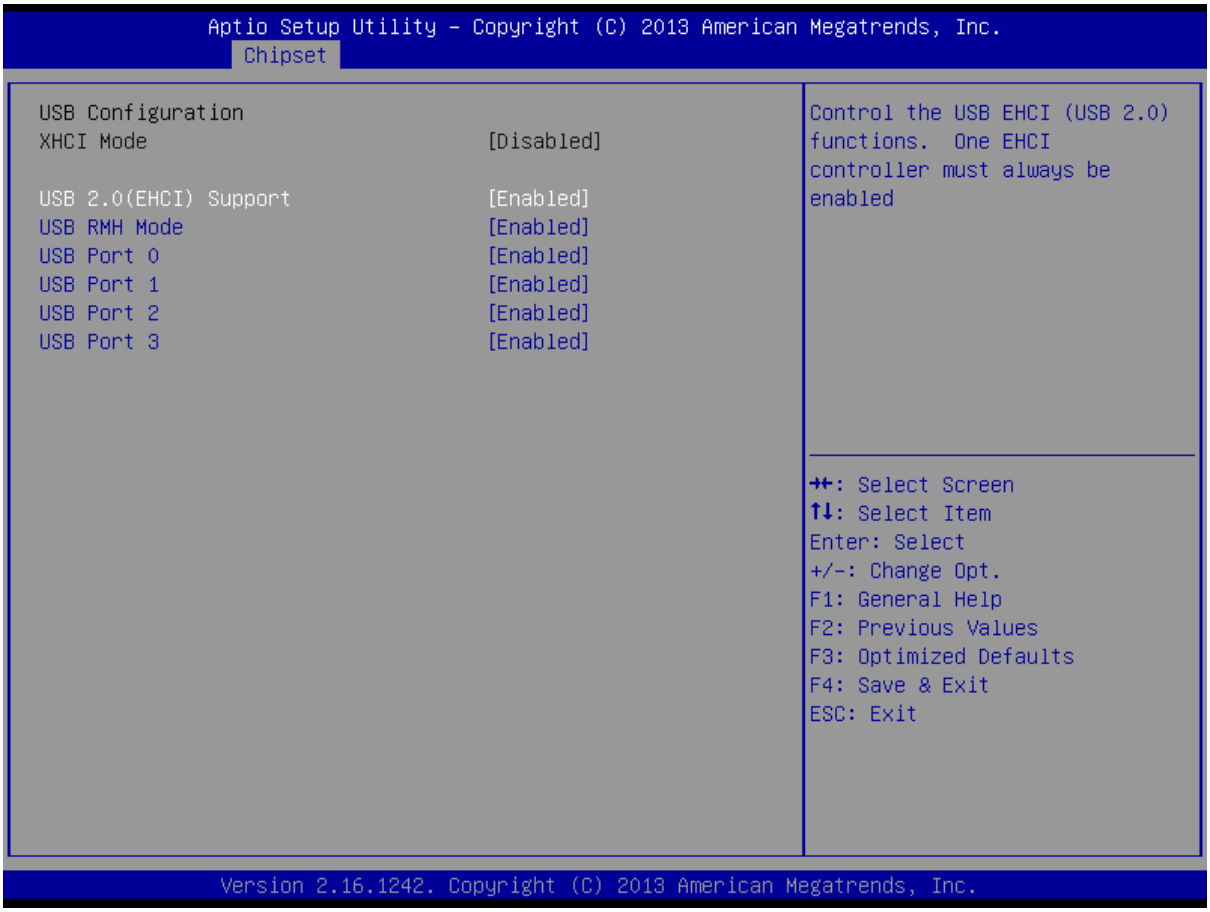
- 3.3.2.1 Azalia HD Audio

Description: Audio Azalia device can be configured at this menu.



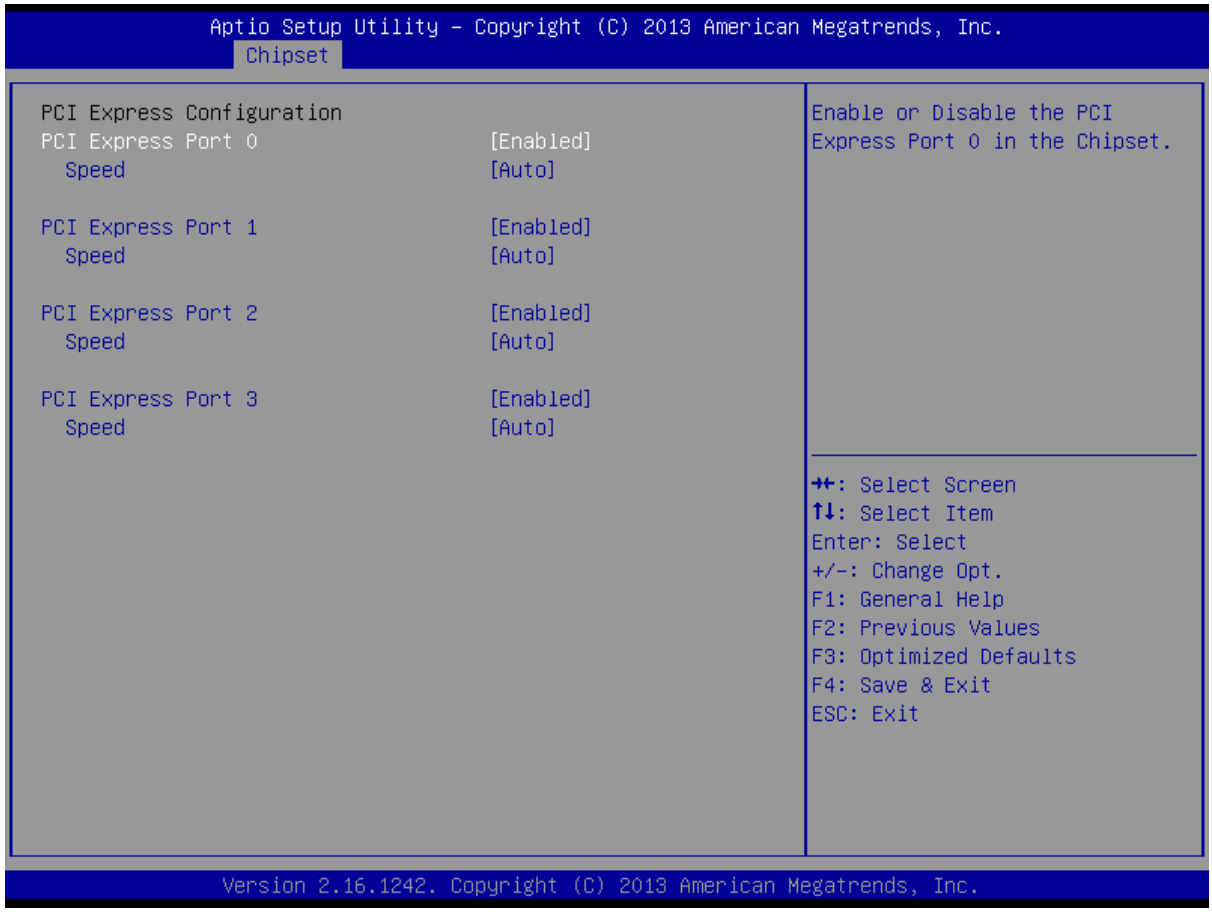
- 3.3.2.2 USB Configuration

Description: USBs configuration can be set at this menu.



- 3.3.2.3 PCI Express Configuration

Description: Function and speed of PCI Express ports can be configured at this menu.

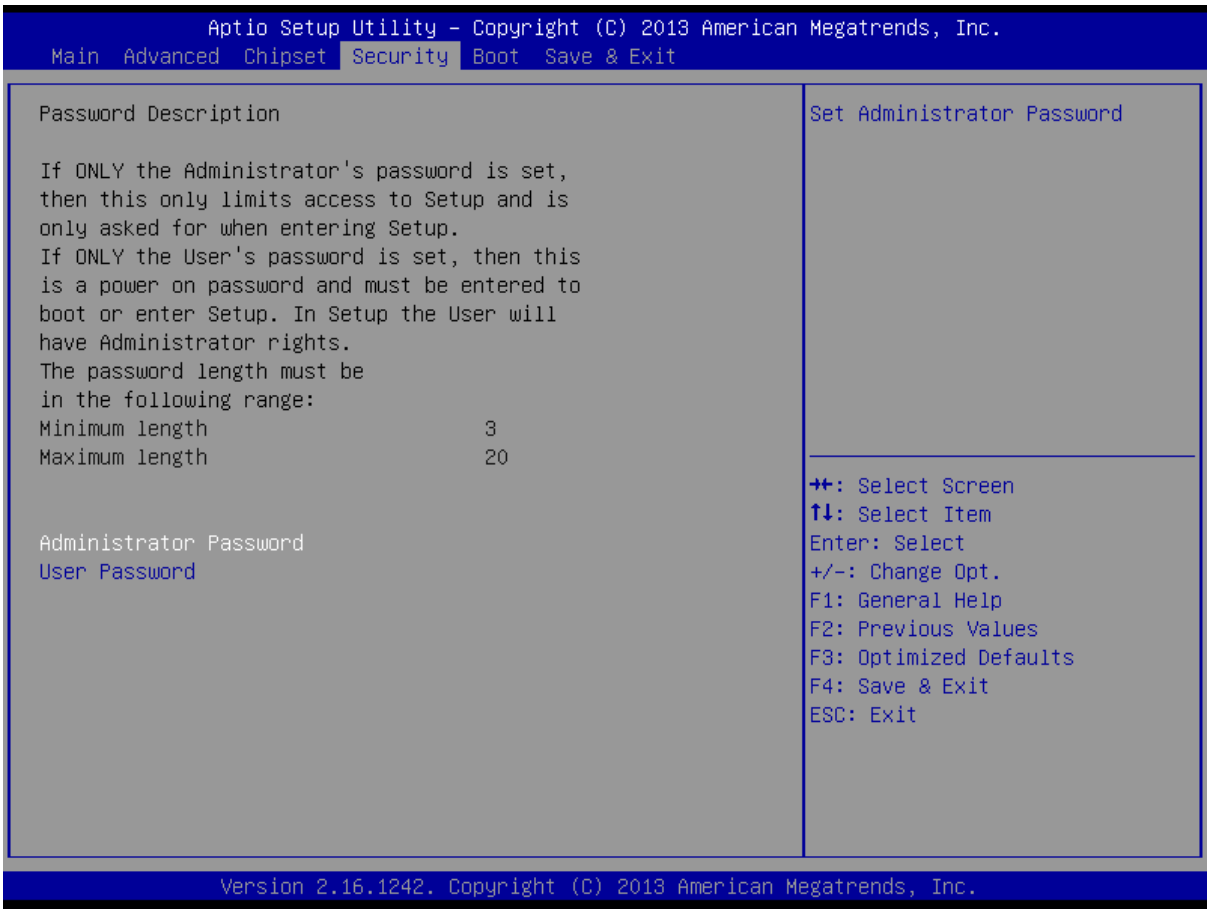


- 3.4 Security Menu

Description:

Administrator Password: Press Enter to create a new, or change an existing Administrator password.

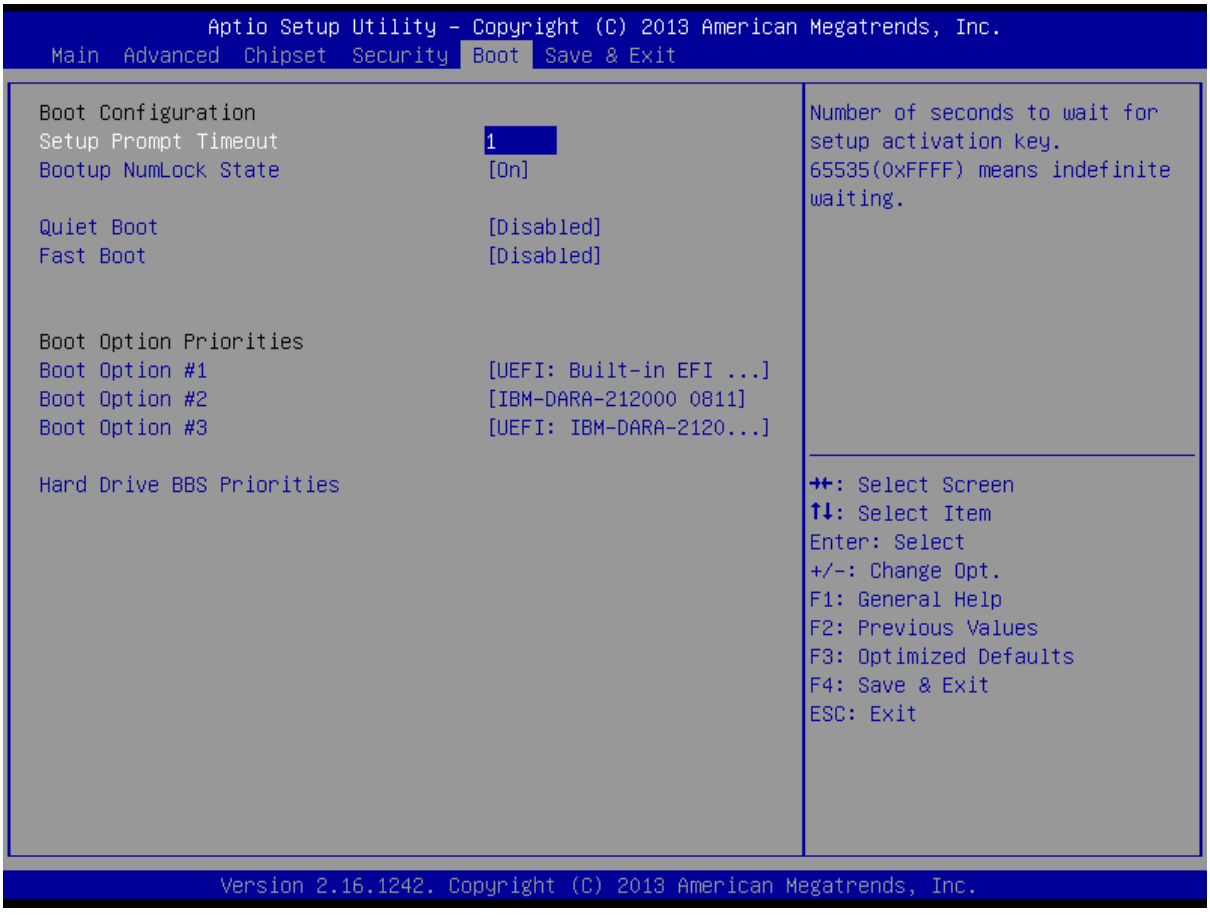
User Password: Press Enter to create a new, or change an existing User password.



- 3.5 Boot Menu

Boot Configuration:
This feature allows the user to specify which devices are boot devices and the order of priority from which the systems boots from during startup.

Quiet Boot:
This feature allows booting with initialization with a minimal set of devices required to launch an active boot option. The options are Disabled, and Enabled.



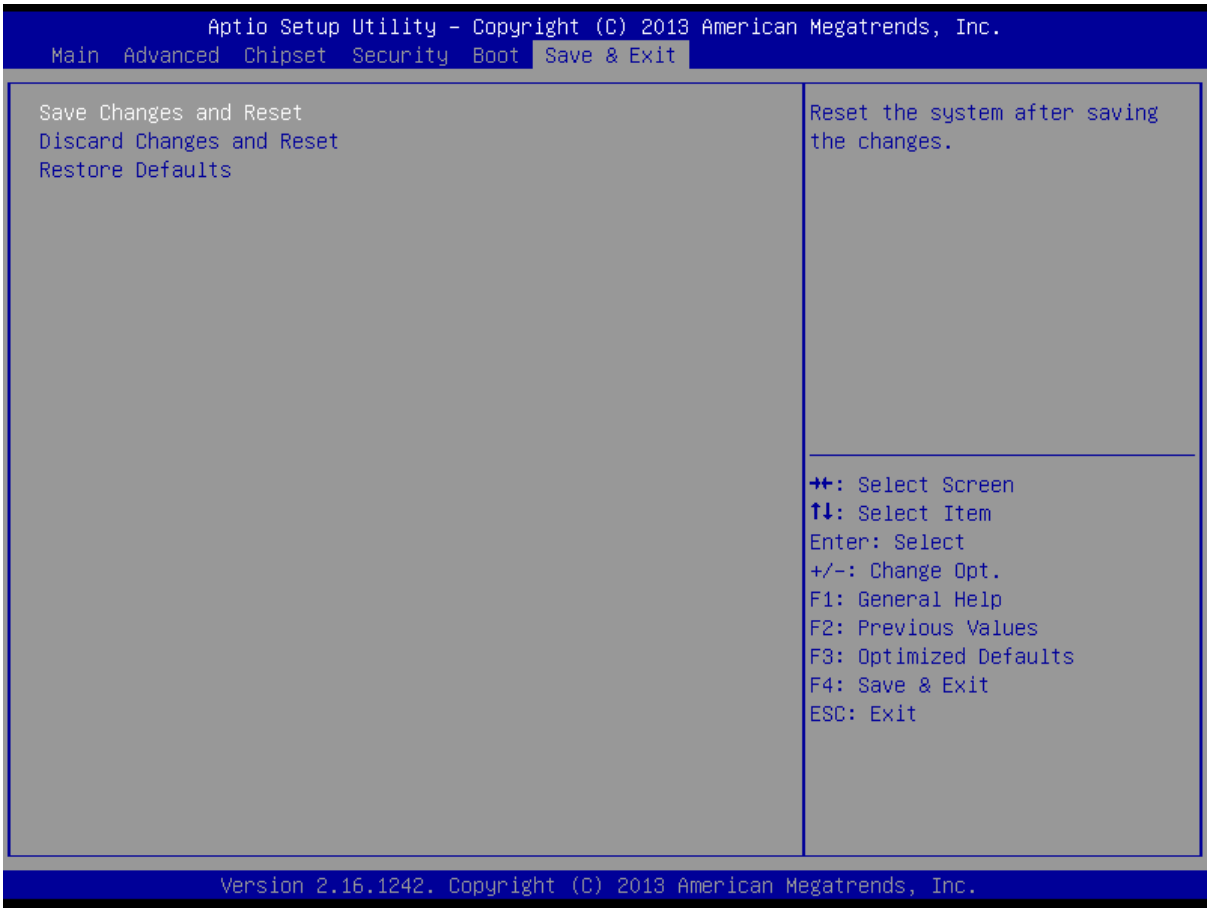
- 3.6 Save & Exit Menu

Description:

Save Change and Reset : When you have completed the system configuration changes, select this option to leave the BIOS Setup Utility and reboot the computer, so the new system configuration parameters can take effect. Select Save Changes and Exit from the Exit menu and press <Enter>

Discard Changes and Reset : Select this option to quit the BIOS Setup without making any permanent changes to the system configuration, and reboot the computer. Select Discard Changes and Exit from the Exit menu and press <Enter>.

Restored Defaults : To set this feature, select Restore Defaults from the Exit menu and press <Enter>. These are factory settings designed for maximum system stability, but not for maximum performance.


Aptio Setup Utility – Copyright (C) 2013 American Megatrends, Inc.
Main Advanced Chipset Security Boot Save & Exit
Save Changes and Reset
Discard Changes and Reset
Restore Defaults
Reset the system after saving the changes.
→←: Select Screen
↑↓: Select Item
Enter: Select
+/-: Change Opt.
F1: General Help
F2: Previous Values
F3: Optimized Defaults
F4: Save & Exit
ESC: Exit
Version 2.16.1242. Copyright (C) 2013 American Megatrends, Inc.

- 3.7 Flash BIOS

To update BIOS, please follow the steps as below to flash BIOS:

1. Press Delete key on BIOS POST time to enter into BIOS setup screen.
2. Select all boot device to Disabled in Boot page.
3. Save Changes and Reset.
4. BIOS will boot to shell.
5. Select your drive contain BIOS bin file and FPT64.efi and fparts.txt(the 3 files need in same folder).
   (normally drive name is : fs0 , fs1 , fs2 .... ; if the files in fs0 just type fs0: then press Enter key)
6. shell command "ls" = DOS command "dir"
7. shell command "cd" = DOS command "cd"
8. Flash BIOS command : fpt64 -f filename.bin
9. Wait for flash BIOS, with success, power off and restart your system.

4. GPIO and WatchDog Timer sample code

- 4.1 GPIO Sample Program for DOS environment

```
//WIN GPIO Program for MB-80580 (Linux Version)
#include <stdio.h>
#include <sys/io.h>/* linux-specific */
#include <stdlib.h> /* linux-specific */
#include <unistd.h>       /* linux-specific */
#include <fcntl.h>  /* linux-specific */
#include <sys/types.h>   /* linux-specific */
#include <sys/mman.h> /* linux-specific */
#include <string.h>

#define index_port  0x4E
#define data_port       0x4f

void help();
void Enter_SIO();
void Exit_SIO();

// Linux-start
void outportb(int reg, int val)
{
            outb_p(val, reg);
}

int    inportb(int reg)
{
            return inb_p(reg);
}

void outportl(int reg, int val)
{
            outl_p(val, reg);
}
```

```
int inportl(int reg)
{
        return inl_p(reg);
}

void delay(int val)
{
         usleep(val*1000);
}
// Linux-end

int    main(int argc, char *argv[])
{
         int data_rw8, val;

         if (argc!=2)
         {
                  help();
                  return;
         }

         iopl(3);

         Enter_SIO();
         if(strcmp(argv[1], "-80h") == 0)
         {
                  val = 0x01;
                  outportb(index_port, 0x89);
                  data_rw8 = inportb(data_port)&(~val);
                  data_rw8 |= val;
                  outportb(data_port, data_rw8);
                  printf("(Set GP80 to high)\n");
         }
         else if(strcmp(argv[1], "-80l") == 0)
         {
                  val = 0x01;
                  outportb(index_port, 0x89);
```

```
        data_rw8 = inportb(data_port)&(~val);
        outportb(data_port, data_rw8);
        printf("(Set GP80 to low)\n");
}
else if(strcmp(argv[1], "-81h") == 0)
{
        val = 0x02;
        outportb(index_port, 0x89);
        data_rw8 = inportb(data_port)&(~val);
        data_rw8 |= val;
        outportb(data_port, data_rw8);
        printf("(Set GP81 to high)\n");
}
else if(strcmp(argv[1], "-81l") == 0)
{
        val = 0x02;
        outportb(index_port, 0x89);
        data_rw8 = inportb(data_port)&(~val);
        outportb(data_port, data_rw8);
        printf("(Set GP81 to low)\n");
}
else if(strcmp(argv[1], "-82h") == 0)
{
        val = 0x04;
        outportb(index_port, 0x89);
        data_rw8 = inportb(data_port)&(~val);
        data_rw8 |= val;
        outportb(data_port, data_rw8);
        printf("(Set GP82 to high)\n");
}
else if(strcmp(argv[1], "-82l") == 0)
{
        val = 0x04;
        outportb(index_port, 0x89);
        data_rw8 = inportb(data_port)&(~val);
        outportb(data_port, data_rw8);
        printf("(Set GP82 to low)\n");
```

```
}
else if(strcmp(argv[1], "-83h") == 0)
{
        val = 0x08;
        outportb(index_port, 0x89);
        data_rw8 = inportb(data_port)&(~val);
        data_rw8 |= val;
        outportb(data_port, data_rw8);
        printf("(Set GP83 to high)\n");
}
else if(strcmp(argv[1], "-83l") == 0)
{
        val = 0x08;
        outportb(index_port, 0x89);
        data_rw8 = inportb(data_port)&(~val);
        outportb(data_port, data_rw8);
        printf("(Set GP83 to low)\n");
}
else if(strcmp(argv[1], "-84h") == 0)
{
        val = 0x10;
        outportb(index_port, 0x89);
        data_rw8 = inportb(data_port)&(~val);
        data_rw8 |= val;
        outportb(data_port, data_rw8);
        printf("(Set GP84 to high)\n");
}
else if(strcmp(argv[1], "-84l") == 0)
{
        val = 0x10;
        outportb(index_port, 0x89);
        data_rw8 = inportb(data_port)&(~val);
        outportb(data_port, data_rw8);
        printf("(Set GP84 to low)\n");
}
else if(strcmp(argv[1], "-85h") == 0)
{
```

```
            val = 0x20;
            outportb(index_port, 0x89);
            data_rw8 = inportb(data_port)&(~val);
            data_rw8 |= val;
            outportb(data_port, data_rw8);
            printf("(Set GP85 to high)\n");
    }
    else if(strcmp(argv[1], "-85l") == 0)
    {
            val = 0x20;
            outportb(index_port, 0x89);
            data_rw8 = inportb(data_port)&(~val);
            outportb(data_port, data_rw8);
            printf("(Set GP85 to low)\n");
    }
    else if(strcmp(argv[1], "-86h") == 0)
    {
            val = 0x40;
            outportb(index_port, 0x89);
            data_rw8 = inportb(data_port)&(~val);
            data_rw8 |= val;
            outportb(data_port, data_rw8);
            printf("(Set GP86 to high)\n");
    }
    else if(strcmp(argv[1], "-86l") == 0)
    {
            val = 0x40;
            outportb(index_port, 0x89);
            data_rw8 = inportb(data_port)&(~val);
            outportb(data_port, data_rw8);
            printf("(Set GP86 to low)\n");
    }
    else if(strcmp(argv[1], "-87h") == 0)
    {
            val = 0x80;
            outportb(index_port, 0x89);
            data_rw8 = inportb(data_port)&(~val);
```

```
            data_rw8 |= val;
            outportb(data_port, data_rw8);
            printf("(Set GP87 to high)\n");
    }
    else if(strcmp(argv[1], "-87l") == 0)
    {
            val = 0x80;
            outportb(index_port, 0x89);
            data_rw8 = inportb(data_port)&(~val);
            outportb(data_port, data_rw8);
            printf("(Set GP87 to low)\n");
    }
    else if(strcmp(argv[1], "-hhh") == 0)
    {
            val = 0xFF;
            outportb(index_port, 0x89);
            data_rw8 = inportb(data_port);
            data_rw8 |= val;
            outportb(data_port, data_rw8);
            printf("(Set GP80~87 to high)\n");
    }
    else if(strcmp(argv[1], "-lll") == 0)
    {
            val = 0x00;
            outportb(index_port, 0x89);
            data_rw8 = inportb(data_port);
            data_rw8 &= val;
            outportb(data_port, data_rw8);
            printf("(Set GP80~87 to low)\n");
    }
    else if(strcmp(argv[1], "-lte") == 0)
    {
            val = 0x0F;
            outportb(index_port, 0x89);
            data_rw8 = inportb(data_port)&0x00;
            data_rw8 |= val;
            outportb(data_port, data_rw8);
```

```
        printf("(Set GP80~83 to high, GP84~87 to low)\n");
}
else if(strcmp(argv[1], "-ltd") == 0)
{
        val = 0xF0;
        outportb(index_port, 0x89);
        data_rw8 = inportb(data_port)&0x00;
        data_rw8 |= val;
        outportb(data_port, data_rw8);
        printf("(Set GP80~83 to low, GP84~87 to high)\n");
}
else if(strcmp(argv[1], "-gpr") == 0)
{
        outportb(index_port, 0x8A);

        val = 0x01;
        data_rw8 = inportb(data_port)&(val);
        if (data_rw8==val)
                printf("(GP80 is high)\n");
        else printf("(GP80 is low)\n");

        val = 0x02;
        data_rw8 = inportb(data_port)&(val);
        if (data_rw8==val)
                printf("(GP81 is high)\n");
        else printf("(GP81 is low)\n");

        val = 0x04;
        data_rw8 = inportb(data_port)&(val);
        if (data_rw8==val)
                printf("(GP82 is high)\n");
        else printf("(GP82 is low)\n");

        val = 0x08;
        data_rw8 = inportb(data_port)&(val);
        if (data_rw8==val)
                printf("(GP83 is high)\n");
```

```
                else printf("(GP83 is low)\n");

                val = 0x10;
                data_rw8 = inportb(data_port)&(val);
                if (data_rw8==val)
                        printf("(GP84 is high)\n");
                else printf("(GP84 is low)\n");

                val = 0x20;
                data_rw8 = inportb(data_port)&(val);
                if (data_rw8==val)
                        printf("(GP85 is high)\n");
                else printf("(GP85 is low)\n");

                val = 0x40;
                data_rw8 = inportb(data_port)&(val);
                if(data_rw8==val)
                        printf("(GP86 is high)\n");
                else printf("(GP86 is low)\n");

                val = 0x80;
                data_rw8 = inportb(data_port)&(val);
                if (data_rw8==val)
                        printf("(GP87 is high)\n");
                else printf("(GP87 is low)\n");
        }
        else
        {
                help();
        }
        Exit_SIO();

    iopl(0);

        return 0;
}
```

```
void Enter_SIO()
{
        outportb(index_port, 0x87);
        delay(1);
        outportb(index_port, 0x87);
        outportb(index_port, 0x07);
        outportb(data_port,    0x06);
}

void Exit_SIO()
{
        outportb(index_port, 0xAA);
}

void help()
{
        printf("WIN GPIO Test Program\n");
        printf("=============================================\n");
        printf("GPIO -hhh (Set GP80~87 to high)\n");
        printf("GPIO -lll (Set GP80~87 to low )\n");
        printf("GPIO -lte (Set GP80~83 to high, GP84~87 to low)\n");
        printf("GPIO -ltd (Set GP80~83 to low, GP84~87 to high)\n");
        printf("GPIO -gpr (Read GPIO status)\n");
        printf("GPIO -xxh (Set xx to high)\n");
        printf("GPIO -xxl (Set xx to low)\n");
        printf("          (xx=80~87)\n");
        printf("=============================================\n");
}
```

4.2 Watchdog timer Sample Program for DOS environment

```
//WIN Watch dog program for MB-80580 (Linux Version) PCB:
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <sys/io.h> /* linux-specific */
#include <unistd.h>

#define index_port  0x4E                    //Super IO Index port address
#define data_port    0x4F                   //Super IO Data port address

void Enter_sio_config();
void Exit_sio_config();
void help();

//Linux-start
void outportb(int reg, int val)
{
      outb_p(val, reg);
}

int inportb(int reg)
{
      return inb_p(reg);
}

void outportl(int reg, int val)
{
      outl_p(val, reg);
}

int inportl(int reg)
{
      return inl_p(reg);
}
```

```
void delay(int val)
{
	usleep(val*1000);
}
//Linux-end


int main(int argc, char *argv[])
{
	int data_rw8, time;

		if (argc<2){
		help();
		return 0;
	}

	iopl(3);

		if(strcmp(argv[1], "-s") == 0){
			//Show Watchdog Register Settings

		Enter_sio_config();
		outportb(index_port, 0xF5);
		data_rw8 = inportb(data_port)&0x08;
		if(data_rw8 == 0x00){
			//second mode
			outportb(index_port, 0xF6);
			data_rw8 = inportb(data_port);
			printf("Second mode: %d second\n", data_rw8);
		}
		else{
			//minute mode
			outportb(index_port, 0xF6);
			data_rw8 = inportb(data_port);
			printf("Minute mode: %d minute\n", data_rw8);
		}
```

```
        }
else if(strcmp(argv[1], "-t") ==0 ){
        //Set Time-out Value
        if(argv[2] == NULL){
                help();
                return;
        }
        else{
                Enter_sio_config();
                // 2014/06/19 Debug
                //outportb(index_port, 0xF6);
                //outportb(data_port , 0x00);                   //Clear Status
                outportb(index_port, 0xF5);
                // 2014/06/19 Debug ;
                //data_rw8 = 0x32;
                data_rw8 = 0x12;                                //Disable WDT
                outportb(data_port, data_rw8);
                // 2014/06/19 Debug
                outportb(index_port, 0xFA);                     //Disable reset function
                outportb(data_port , 0x00);

                sscanf(argv[2], "%d", &time);
                outportb(index_port, 0xF6);
                outportb(data_port,    time);
                if(time==0){
                        //Disable count
                        outportb(index_port, 0xF5);
                        data_rw8 = inportb(data_port)&0xCF;
                        outportb(data_port , data_rw8);
                }
                else{
                        // 2014/06/19 Debug
                        outportb(index_port, 0xF5);
                        data_rw8 = inportb(data_port) | 0x20;
                        outportb(data_port , data_rw8);
                        outportb(index_port, 0xFA);
                        outportb(data_port , 0x01);             //Enable reset function
```

```
            }
                printf("Watchdog Timer will count down for %d second(s)\n", time);
        }
    }
    else if(strcmp(argv[1], "-m") ==0 ){
        //Set Time-out Value
        if(argv[2] == NULL){
            help();
            return;
        }
        else{
            Enter_sio_config();
            outportb(index_port, 0xF6);
            outportb(data_port , 0x00);                    //Clear Status
            outportb(index_port, 0xF5);
            data_rw8 = 0x32;
            data_rw8 |= 0x08;
            outportb(data_port, data_rw8);
                sscanf(argv[2], "%d", &time);
            outportb(index_port, 0xF6);
            outportb(data_port,    time);
            if(time==0){
                //Disable count
                outportb(index_port, 0xF5);
                data_rw8 = inportb(data_port)&0xCF;
                outportb(data_port , data_rw8);
            }
            else{
                outportb(index_port, 0xFA);
                outportb(data_port , 0x01);                //Enable reset function
            }
                printf("Watchdog Timer will count down for %d minute(s)\n", time);
        }
    }
    Exit_sio_config();
    return;
}
```

```
void Enter_sio_config()                         //Enter W83627EHF Configuration
{
      outportb(index_port, 0x87);
      delay(1);                          //Delay some time
      outportb(index_port, 0x87);
      delay(1);
      outportb(index_port, 0x07);               //Super IO Selct Bank Register Number
      delay(1);
      outportb(data_port , 0x07);               //Select logical device 7
}

void Exit_sio_config()                          //Exit W83627EHF Configuration
{
      outportb(index_port, 0xAA);
}

void help()
{
      printf("WIN Watchdog Timer Program\n");
      printf("Usage: WDT -s        (Show Watchdog Register Settings)\n");
      printf("Usage: WDT -t xxx (Set Time-out Value)\n");
      printf("                   xxx = 1 ~ 255 seconds\n");
      printf("                   xxx = 0 : Time-out Disable \n");
      printf("Usage: WDT -m xxx (Set Time-out Value)\n");
      printf("                   xxx = 1 ~ 255 minutes\n");
      printf("                   xxx = 0 : Time-out Disable \n");
}
```

Device Resources

		Resource	**Share**	**Device Description**
		IRQ 00	Exclusive	System timer
		IRQ 01	Exclusive	Standard PS/2 Keyboard
		IRQ 03	Exclusive	Communications Port (COM2)
		IRQ 04	Exclusive	Communications Port (COM1)
		IRQ 05	Shared	Intel(R) Atom(TM)/Celeron(R)/Pentium(R) Processor Platform Control Unit - SMBus Port - 0F12
		IRQ 05	Shared	Intel(R) Atom(TM)/Celeron(R)/Pentium(R) Processor Trusted Execution Engine Interface - 0F18
		IRQ 07	Exclusive	Communications Port (COM3)
		IRQ 08	Exclusive	High precision event timer
		IRQ 100	Exclusive	Microsoft ACPI-Compliant System
		IRQ 101	Exclusive	Microsoft ACPI-Compliant System
		IRQ 102	Exclusive	Microsoft ACPI-Compliant System
		IRQ 103	Exclusive	Microsoft ACPI-Compliant System
		IRQ 104	Exclusive	Microsoft ACPI-Compliant System
		IRQ 105	Exclusive	Microsoft ACPI-Compliant System
		IRQ 106	Exclusive	Microsoft ACPI-Compliant System
		IRQ 107	Exclusive	Microsoft ACPI-Compliant System
		IRQ 108	Exclusive	Microsoft ACPI-Compliant System
		IRQ 109	Exclusive	Microsoft ACPI-Compliant System
		IRQ 11	Shared	Communications Port (COM4)
		IRQ 110	Exclusive	Microsoft ACPI-Compliant System
		IRQ 111	Exclusive	Microsoft ACPI-Compliant System
		IRQ 112	Exclusive	Microsoft ACPI-Compliant System
		IRQ 113	Exclusive	Microsoft ACPI-Compliant System

		IRQ 114	Exclusive	Microsoft ACPI-Compliant System
		IRQ 115	Exclusive	Microsoft ACPI-Compliant System
		IRQ 116	Exclusive	Microsoft ACPI-Compliant System
		IRQ 117	Exclusive	Microsoft ACPI-Compliant System
		IRQ 118	Exclusive	Microsoft ACPI-Compliant System
		IRQ 119	Exclusive	Microsoft ACPI-Compliant System
		IRQ 12	Exclusive	Microsoft PS/2 Mouse
		IRQ 120	Exclusive	Microsoft ACPI-Compliant System
		IRQ 121	Exclusive	Microsoft ACPI-Compliant System
		IRQ 122	Exclusive	Microsoft ACPI-Compliant System
		IRQ 123	Exclusive	Microsoft ACPI-Compliant System
		IRQ 124	Exclusive	Microsoft ACPI-Compliant System
		IRQ 125	Exclusive	Microsoft ACPI-Compliant System
		IRQ 126	Exclusive	Microsoft ACPI-Compliant System
		IRQ 127	Exclusive	Microsoft ACPI-Compliant System
		IRQ 128	Exclusive	Microsoft ACPI-Compliant System
		IRQ 129	Exclusive	Microsoft ACPI-Compliant System
		IRQ 130	Exclusive	Microsoft ACPI-Compliant System
		IRQ 131	Exclusive	Microsoft ACPI-Compliant System
		IRQ 132	Exclusive	Microsoft ACPI-Compliant System
		IRQ 133	Exclusive	Microsoft ACPI-Compliant System
		IRQ 134	Exclusive	Microsoft ACPI-Compliant System
		IRQ 135	Exclusive	Microsoft ACPI-Compliant System
		IRQ 136	Exclusive	Microsoft ACPI-Compliant System
		IRQ 137	Exclusive	Microsoft ACPI-Compliant System
		IRQ 138	Exclusive	Microsoft ACPI-Compliant System
		IRQ 139	Exclusive	Microsoft ACPI-Compliant System
		IRQ 140	Exclusive	Microsoft ACPI-Compliant System
		IRQ 141	Exclusive	Microsoft ACPI-Compliant System
		IRQ 142	Exclusive	Microsoft ACPI-Compliant System

		IRQ 143	Exclusive	Microsoft ACPI-Compliant System
		IRQ 144	Exclusive	Microsoft ACPI-Compliant System
		IRQ 145	Exclusive	Microsoft ACPI-Compliant System
		IRQ 146	Exclusive	Microsoft ACPI-Compliant System
		IRQ 147	Exclusive	Microsoft ACPI-Compliant System
		IRQ 148	Exclusive	Microsoft ACPI-Compliant System
		IRQ 149	Exclusive	Microsoft ACPI-Compliant System
		IRQ 150	Exclusive	Microsoft ACPI-Compliant System
		IRQ 151	Exclusive	Microsoft ACPI-Compliant System
		IRQ 152	Exclusive	Microsoft ACPI-Compliant System
		IRQ 153	Exclusive	Microsoft ACPI-Compliant System
		IRQ 154	Exclusive	Microsoft ACPI-Compliant System
		IRQ 155	Exclusive	Microsoft ACPI-Compliant System
		IRQ 156	Exclusive	Microsoft ACPI-Compliant System
		IRQ 157	Exclusive	Microsoft ACPI-Compliant System
		IRQ 158	Exclusive	Microsoft ACPI-Compliant System
		IRQ 159	Exclusive	Microsoft ACPI-Compliant System
		IRQ 16	Shared	Intel(R) Atom(TM)/Celeron(R)/Pentium(R) Processor PCI Express - Root Port 1 - 0F48
		IRQ 160	Exclusive	Microsoft ACPI-Compliant System
		IRQ 161	Exclusive	Microsoft ACPI-Compliant System
		IRQ 162	Exclusive	Microsoft ACPI-Compliant System
		IRQ 163	Exclusive	Microsoft ACPI-Compliant System
		IRQ 164	Exclusive	Microsoft ACPI-Compliant System
		IRQ 165	Exclusive	Microsoft ACPI-Compliant System
		IRQ 166	Exclusive	Microsoft ACPI-Compliant System
		IRQ 167	Exclusive	Microsoft ACPI-Compliant System
		IRQ 168	Exclusive	Microsoft ACPI-Compliant System
		IRQ 169	Exclusive	Microsoft ACPI-Compliant System
		IRQ 17	Shared	Intel(R) Atom(TM)/Celeron(R)/Pentium(R) Processor PCI Express - Root Port 2 - 0F4A

		IRQ 170	Exclusive	Microsoft ACPI-Compliant System
		IRQ 171	Exclusive	Microsoft ACPI-Compliant System
		IRQ 172	Exclusive	Microsoft ACPI-Compliant System
		IRQ 173	Exclusive	Microsoft ACPI-Compliant System
		IRQ 174	Exclusive	Microsoft ACPI-Compliant System
		IRQ 175	Exclusive	Microsoft ACPI-Compliant System
		IRQ 176	Exclusive	Microsoft ACPI-Compliant System
		IRQ 177	Exclusive	Microsoft ACPI-Compliant System
		IRQ 178	Exclusive	Microsoft ACPI-Compliant System
		IRQ 179	Exclusive	Microsoft ACPI-Compliant System
		IRQ 18	Shared	Intel(R) Pentium(R) processor N- and J-series / Intel(R) Celeron(R) processor N- and J-series PCI Express - Root Port 3 - 0F4C
		IRQ 180	Exclusive	Microsoft ACPI-Compliant System
		IRQ 181	Exclusive	Microsoft ACPI-Compliant System
		IRQ 182	Exclusive	Microsoft ACPI-Compliant System
		IRQ 183	Exclusive	Microsoft ACPI-Compliant System
		IRQ 184	Exclusive	Microsoft ACPI-Compliant System
		IRQ 185	Exclusive	Microsoft ACPI-Compliant System
		IRQ 186	Exclusive	Microsoft ACPI-Compliant System
		IRQ 187	Exclusive	Microsoft ACPI-Compliant System
		IRQ 188	Exclusive	Microsoft ACPI-Compliant System
		IRQ 189	Exclusive	Microsoft ACPI-Compliant System
		IRQ 19	Shared	Intel(R) Pentium(R) processor N- and J-series / Intel(R) Celeron(R) processor N- and J-series PCI Express - Root Port 4 - 0F4E
		IRQ 19	Shared	Intel(R) Atom(TM)/Celeron(R)/Pentium(R) Processor AHCI - 0F23
		IRQ 190	Exclusive	Microsoft ACPI-Compliant System
		IRQ 22	Shared	High Definition Audio Controller
		IRQ 23	Shared	Intel(R) Atom(TM)/Celeron(R)/Pentium(R) Processor EHCI USB - 0F34
		IRQ 65536	Exclusive	Intel(R) I210 Gigabit Network Connection
		IRQ 65536	Exclusive	Intel(R) I210 Gigabit Network Connection

		IRQ 65536	Exclusive	Intel(R) I210 Gigabit Network Connection
		IRQ 65536	Exclusive	Intel(R) I210 Gigabit Network Connection
		IRQ 65536	Exclusive	Intel(R) I210 Gigabit Network Connection
		IRQ 65536	Exclusive	Intel(R) I210 Gigabit Network Connection
		IRQ 65536	Exclusive	Intel(R) I210 Gigabit Network Connection #2
		IRQ 65536	Exclusive	Intel(R) I210 Gigabit Network Connection #2
		IRQ 65536	Exclusive	Intel(R) I210 Gigabit Network Connection #2
		IRQ 65536	Exclusive	Intel(R) I210 Gigabit Network Connection #2
		IRQ 65536	Exclusive	Intel(R) I210 Gigabit Network Connection #2
		IRQ 65536	Exclusive	Intel(R) I210 Gigabit Network Connection #2
		IRQ 65536	Exclusive	Intel(R) Atom(TM) Processor E3800 Series/Intel(R) Celeron(R) Processor N2920/J1900
		IRQ 81	Exclusive	Microsoft ACPI-Compliant System
		IRQ 82	Exclusive	Microsoft ACPI-Compliant System
		IRQ 83	Exclusive	Microsoft ACPI-Compliant System
		IRQ 84	Exclusive	Microsoft ACPI-Compliant System
		IRQ 85	Exclusive	Microsoft ACPI-Compliant System
		IRQ 86	Exclusive	Microsoft ACPI-Compliant System
		IRQ 87	Exclusive	Microsoft ACPI-Compliant System
		IRQ 88	Exclusive	Microsoft ACPI-Compliant System
		IRQ 89	Exclusive	Microsoft ACPI-Compliant System
		IRQ 90	Exclusive	Microsoft ACPI-Compliant System
		IRQ 91	Exclusive	Microsoft ACPI-Compliant System
		IRQ 92	Exclusive	Microsoft ACPI-Compliant System
		IRQ 93	Exclusive	Microsoft ACPI-Compliant System
		IRQ 94	Exclusive	Microsoft ACPI-Compliant System
		IRQ 95	Exclusive	Microsoft ACPI-Compliant System
		IRQ 96	Exclusive	Microsoft ACPI-Compliant System
		IRQ 97	Exclusive	Microsoft ACPI-Compliant System
		IRQ 98	Exclusive	Microsoft ACPI-Compliant System

		IRQ 99	Exclusive	Microsoft ACPI-Compliant System
		Memory 000A0000-000BFFFF	Shared	Intel(R) Atom(TM) Processor E3800 Series/Intel(R) Celeron(R) Processor N2920/J1900
		Memory 000A0000-000BFFFF	Shared	PCI bus
		Memory 000C0000-000DFFFF	Shared	PCI bus
		Memory 000E0000-000FFFFF	Shared	PCI bus
		Memory 80000000-8FFFFFFF	Exclusive	Intel(R) Atom(TM) Processor E3800 Series/Intel(R) Celeron(R) Processor N2920/J1900
		Memory 80000000-90C06FFE	Shared	PCI bus
		Memory 90000000-903FFFFF	Exclusive	Intel(R) Atom(TM) Processor E3800 Series/Intel(R) Celeron(R) Processor N2920/J1900
		Memory 90400000-904FFFFF	Exclusive	Intel(R) Atom(TM)/Celeron(R)/Pentium(R) Processor Trusted Execution Engine Interface - 0F18
		Memory 90500000-905FFFFF	Exclusive	Intel(R) Atom(TM)/Celeron(R)/Pentium(R) Processor Trusted Execution Engine Interface - 0F18
		Memory 90600000-908FFFFF	Exclusive	Intel(R) Atom(TM)/Celeron(R)/Pentium(R) Processor PCI Express - Root Port 2 - 0F4A
		Memory 90700000-907FFFFF	Exclusive	Intel(R) I210 Gigabit Network Connection
		Memory 90800000-90803FFF	Exclusive	Intel(R) I210 Gigabit Network Connection
		Memory 90900000-90BFFFFF	Exclusive	Intel(R) Atom(TM)/Celeron(R)/Pentium(R) Processor PCI Express - Root Port 1 - 0F48
		Memory 90A00000-90AFFFFF	Exclusive	Intel(R) I210 Gigabit Network Connection #2
		Memory 90B00000-90B03FFF	Exclusive	Intel(R) I210 Gigabit Network Connection #2
		Memory 90C00000-90C03FFF	Exclusive	High Definition Audio Controller

		Memory 90C04000-90C0401F	Exclusive	Intel(R) Atom(TM)/Celeron(R)/Pentium(R) Processor Platform Control Unit - SMBus Port - 0F12
		Memory 90C05000-90C053FF	Exclusive	Intel(R) Atom(TM)/Celeron(R)/Pentium(R) Processor EHCI USB - 0F34
		Memory 90C06000-90C067FF	Exclusive	Intel(R) Atom(TM)/Celeron(R)/Pentium(R) Processor AHCI - 0F23
		Memory E0000000-EFFFFFFF	Exclusive	Motherboard resources
		Memory FED00000-FED003FF	Exclusive	High precision event timer
		Memory FED01000-FED01FFF	Exclusive	Motherboard resources
		Memory FED03000-FED03FFF	Exclusive	Motherboard resources
		Memory FED04000-FED04FFF	Exclusive	Motherboard resources
		Memory FED08000-FED08FFF	Exclusive	Motherboard resources
		Memory FED1C000-FED1CFFF	Exclusive	Motherboard resources
		Memory FEE00000-FEEFFFFF	Exclusive	Motherboard resources
		Memory FEF00000-FEFFFFFF	Exclusive	Motherboard resources
		Memory FF000000-FFFFFFFF	Exclusive	Intel(R) 82802 Firmware Hub Device

sales@win-ent.com