CI370D

Intel® Coffee Lake-S Core™ I processor + Intel Q370, DDR4 LAN / Fiber / DP / HDMI / USB / PCIe / M.2

All-In-One

8th gen. Intel Coffee Lake-S Core™ I CPU

HDMI, DP, eDP, PCIe, PCIe mini card, M.2, SIM

Multi-LAN, Fiber, COM, Audio, SATA, USB

CAUTION

RISK OF EXPLOSION IF BATTERY IS REPLACED BY AN INCORRECT TYPE.
DISPOSE OF USED BATTERIES ACCORDING TO THE INSTRUCTIONS

NO. C370D
Release date: NOV. 21. 2019

Contents

Trademarks

User Manual edition 0.1, NOV. 21. 2019

Warning !

1. Battery
   Batteries on board are consumables.
   The life time of them are not guaranteed.

2. Fanless solution with HDD
   The specification & limitation of HDD should be considered carefully when the fanless solution is implemented.

3. We will not give further notification in case of changes of product information and manual.

4. SATA interface does not support Hot SWAP function.

5. There might be a 20% inaccuracy of WDT at room temperature.

6. Please make sure the voltage specification meets the requirement of equipment before plugging in.

7. Caution! Please notice that the heat dissipation problem could cause the MB system unstable. Please deal with heat dissipation properly when buying single MB set.

8. Please avoid approaching the heat sink area to prevent users from being scalded with fanless products.

9. If users repair, modify or destroy any component of product unauthorizedly, We will not take responsibility or provide warranty anymore.

10. DO NOT apply any other material which may reduce cooling performance onto the thermal pad.

11. It is important to install a system fan toward the CPU to decrease the possibility of overheating/system hanging up issues, or customer is suggested to have a fine cooling system to dissipate heat from CPU.

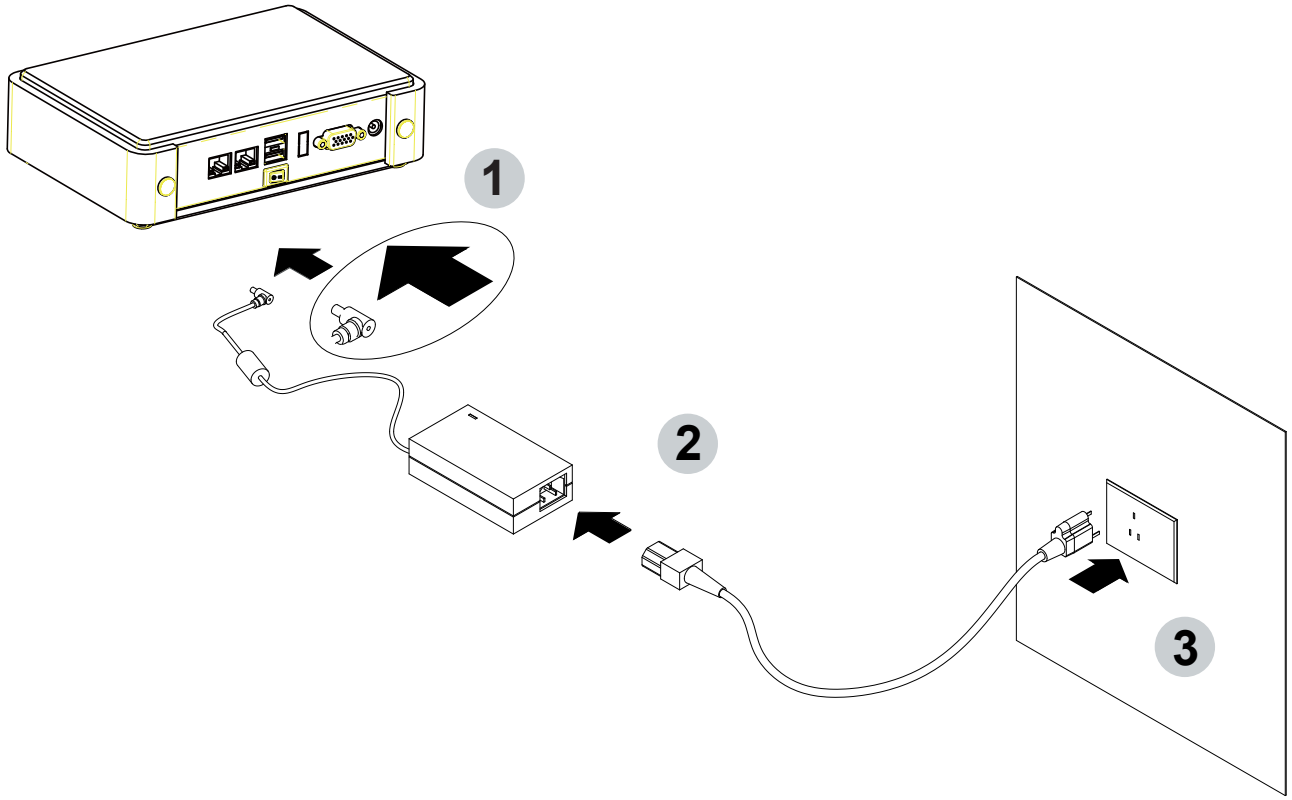
✻ Hardware Notice Guide

1. Before linking power supply with the motherboard, please attach DC-in adapter to the motherboard first. Then plug the adapter power to AC outlet.
   Always shut down the computer normally before you move the system unit or remove the power supply from the motherboard. Please unplug the DC-in adapter first and then unplug the adapter from the AC outlet.
   Please refer photo 1 as standard procedures.
2. In case of using DIRECT DC-in (without adapter), please check the allowed range for voltage & current of cables. And make sure you have the safety protection for outer issues such as short/broken circuit, overvoltage, surge, lightning strike.
3. In case of using DC-out to an external device, please make sure its voltage and current comply with the motherboard specification.
4. The total power consumption is determined by various conditions (CPU/motherboard type, device, application, etc.). Be cautious to the power cable you use for the system, one with UL standard will be highly recommended.
5. It's highly possible to burn out the CPU if you change / modify any parts of the CPU cooler.
6. Please wear wrist strap and attach it to a metal part of the system unit before handling a component. You can also touch an object which is ground connected or attached with metal surface if you don't have wrist strap.
7. Please be careful to handle & don't touch the sharp-pointed components on the bottom of PCBA.
8. Remove or change any components form the motherboard will VOID the warranty of the motherboard.
9. Before you install/remove any components or even make any jumper setting on the motherboard, please make sure to disconnect the power supply first. (follow the aforementioned instruction guide)
10. "POWERON after PWR-Fail" function must be used carefully as below:
    When the DC power adaptor runs out of power, unplug it from the DC current;
    Once power returns, plug it back after 5 seconds.
    If there is a power outage, unplug it from the AC current, once power returns, plug it back after 30 seconds. Otherwise it will cause system locked or made a severe damage.

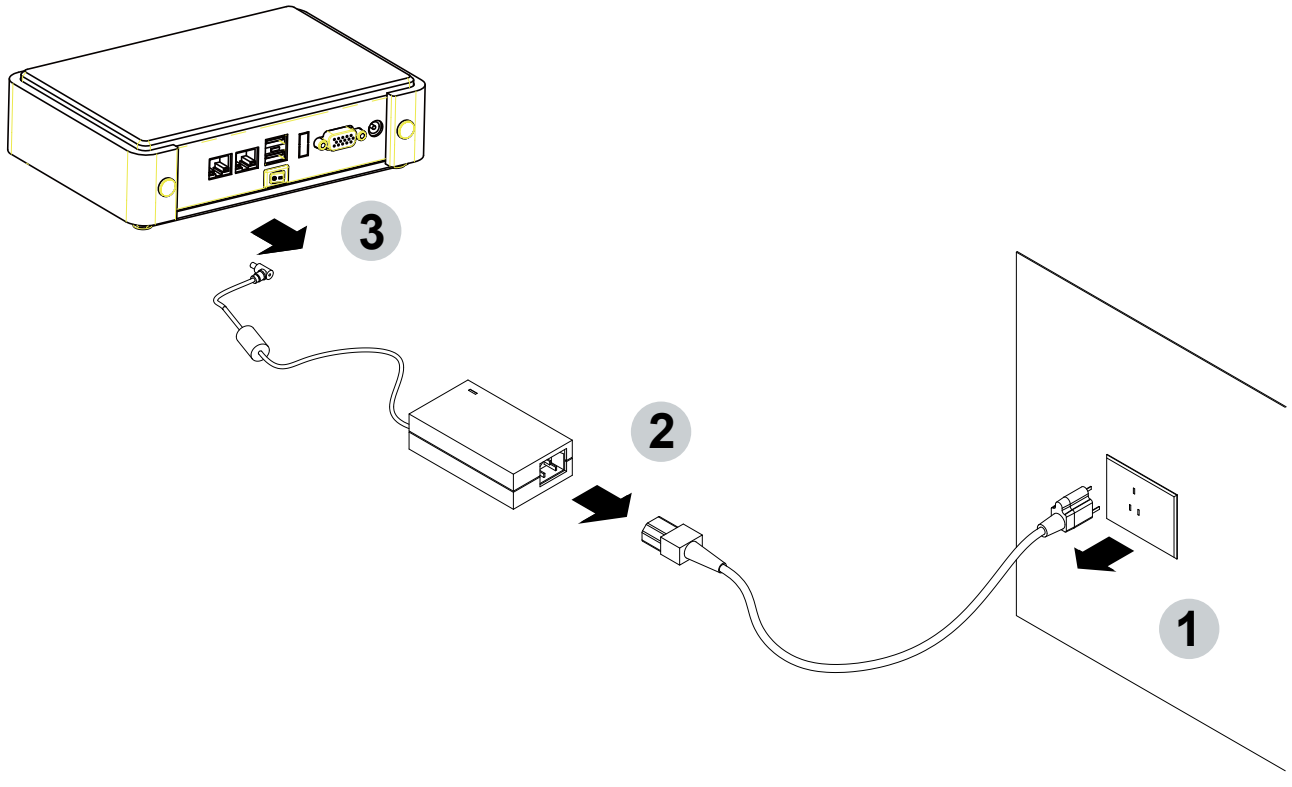
Remark 1:
Always insert/unplug the DC-in horizontally & directly to/from the motherboard. DO NOT twist, it is designed to fit snugly.
Moreover, erratic pull/push action might cause an unpredictable damage to the component & system unit.

Photo 1

Insert



Unplug



Chapter-1

General Information

The CI370D with latest Intel 8th Gen Coffee Lake-S Core i CPU and Pentium® / Celeron® Processor All-In-One board in the LGA1151 package with Intel® Q370 Express chipset. The CI370D supports high-speed data transfer interfaces such as PCIe3.0, USB 3.1, and SATA 6 Gb/s (SATA III), with dual-channel DDR4 memory up to 32 GB in two SO-DIMM slots, as well as graphics interface for HDMI and DisplayPort displays.

High-performance and power-efficient communication platform, the embedded motherboard of CI370D is specially designed for IoT, industrial, digital signage, medial, POS, retail and factory automation applications.CI370D with a wide variety of expansion options including PCIe(x16), PCIe(x8), PCIe(x4), PCIe(x1), or 2 PCIe(x8), 2 PCIe(x4), 2 PCIe(x1), selectable by raiser cards. The platform comes with Intel Gigabit Ethernet controllers and supports Wake-On LAN, vPro with TPM 2.0 and the PXE function in BIOS for Intel LAN chipset, it is perfect control board for networking devices.

The CI370D SBC equipped with a variety of interfaces : 9 x LAN,2x Fiber, 2 x RS232 / RS422 / 485 serial, 11 x USB, DP, HDMI & eDP) and expansion slots such as PCIe / mPCIe / M.2 slots for I/O modules (Digital I/O, WiFi, 3G / 4G, LoRA, Bluetooth) Connection options, is ideal solution to integrate with diagnostic test tools and control equipment, with excellent flexibility to meet any needs.
In addition to industrial grade components, TWISTER CI370D & TINO CI370D are with excellent thermal dispatched design, enable to operate smoothly with automation, machine vision, industrial communication, military and intelligent transportation applications in various harsh environments such as mountains, cold polar climate, mines, oil fields and deserts.

1-1 Major Feature

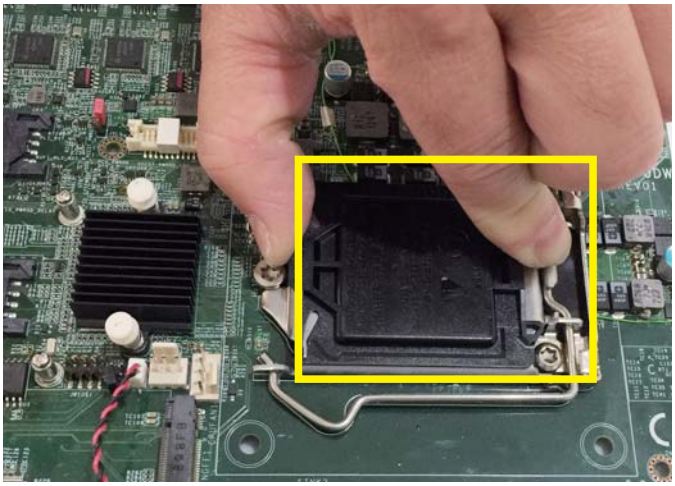
1. The Desktop Coffee Lake-Platform processor includes Integrated Display Engine, GPU and Integrated Memory Controller. The processor is designed be offered in a LGA1151 package.
2. Intel Q370 Chipset Family Platform Controller Hub (PCH)
3. Supports Two Channels of DDR4 SO-DIMM SDRAM, Max. 32GB, data transfer rates of 2400MT/s
4. Intel Desktop Coffee Lake-Platform Processor Integrated Graphics. GEN 9 architecture supports up to 72 Execution Units (EUs), depending on the processor SKU.
5. Integrated Gigabit LAN Controller with Intel I219LM Gigabit Ethernet PHY supports vPro. Support 8 x 10 / 100 / 1000 Mbps Intel LAN ports & 2 x Gbps Intel LAN port for fiber
6. Support DP, HDMI, eDP1.4 2 lanes on Board.
7. Support 2 x RS232 auto switch to RS485 / RS422 by BIOS
8. 4 x type A USB3.1 external and 7 x USB 2.0 internal
9. ALC886 HD Audio Specification 1.0 Two channels sound.
10. Four SATA ports 3.0 Data transfer rates up to 6.0 Gb/s (600 MB/s)
11. Support extended 2 x Mini PCIe card for PCIe x 1, mSATA and USB interface, There are 2 x SIM Card Socket for these two mini cards. (3G/4G LTE module)
12. One M.2 B-Key 2242, 2280, 3042 for PCIe x 2 & USB device. Support PCIe NVMe storage
13. Hardware digital Input & Output, 8 x DI / 8 x DO, Hardware Watch Dog Timer, 0~255 sec programmable
14. 1 PCIe x 16 & 1 PCIe x 4 Golden Finger supports 1 PCIe x 1, 4, 8,16 or 1 PCIe x 1, 4 with Riser Cards
15. Support TPM 2.0
16. Support SPI

1-2 Specification

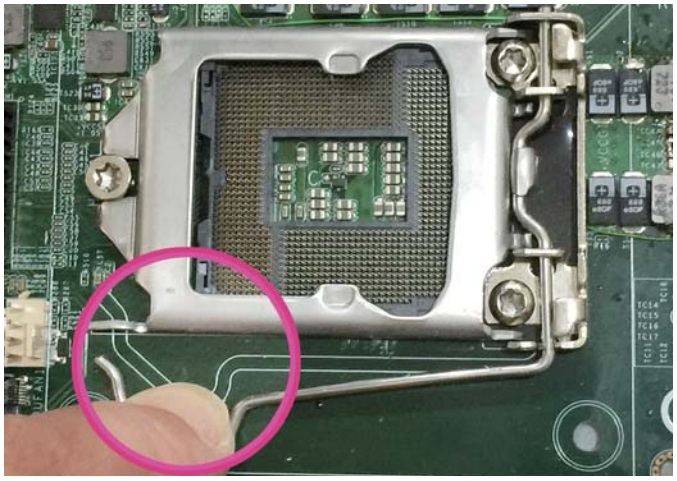
1. **CPU:** Desktop Coffee Lake Platform processor. The processor is designed be offered in a LGA1151 package.
2. **Memory:** Two SO-DIMM slots for DDR4 SDRAM, Max. 32GB, data transfer rates of 2400MT/s
3. **Graphics:** Intel 9th generation (Gen 9) LP graphics and media encode / decode engine supports OpenGL x 4.5, OpenCL x 2.1, Direct3D x 2015, Direct3D 11.2, Direct2D eDP 1.4 2 Lanes up to 1920 x 1080, DP 1.2 4096 x 2160, HDMI 1.4 up to 3840 x 2160
4. **SATA:** Integrated Serial ATA Host Controller Up to 4 SATA port, SATA Gen3 Data transfer rates up to 6.0 Gb/s (600 MB/s).
5. **LAN:** LAN1 Intel I219LM Gigabit Ethernet PHY. LAN2~9 Intel I210-IT LAN chipset or Intel I211-AT LAN chipset (Option) with 10 / 100 / 1000 Mbps. LAN10~11 Intel I210-IS LAN chipset for fiber.
6. **I/O Chip:** Chipsets for 2 ports RS232 / 422 / 485
7. **USB:** 4 type A USB 3.1 connector onboard and 7 USB 2.0 (internal)
8. **Sound:** Support line in, line out and MIC in
9. **eDP:** Support eDP 1.4 2 Lanes up to 1920 x 1080
10. **WDT / DIO:** Hardware digital Input & Output, 8 x DI / 8 x DO / Hardware Watch Dog Timer, 0~255 sec programmable
11. **Expansion interface:** Two full-size PCIe Mini card for PCIe x 1, mSATA and USB interface with 2 SIM sockets. 1 M.2 B-key for PCIe x 2 & USB devices.
12. **Golden Finger:** PCIe x 16 Golden Finger supports 1 PCIe x 1, 4, 8,16 or 2 PCIe x 1, 4, 8 with Riser Cards
13. **TPM:** Infineon SLB 9665 TT 2.0 Trusted Platform Module
14. **BIOS:** AMI UEFI BIOS
15. **Dimension:** 217 x 185 mm
16. **Power:** DC IN +24V

1-3 Installing the CPU (Socket Version)

1. Remove the CPU socket cover from bottom to top.

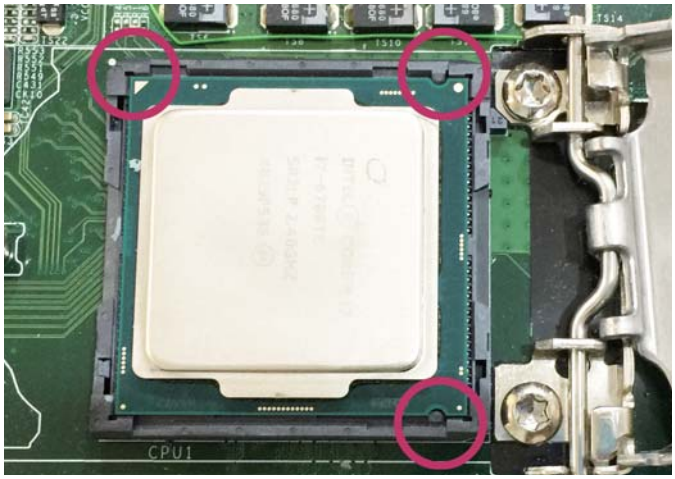


2. Release the small metal lever holding the CPU retention bracket downwards by the outside



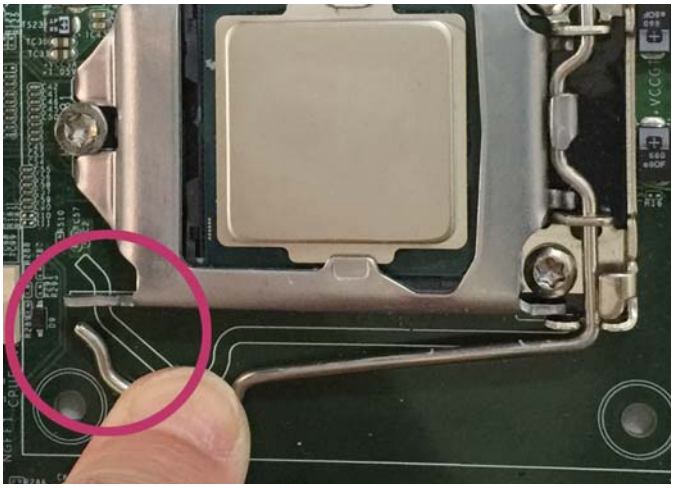
3. Open the CPU bracket upwards.

4. Insert the CPU on the CPU socket. Please pay attention to the direction of the CPU.



5. With the CPU seated in the socket, you can lower the retention bracket back into place.

6. Make sure that you slip the notch at the end of the bracket around the single screw at the base of the socket before you use the metal lever to lock the CPU into place.

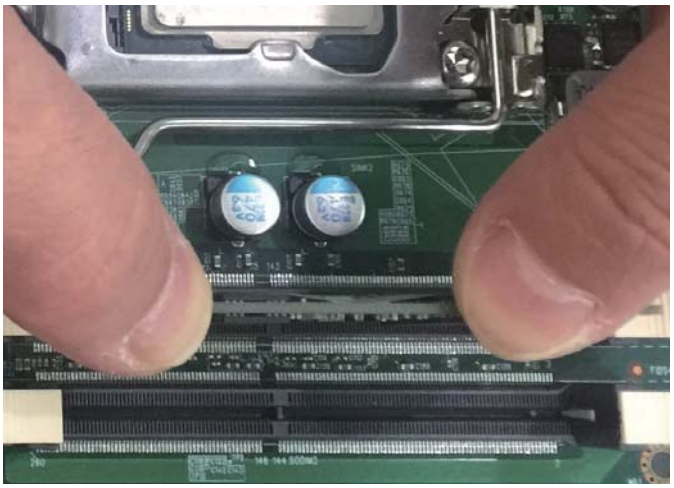


7. Finished.

1-4 Vertical SO-DIMM assembly guide

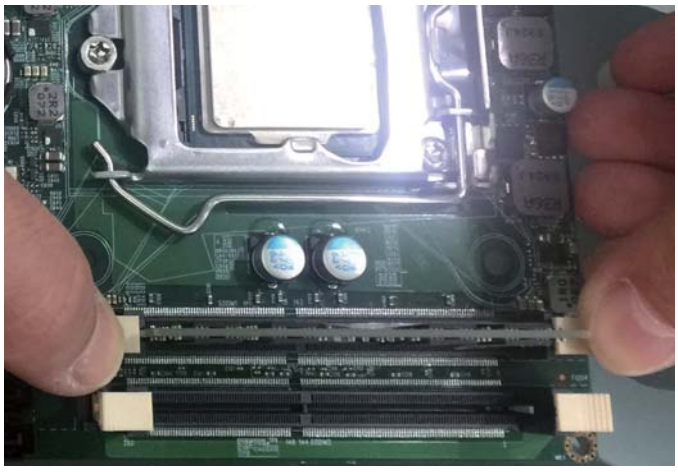
1. Install the memory into SODIMM.

2. Press down firmly to ensure the memory is locked.



Uninstall

1. Pull open both sides of the memory slot.

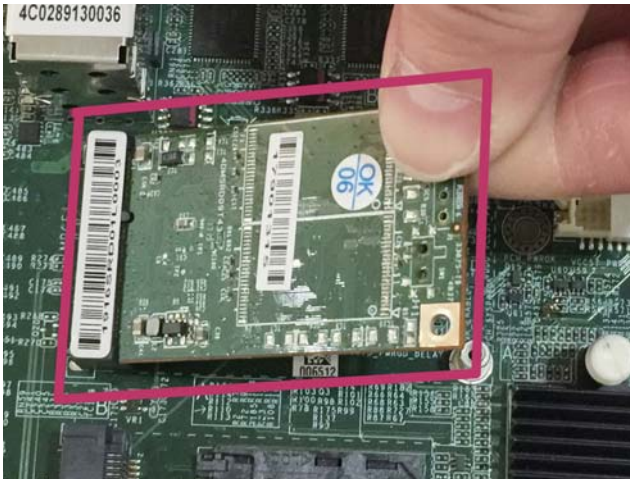


2. Take out the memory.

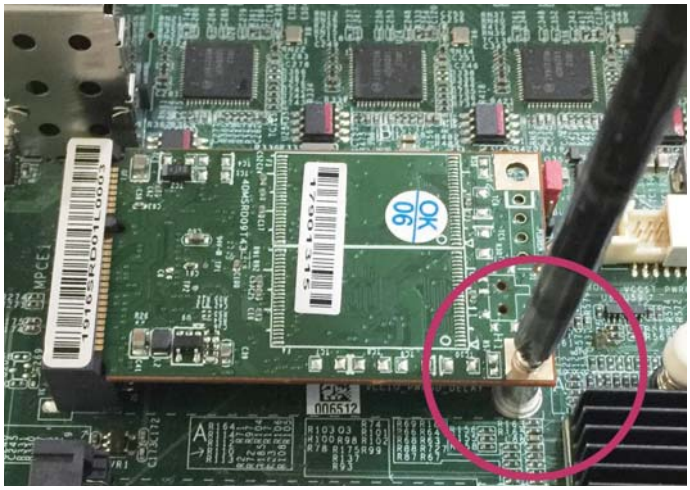
1-5 Directions for installing the Mini PCI-e Card (Full Size)

1. Unscrew the screw on the board

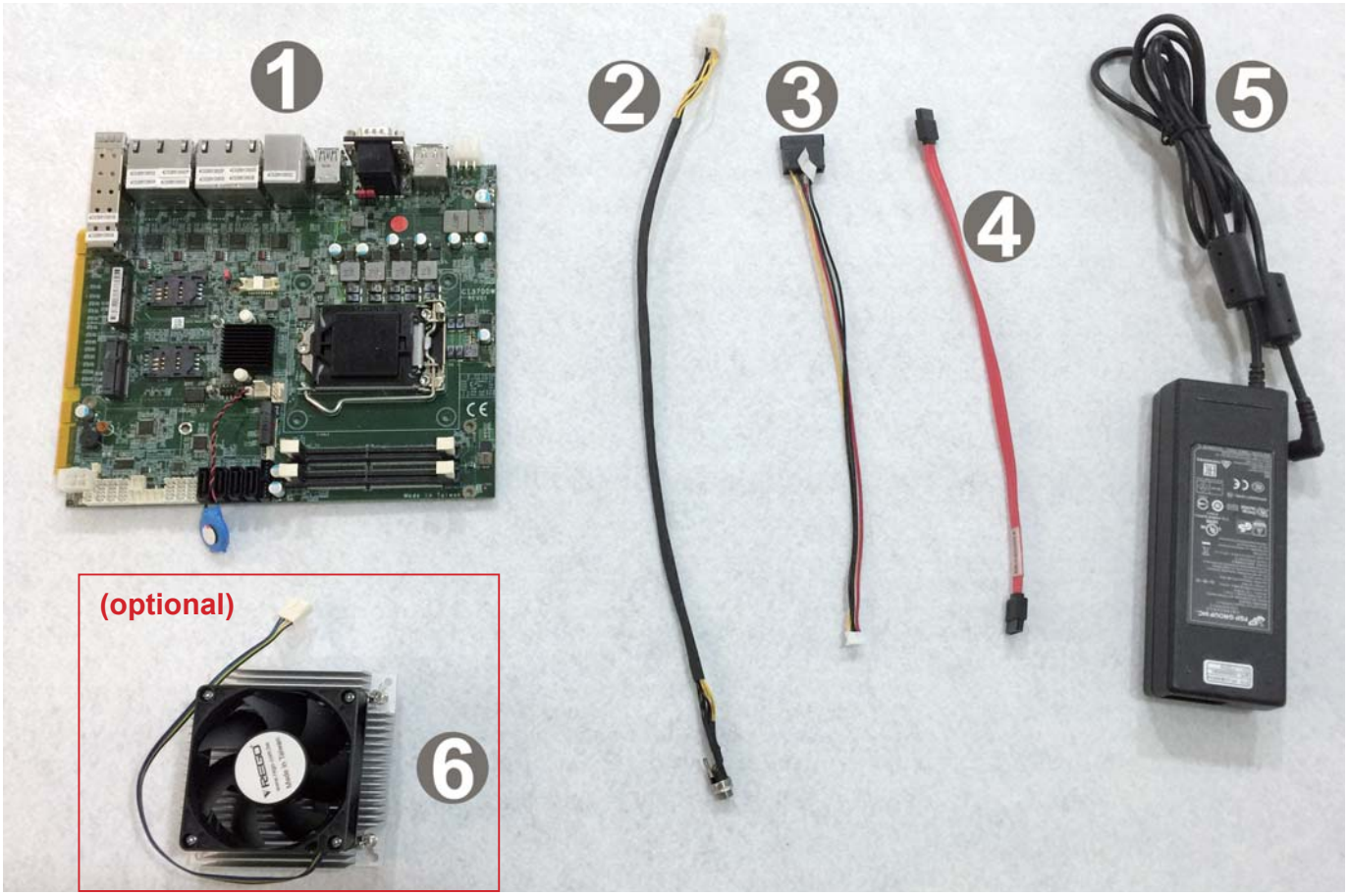
2. Plug in the Mini Card in a 45 angle



3. Gently push down the Mini Card and screw the screw back.



1-7 Packing List



	Material Code	Description	Detail Specification	Quantity
1	7G1901-1780001-0	MB-CI370D-2CXX-001	LF,CI370D-2CXX,Rev.:001	1
2	6G6003-7405-0100	Power Cable	DC JK/ATX 2x3,L=35cm	1
3	6G6003-1009-0100	SATA Power Cable	LF,L=25cm,1*4/2.0 to 180° SATA 15p	1
4	6G6001-2203-0100	SATA DATA Cable (Red)	LF,L=25cm	1
5	6G5212-1209-0100	120W Power Adapter,24V/5A	LF,L Type,EA11011M-240,EDAC	1
6	(optional) 6G7300-5400-0100	CPU Cooler(For LGA 115X),Aluminum Heat Sink+7025 Fan 5400RPM,Up To 65W	LF,Material,For CI170A,REGO RG3215-003	1

Optional accessories (items in addition to motherboard)
are not included in the standard packing.
Please contact your dealer to purchase the optional accessories.

Chapter-2

Hardware Installation

2-1 Unpacking Precaution

This chapter provides the information how to install the hardware of CI370D.
Please follow section 1-7, 2-1 and 2-2 to check the delivery package and unpack carefully. Please follow the jumper setting procedure.

NOTE!

1. Do not touch the board or any other sensitive components without all necessary anti-static protection.
2. Please pay attention to the voltage limitation of DC-IN 24V 5%.
   Overuse of DC-IN voltage limitation or change to another power adapter (not provided with this system) will VOID warranty.

You should follow these
steps to protect the board from the static electric
discharge whenever you handle the board:

1. Ground yourself by a grounded wrist strap at all times when you handle the CI370D.
   Well secure the ALLIGATOR clip of the strap to the end of the shielded wire lead from a grounded object. Please put on and connect the strap before handling the CI370D for harmlessly discharge any static electricity through the strap.

2. Please use anti-static pad to put any components, parts, or tools on the pad whenever you work on them outside the computer. You may also use the anti-static bag instead of the pad. Please ask your local supplier for necessary parts on anti-static requirement.

3. Do not plug any connector or set any jumper when the power is on.

2-2 Unpacking checkup

First of all, please follow all necessary steps of section 2-1 to protect CI370D from electricity discharge. With reference to section 1-7 please check the delivery package again with following steps:

1. Unpack the CI370D board and keep all packing material, manual and driver disc etc, do not dispose !

2. Is there any components lose or drops from the board?
   DO NOT CONTINUE TO INSTALL THIS BOARD!
   CONTACT THE DEALER YOU PURCHASED
   THIS BOARD FROM, IMMEDIATELY.

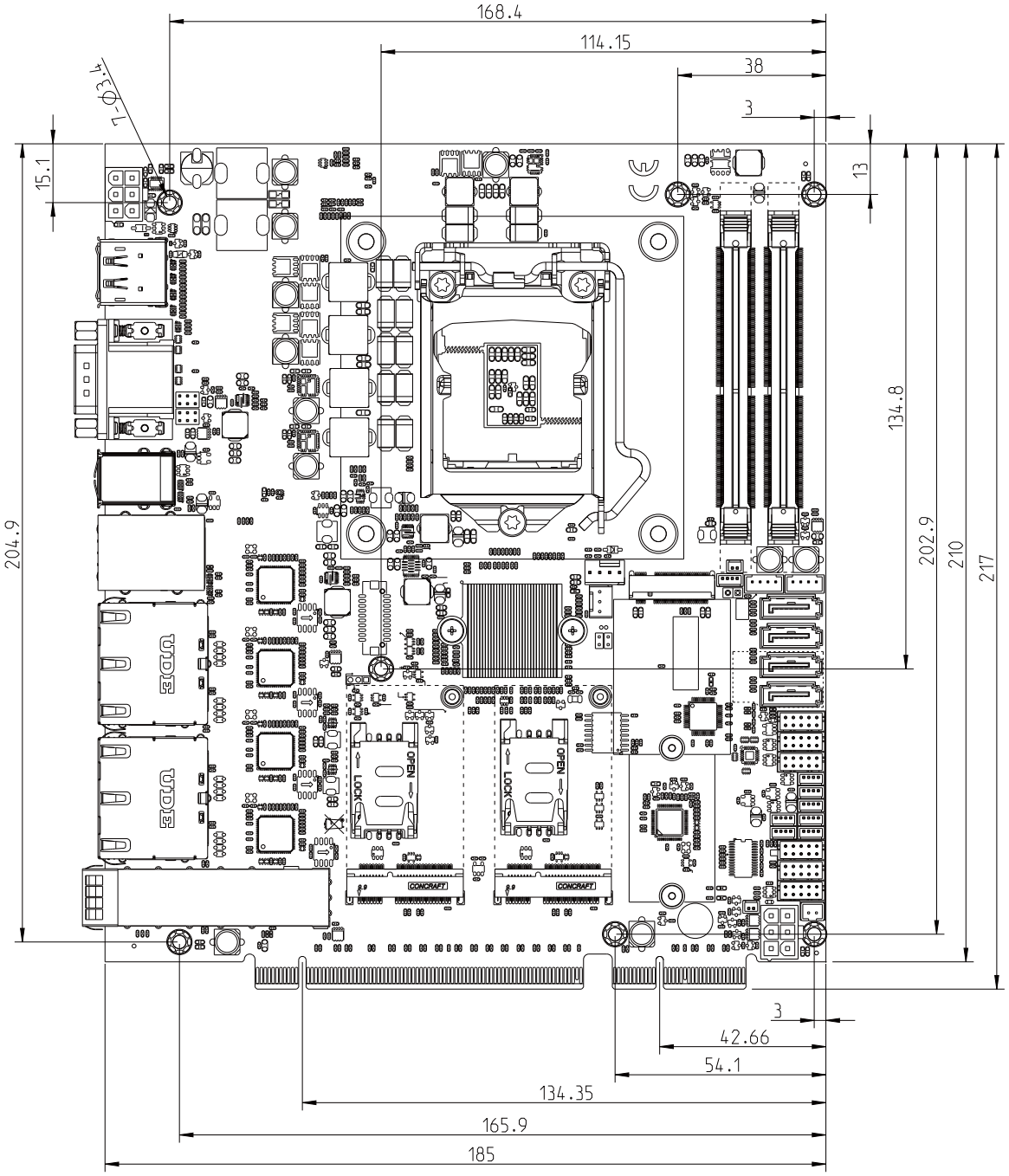
3. Is there any visible damage on the board?
   DO NOT CONTINUE TO INSTALL THIS BOARD!CONTACT
   THE DEALER YOU PURCHASED THIS BOARD FROM, IMMEDIATELY.

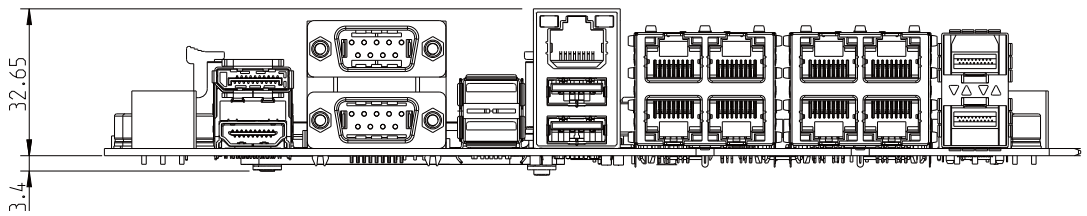
4. Check your optional parts (i.e. DDR, CF etc.), all necessary jumpers setting to jumper pin-set, and CMOS setup correctly.
   Please also refer to all information of jumper settings in this manual.

5. Check your external devices (i.e. Add-On-Card, Driver Type etc.) for complete add-in or connection and CMOS setup correctly.
   Please also refer to all information of connector connection in this manual.

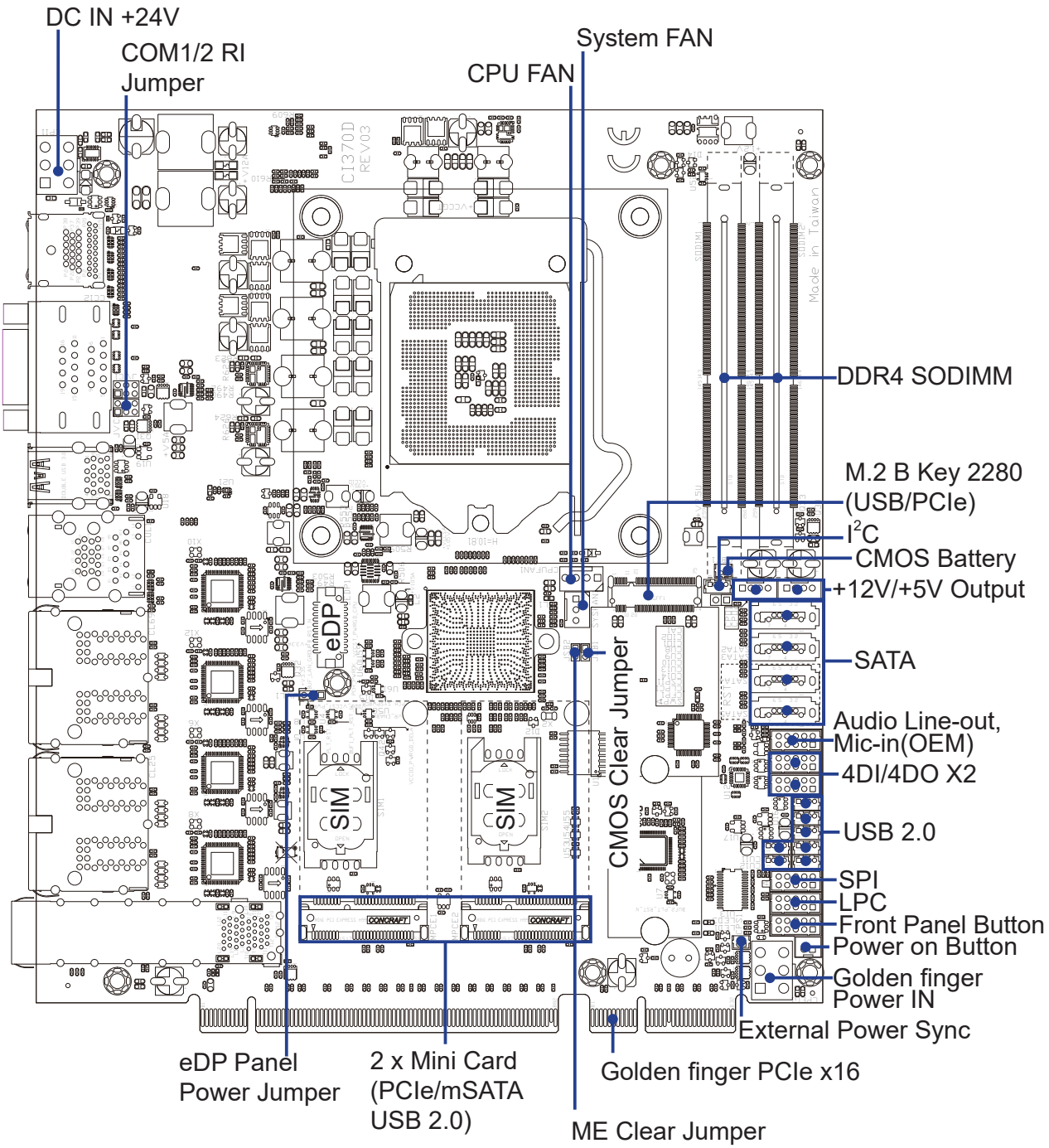
6. Please keep all necessary manual and driver disc in a good condition for future re-installation if you change your Operating System.

2-3 Dimension-CI370D

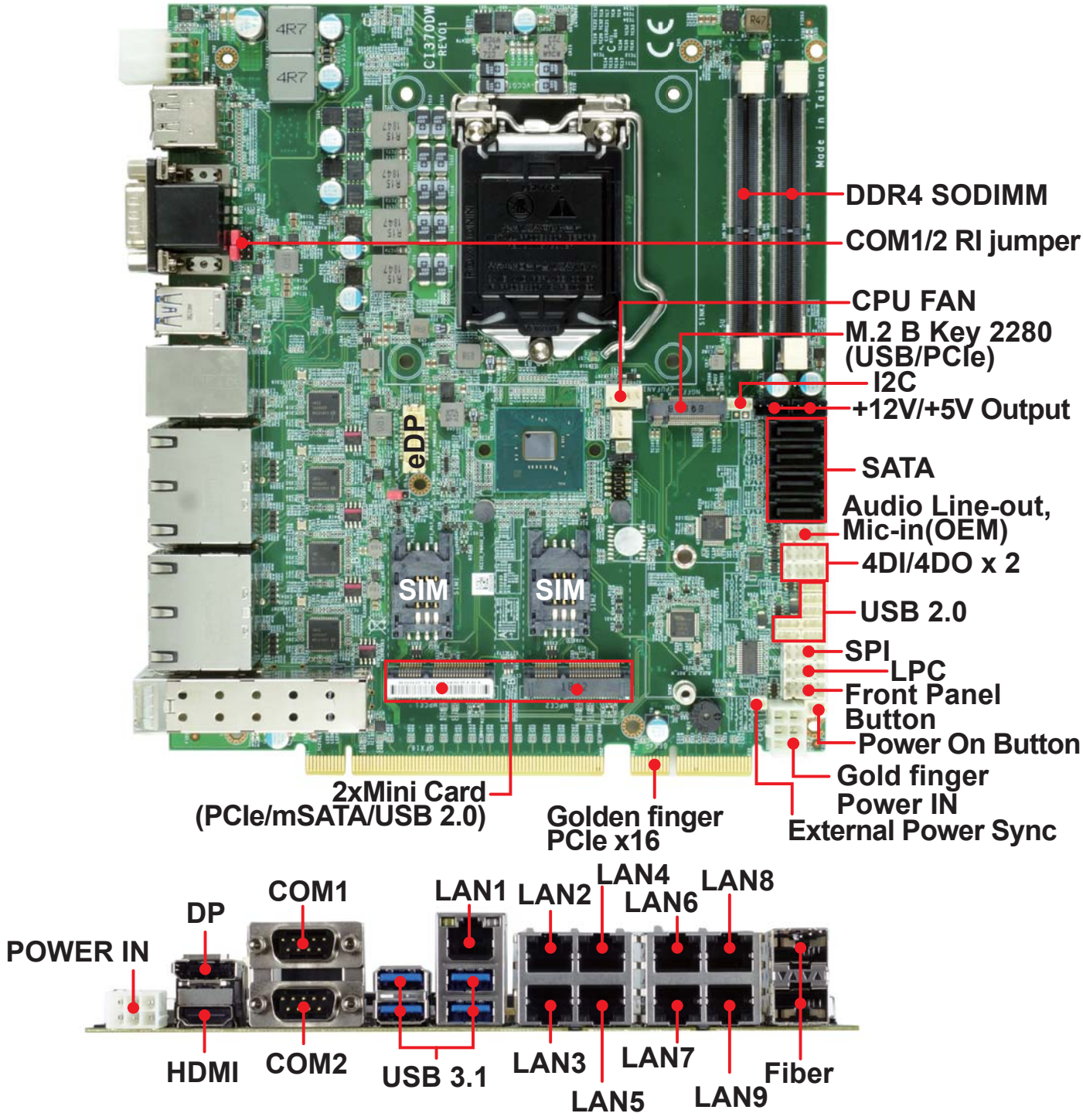




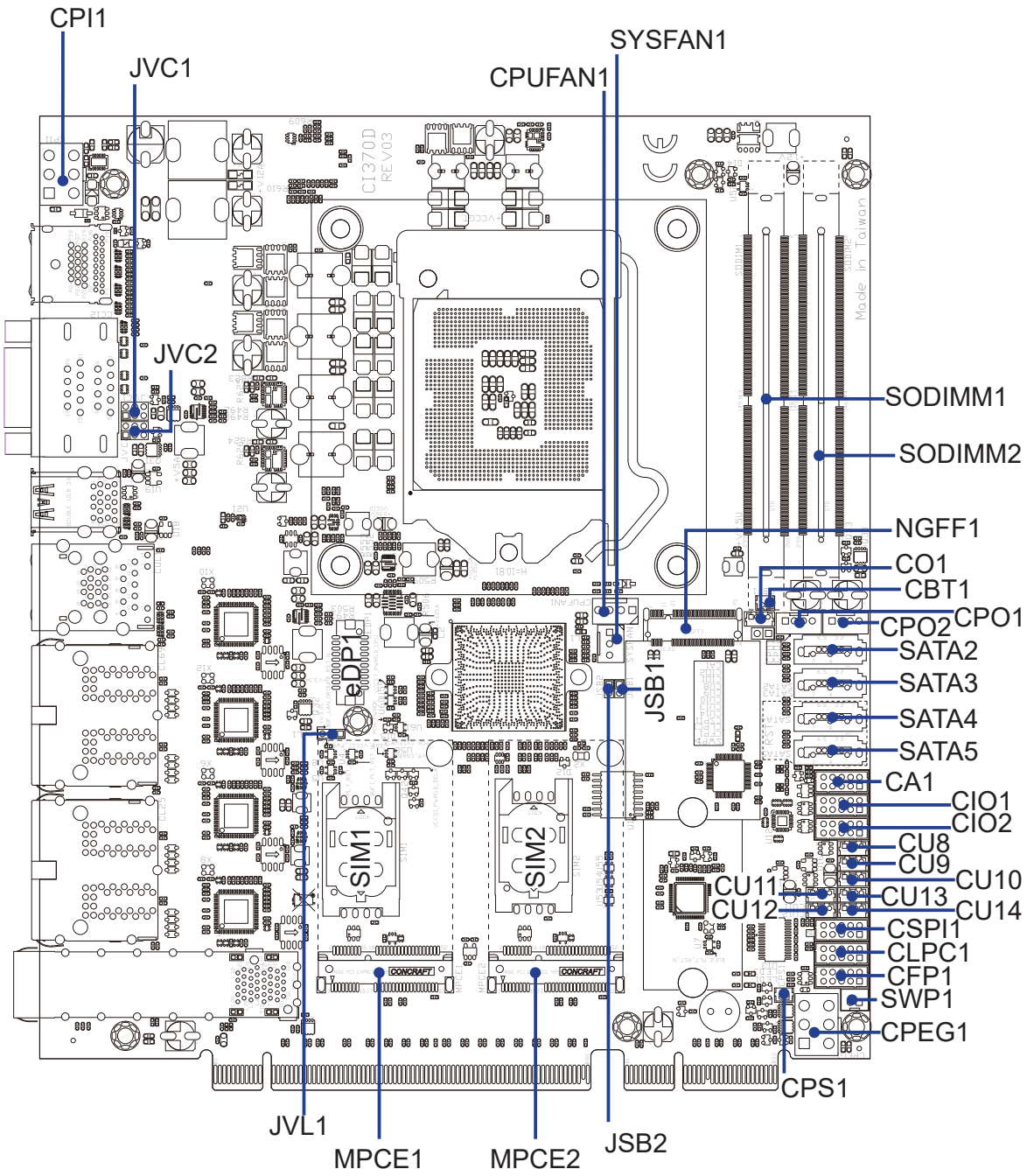
2-4 Layout-Function Map-CI370D



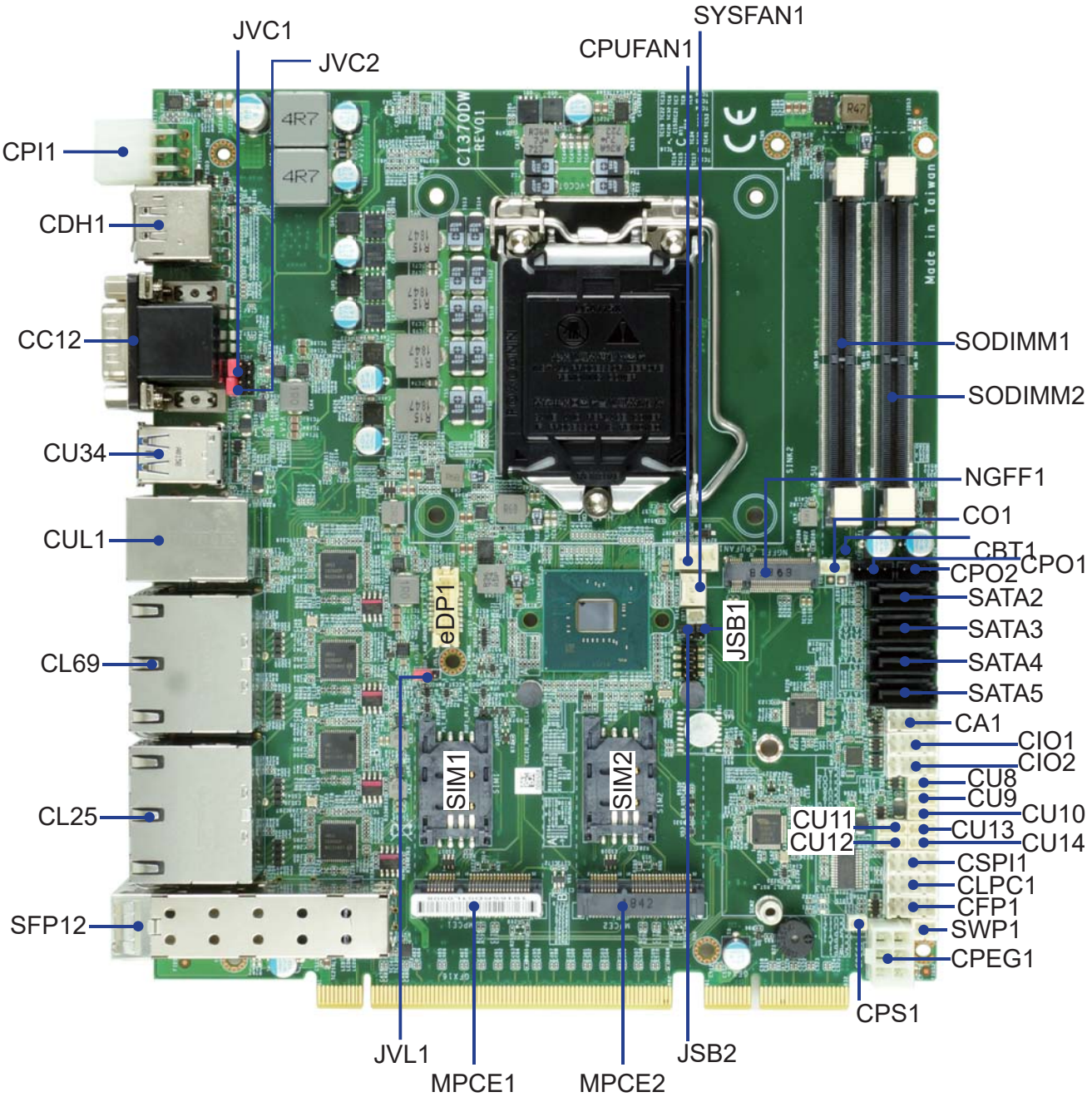
2-5 Function MAP-CI370D



2-6 Connector MAP-CI370D



2-7 Diagram-CI370D



2-8 List of Jumpers

JSB1: CMOS DATA Clear

JSB2: ME DATA Clear

JVL1: eDP panel power select

JVC1/2: COM1/2 PIN9 RI/+12V/+5V Select

2-9 Jumper Setting Description

A jumper is ON as a closed circuit with a plastic cap covering two pins. A jumper is OFF as an open circuit without the plastic cap. Some jumpers have three pins, labeled 1, 2, and 3. You could connect either pin 1 and 2 or 2 and 3.The below figure 2.2 shows the examples of different jumper settings in this manual.

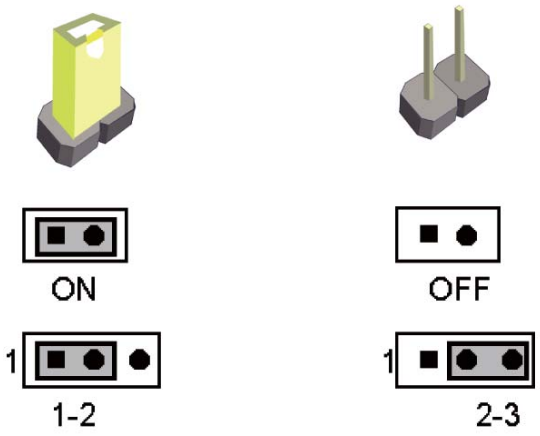


Figure 2.2

All jumpers already have its default setting with the plastic cap inserted as ON, or without the plastic cap as OFF. The default setting may be referred in this manual with a " * " symbol .

2-10 JSB1: CMOS DATA Clear

A battery must be used to retain the motherboard configuration in CMOS RAM.

Close Pin1 and pin 2 of JCMOS1 to store the CMOS data.

To clear the CMOS, follow the procedures below:

1. Turn off the system and unplug the AC power
2. Remove DC IN power cable from DC IN power connector
3. Locate JCMOS1 and close pin 1-2 for few seconds
4. Return to default setting
5. Connect DC IN power cable back to DC IN Power connector

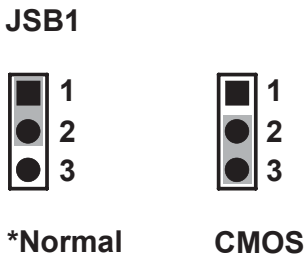
JSB1	Description
*1-x	Normal set
1-2	CMOS data clear

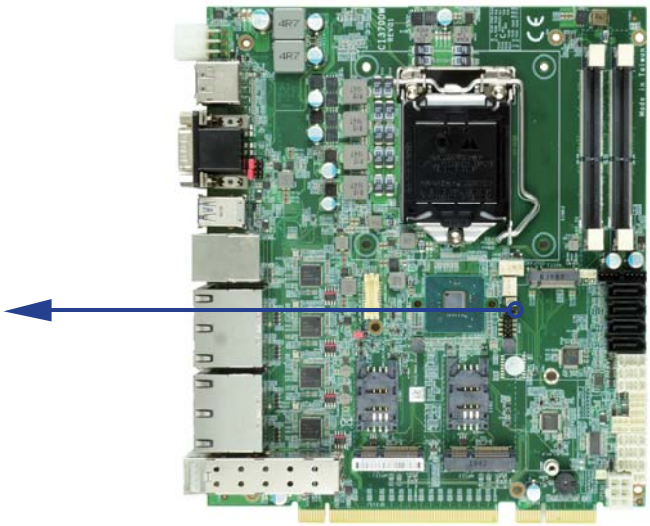
Note: Do not clear CMOS unless

1. Troubleshooting

2. Forget password

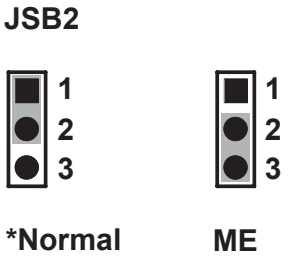
3. You fail over-clocking system

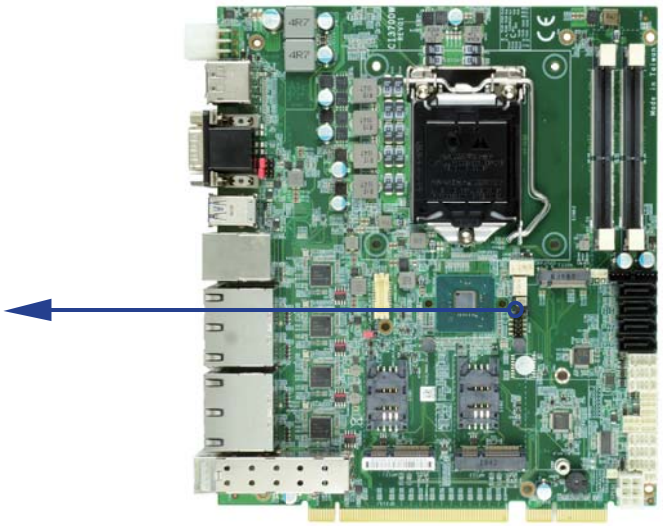




2-11 JSB2: ME DATA Clear

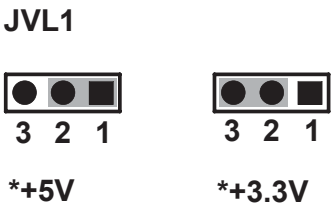
JSB2	Description
*1-x	Normal set
1-2	ME data clear

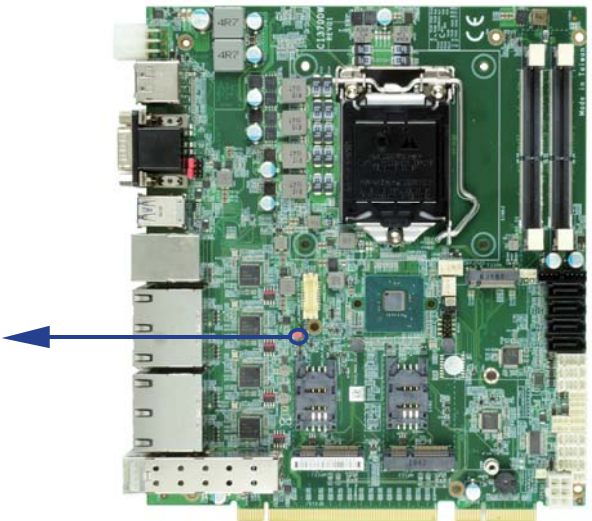




2-12 JVL1: eDP panel power select

JVL1	Description
1-2	+5V
*2-3	+3.3V

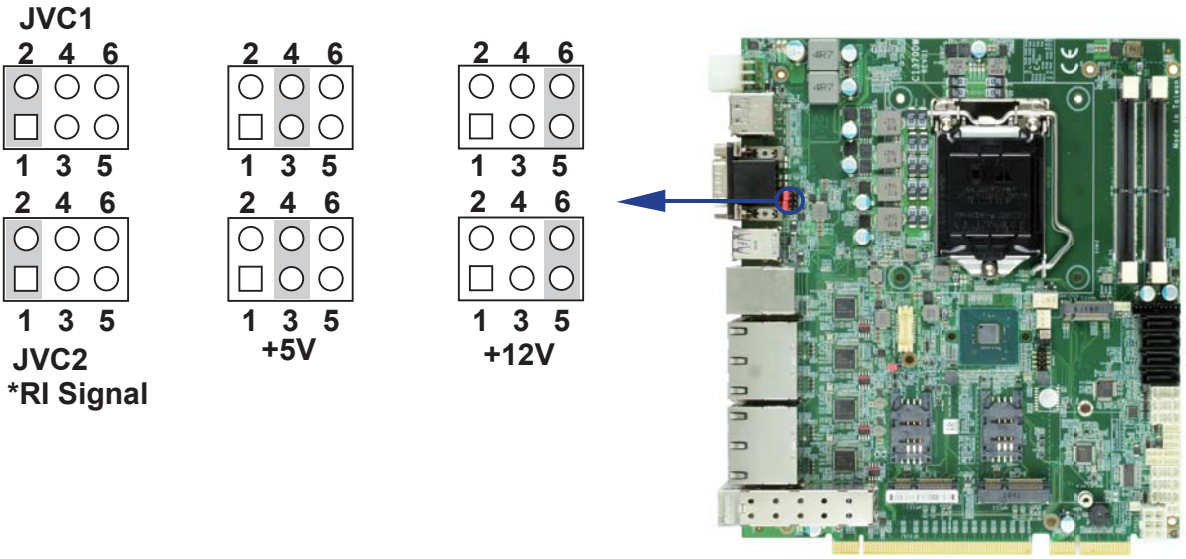




2-13 COM port pin9 select RI signal or Voltage source

JVC1/2: COM1/2 PIN9 RI/+12V/+5V Select

JVC1/2	DESCRIPION
*1-2	COM port pin9 use RI signal
3-4	COM port pin9 use +5V voltage
5-6	COM port pin9 use +12V voltage



Chapter-3

Connection

This chapter provides all necessary information of the peripheral's connections, switches and indicators. Always power off the board before you install the peripherals.

3-1 List of Connectors

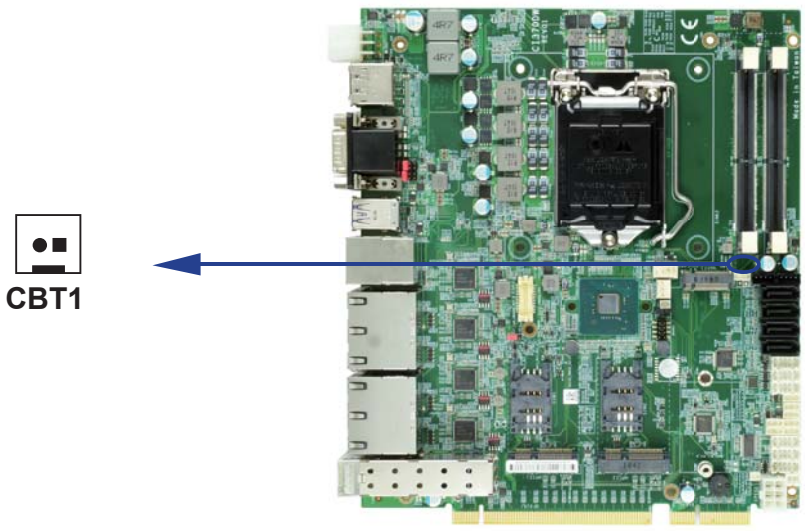
CBT1:	COMS battery 1x2 pin (1.25mm) wafer
CA1:	Line-out/Line-in/Mic-in/SPDIF-out 2x5 pin (2.0mm) Wafer
CC12:	Dual D-SUB COM port connector.
CDH1:	Display Port and HDMI connector.
CFP1:	Front Panel connector 2x5pin (2.0mm) wafer
CIO1:	4DI/4DO 2x5 pin (2.0mm) Wafer
CIO2:	4DI/4DO 2x5 pin (2.0mm) Wafer
CU34:	Dual USB3.0 Port 3,4 connector.
CU8:	USB 2.0 port 4pin (1.25mm) Wafer
CU9:	USB 2.0 port 4pin (1.25mm) Wafer
CU10:	USB 2.0 port 4pin (1.25mm) Wafer
CU11:	USB 2.0 port 4pin (1.25mm) Wafer
CU12:	USB 2.0 port 4pin (1.25mm) Wafer
CU13:	USB 2.0 port 4pin (1.25mm) Wafer
CU14:	USB 2.0 port 4pin (1.25mm) Wafer
CUL1:	LAN1 and USB3.0 Port 1,2 connector.
CL25:	LAN port 6 to port 9 RJ45 Connector
CL69:	LAN port 2 to port 5 RJ45 Connector
CO1:	I^2C Bus 4pin (1.25mm) Wafer
CLPC1:	LPC 2x5 pin (2.00mm) Wafer
CPEG:	Gold finger DC 12V-IN 2x3 pin (2.54mm) ATX wafer
CPI1:	Motherboard DC 24V-IN 2x3 pin (2.54mm) ATX wafer
CPO1:	+12V/+5V power output 4 pin (2.0mm) Black wafer
CPO2:	+12V/+5V power output 4 pin (2.0mm) Black wafer
CPS1:	External Power-On sync 2 pin (1.25mm) wafer.
CSPI1:	SPI 2x5 pin (2.00mm) Wafer
EDP1:	eDP 2x10 pin (1.25mm) connector
SATA2:	SATA port 2 connector 7pin
SATA3:	SATA port 3 connector 7pin
SATA4:	SATA port 4 connector 7pin

SATA5:	SATA port 5 connector 7pin
SFP12:	Dual Fiber cage for LAN10,LAN11.
SIM1:	With MPCE1 3G/4G/LTE module.
SIM2:	With MPCE2 3G/4G/LTE module.
SODIMM1:	DDR4 Channel 0 SODIMM
SODIMM2:	DDR4 Channel 1 SODIMM
SWP1:	Power On-Off 1x2 pin Wafer
MPCE1 :	Full size mini card port 1 sockets 52pin
MPCE2 :	Full size mini card port 2 sockets 52pin
M.2 :	M.2 B Key 2280/2242/3042 sockets.

3-2 CBT1: CMOS Battery 1x2 pin (1.25mm) Wafer

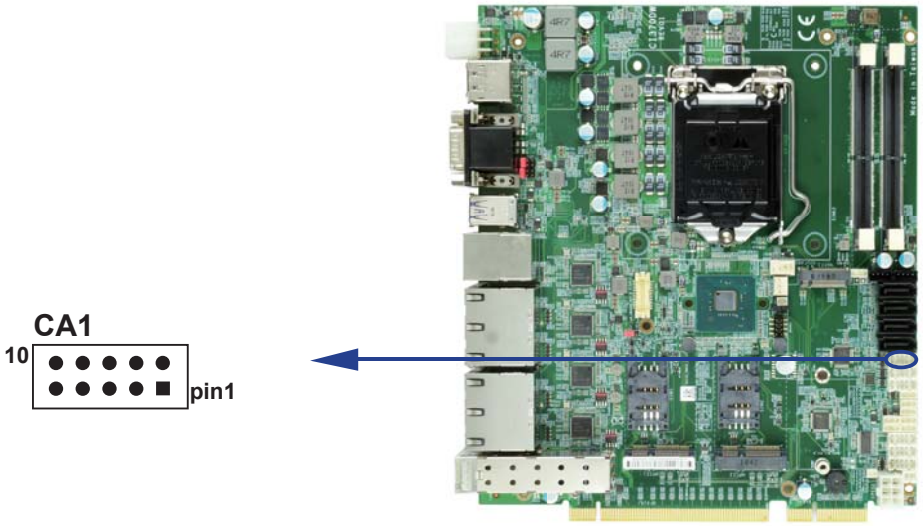
PIN NO.	Description
1	GND
2	+3V

Note:
1. When the board without Adaptor plug in, this board power RTC consumption about 2.7uA
2. If adaptor always plug in RTC power consumption about 0.1uA



3-3 CA1: Line-out/Line-in/Mic-in 2x5 pin (2.0mm) Wafer

PIN NO.	Description	PIN NO.	Description
1	Line-out-R	2	MIC-IN
3	Line-in-R	4	GND
5	GND	6	GND
7	Line-in-L	8	+5V
9	Line-out-L	10	MIC-IN



3-4 CC12 COM1/2 Dual COM port connector.

- **RS232 Mode**

PIN NO.	Description	PIN NO.	Description
1	DCD	6	DSR
2	RXD	7	RTS
3	TXD	8	CTS
4	DTR	9	RI
5	GND		

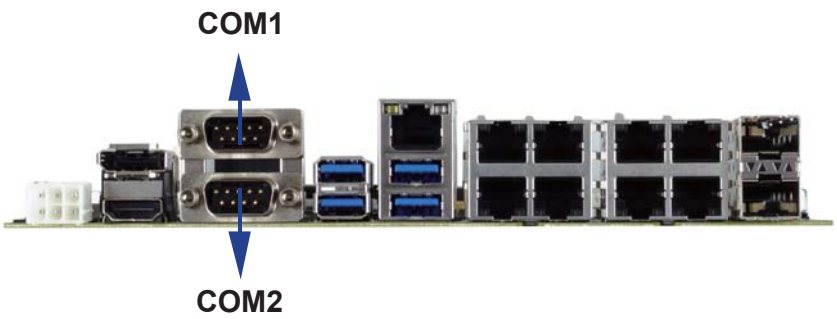
- **RS485 Mode**

PIN NO.	Description	PIN NO.	Description
1	Data-	6	NC
2	Data+	7	NC
3	NC	8	NC
4	NC	9	NC
5	GND		

- **RS422 Mode**

PIN NO.	Description	PIN NO.	Description
1	TX-	6	NC
2	TX+	7	NC
3	RX+	8	NC
4	RX-	9	NC
5	GND		

Note:
1. Pin 9 RI and Voltage setting only for COM 12 ports, JVC1 for COM1, JVC2 for COM2
2. Default support RS232/RS422/RS485 by BIOS selected



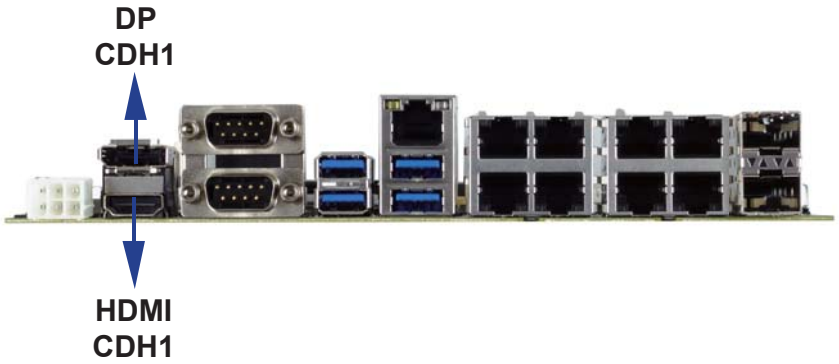
3-5 CDH1 Display Port and HDMI Connector

- **Display Port**

PIN	Description	PIN	Description
1	Lane 0+	2	GND
3	Lane 0-	4	Lane 1+
5	GND	6	Lane 1-
7	Lane 2+	8	GND
9	Lane 2-	10	Lane 3+
11	GND	12	Lane 3-
13	GND	14	GND
15	AUX+	16	GND
17	AUX-	18	H.P. Detect
19	GND	20	+3.3V

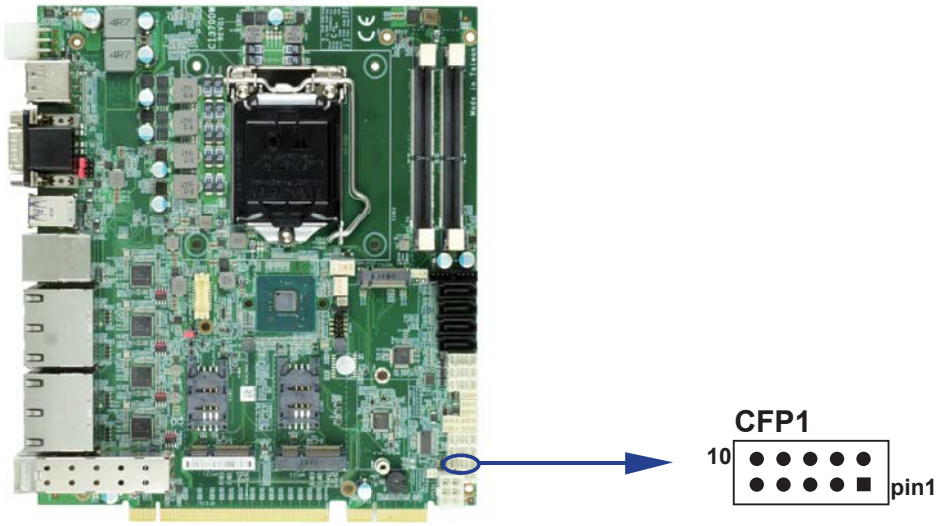
- **HDMI**

PIN	Description	PIN	Description
1	TMDS DATA2+	2	GND
3	TMDS DATA2-	4	TMDS DATA1+
5	GND	6	TMDS DATA1-
7	TMDS DATA0+	8	GND
9	TMDS DATA0-	10	TMDS CLK+
11	GND	12	TMDS CLK-
13	NC	14	NC
15	DDC CLOCK	16	DDC DATA
17	GND	18	+5V
19	H.P. Detect		



3-6 CFP1 Front Panel connector 2x5 pin (2.0mm) wafer

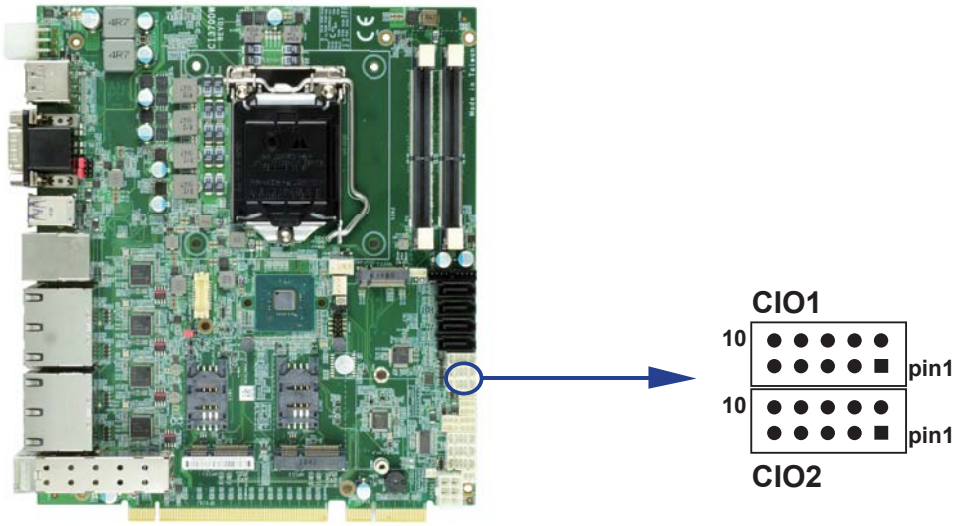
PIN NO.	DESCRIPTION	PIN NO.	DESCRIPTION
1	Power button pin	2	Power button GND
3	Reset pin	4	Reset GND
5	Power LED-	6	Power LED+
7	HDD LED-	8	HDD LED+
9	LAN LED-	10	LAN LED+



3-7 CIO1/2 DIO 0--7 (2x5 pin 2.0mm wafer)

PIN NO.	DESCRIPTION	PIN NO.	DESCRIPTION
1	DI-0,4	2	DO-3,7
3	DI-1,5	4	DO-2,6
5	DI-2,6	6	DO-1,5
7	DI-3,7	8	DO-0,4
9	GND	10	+5V

Note:
1. DI pin default pull up 10KΩ to +5V
2. If use need isolate circuit to control external device
3. F75111N-1 I2C bus address 0x9c



- **For F75111N I^2C watch dog timer device:**

DC spec:
Input low Voltage (VIL): +0.8 Max
Input High Voltage (VIH): +2V Min
Output low Current (IOL): 10mA (Min) VOL=0.4V
Output High Current (IOH): -10mA (Min) VOH=2.4V
Watch Dog Time value 0~255 sec
The system will be issued reset. When WDT is enable the hardware start down counter to zero. The reset timer have 10~20% tolerance upon the Temperature.
Note: If want to SDK support. Please contact to sales window.

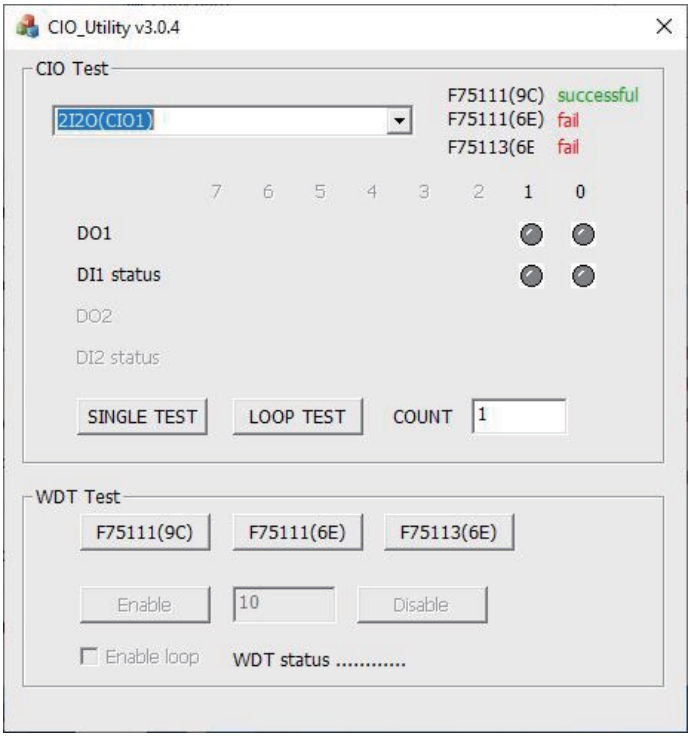
3-7-1 IO Device: F75111 CIO Utility

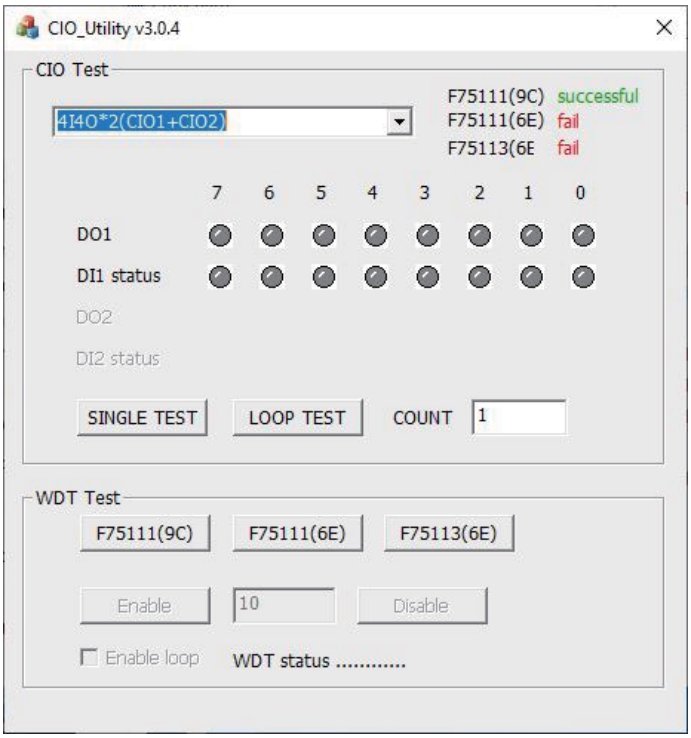
The Sample code source you can download from

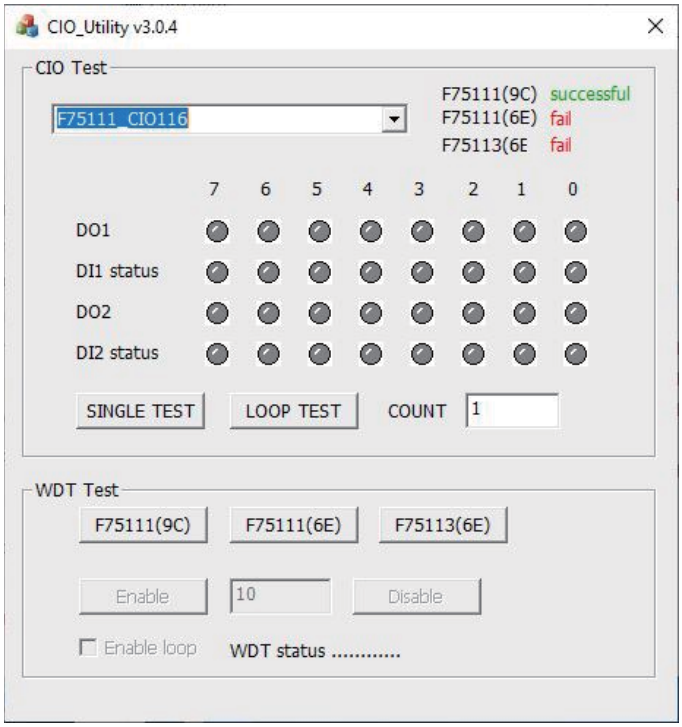
Source file: CIO_Utility_Src_v3.0.5_w.zip http://tprd.info/lexwiki/index.php/IO_Device:F75111_CIO_Utility

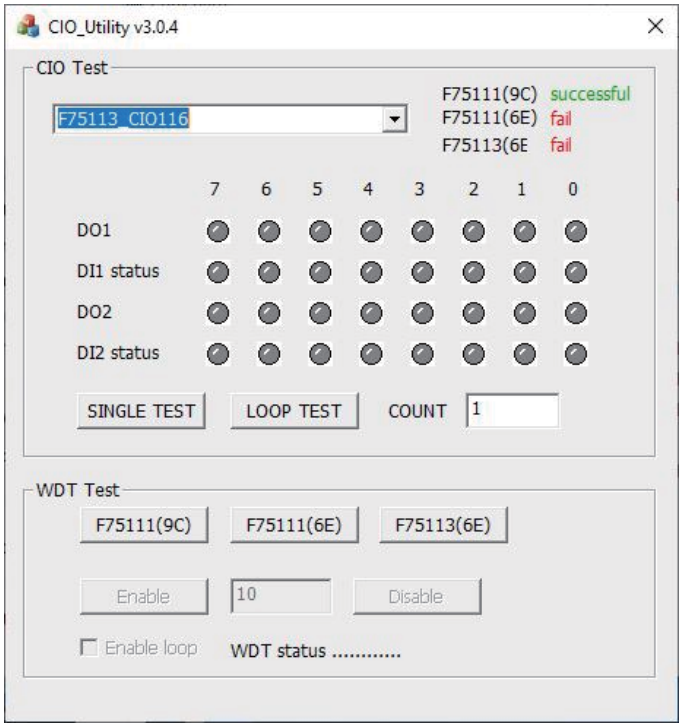
Binary file: CIO_Utility_Bin_v3.0.5_x32_w.zip CIO_Utility_Bin_v3.0.5_x64_w.zip

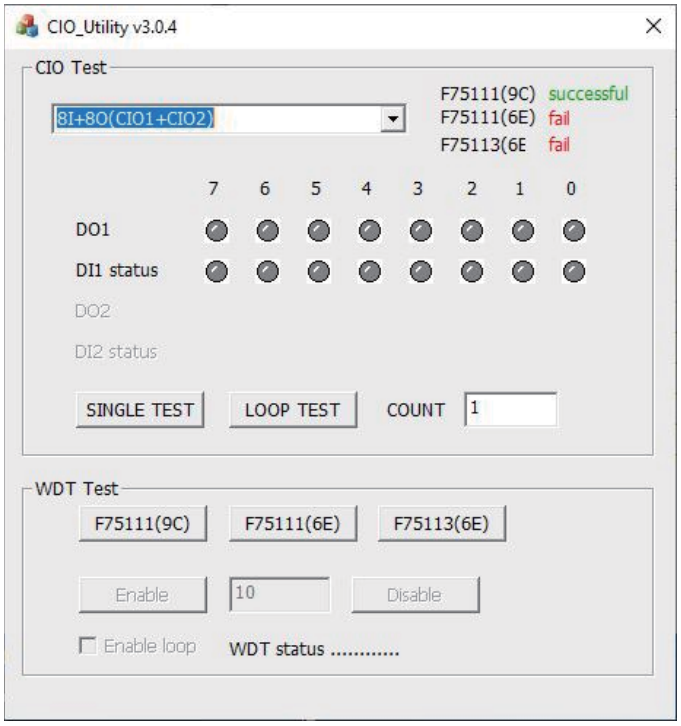
How to use this Demo Application







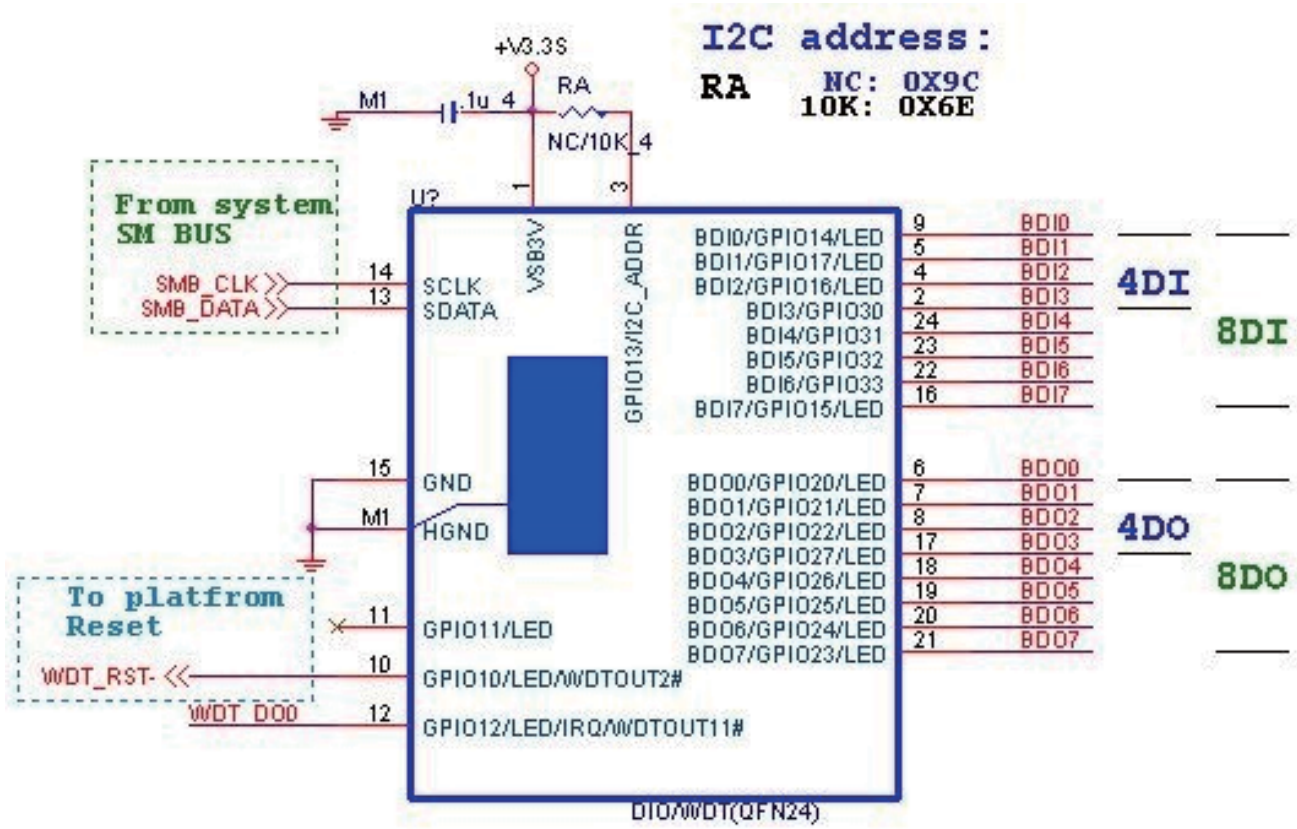




Attention Please:You must be install vcredist_x86.exe when first time you run the F75111_DIO.exe DEMO AP,The vcredist_x86.exe include all required DLL file.

1. Press the select your test "2i2o"，"4i4o"，"4i4o*2"，"F75111CIO116"，"F75113CIO116"，"8i+8o"
2. start test，select single mode or looptest

F75111 Layout Picture



Introduction F75111

Initial Internal F75111 port address (0x9c)

define GPIO1X, GPIO2X, GPIO3X to input or output

and Enable WDT function pin

Set F75111 DI/DO (sample code as below Get Input value / Set output value)

DO: InterDigitalOutput(BYTE byteValue))

DI: InterDigitalInput()

PULSE mode

Sample to setting GP33, 32, 31, 30 output 1mS low pulse signal.

```
{
  this->Write_Byte(F75111_INTERNAL_ADDR, GPIO3X_PULSE_CONTROL,          0x00);   //This is setting low,Level output
  this->Write_Byte(F75111_INTERNAL_ADDR, GPIO3X_PULSE_WIDTH_CONTROL,    0x01);   //This selects the pulse width to 1mS
  this->Write_Byte(F75111_INTERNAL_ADDR, GPIO3X_CONTROL_MODE,           0x0F);    //This is setting the GP33, 32, 31, 30 to output function.
  this->Write_Byte(F75111_INTERNAL_ADDR, GPIO3X_Output_Data             0x0F);   //This is setting the GP33, 32, 31, 30 output data.
}
```

Initial internal F75111

```
void F75111::InitInternalF75111()
{
   this->Write_Byte(F75111_INTERNAL_ADDR,GPIO1X_CONTROL_MODE  ,0x00);        //set GPIO1X to Input  function
   this->Write_Byte(F75111_INTERNAL_ADDR,GPIO3X_CONTROL_MODE  ,0x00);        //set GPIO3X to Input  function
   this->Write_Byte(F75111_INTERNAL_ADDR,GPIO2X_CONTROL_MODE  ,0xFF);        //set GPIO2X to Output function

   this->Write_Byte(F75111_INTERNAL_ADDR,F75111_CONFIGURATION    ,0x03);        //Enable WDT OUT function
}
```

Set output value

```
void F75111::InterDigitalOutput(BYTE byteValue)
{
    BYTE byteData = 0;
    byteData = (byteData & 0x01 )? byteValue + 0x01 : byteValue;
    byteData = (byteData & 0x02 )? byteValue + 0x02 : byteValue;
    byteData = (byteData & 0x04 )? byteValue + 0x04 : byteValue;
    byteData = (byteData & 0x80 )? byteValue + 0x08 : byteValue;
    byteData = (byteData & 0x40 )? byteValue + 0x10 : byteValue;
    byteData = (byteData & 0x20 )? byteValue + 0x20 : byteValue;
    byteData = (byteData & 0x10 )? byteValue + 0x40 : byteValue;
    byteData = (byteData & 0x08 )? byteValue + 0x80 : byteValue;                    // get value bit by bit

    this->Write_Byte(F75111_INTERNAL_ADDR,GPIO2X_OUTPUT_DATA,byteData);        // write byteData value via GPIO2X output pin
}
```

Get Input value

```
BYTE F75111::InterDigitalInput()
{
   BYTE byteGPIO1X = 0;
   BYTE byteGPIO3X = 0;
   BYTE byteData   = 0;

   this->Read_Byte(F75111_INTERNAL_ADDR,GPIO1X_INPUT_DATA,&byteGPIO1X) ;          // Get value from GPIO1X
   this->Read_Byte(F75111_INTERNAL_ADDR,GPIO3X_INPUT_DATA,&byteGPIO3X) ;          // Get value from GPIO3X

   byteGPIO1X = byteGPIO1X  & 0xF0;                                               // Mask unuseful value
   byteGPIO3X = byteGPIO3X  & 0x0F;                                               // Mask unuseful value

   byteData = ( byteGPIO1X & 0x10 )? byteData + 0x01 : byteData;
   byteData = ( byteGPIO1X & 0x80 )? byteData + 0x02 : byteData;
   byteData = ( byteGPIO1X & 0x40 )? byteData + 0x04 : byteData;
   byteData = ( byteGPIO3X & 0x01 )? byteData + 0x08 : byteData;

   byteData = ( byteGPIO3X & 0x02 )? byteData + 0x10 : byteData;
   byteData = ( byteGPIO3X & 0x04 )? byteData + 0x20 : byteData;
   byteData = ( byteGPIO3X & 0x08 )? byteData + 0x40 : byteData;
   byteData = ( byteGPIO1X & 0x20 )? byteData + 0x80 : byteData;                  // Get correct DI value from GPIO1X & GPIO3X

   return byteData;
}
```

define F75111 pin in F75111.h

```
//-------------------------------------------------------------------------------------------------------------------------------------------------
#define   F75111_INTERNAL_ADDR              0x9C              //  OnBoard  F75111 Chipset
#define   F75111_EXTERNAL_ADDR              0x6E              //  External F75111 Chipset
//-------------------------------------------------------------------------------------------------------------------------------------------------
#define   F75111_CONFIGURATION              0x03              //  Configure GPIO13 to WDT2 Function
//-------------------------------------------------------------------------------------------------------------------------------------------------
#define   GPIO1X_CONTROL_MODE               0x10              //  Select Output Mode or Input Mode
#define   GPIO2X_CONTROL_MODE               0x20              //  Select GPIO2X Output Mode or Input Mode
#define   GPIO3X_CONTROL_MODE               0x40              //  Select GPIO3X Output Mode or Input Mode
//-------------------------------------------------------------------------------------------------------------------------------------------------
#define   GPIO1X_INPUT_DATA                 0x12              //  GPIO1X Input
#define   GPIO3X_INPUT_DATA                 0x42              //  GPIO3X Input
//-------------------------------------------------------------------------------------------------------------------------------------------------
#define   GPIO2X_OUTPUT_DATA                0x21              //  GPIO2X Output
//-------------------------------------------------------------------------------------------------------------------------------------------------
#define   GPIO1X_PULSE_CONTROL              0x13              //  GPIO1x Level/Pulse Control Register
                                                              //  0:Level Mode
                                                              //  1:Pulse Mode
#define   GPIO1X_PULSE_WIDTH_CONTROL        0x14              //  GPIO1x Pulse Width Control Register
#define   GP1_PSWIDTH_500US                 0x00              //  When select Pulse mode:       500         us.
#define   GP1_PSWIDTH_1MS                   0x01              //  When select Pulse mode:       1           ms.
#define   GP1_PSWIDTH_20MS                  0x02              //  When select Pulse mode:       20          ms.
#define   GP1_PSWIDTH_100MS                 0x03              //  When select Pulse mode:       100         ms.
//-------------------------------------------------------------------------------------------------------------------------------------------------
#define   GPIO2X_PULSE_CONTROL              0x23              //  GPIO2x Level/Pulse Control Register
                                                              //  0:Level Mode
                                                              //  1:Pulse Mode
#define   GPIO2X_PULSE_WIDTH_CONTROL        0x24              //  GPIO2x Pulse Width Control Register
#define   GP2_PSWIDTH_500US                 0x00              //  When select Pulse mode:       500         us.
#define   GP2_PSWIDTH_1MS                   0x01              //  When select Pulse mode:       1           ms.
#define   GP2_PSWIDTH_20MS                  0x02              //  When select Pulse mode:       20          ms.
#define   GP2_PSWIDTH_100MS                 0x03              //  When select Pulse mode:       100         ms.
//-------------------------------------------------------------------------------------------------------------------------------------------------
#define   GPIO3X_PULSE_CONTROL              0x43              //  GPIO3x Level/Pulse Control Register
                                                              //  0:Level Mode
                                                              //  1:Pulse Mode
#define   GPIO3X_Output_Data                0x41              //  GPIO3x Output Data Register
#define   GPIO3X_PULSE_WIDTH_CONTROL        0x44              //  GPIO3x Pulse Width Control Register
#define   GP3_PSWIDTH_500US                 0x00              //  When select Pulse mode:       500         us.
#define   GP3_PSWIDTH_1MS                   0x01              //  When select Pulse mode:       1           ms.
#define   GP3_PSWIDTH_20MS                  0x02              //  When select Pulse mode:       20          ms.
#define   GP3_PSWIDTH_100MS                 0x03              //  When select Pulse mode:       100         ms.
//-------------------------------------------------------------------------------------------------------------------------------------------------
```

3-7-2 IO Device: F75111 CIO Utility CIO116

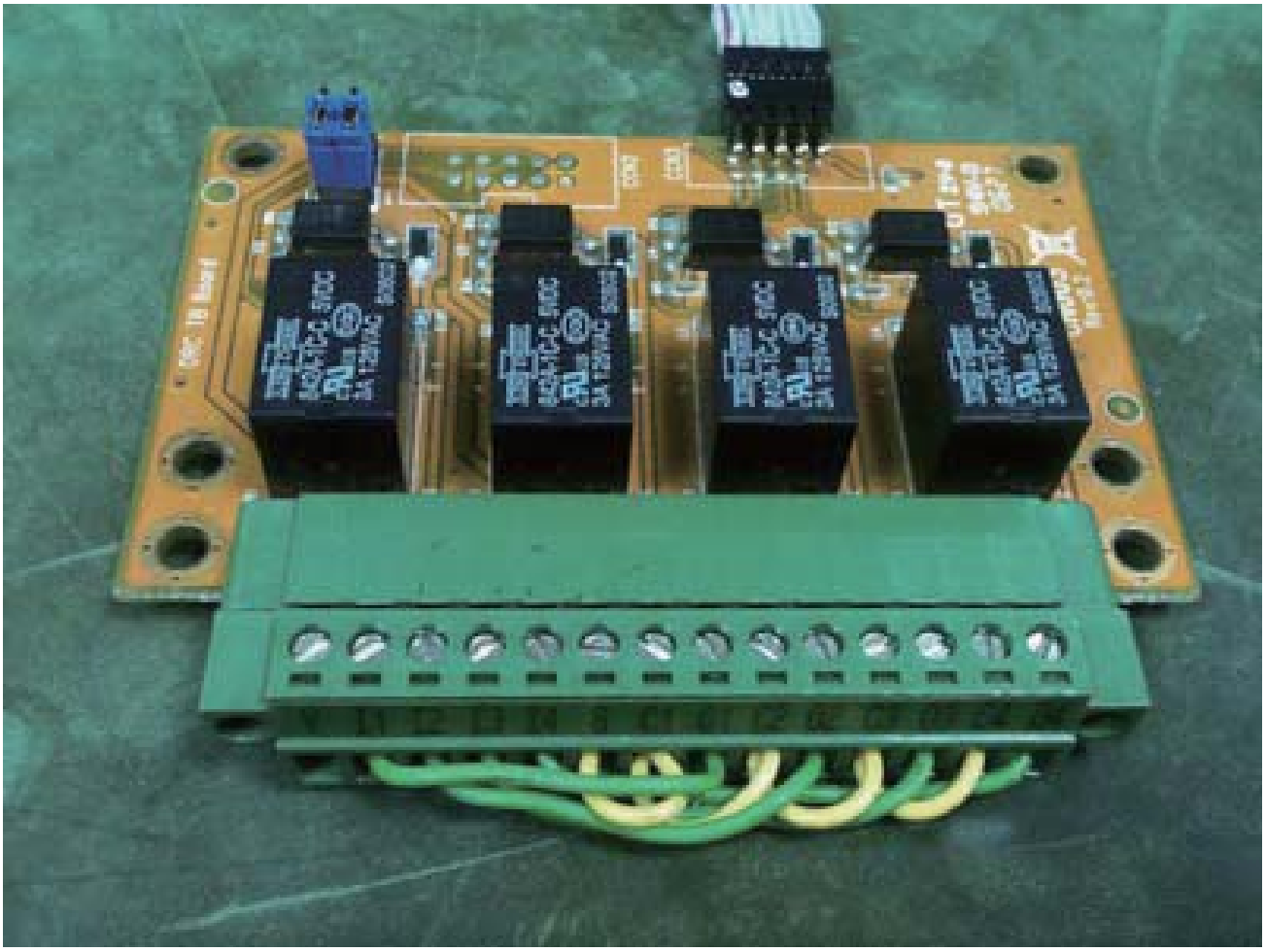
The Sample code source you can download from

Source file: CIO_Utility_Src_v3.0.3.tar.gz http://tprd.info/lexwiki/index.php/IO_Device:F75111_CIO_Utility_CIO116

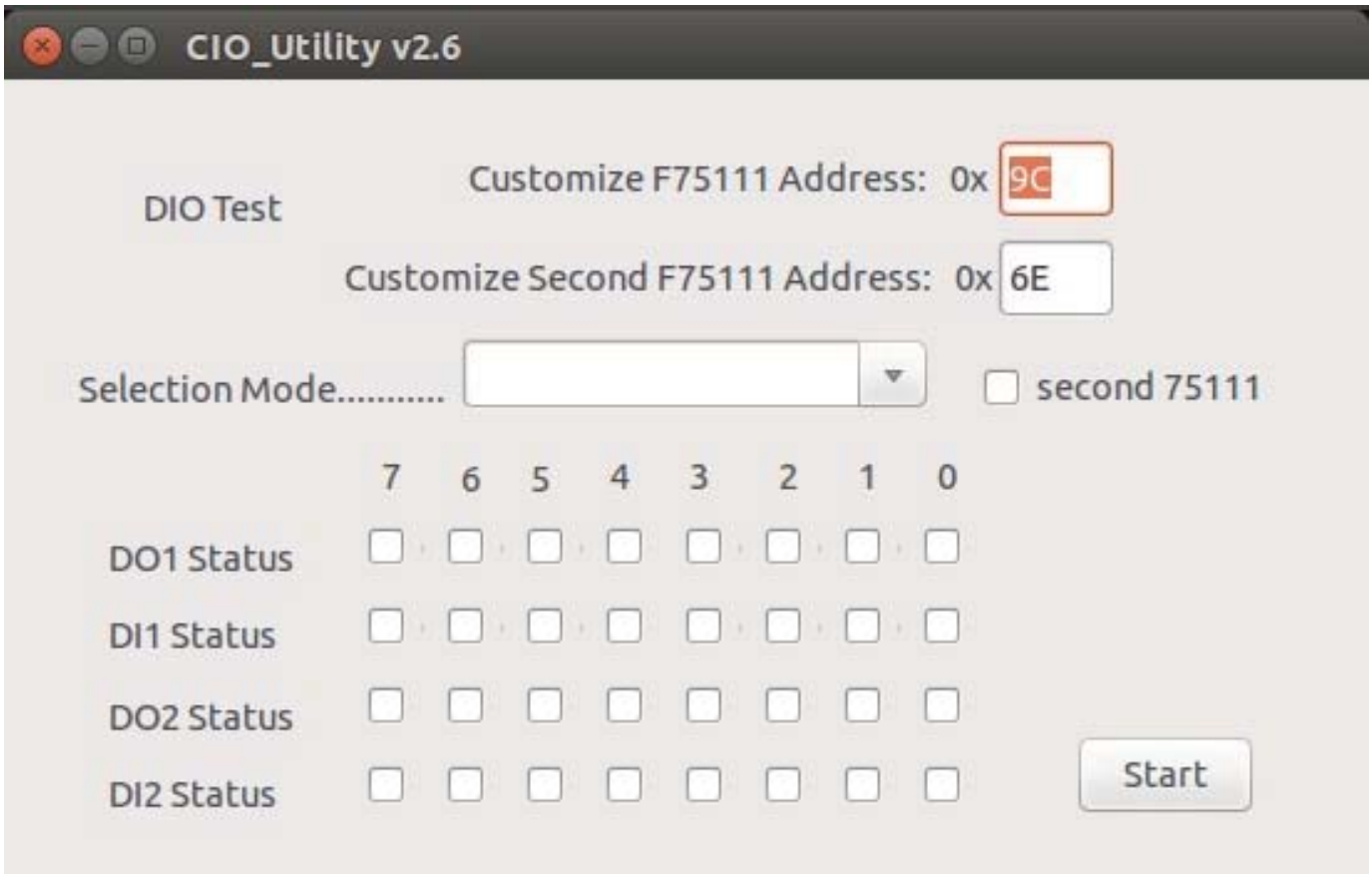
Binary file: CIO_Utility_Bin_v3.0.3_x32.tar.gz

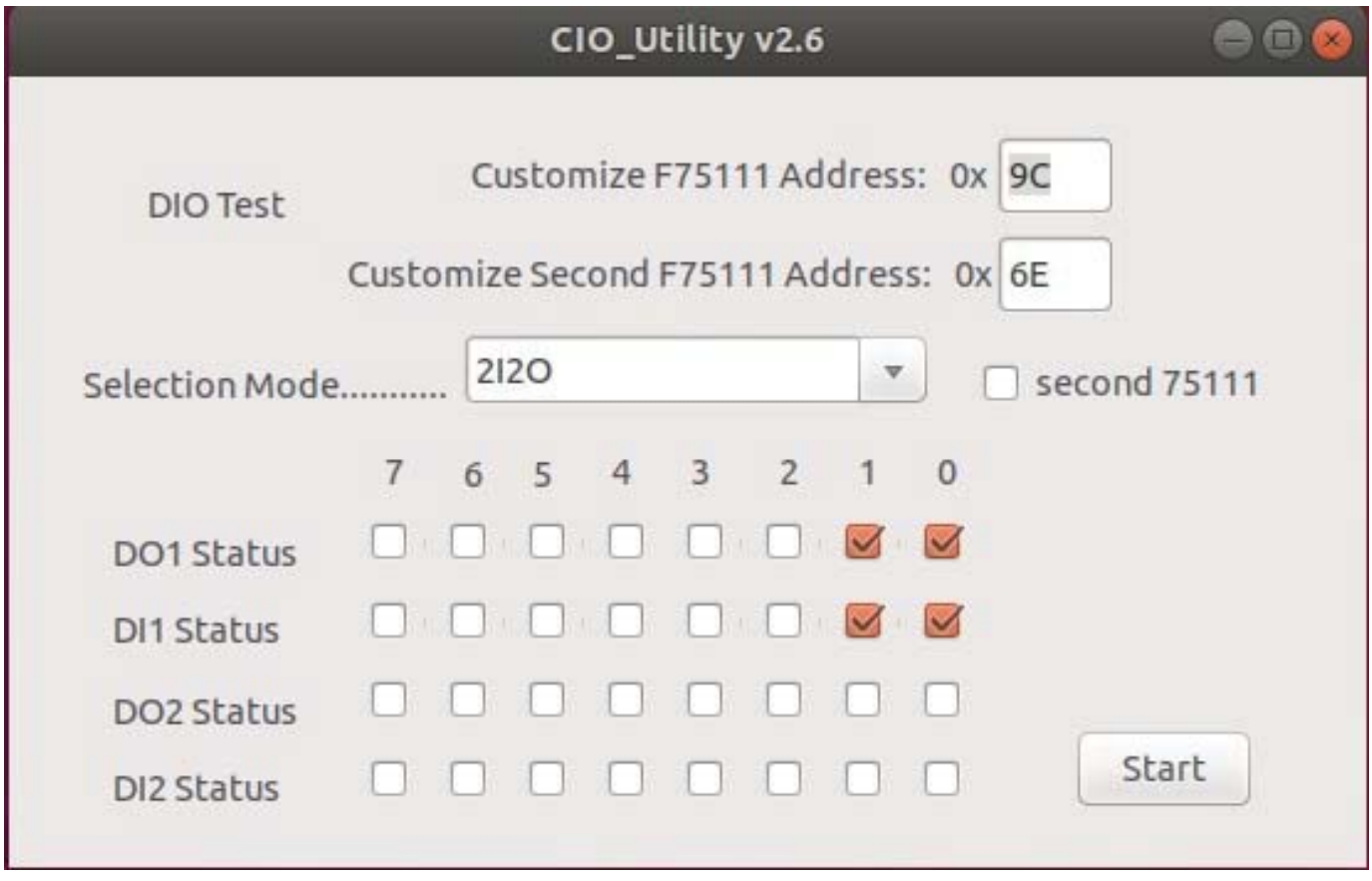
CIO_Utility_Bin_v3.0.3_x64.tar.gz

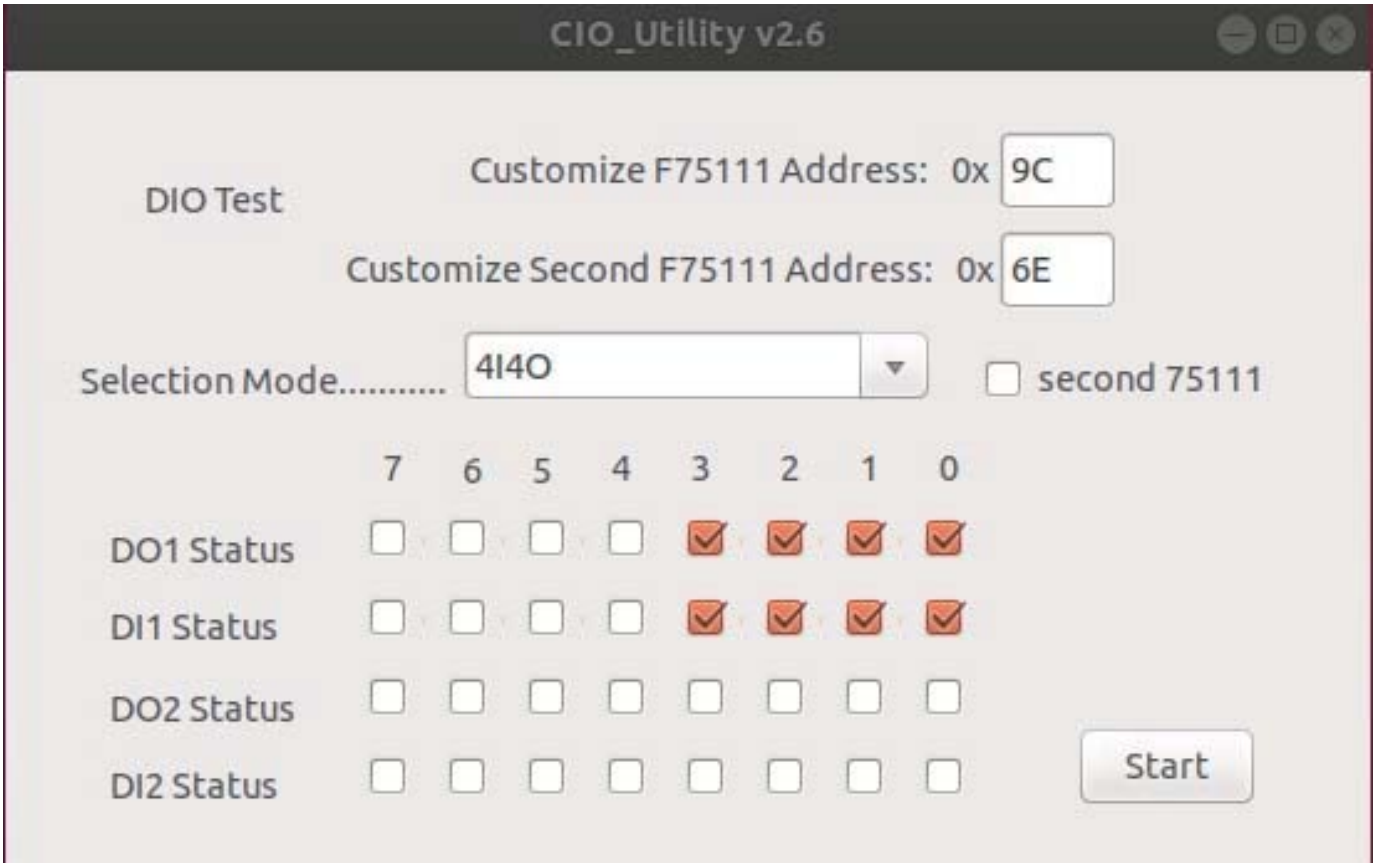
We do the demo test with a test tool which DIx connect to DOx with Relay.

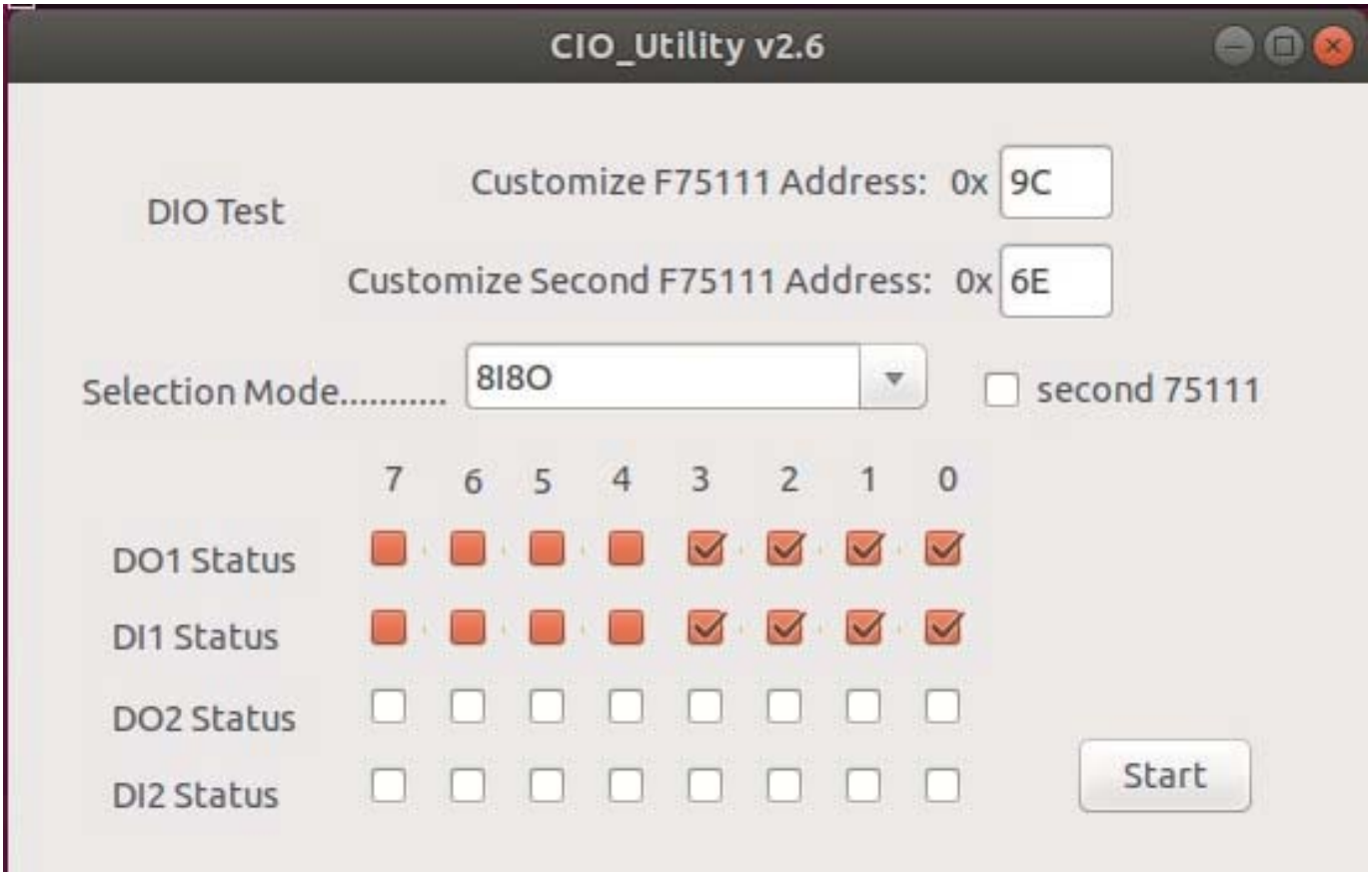


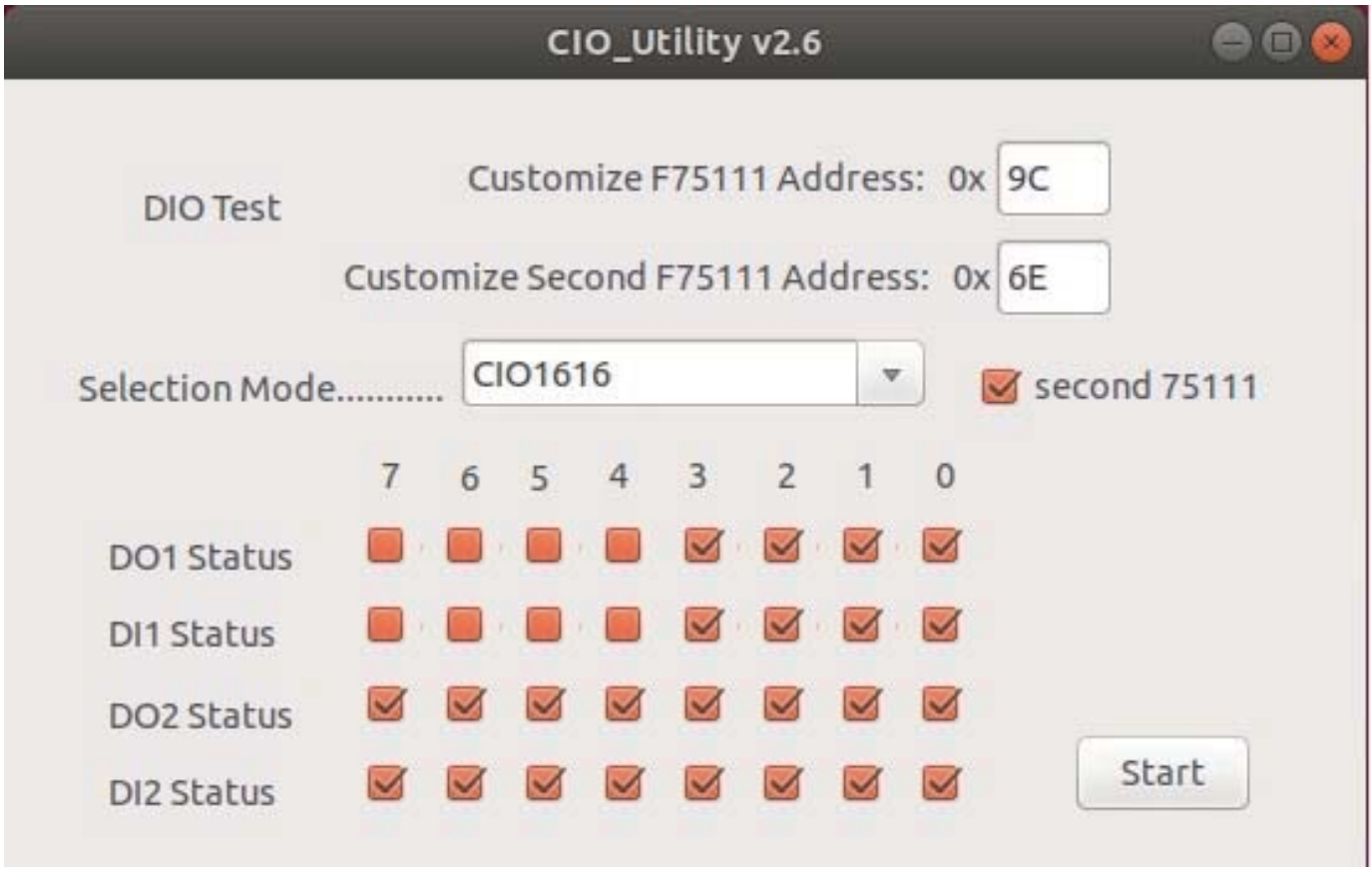
How to use this Demo Application






CIO_Utility v2.6
DIO Test
Customize F75111 Address: 0x 9C
Customize Second F75111 Address: 0x 6E
Selection Mode.....
4I4O
second 75111
7 6 5 4 3 2 1 0
DO1 Status
DI1 Status
DO2 Status
DI2 Status
Start

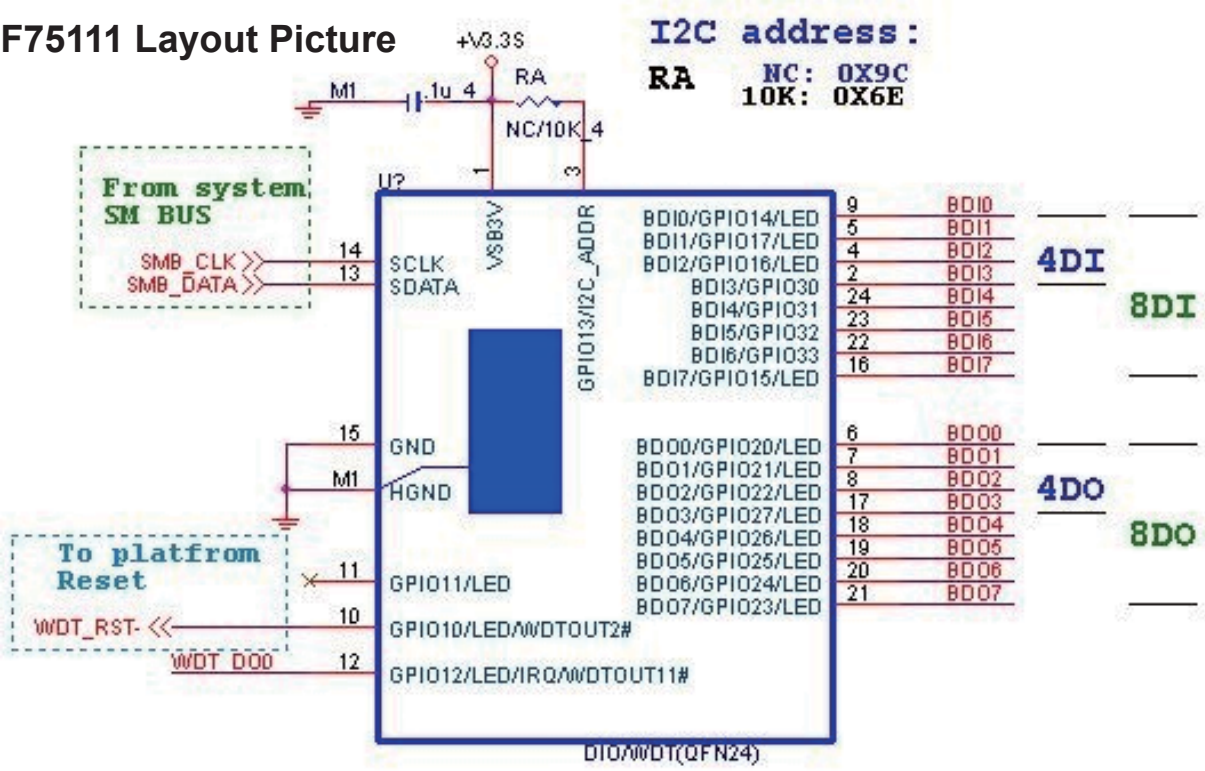

CIO_Utility v2.6
DIO Test
Customize F75111 Address: 0x 9C
Customize Second F75111 Address: 0x 6E
Selection Mode.....
8I8O
second 75111
7 6 5 4 3 2 1 0
DO1 Status
DI1 Status
DO2 Status
DI2 Status
Start



**Before executing the program began, Please switch to the highest authority , continued second F75111, chmod 777 and root: **

1. Press the select your test "2i2o","4i4o","8i8o","CIO1616"
2. If you test CIO1616 checkbutton second 75111
3. start button，select single mode or looptest

F75111 Layout Picture



Introduction

Initial Internal F75111 port address (0x9c)

```
define GPIO1X, GPIO2X, GPIO3X to input or output
and Enable WDT function pin
```

Set F75111 DI/DO (sample code as below Get Input value/Set output value)

```
DO: InterDigitalOutput(BYTE byteValue))
DI: InterDigitalInput()
```

PULSE mode

Sample to setting GP33, 32, 31, 30 output 1mS low pulse signal.

```
{
  this->Write_Byte(F75111_INTERNAL_ADDR, GPIO3X_PULSE_CONTROL,             0x00);  //This is setting low pulse output
  this->Write_Byte(F75111_INTERNAL_ADDR, GPIO3X_PULSE_WIDTH_CONTROL, 0x01);  //This selects the pulse width to 1mS
  this->Write_Byte(F75111_INTERNAL_ADDR, GPIO3X_CONTROL_MODE,              0x0F);  //This is setting the GP33, 32, 31, 30 to output function.
  this->Write_Byte(F75111_INTERNAL_ADDR, GPIO3X_Output_Data ,              0x0F);  //This is setting the GP33, 32, 31, 30 output data.
}
```

Initial internal F75111

```
void F75111::InitInternalF75111()
{
  this->Write_Byte(F75111_INTERNAL_ADDR,GPIO1X_CONTROL_MODE  ,0x00);      //set GPIO1X to Input  function
  this->Write_Byte(F75111_INTERNAL_ADDR,GPIO3X_CONTROL_MODE  ,0x00);      //set GPIO3X to Input  function
  this->Write_Byte(F75111_INTERNAL_ADDR,GPIO2X_CONTROL_MODE  ,0xFF);      //set GPIO2X to Output function

  this->Write_Byte(F75111_INTERNAL_ADDR,F75111_CONFIGURATION, 0x03);      //Enable WDT OUT function
}
```

Set output value

```
void F75111::InterDigitalOutput(BYTE byteValue)
{
   BYTE byteData = 0;
   byteData = (byteData & 0x01 )? byteValue + 0x01 : byteValue;
   byteData = (byteData & 0x02 )? byteValue + 0x02 : byteValue;
   byteData = (byteData & 0x04 )? byteValue + 0x04 : byteValue;
   byteData = (byteData & 0x80 )? byteValue + 0x08 : byteValue;
   byteData = (byteData & 0x40 )? byteValue + 0x10 : byteValue;
   byteData = (byteData & 0x20 )? byteValue + 0x20 : byteValue;
   byteData = (byteData & 0x10 )? byteValue + 0x40 : byteValue;
   byteData = (byteData & 0x08 )? byteValue + 0x80 : byteValue;                  // get value bit by bit

   this->Write_Byte(F75111_INTERNAL_ADDR,GPIO2X_OUTPUT_DATA,byteData);   // write byteData value via GPIO2X output pin
}
```

Get Input value

```
BYTE F75111::InterDigitalInput()
{
    BYTE byteGPIO1X = 0;
    BYTE byteGPIO3X = 0;
    BYTE byteData   = 0;

    this->Read_Byte(F75111_INTERNAL_ADDR,GPIO1X_INPUT_DATA,&byteGPIO1X) ;   // Get value from GPIO1X
    this->Read_Byte(F75111_INTERNAL_ADDR,GPIO3X_INPUT_DATA,&byteGPIO3X) ;   // Get value from GPIO3X

    byteGPIO1X = byteGPIO1X  & 0xF0;                        // Mask unuseful value
    byteGPIO3X = byteGPIO3X  & 0x0F;                        // Mask unuseful value

    byteData = ( byteGPIO1X & 0x10 )? byteData + 0x01 : byteData;
    byteData = ( byteGPIO1X & 0x80 )? byteData + 0x02 : byteData;
    byteData = ( byteGPIO1X & 0x40 )? byteData + 0x04 : byteData;
    byteData = ( byteGPIO3X & 0x01 )? byteData + 0x08 : byteData;

    byteData = ( byteGPIO3X & 0x02 )? byteData + 0x10 : byteData;
    byteData = ( byteGPIO3X & 0x04 )? byteData + 0x20 : byteData;
    byteData = ( byteGPIO3X & 0x08 )? byteData + 0x40 : byteData;
    byteData = ( byteGPIO1X & 0x20 )? byteData + 0x80 : byteData;          // Get correct DI value from GPIO1X & GPIO3X

    return byteData;
}
```

define F75111 pin in F75111.h

```
//-------------------------------------------------------------------------------------------------
#define   F75111_INTERNAL_ADDR                  0x9C        // OnBoard  F75111 Chipset
#define   F75111_EXTERNAL_ADDR                  0x6E        // External F75111 Chipset
//-------------------------------------------------------------------------------------------------
#define   F75111_CONFIGURATION                  0x03        // Configure GPIO13 to WDT2 Function
//-------------------------------------------------------------------------------------------------
#define   GPIO1X_CONTROL_MODE                   0x10        // Select Output Mode or Input Mode
#define   GPIO2X_CONTROL_MODE                   0x20        // Select GPIO2X Output Mode or Input Mode
#define   GPIO3X_CONTROL_MODE                   0x40        // Select GPIO3X Output Mode or Input Mode
//-------------------------------------------------------------------------------------------------
#define   GPIO1X_INPUT_DATA                     0x12        // GPIO1X Input
#define   GPIO3X_INPUT_DATA                     0x42        // GPIO3X Input
//-------------------------------------------------------------------------------------------------
#define   GPIO2X_OUTPUT_DATA                    0x21        // GPIO2X Output
//-------------------------------------------------------------------------------------------------
#define   GPIO1X_PULSE_CONTROL                  0x13        // GPIO1x Level/Pulse Control Register
                                                            // 0:Level Mode
                                                            // 1:Pulse Mode
#define   GPIO1X_PULSE_WIDTH_CONTROL            0x14        // GPIO1x Pulse Width Control Register
#define   GP1_PSWIDTH_500US                     0x00        // When select Pulse mode: 500     us.
#define   GP1_PSWIDTH_1MS                       0x01        // When select Pulse mode: 1       ms.
#define   GP1_PSWIDTH_20MS                      0x02        // When select Pulse mode: 20      ms.
#define   GP1_PSWIDTH_100MS                     0x03        // When select Pulse mode: 100     ms.
//-------------------------------------------------------------------------------------------------
#define   GPIO2X_PULSE_CONTROL                  0x23        // GPIO2x Level/Pulse Control Register
                                                            // 0:Level Mode
                                                            // 1:Pulse Mode
#define   GPIO2X_PULSE_WIDTH_CONTROL            0x24        // GPIO2x Pulse Width Control Register
#define   GP2_PSWIDTH_500US                     0x00        // When select Pulse mode: 500     us.
#define   GP2_PSWIDTH_1MS                       0x01        // When select Pulse mode: 1       ms.
#define   GP2_PSWIDTH_20MS                      0x02        // When select Pulse mode: 20      ms.
#define   GP2_PSWIDTH_100MS                     0x03        // When select Pulse mode: 100     ms.
//-------------------------------------------------------------------------------------------------
#define   GPIO3X_PULSE_CONTROL                  0x43        // GPIO3x Level/Pulse Control Register
                                                            // 0:Level Mode
                                                            // 1:Pulse Mode
#define   GPIO3X_Output_Data                    0x41        // GPIO3x Output Data Register
#define   GPIO3X_PULSE_WIDTH_CONTROL            0x44        // GPIO3x Pulse Width Control Register
#define   GP3_PSWIDTH_500US                     0x00        // When select Pulse mode: 500     us.
#define   GP3_PSWIDTH_1MS                       0x01        // When select Pulse mode: 1       ms.
#define   GP3_PSWIDTH_20MS                      0x02        // When select Pulse mode: 20      ms.
#define   GP3_PSWIDTH_100MS                     0x03        // When select Pulse mode: 100     ms.
//-------------------------------------------------------------------------------------------------
```

3-7-3 IO Device:F75111 CIO Utility Console under linux

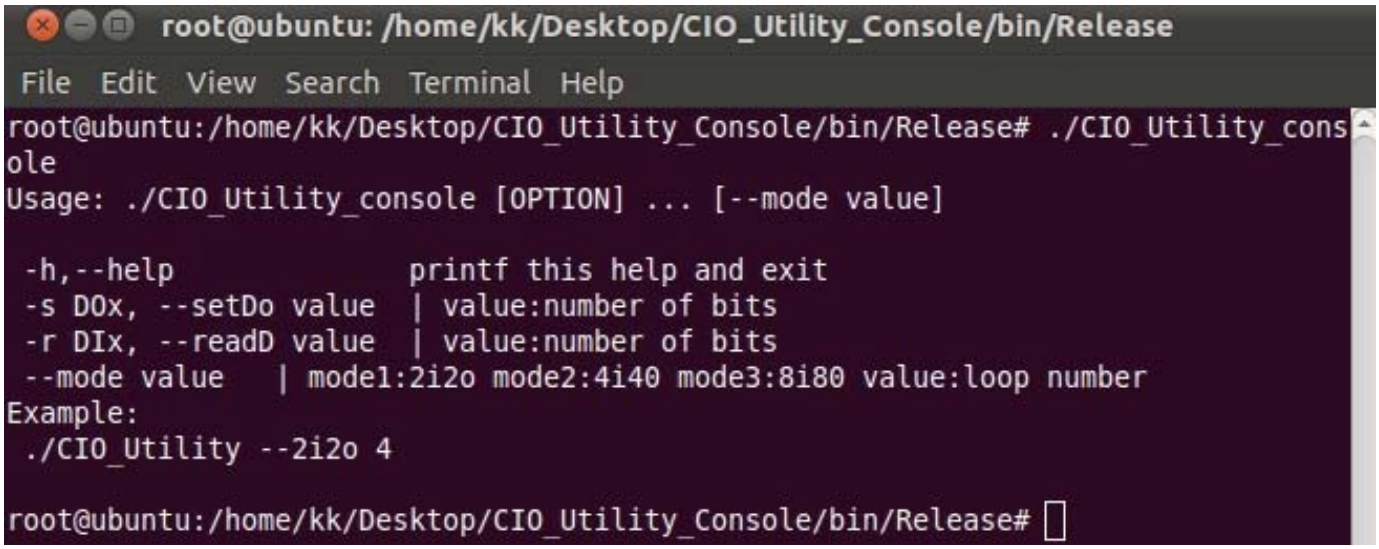
The Sample code source you can download from

Source file: CIO_Utility_Console_Src_v1.2.tar.gz

Binary file: CIO_Utility_Console_Src_v1.3.tar.gz

http://tprd.info/lexwiki/index.php/IO_Device:F75111_CIO_Utility_Console_under_linux

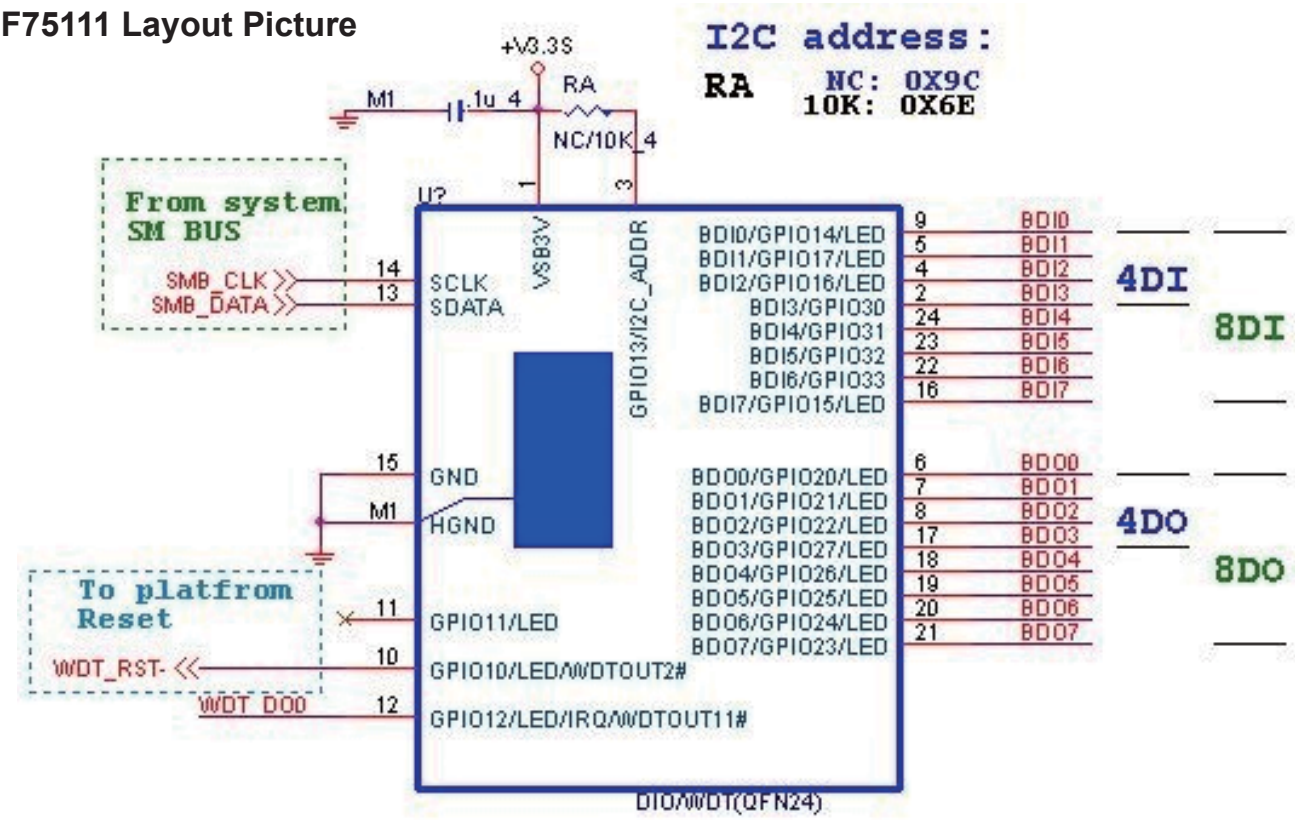
How to use this Demo Application



1. The program must control I/O device, when you use this you must change user to root, you can use this command "sudo su"
2. enter "./CIO_Utility_console -h"show help function

Example:
CIO_Utility_console --4i4o 1

F75111 Layout Picture



Introduction

Initial Internal F75111 port address (0x9c)

define GPIO1X, GPIO2X, GPIO3X to input or output

and Enable WDT function pin

Set F75111 DI/DO (sample code as below Get Input value/Set output value)

DO: InterDigitalOutput(BYTE byteValue))

DI: InterDigitalInput()

PULSE mode

Sample to setting GP33, 32, 31, 30 output 1mS low pulse signal.

```
{
  this->Write_Byte(F75111_INTERNAL_ADDR, GPIO3X_PULSE_CONTROL,          0x00);  //This is setting low pulse output
  this->Write_Byte(F75111_INTERNAL_ADDR, GPIO3X_PULSE_WIDTH_CONTROL, 0x01);  //This selects the pulse width to 1mS
  this->Write_Byte(F75111_INTERNAL_ADDR, GPIO3X_CONTROL_MODE,          0x0F);  //This is setting the GP33, 32, 31, 30 to output function.
  this->Write_Byte(F75111_INTERNAL_ADDR, GPIO3X_Output_Data ,          0x0F);  //This is setting the GP33, 32, 31, 30 output data.
}
```

Initial internal F75111

```
void F75111::InitInternalF75111()
{
    this->Write_Byte(F75111_INTERNAL_ADDR,GPIO1X_CONTROL_MODE     ,0x00);      //set GPIO1X to Input  function
    this->Write_Byte(F75111_INTERNAL_ADDR,GPIO3X_CONTROL_MODE     ,0x00);      //set GPIO3X to Input  function
    this->Write_Byte(F75111_INTERNAL_ADDR,GPIO2X_CONTROL_MODE     ,0xFF);      //set GPIO2X to Output function

    this->Write_Byte(F75111_INTERNAL_ADDR,F75111_CONFIGURATION       ,0x03);      //Enable WDT OUT function
}
```

Set output value

```
void F75111::InterDigitalOutput(BYTE byteValue)
{
    BYTE byteData = 0;
    byteData = (byteData & 0x01 )? byteValue + 0x01 : byteValue;
    byteData = (byteData & 0x02 )? byteValue + 0x02 : byteValue;
    byteData = (byteData & 0x04 )? byteValue + 0x04 : byteValue;
    byteData = (byteData & 0x80 )? byteValue + 0x08 : byteValue;
    byteData = (byteData & 0x40 )? byteValue + 0x10 : byteValue;
    byteData = (byteData & 0x20 )? byteValue + 0x20 : byteValue;
    byteData = (byteData & 0x10 )? byteValue + 0x40 : byteValue;
    byteData = (byteData & 0x08 )? byteValue + 0x80 : byteValue;                    // get value bit by bit

    this->Write_Byte(F75111_INTERNAL_ADDR,GPIO2X_OUTPUT_DATA,byteData);          // write byteData value via GPIO2X output pin
}
```

Get Input value

```
BYTE F75111::InterDigitalInput()
{
    BYTE byteGPIO1X = 0;
    BYTE byteGPIO3X = 0;
    BYTE byteData   = 0;

    this->Read_Byte(F75111_INTERNAL_ADDR,GPIO1X_INPUT_DATA,&byteGPIO1X) ;   // Get value from GPIO1X
    this->Read_Byte(F75111_INTERNAL_ADDR,GPIO3X_INPUT_DATA,&byteGPIO3X) ;   // Get value from GPIO3X

    byteGPIO1X = byteGPIO1X  & 0xF0;                        // Mask unuseful value
    byteGPIO3X = byteGPIO3X  & 0x0F;                        // Mask unuseful value

    byteData = ( byteGPIO1X & 0x10 )? byteData + 0x01 : byteData;
    byteData = ( byteGPIO1X & 0x80 )? byteData + 0x02 : byteData;
    byteData = ( byteGPIO1X & 0x40 )? byteData + 0x04 : byteData;
    byteData = ( byteGPIO3X & 0x01 )? byteData + 0x08 : byteData;

    byteData = ( byteGPIO3X & 0x02 )? byteData + 0x10 : byteData;
    byteData = ( byteGPIO3X & 0x04 )? byteData + 0x20 : byteData;
    byteData = ( byteGPIO3X & 0x08 )? byteData + 0x40 : byteData;
    byteData = ( byteGPIO1X & 0x20 )? byteData + 0x80 : byteData;       // Get correct DI value from GPIO1X & GPIO3X

    return byteData;
}
```

define F75111 pin in F75111.h

```
//-------------------------------------------------------------------------------------------------
#define   F75111_INTERNAL_ADDR                  0x9C        // OnBoard  F75111 Chipset
#define   F75111_EXTERNAL_ADDR                  0x6E        // External F75111 Chipset
//-------------------------------------------------------------------------------------------------
#define   F75111_CONFIGURATION                  0x03        // Configure GPIO13 to WDT2 Function
//-------------------------------------------------------------------------------------------------
#define   GPIO1X_CONTROL_MODE                   0x10        // Select Output Mode or Input Mode
#define   GPIO2X_CONTROL_MODE                   0x20        // Select GPIO2X Output Mode or Input Mode
#define   GPIO3X_CONTROL_MODE                   0x40        // Select GPIO3X Output Mode or Input Mode
//-------------------------------------------------------------------------------------------------
#define   GPIO1X_INPUT_DATA                     0x12        // GPIO1X Input
#define   GPIO3X_INPUT_DATA                     0x42        // GPIO3X Input
//-------------------------------------------------------------------------------------------------
#define   GPIO2X_OUTPUT_DATA                    0x21        // GPIO2X Output
//-------------------------------------------------------------------------------------------------
#define   GPIO1X_PULSE_CONTROL                  0x13        // GPIO1x Level/Pulse Control Register
                                                            // 0:Level Mode
                                                            // 1:Pulse Mode
#define   GPIO1X_PULSE_WIDTH_CONTROL            0x14        // GPIO1x Pulse Width Control Register
#define   GP1_PSWIDTH_500US                     0x00        // When select Pulse mode:   500     us.
#define   GP1_PSWIDTH_1MS                       0x01        // When select Pulse mode:   1       ms.
#define   GP1_PSWIDTH_20MS                      0x02        // When select Pulse mode:   20      ms.
#define   GP1_PSWIDTH_100MS                     0x03        // When select Pulse mode:   100     ms.
//-------------------------------------------------------------------------------------------------
#define   GPIO2X_PULSE_CONTROL                  0x23        // GPIO2x Level/Pulse Control Register
                                                            // 0:Level Mode
                                                            // 1:Pulse Mode
#define   GPIO2X_PULSE_WIDTH_CONTROL            0x24        // GPIO2x Pulse Width Control Register
#define   GP2_PSWIDTH_500US                     0x00        // When select Pulse mode:   500     us.
#define   GP2_PSWIDTH_1MS                       0x01        // When select Pulse mode:   1       ms.
#define   GP2_PSWIDTH_20MS                      0x02        // When select Pulse mode:   20      ms.
#define   GP2_PSWIDTH_100MS                     0x03        // When select Pulse mode:   100     ms.
//-------------------------------------------------------------------------------------------------
#define   GPIO3X_PULSE_CONTROL                  0x43        // GPIO3x Level/Pulse Control Register
                                                            // 0:Level Mode
                                                            // 1:Pulse Mode
#define   GPIO3X_Output_Data                    0x41        // GPIO3x Output Data Register
#define   GPIO3X_PULSE_WIDTH_CONTROL            0x44        // GPIO3x Pulse Width Control Register
#define   GP3_PSWIDTH_500US                     0x00        // When select Pulse mode:   500     us.
#define   GP3_PSWIDTH_1MS                       0x01        // When select Pulse mode:   1       ms.
#define   GP3_PSWIDTH_20MS                      0x02        // When select Pulse mode:   20      ms.
#define   GP3_PSWIDTH_100MS                     0x03        // When select Pulse mode:   100     ms.
//-------------------------------------------------------------------------------------------------
```

3-8 USB3.0 and USB2.0

- **CU34: Dual USB3.0 Port 3/4 Type A Connector**

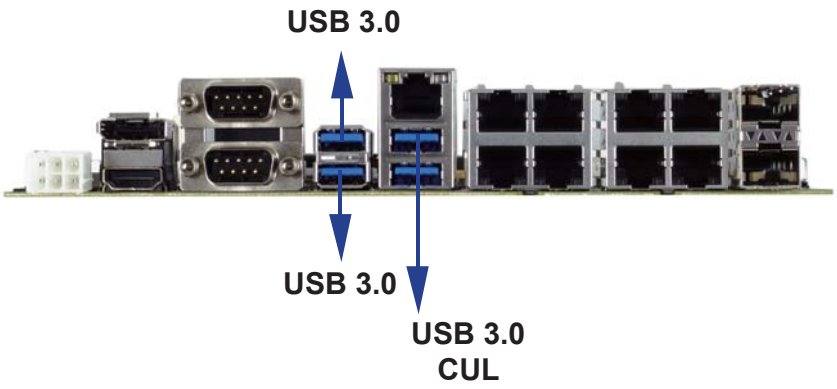
PIN NO.	DESCRIPTION	PIN NO.	DESCRIPTION
1	VBUS	5	SS_RX-
2	D-	6	SS_RX+
3	D+	7	GND
4	GND	8	SS_TX-
		9	SS_TX+

Note: the power supply 0.9A for each USB3.0 respect specification.

- **CUL: Dual USB3.0 Port 0/1 Type A Connector**

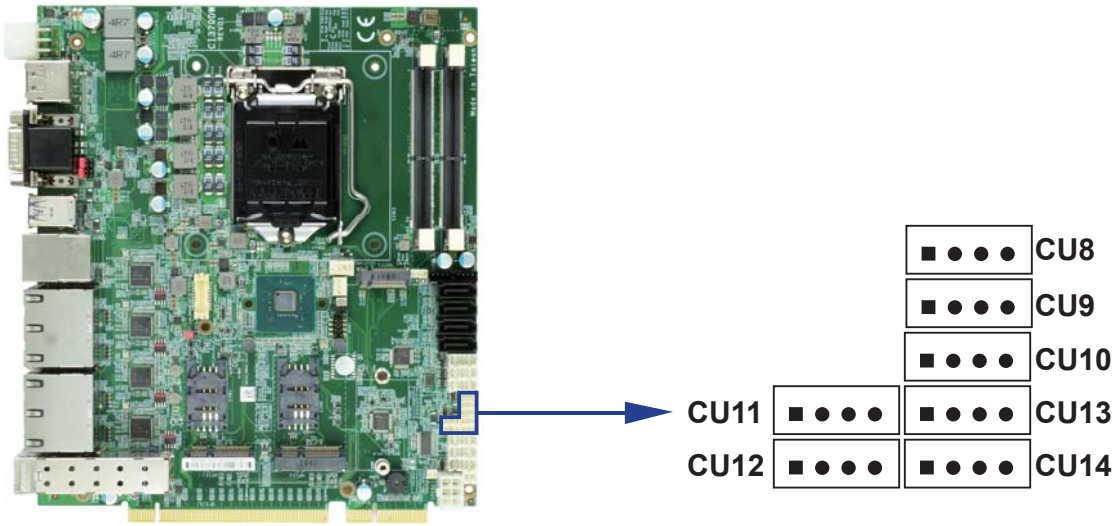
PIN NO.	DESCRIPTION	PIN NO.	DESCRIPTION
1	VBUS	5	SS_RX-
2	D-	6	SS_RX+
3	D+	7	GND
4	GND	8	SS_TX-
		9	SS_TX+

Note: the power supply 0.9A for each USB3.0 respect specification.



- **CU8/9/10/11/12/13/14 USB2.0 port (1x4pin 1.25mm Wafer)**

PIN NO.	DESCRIPTION
1	+5V
2	DATA-
3	DATA+
4	GND



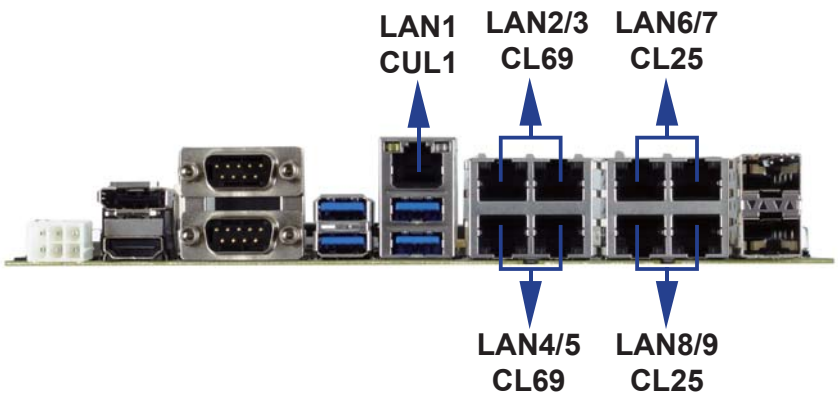
3-9 LAN

- **CUL1: RJ45 LAN1 Connector**

PIN NO.	DESCRIPTION	PIN NO.	DESCRIPTION
1	MDI0+	5	MDI2-
2	MDI0-	6	MDI1-
3	MDI1+	7	MDI3+
4	MID2+	8	MDI3-

- **CL69, CL25: RJ45 LAN2~5, 6~9 Connector**

PIN NO.	DESCRIPTION	PIN NO.	DESCRIPTION
1	MDI0+	5	MDI2-
2	MDI0-	6	MDI1-
3	MDI1+	7	MDI3+
4	MID2+	8	MDI3-

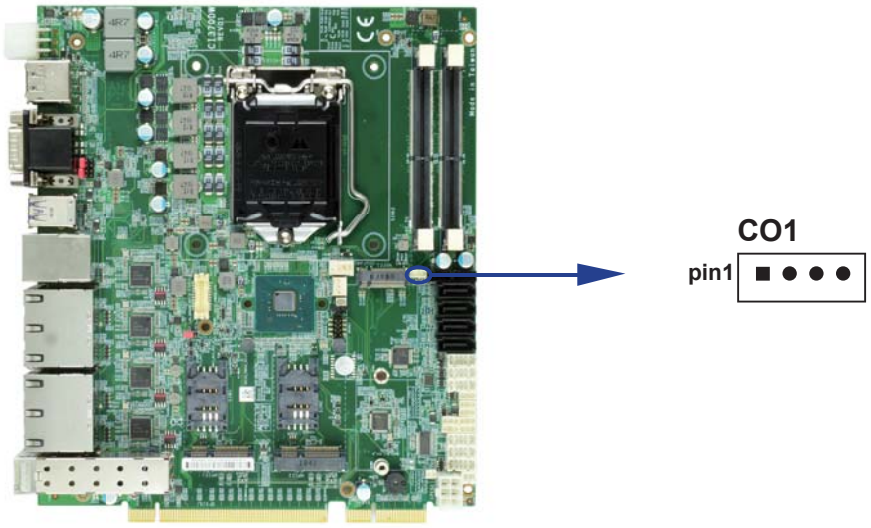


- **SFP12 LAN10, 11 Fiber 1000MB Cage**

PIN NO.	DESCRIPTION	PIN NO.	DESCRIPTION
1	GND	2	SFP_TX_FAULT
3	SFP_TX_DISABLE	4	SFP_SDA
5	SFP_SCL	6	SFP_MOD_ABS
7	SFP_RS0	8	SFP_SIG_DET
9	SFP_RS1	10	GND
11	GND	12	SFP_RD_N
13	SFP_RD_P	14	GND
15	+3.3V	16	+3.3V
17	GND	18	SFP_TD_P
19	SFP_TD_N	20	GND
21	NC	22	NC

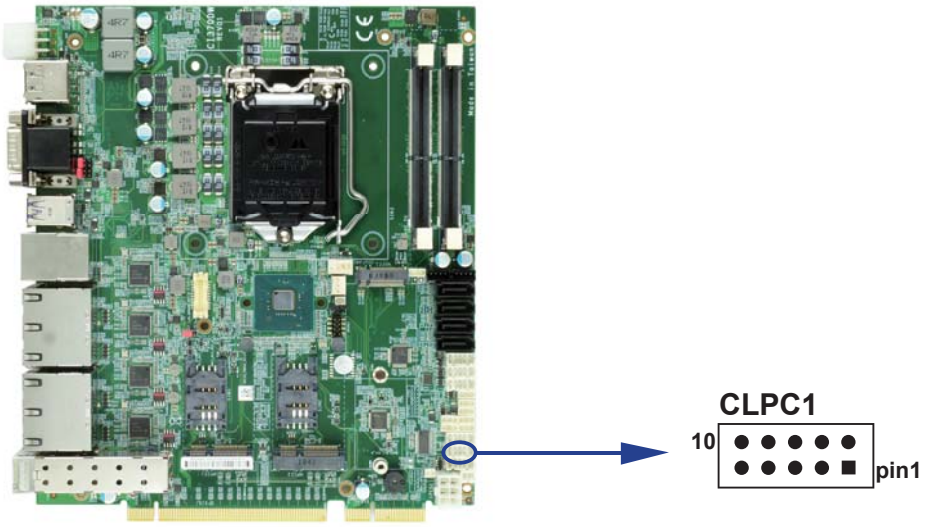
3-10 CO1: I2C Bus 4pin (1.25mm) Wafer

PIN NO.	DESCRIPTION
1	+3.3V
2	GND
3	I2C Clock
4	I2C DATA



3-11 CLPC1: for LPC signal 2x5 pin wafer (2.0mm).

PIN NO.	DESCRIPTION	PIN NO.	DESCRIPTION
1	AD3	2	CLK
3	AD2	4	FRAME
5	AD1	6	RESET
7	AD0	8	SERIAL IRQ
9	GND	10	+3.3V

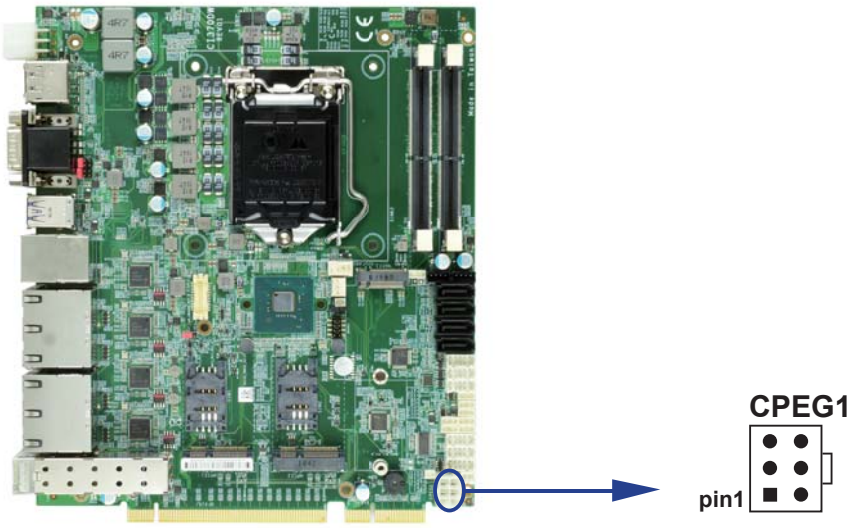


3-12 Power

- **CPEG1: Gold finger DC Power input (ATX 2x3 pin 2.54mm Wafer)**

PIN NO.	DESCRIPTION
1,3,5	DC-IN
2,4,6	GND

Note: Very important check DC-in Voltage that is only +12V for Gold finger device used.



• CPI1: Motherboard DC Power input (ATX 2x3 pin 2.54mm Wafer)

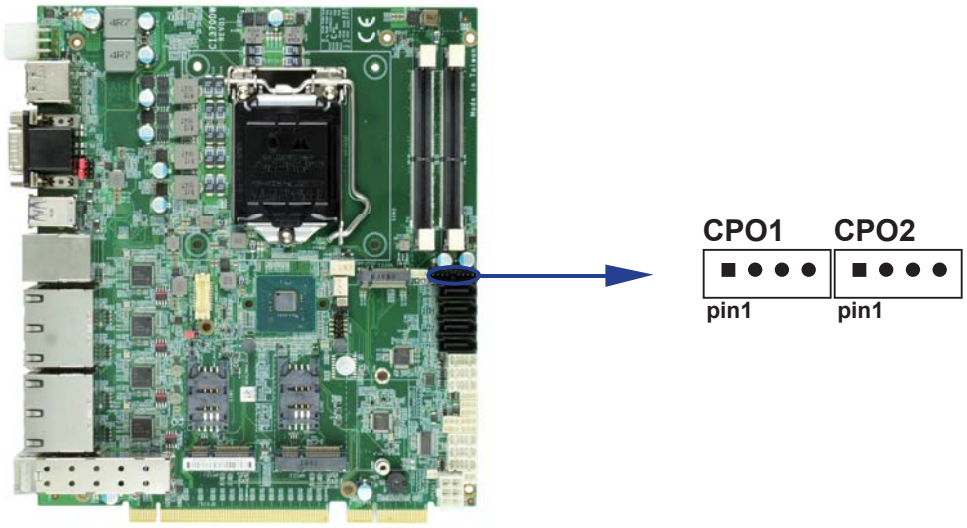
PIN NO.	DESCRIPTION
1,3,5	DC-IN
2,4,6	GND

Note: Very important check DC-in Voltage that is only +24V for Motherboard power used.

• CPO1/2: +12V/+5V DC voltage output wafer connector (Black) (1x4 pin 2.0mm)

PIN NO.	DESCRIPTION
1	+5V
2	GND
3	GND
4	+12V*

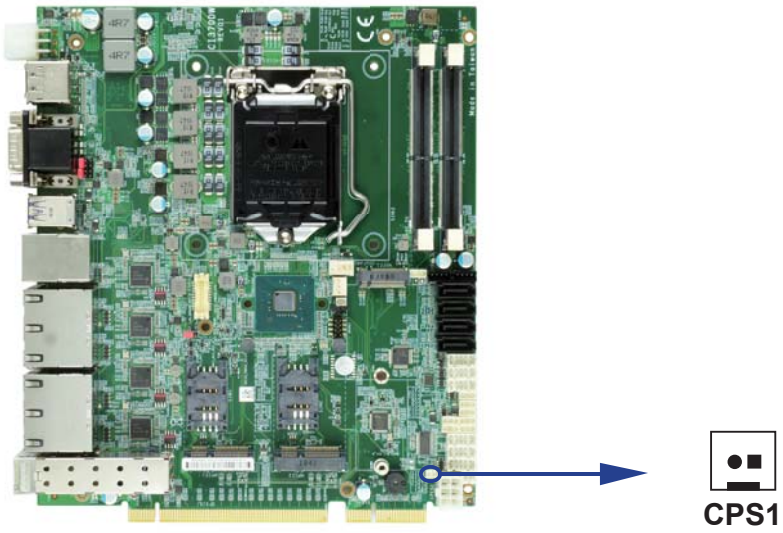
* Note: Attention! Check Device Power in spec



• CPS1: External Power-On Sync contrl

PIN NO.	DESCRIPTION
1	GND
2	PS_ON_N

* Note: The sync signal is Low active. When Motherboard powered on that is Low, Power-off is high.
The signal is +3.3V tolerance.
It can be used for gold finger power to sync mother power sequence.

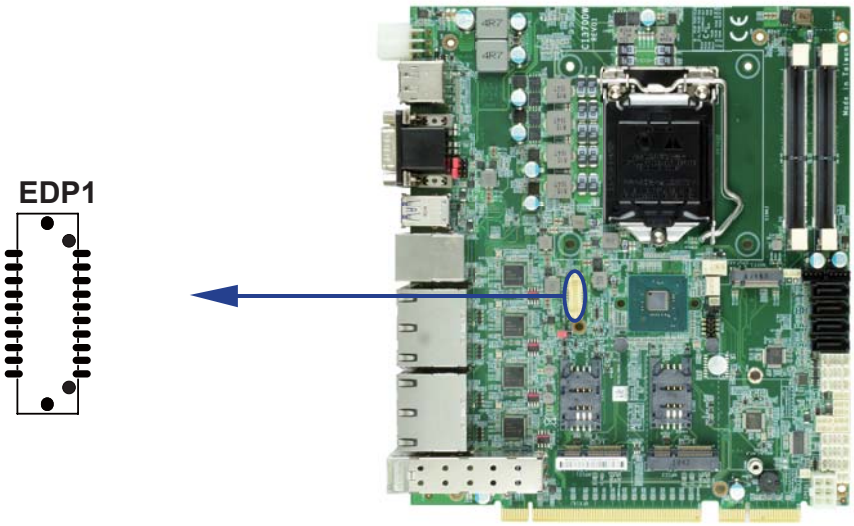


3-13 EDP interface

● **EDP1: eDP interface (2x10 pin 1.25mm wafer)**

PIN NO.	DESCRIPTION	PIN NO.	DESCRIPTION
1	Lane-0-DATA-	2	+12V
3	Lane-0-DATA+	4	+12V
5	Lane-1-DATA-	6	GND
7	Lane-1-DATA+	8	GND
9	Backlight Enable	10	GND
11	PWM dimming	12	GND
13	I2C Clock	14	+LCD (5V or 3.3V)
15	I2C Data	16	+LCD (5V or 3.3V)
17	eDP Aux+	18	+LCD (5V or 3.3V)
19	eDP Aux-	20	EDP_HPD

Note:
1. eDP interface support 2 lanes.
2. JVL1: eDP panel +5V/+3.3V (default) Voltage select.
3. eDP1 PIN 9 for panel backlight enable. +3.3V Level
4. eDP1 PIN 11 for panel backlight dimming control

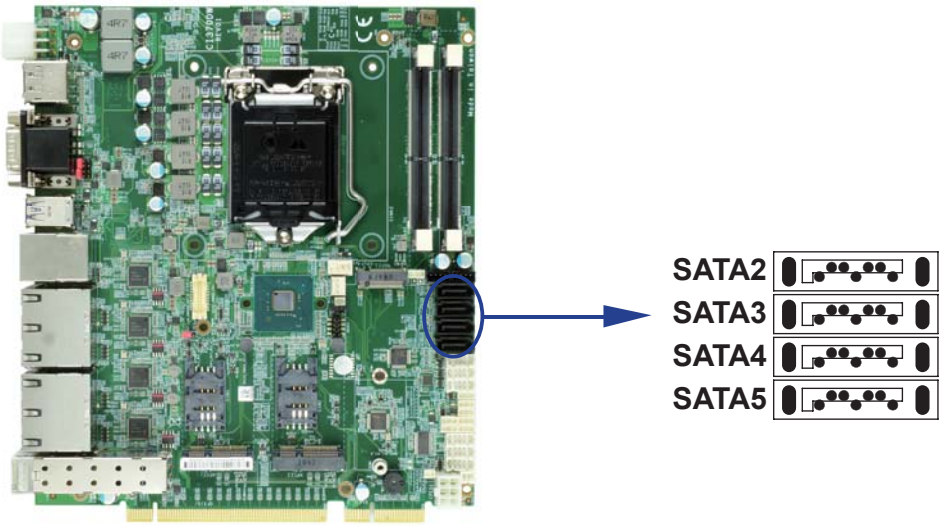


3-14 SATA interface

- **SATA2/3/4/5: SATA port 1x7 pin Connector**

PIN NO.	Description
1	GND
2	TX+
3	TX-
4	GND
5	RX-
6	RX+
7	GND

Note: 1. CPO1 provide SATA HDD power +12V, GND, +5V

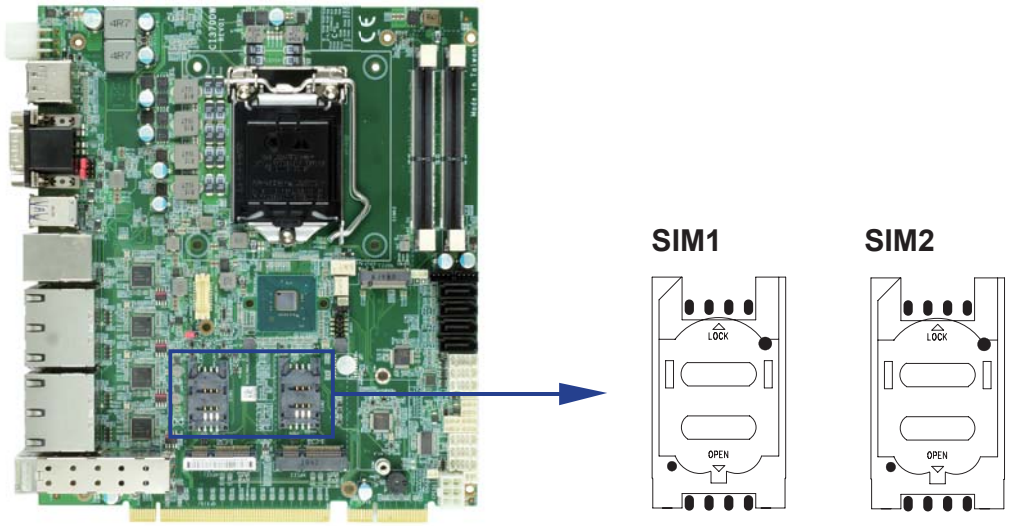


3-15 SIM socket

● SIM1/2 SIM Socket.

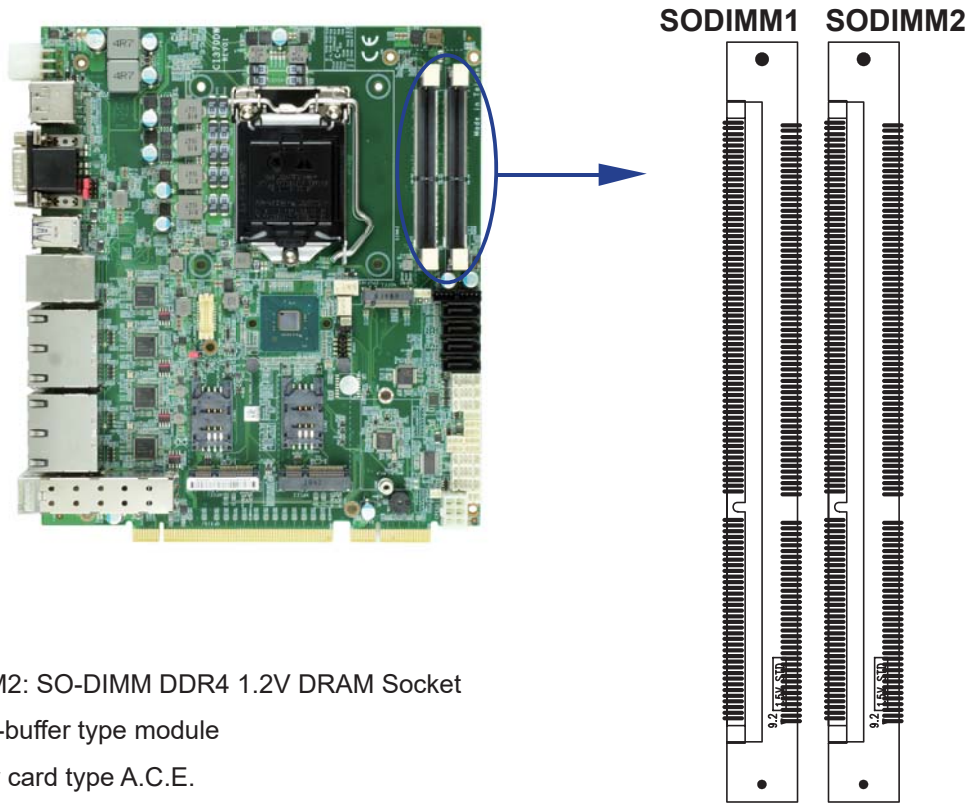
PIN NO.	DESCRIPTION
1	PWR
2	Reset
3	CLK
4	NC
5	GND
6	VPP
7	DATA
8	NC

Note: 1. SIM1 data source from MPCE1. SIM2 data source from MPCE2.



3-16 SODIMM socket

- **SODIMM1/2 socket**

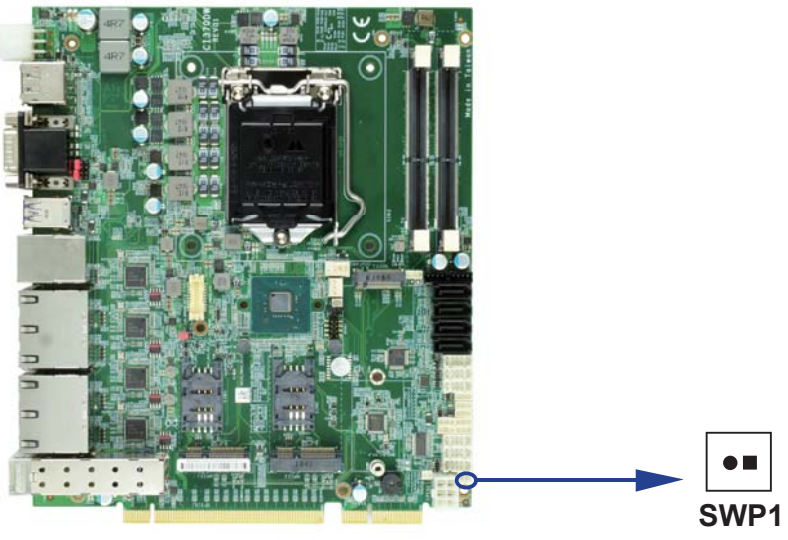


Note:

1. SODIM1/ SODIM2: SO-DIMM DDR4 1.2V DRAM Socket
2. Only Support un-buffer type module
3. Only suport Raw card type A.C.E.

3-17 SWP1 Power On/off switch Wafer (1x2 pin 2.00mm wafer)

PIN NO.	Description
1	Power button pin
2	Power button GND

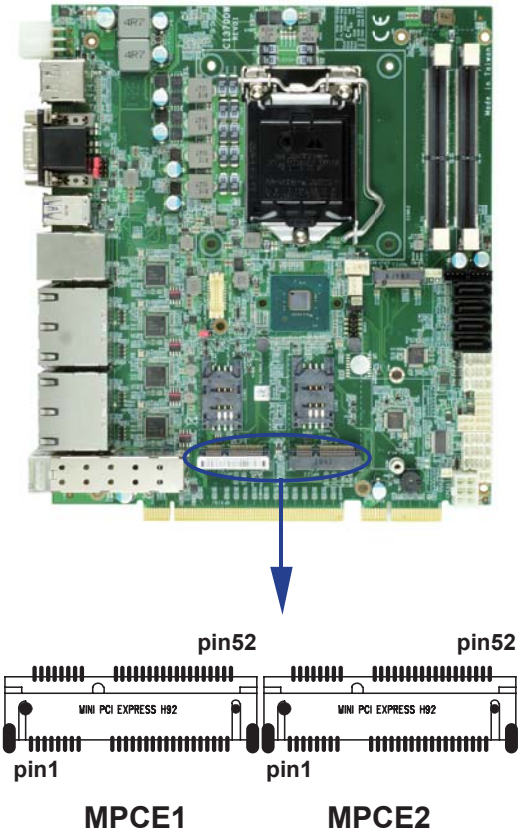


3-18 Module socket

● MPCE1/2 PCI Express Mini card

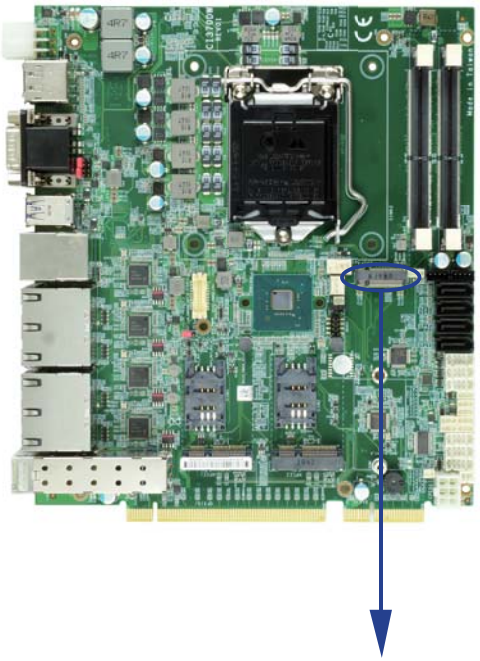
PIN NO.	Description	PIN NO.	Description
1	NC	2	+3.3V
3	NC	4	GND
5	NC	6	+1.5V
7	NC	8	NC
9	GND	10	NC
11	PCIe-CLK-	12	NC
13	PCIe-CLK+	14	NC
15	GND	16	NC
KEY			
17	NC	18	GND
19	NC	20	NC
21	GND	22	Reset
23	PCIe-RX-/mSATA-RX+	24	+3.3V
25	PCIe-RX+/mSATA-RX-	26	GND
27	GND	28	+1.5V
29	GND	30	SMB-CLK
31	PCIe-TX-/mSATA-TX-	32	SMB-DATA
33	PCIe-TX+/mSATA-TX+	34	GND
35	GND	36	USB-DATA-
37	GND	38	USB-DATA+
39	+3.3V	40	GND
41	+3.3V	42	NC
43	GND	44	NC
45	NC	46	NC
47	NC	48	+1.5V
49	NC	50	GND
51	mSATA/PCIe detect	52	+3.3V

Note: 1. Pin51 mSATA/PCIe auto detect function

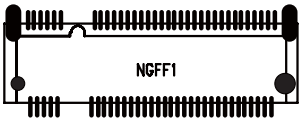


● **NGFF1: B Key size**

PIN NO.	DESCRIPTION	PIN NO.	DESCRIPTION
1	CONFIG_3	2	+3.3V
3	GND	4	+3.3V
5	GND	6	NC
7	USB-DATA-	8	NC
9	USB-DATA+	10	LED#
11	GND		
KEY			
21	CONFIG_0	22	NC
23	NC	24	NC
25	NC	26	NC
27	GND	28	NC
29	NC	30	NC
31	NC	32	NC
33	GND	34	NC
35	NC	36	NC
37	NC	38	NC
39	GND	40	NC
41	PCIe-RX-	42	NC
43	PCIe-RX+	44	NC
45	GND	46	NC
47	PCIe-TX-	48	NC
49	PCIe-TX+	50	RESET#
51	GND	52	NC
53	PCIe-CLK-	54	NC
55	PCIe-CLK+	56	NC
57	GND	58	NC
59	NC	60	NC
61	NC	62	NC
63	NC	64	NC
65	NC	66	NC
67	NC	68	NC
69	CONFIG_1	70	+3.3V
71	GND	72	+3.3V
73	GND	74	+3.3V
75	CONFIG_2		



NGFF1



Note:
1. Only PCIe interface supported.
2. supported PCIe NVMe storage.

3-19 Connector wafer of Compatible Brand and part number list

Location	CKTS	PITCH	Brand Name	Mating connector	Cable housing
CA1	2x5 10Pin	2.00mm	JST	B10B-PHDSS	PHDR-10VS
CFP1	2x5 10Pin	2.00mm	JST	B10B-PHDSS	PHDR-10VS
CIO1	2x5 10Pin	2.00mm	JST	B10B-PHDSS	PHDR-10VS
CIO2	2x5 10Pin	2.00mm	JST	B10B-PHDSS	PHDR-10VS
CU8	1x4 4Pin	1.25mm	MOLEX	53047-0410	51021-0400
CU9	1x4 4Pin	1.25mm	MOLEX	53047-0410	51021-0400
CU10	1x4 4Pin	1.25mm	MOLEX	53047-0410	51021-0400
CU11	1x4 4Pin	1.25mm	MOLEX	53047-0410	51021-0400
CU12	1x4 4Pin	1.25mm	MOLEX	53047-0410	51021-0400
CU13	1x4 4Pin	1.25mm	MOLEX	53047-0410	51021-0400
CU14	1x4 4Pin	1.25mm	MOLEX	53047-0410	51021-0400
CO1	1x4 4Pin	1.25mm	MOLEX	53047-0410	51021-0400
CLPC1	2x5 10Pin	2.00mm	JST	B10B-PHDSS	PHDR-10VS
CIO2	2x5 10Pin	2.00mm	JST	B10B-PHDSS	PHDR-10VS
CPO1	1x4 4Pin	2.00mm	JST	B4B-PH-KL	PHR-4
CPO2	1x4 4Pin	2.00mm	JST	B4B-PH-KL	PHR-4
CBT1	1x2 2Pin	1.25mm	MOLEX	53047-0210	51021-0200
CPS1	1x2 2Pin	1.25mm	MOLEX	53047-0210	51021-0200
CSPI1	2x5 10Pin	2.00mm	JST	B10B-PHDSS	PHDR-10VS
eDP1	2x10 20Pin	1.25mm	HIROSE	DF13-20DS-1.25C	DF13-20DP-1.25V
SWP1	1x2 2Pin	2.00mm	JST	B2B-PH-KL	PHR-2

Chapter-4

Introduction of BIOS

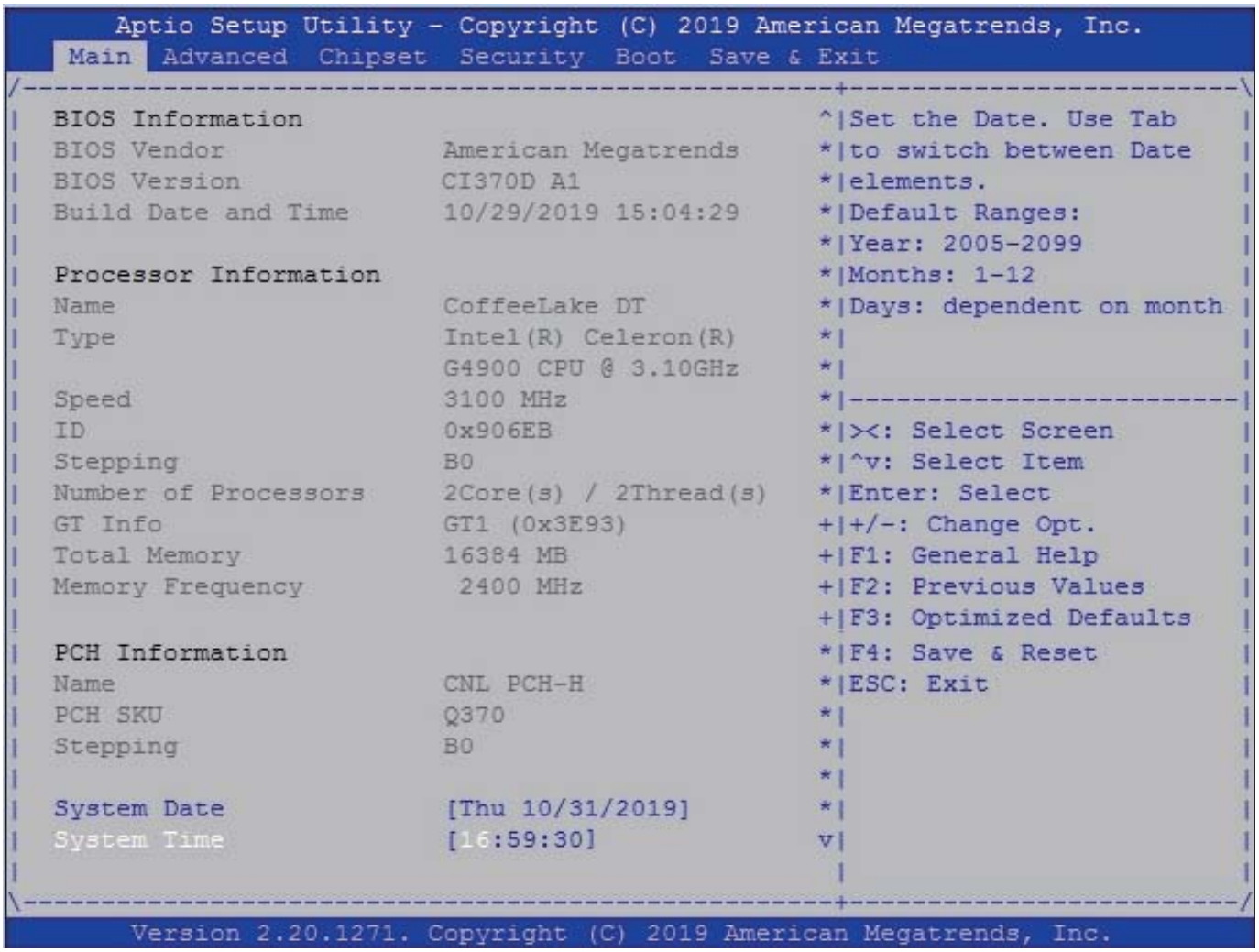
The BIOS is a program located in the Flash Memory on the motherboard.
This program is a bridge between motherboard and operating system.
When you start the computer, the BIOS program gains control.
The BIOS first operates an auto-diagnostic test called POST (Power on Self Test) for all the necessary hardware, it detects the entire hardware devices and configures the parameters of the hardware synchronization. After these tasks are completed, BIOS will give control of the computer back to operating system (OS). Since the BIOS is the only channel for hardware and software to communicate with, it is the key factor of system stability and of ensuring your system performance at best.
In the BIOS Setup main menu, you can see several options. We will explain these options in the following pages. First, let us see the function keys you may use here:

Press <Esc> to quit the BIOS Setup.
Press ↑↓←→(up, down, left, right) to choose the option you want to confirm or modify.
Press <F10> to save these parameters and to exit the BIOS Setup menu after you complete the setup of BIOS parameters.
Press Page Up/Page Down or +/- keys to modify the BIOS parameters for the active option.

4-1 Enter Setup

Power on the computer and press <Del> key immediately to enter Setup.
If the message disappears before your respond but you still wish to enter Setup, restart the system by turning it OFF then ON or pressing the "RESET" button on the system case. You may also restart the system by simultaneously pressing <Ctrl>, <Alt> and <Delete> keys.

4-2 BIOS Menu Screen & Function Keys



In the above BIOS Setup main menu of, you can see several options.
We will explain these options step by step in the following pages of this chapter, but let us first see a short description of the function keys you may use here:

- Press >< (right, left) to select screen;
- Press ↑↓ (up, down) to choose, in the main menu, the option you want to confirm or to modify.
- Press <Enter> to select.
- Press <+>/<–> or <F5>/<F6> keys when you want to modify the BIOS parameters for the active option.
- [F1]: General help.
- [F2]: Previous values.
- [F3]: Optimized defaults.
- [F4]: Save & Exit.
- Press <Esc> to quit the BIOS Setup.

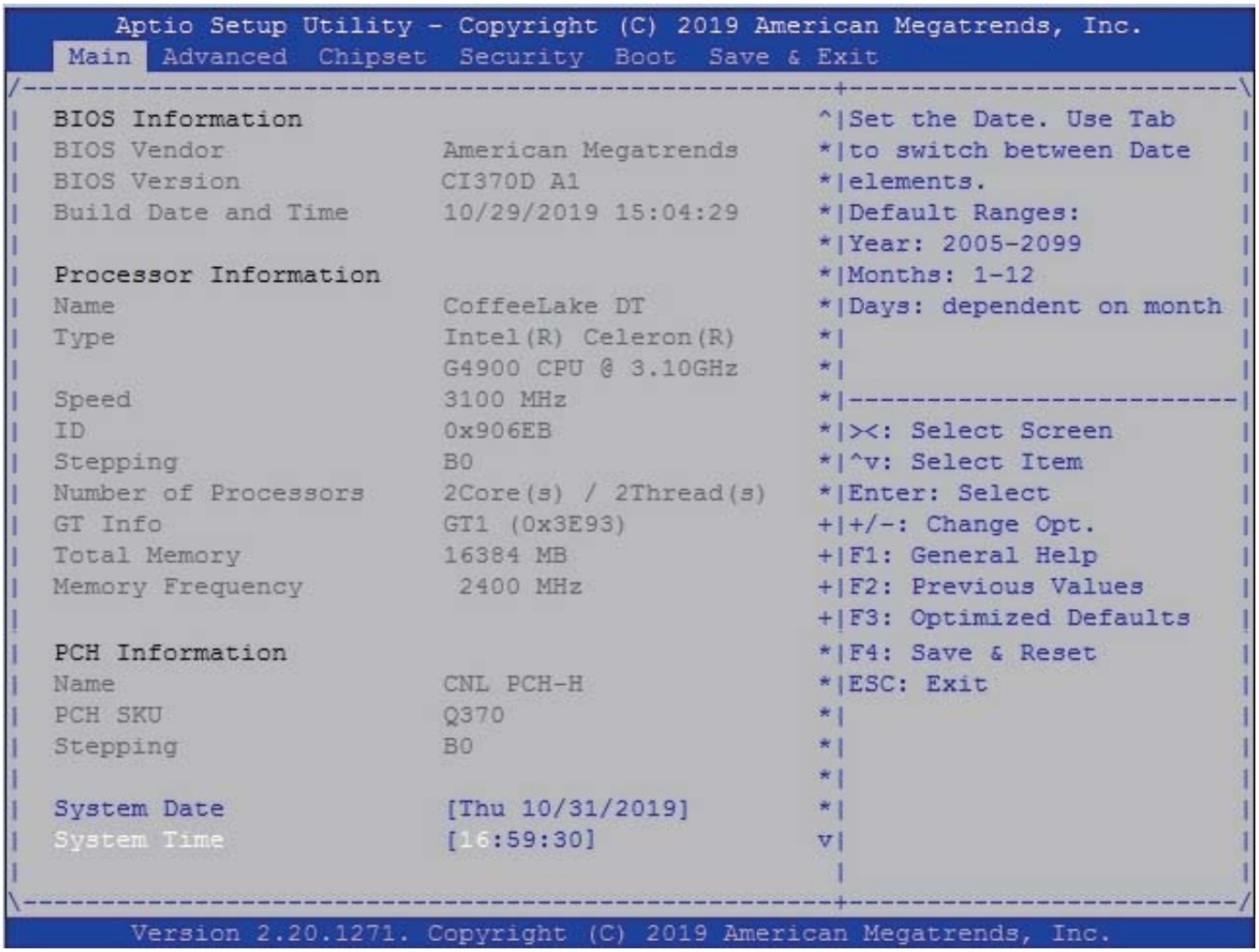
4-3 Getting Help

Status Page Setup Menu / Option Page Setup Menu
Press F1 to pop up a help window that describes the appropriate keys to use and the possible selections for the highlighted item. To exit the Help Window, press <Esc>.

4-4 Menu Bars

There are six menu bars on top of BIOS screen:
Main To change system basic configuration
Advanced To change system advanced configuration
Chipset To change PCH IO configuration
Security Password settings
Boot Quiet boot or boot from USB selected.
Save & Exit Save setting, loading and exit options.
User can press the right or left arrow key on the keyboard to switch from menu bar.
The selected one is highlighted.

4-5 Main



Main menu screen includes some basic system information. Highlight the item and then use the <+> or <-> and numerical keyboard keys to select the value you want in each item.

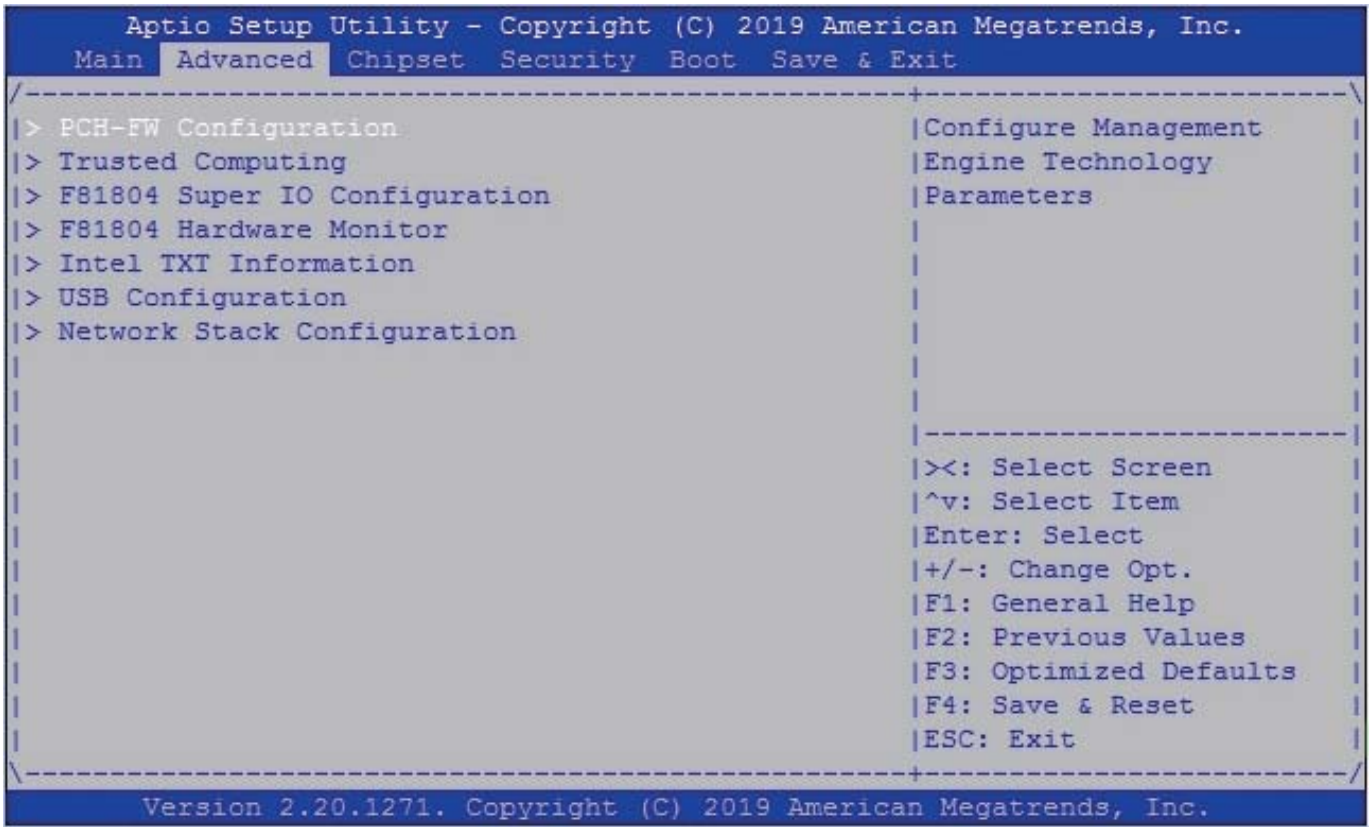
System Date

Set the Date. Please use [Tab] to switch between data elements.

System Time

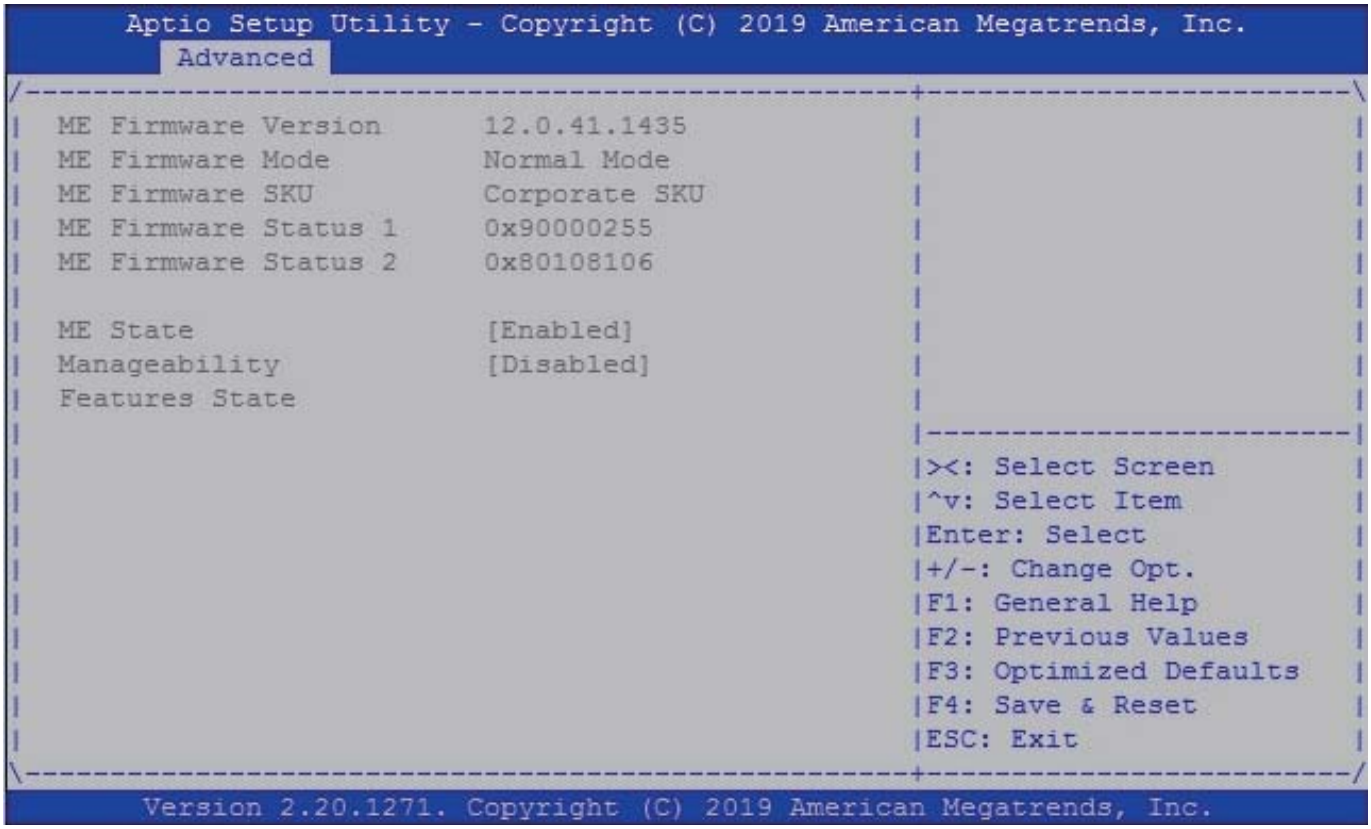
Set the Time. Please use [Tab] to switch between data elements.

4-6 Advanced

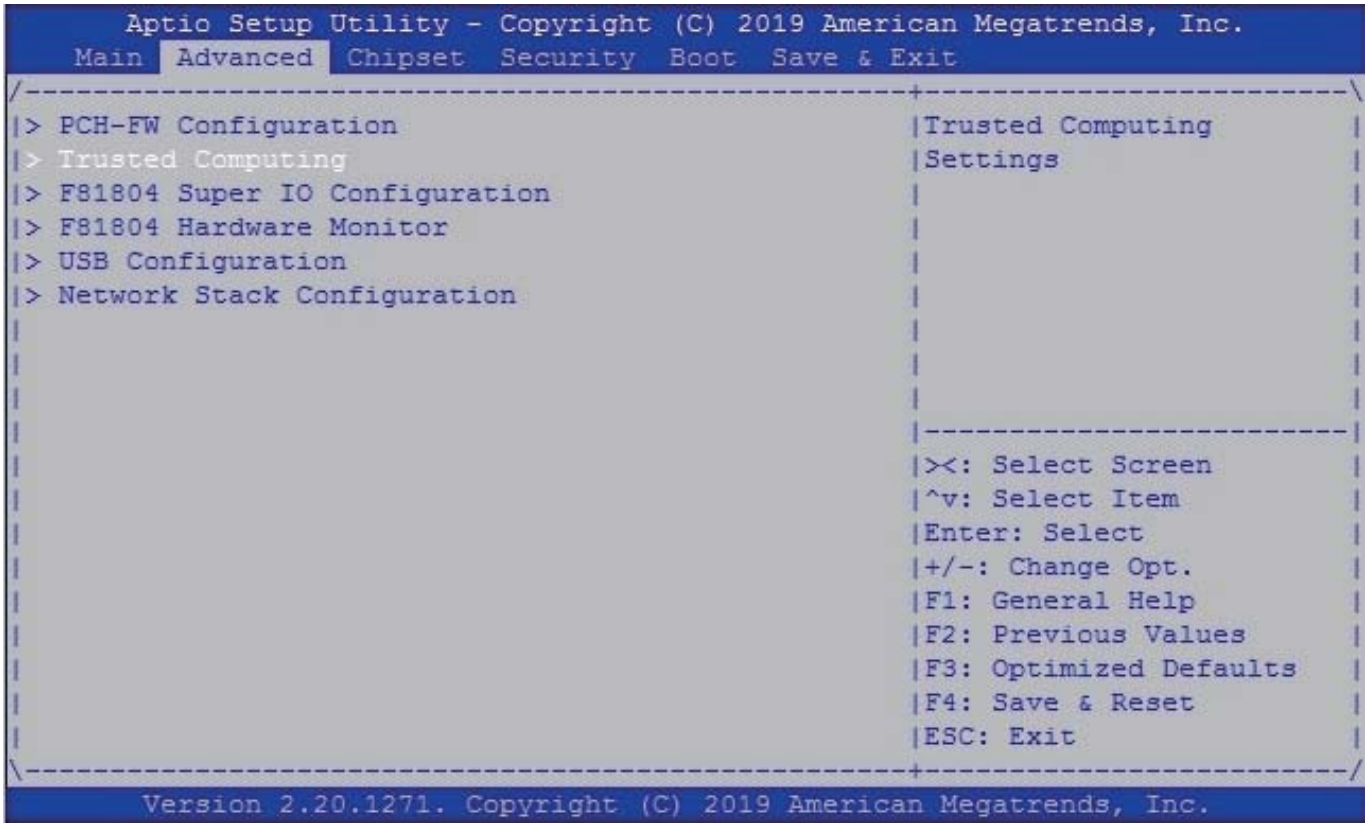


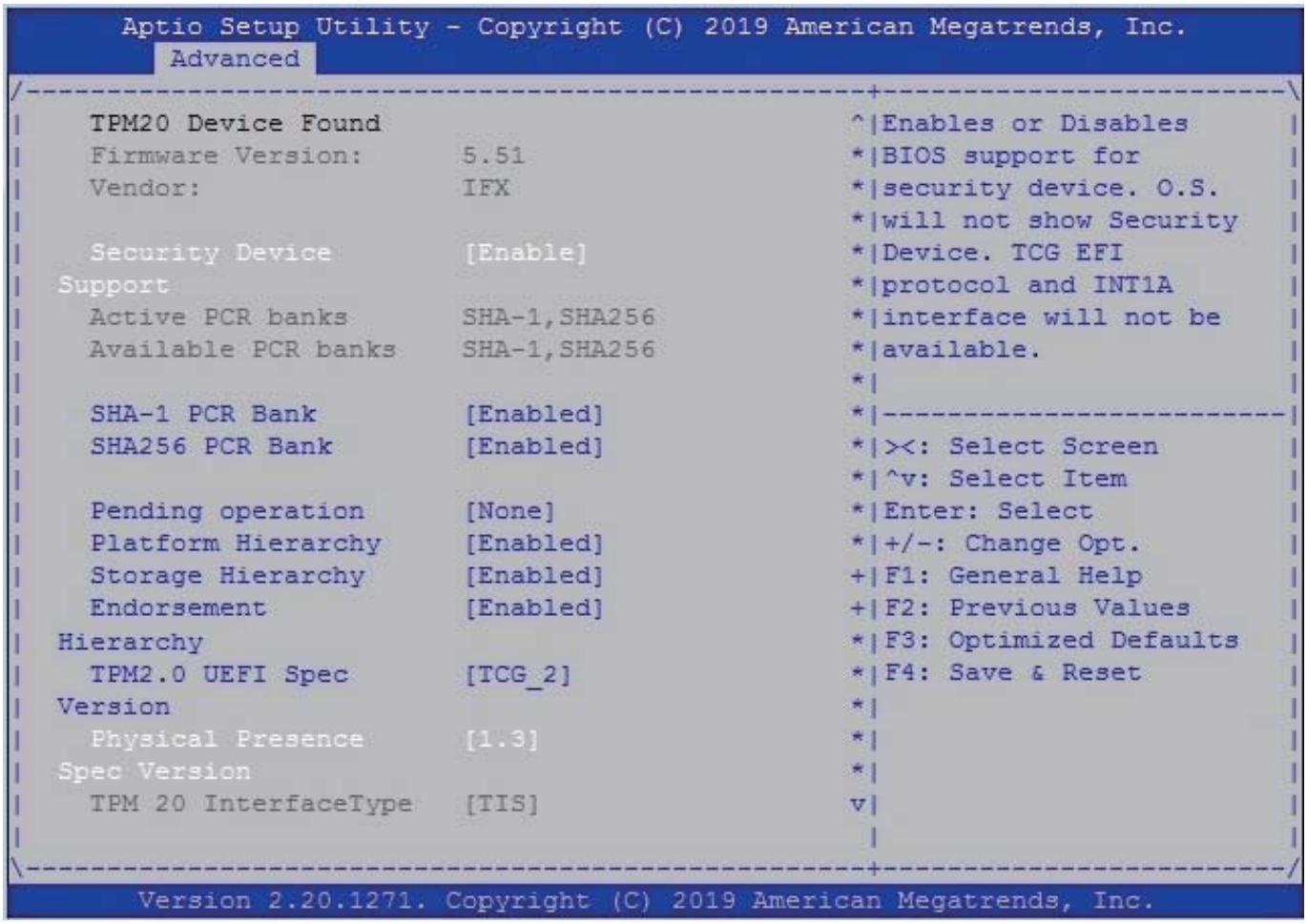
PCH-FW Configuration
Please refer section 4-6-1
Trusted Computing
Please refer section 4-6-2
F81804 Super IO Configuration
Please refer section 4-6-3
F81804 Hardware Monitor
Please refer section 4-6-4
Intel TXT Information
Please refer section 4-6-5
USB Configuration
Please refer section 4-6-6
Network Stack Configuration
Please refer section 4-6-7

4-6-1 PCH-FW Configuration



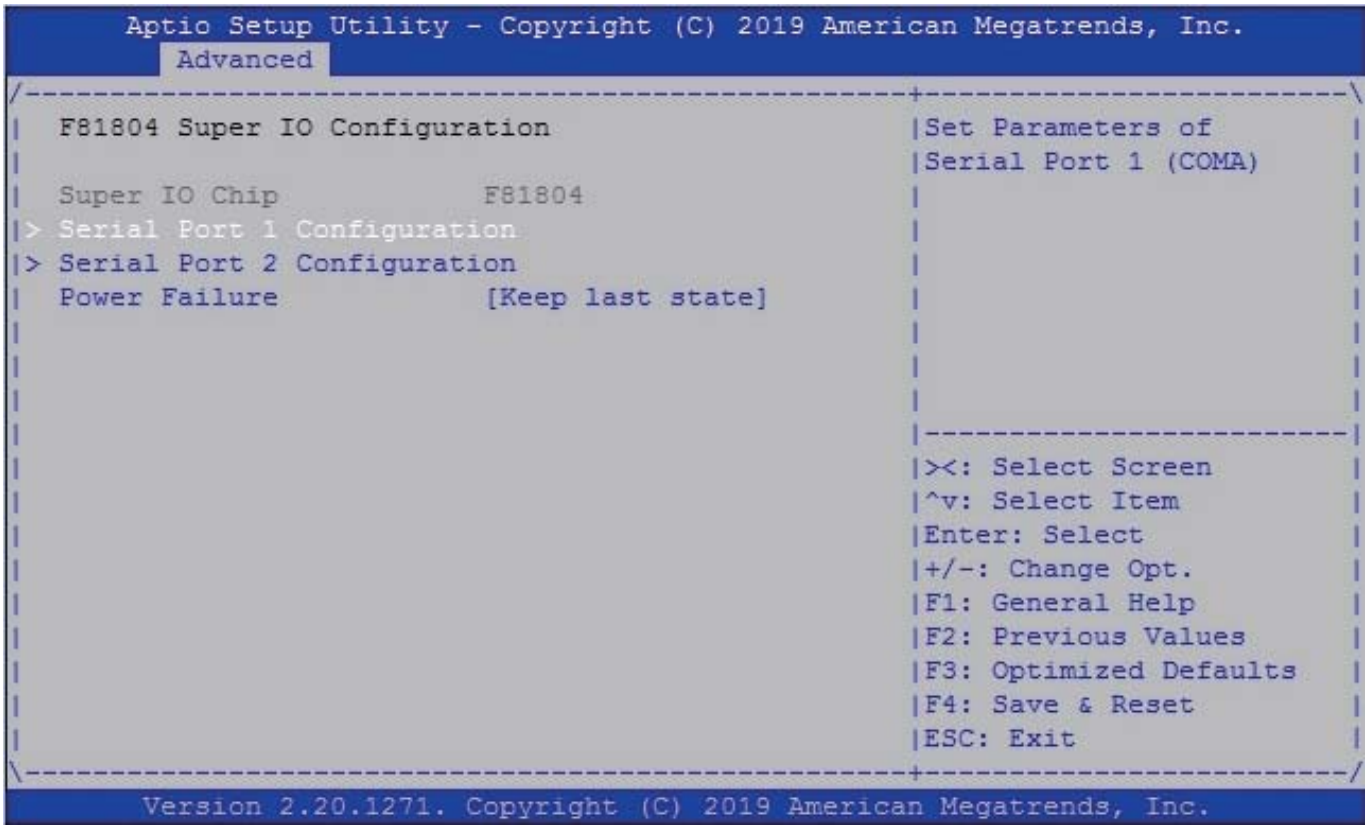
4-6-2 Trusted Computing





This page for TPM Device only. (For Option Model)

4-6-3 F81804 Super IO Configuration



Serial Port 1 Configuration

Please refer section 4-6-3-1

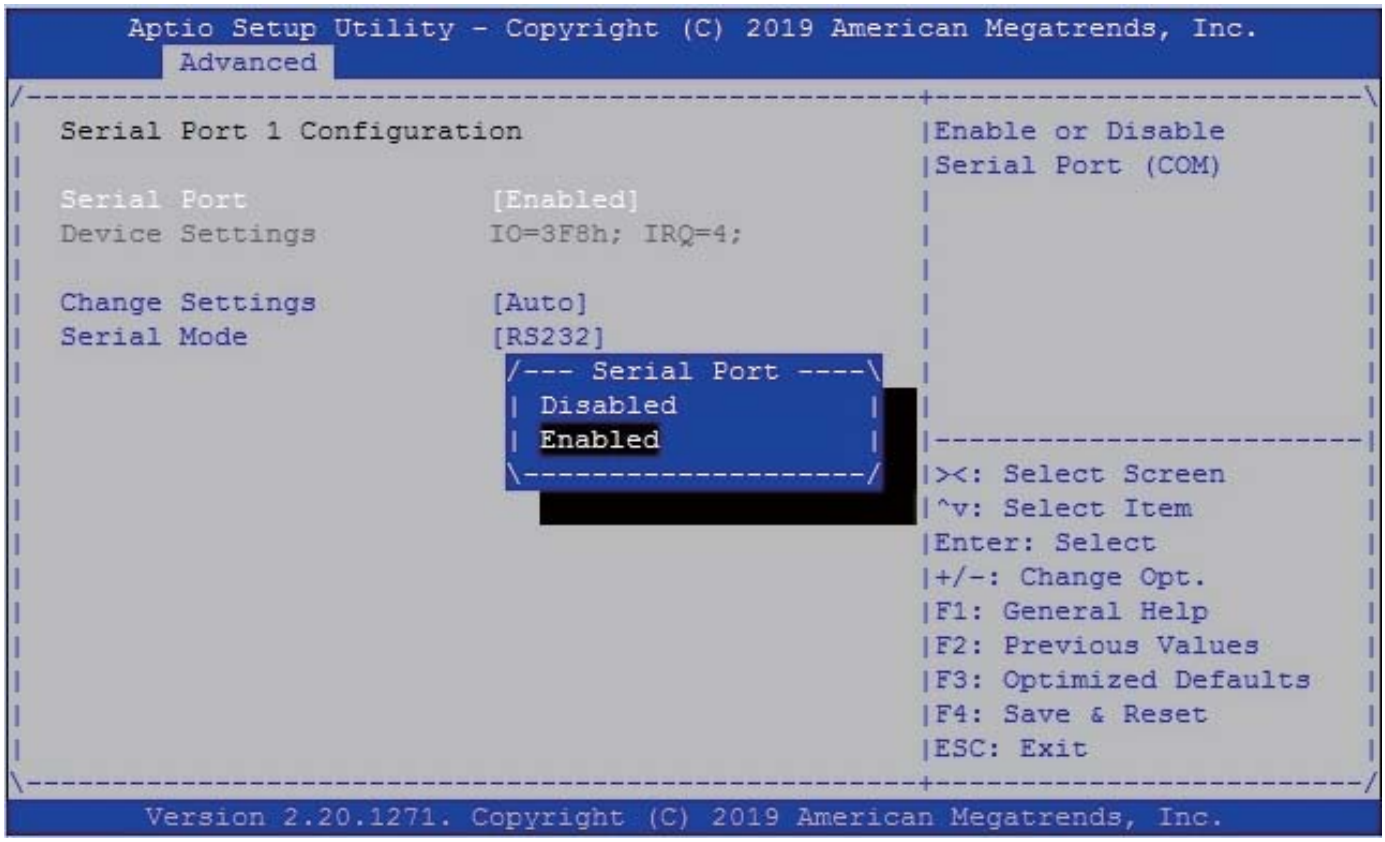
Serial Port 2 Configuration

Please refer section 4-6-3-2

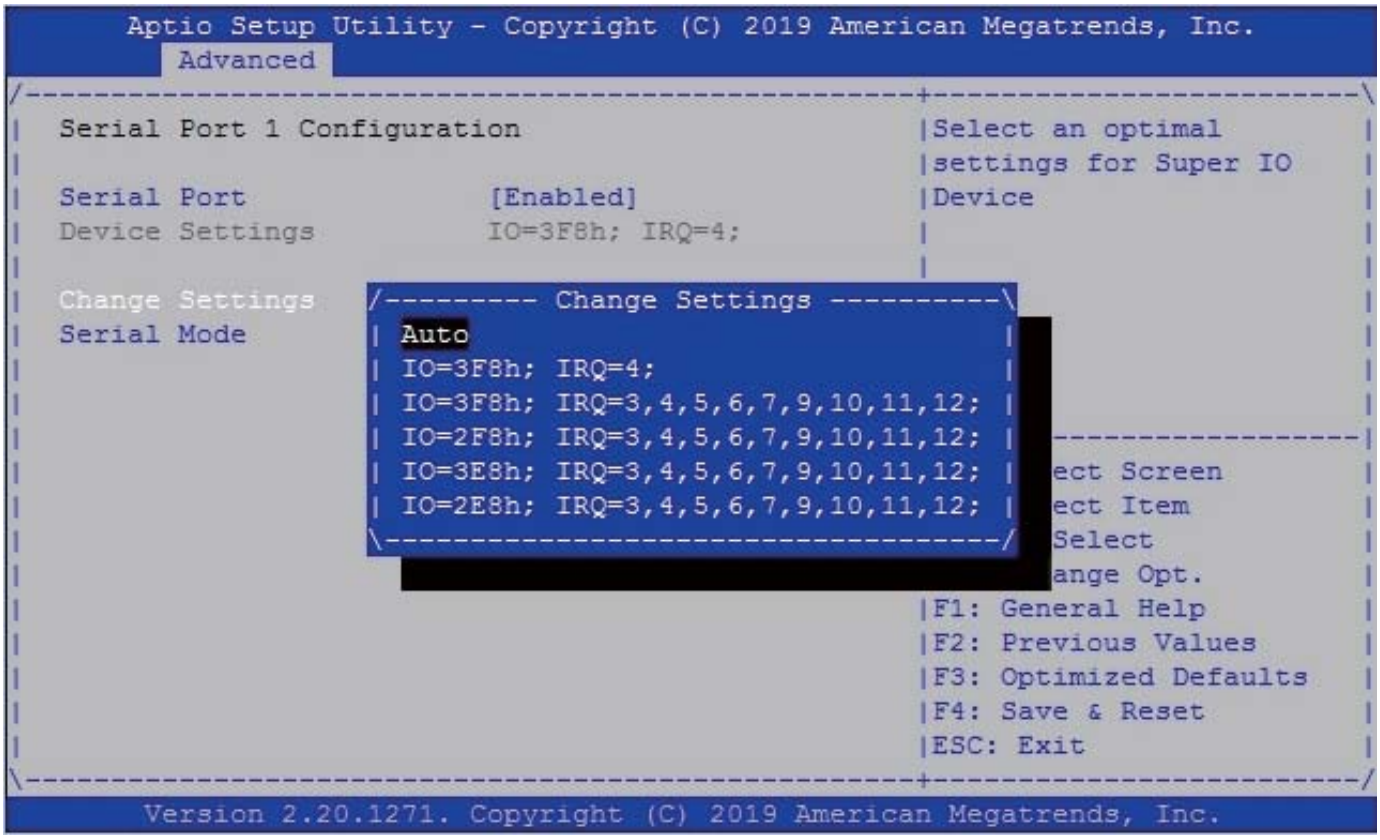
Power Failure

Please refer section 4-6-3-3

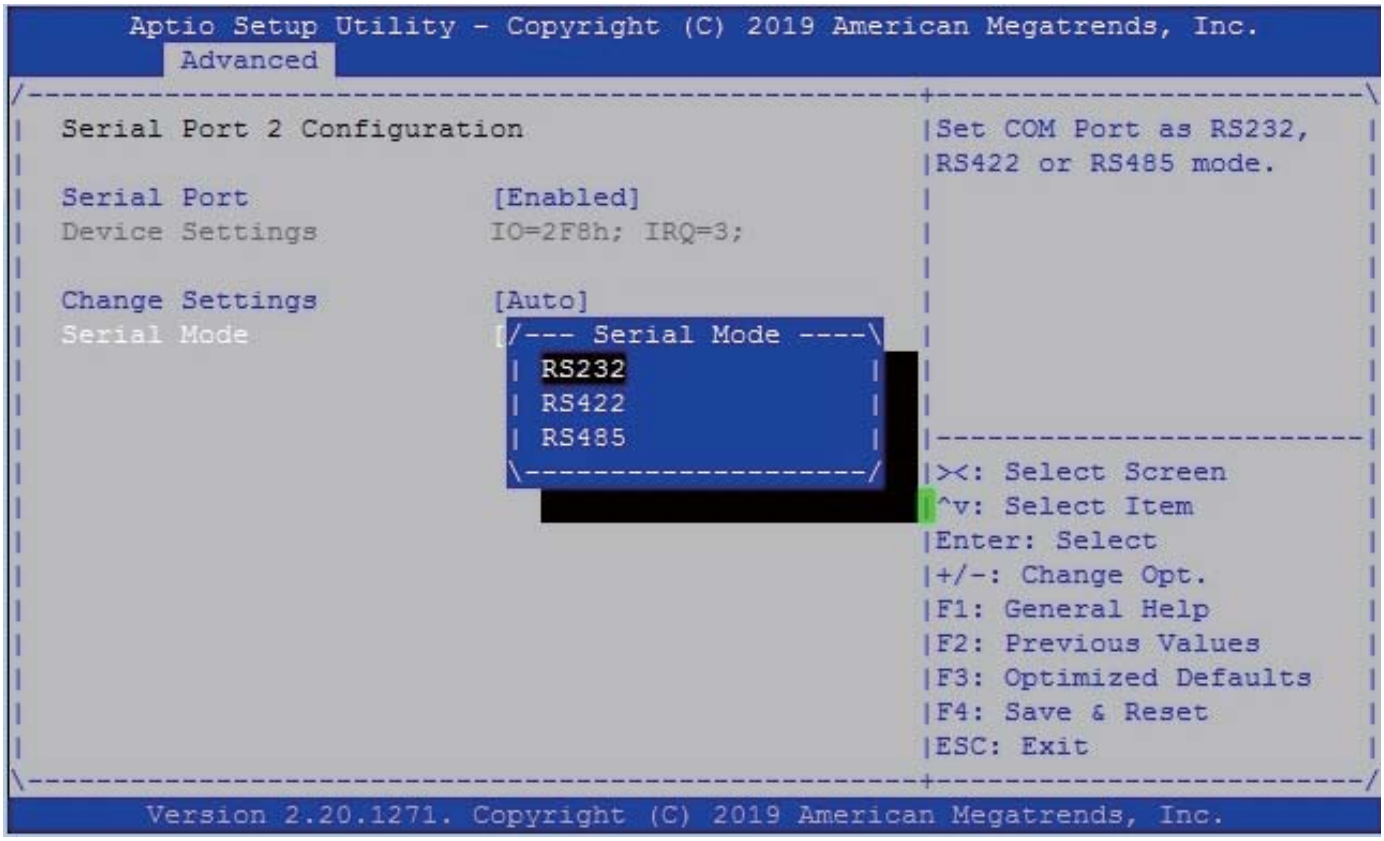
4-6-3-1 ► Serial Port 1 Configuration



To Enable Serial port or not, default is Enabled.

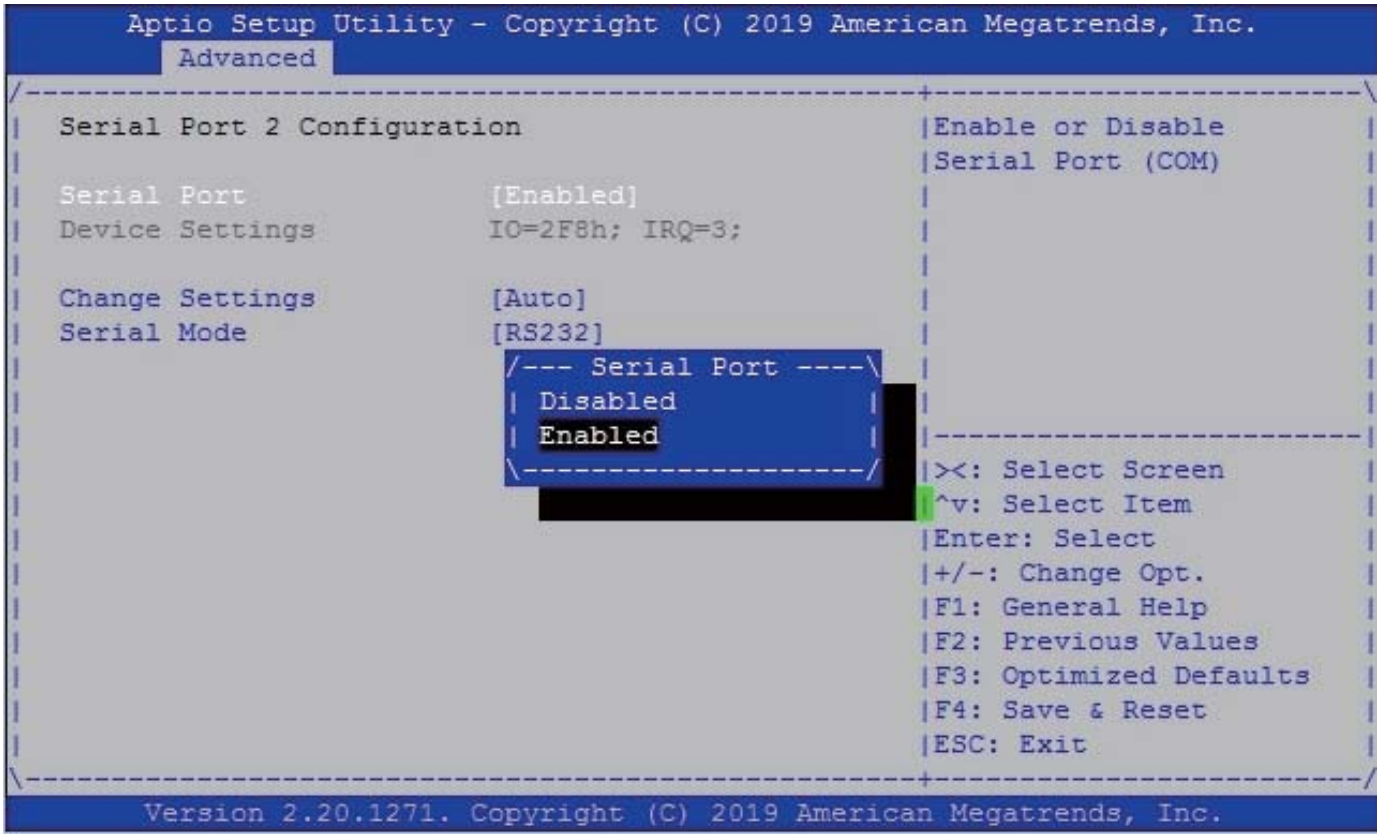


Change Settings, default is Auto.

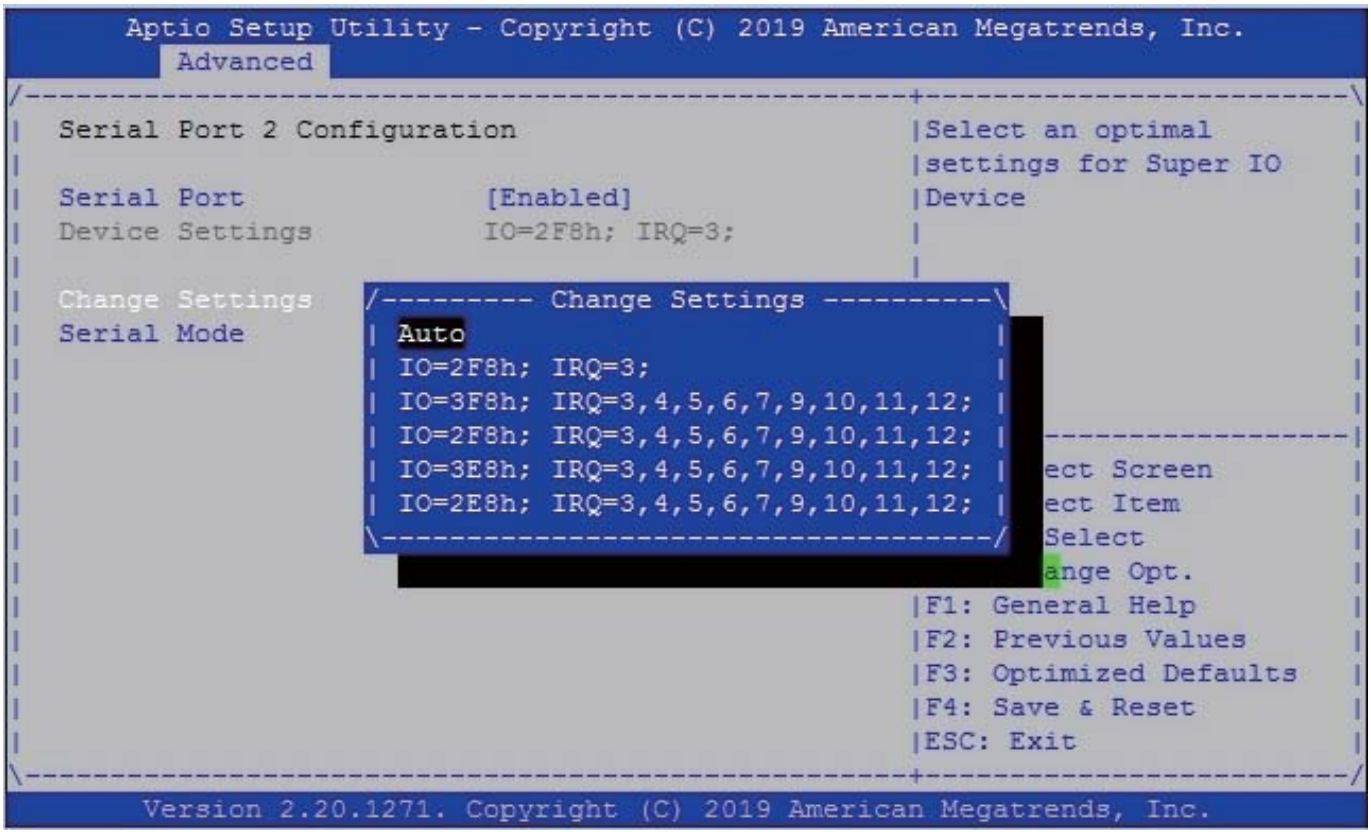


To select the Serial port to RS232 / RS422 / RS485, default is RS232.

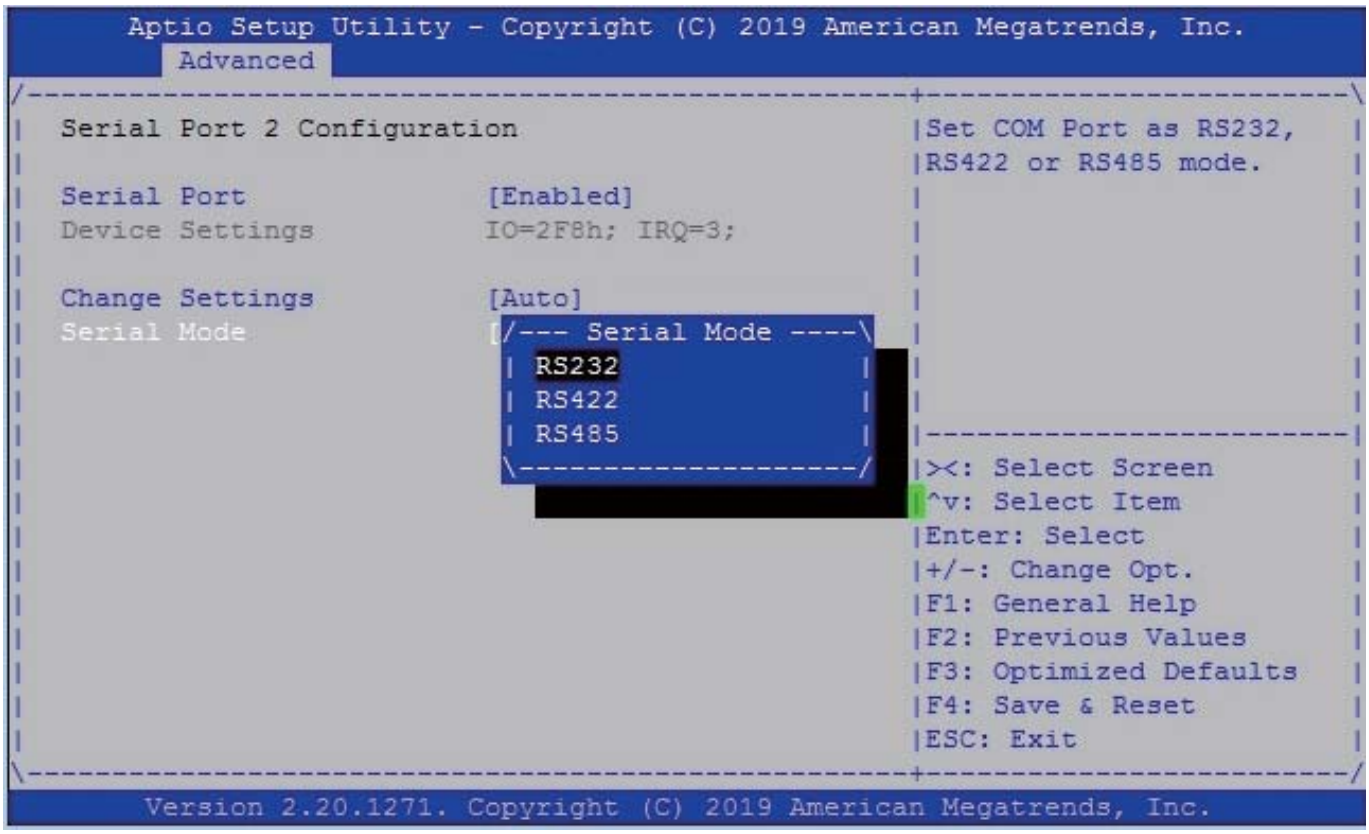
4-6-3-2 ► Serial Port 2 Configuration



To Enable Serial port or not, default is Enabled.

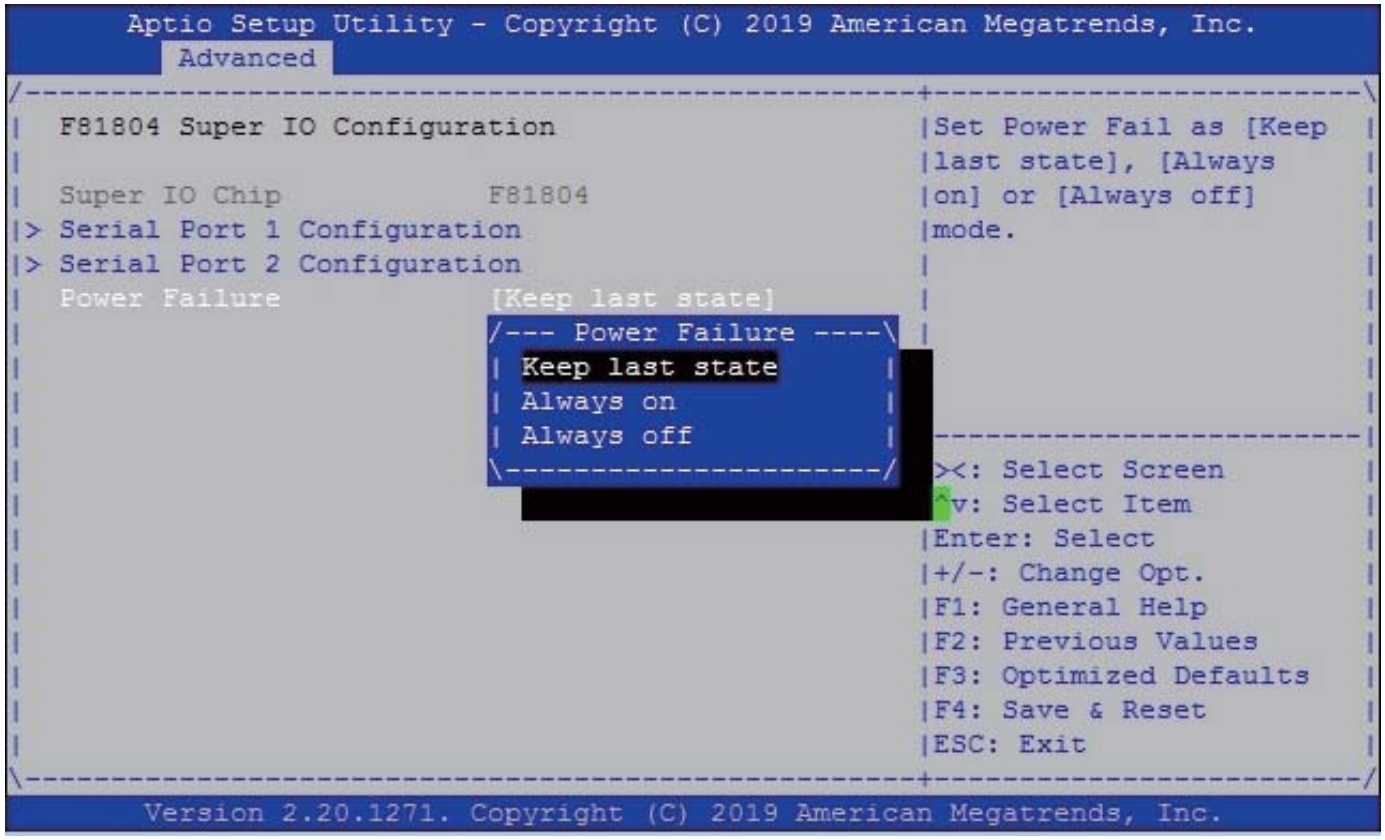


Change Settings, default is Auto.



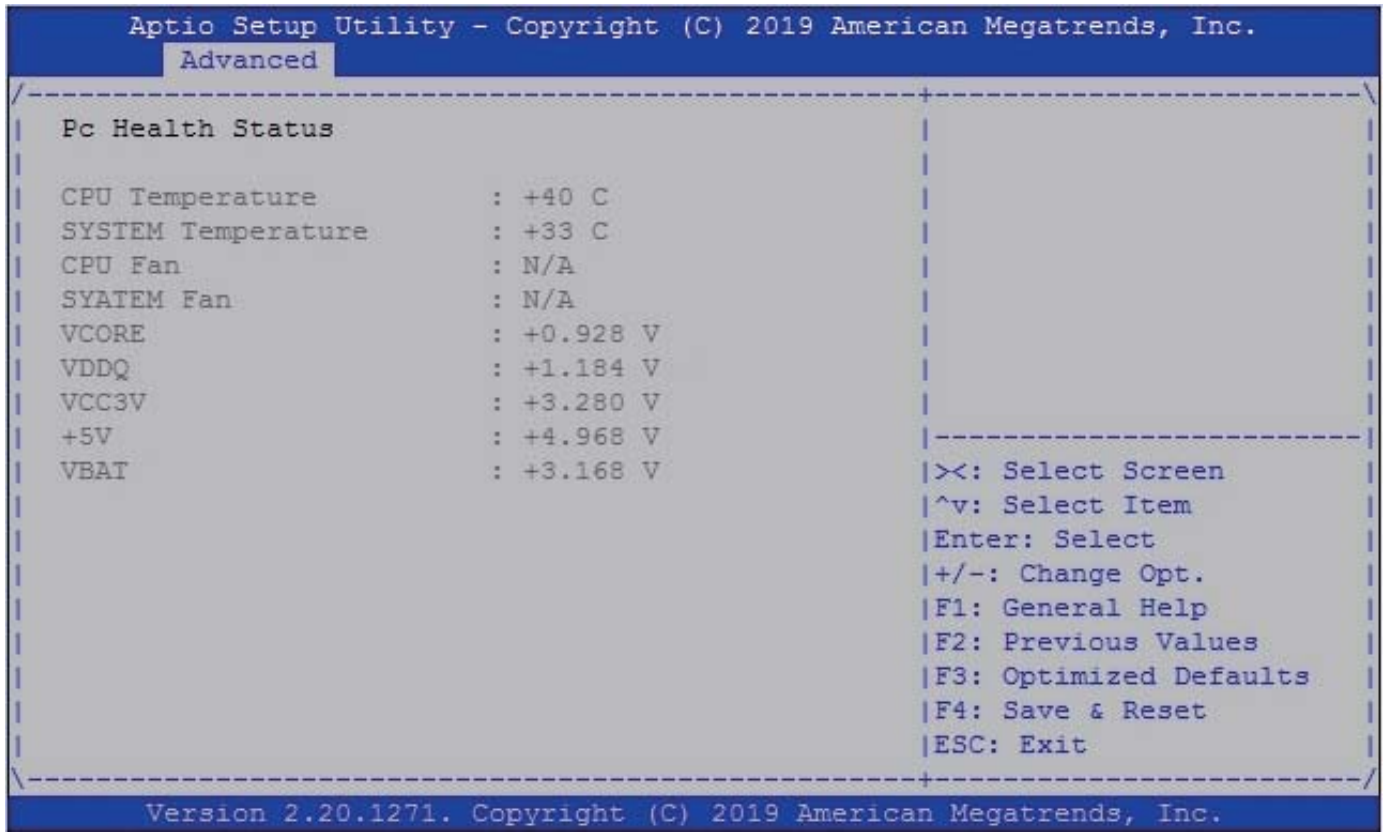
To select the Serial port to RS232 / RS422 / RS485, default is RS232.

4-6-3-3 ► Power Failure



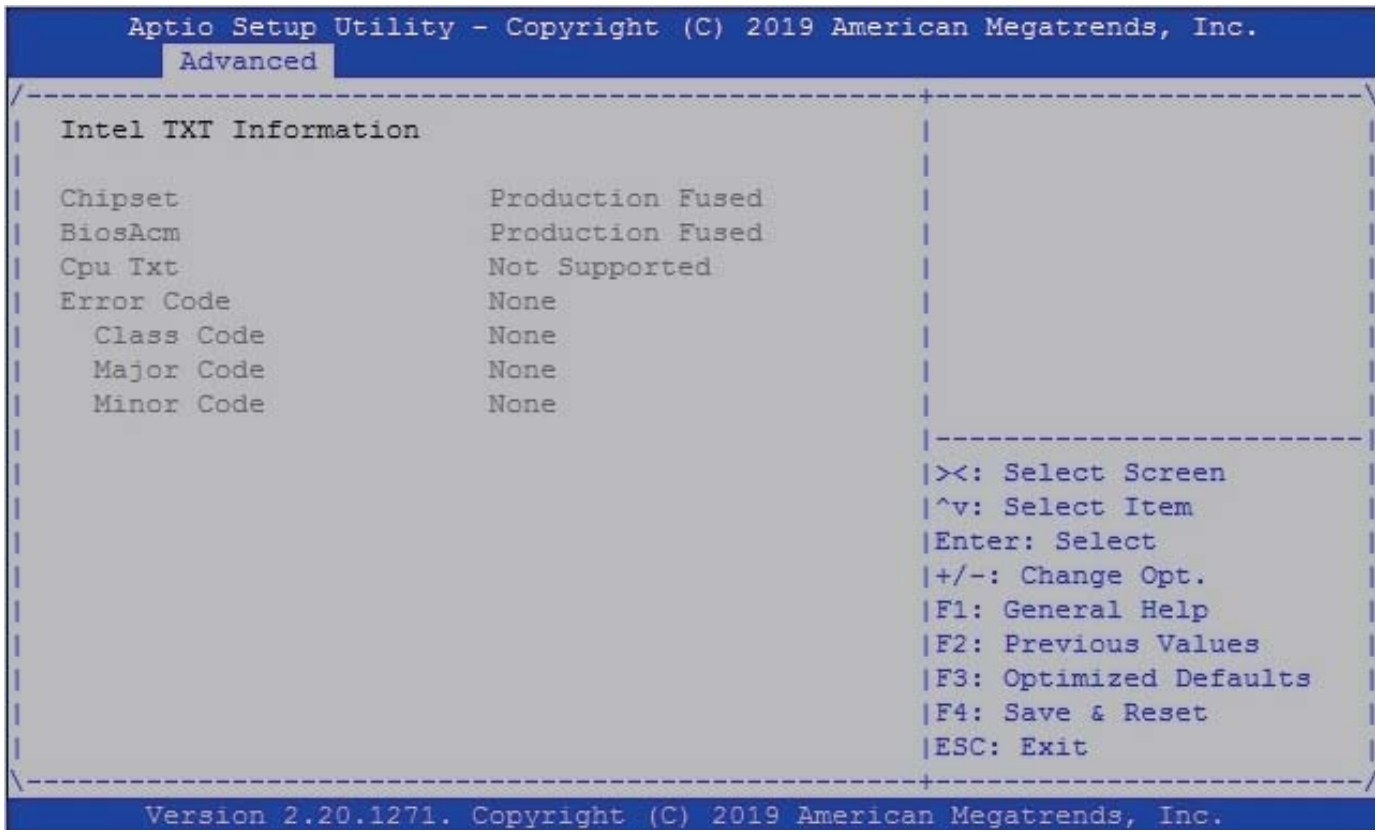
To select the power behavior after power fail, default is Keep last state.

4-6-4 F81804 Hardware Monitor



Press [Enter] to view PC health status.
This section shows the status of your CPU, Fan, and overall system.
This is only available when there is Hardware Monitor function onboard.

4-6-5 Intel TXT Information



Press [Enter] to view Intel TXT Information.

4-6-6 USB Configuration

Legacy USB Support

Enables Legacy USB support. AUTO option disables legacy support if no USB devices are connected. Disable option will keep USB devices available only for EFI applications.

The optional settings are: Enabled (default), Disabled, Auto.

XHCI Hand-off

This is a workaround for OS without XHCI handoff support. The XHCI ownership change should be

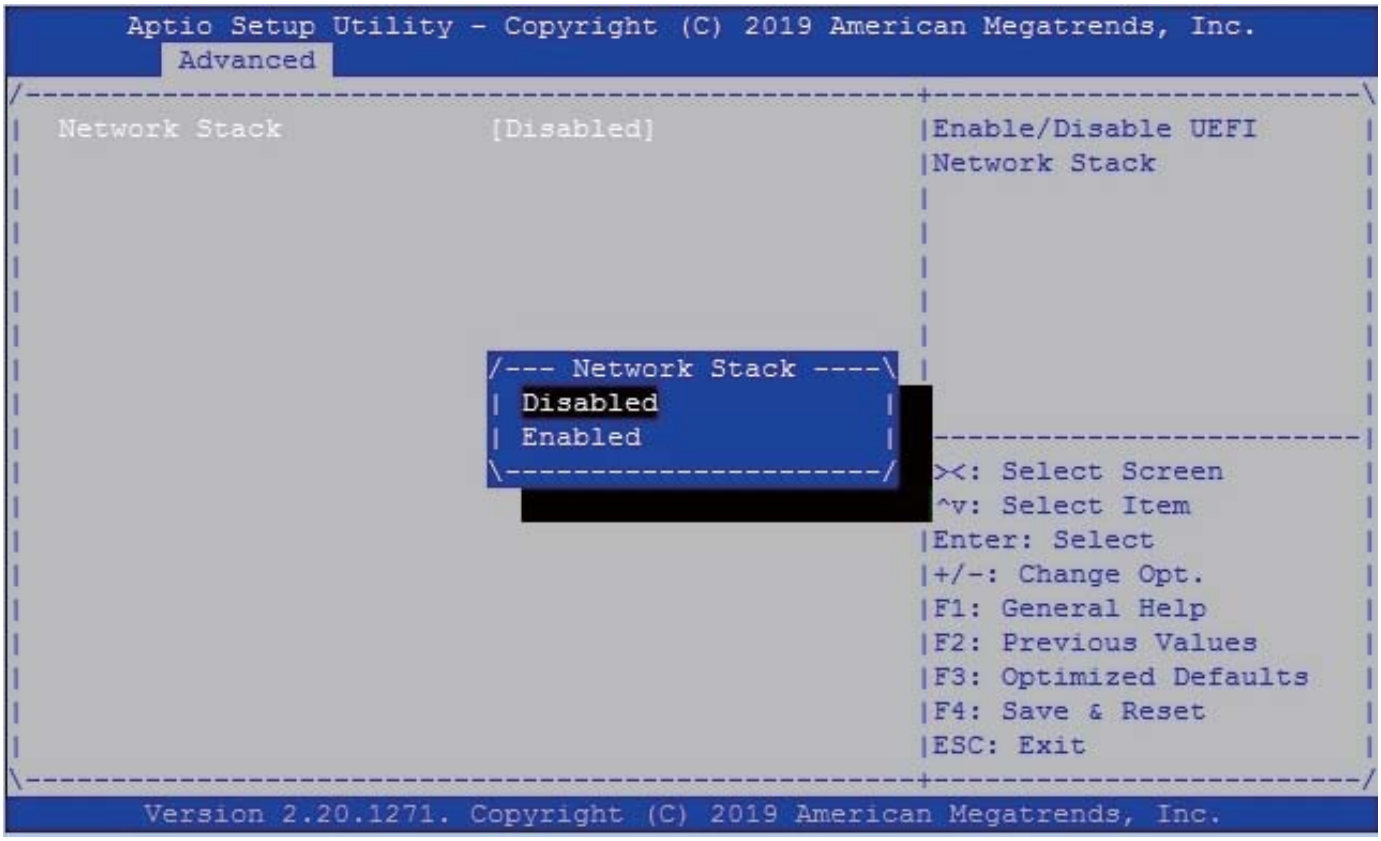
claimed by XHCI driver.

The optional settings are: Enabled, Disabled.

USB Mass Storage Driver Support

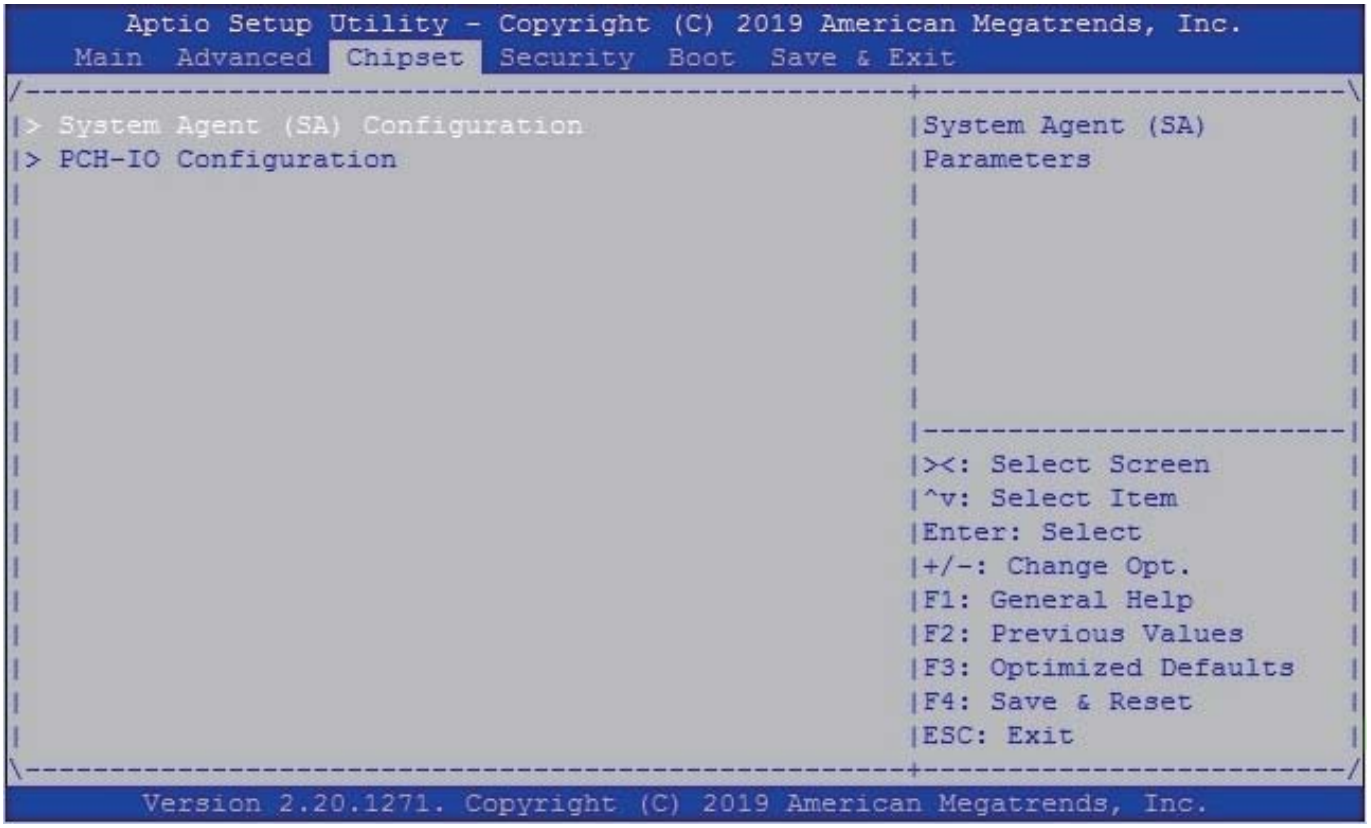
To enable USB mass storage support or not, default is Enabled.

4-6-7 Network Stack Configuration



To enable the UEFI Network stack or not, default is Disabled.

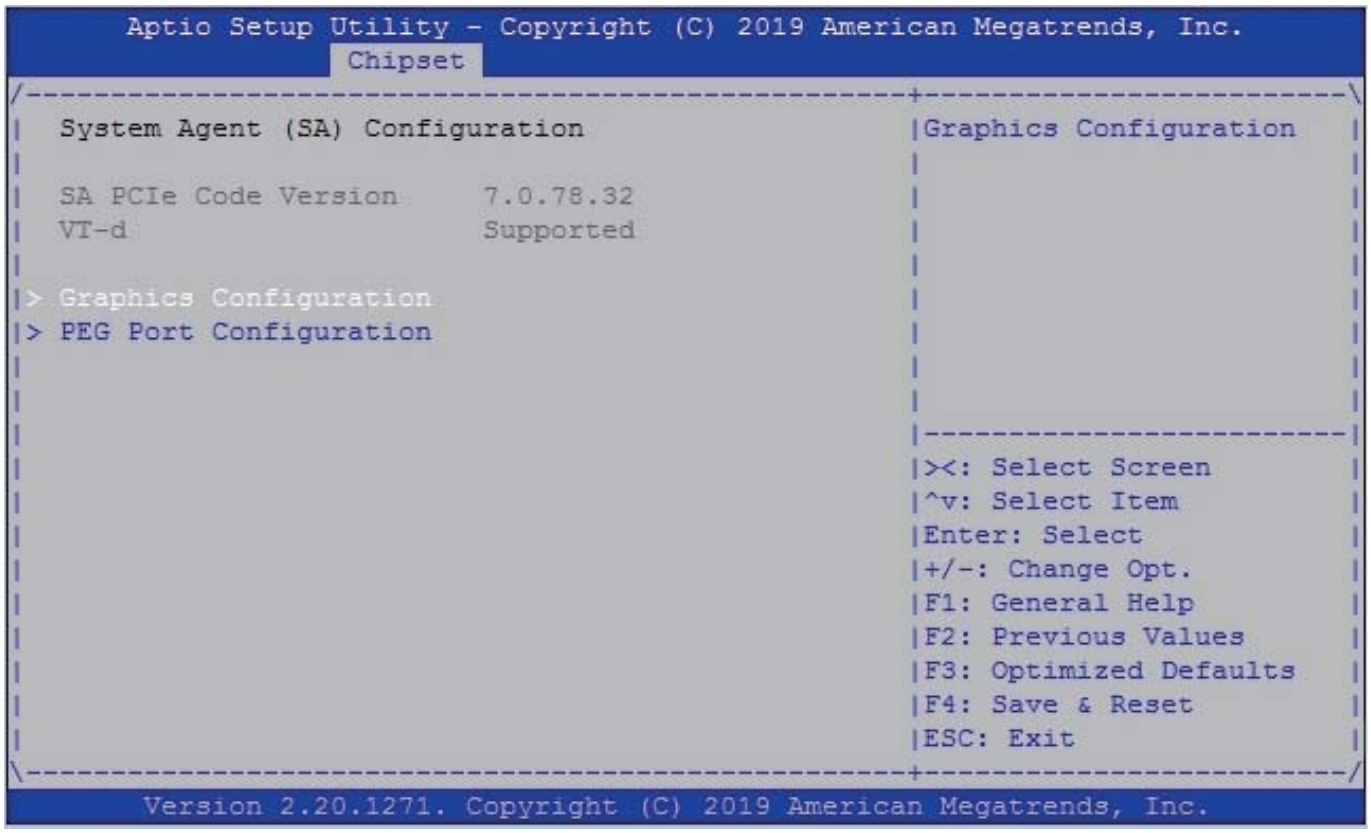
4-7 Chipset



System Agent (SA) Configuration.
Please refer section 4-7-1

PCH-IO Configuration.
Please refer section 4-7-2

4-7-1 System Agent (SA) Configuration



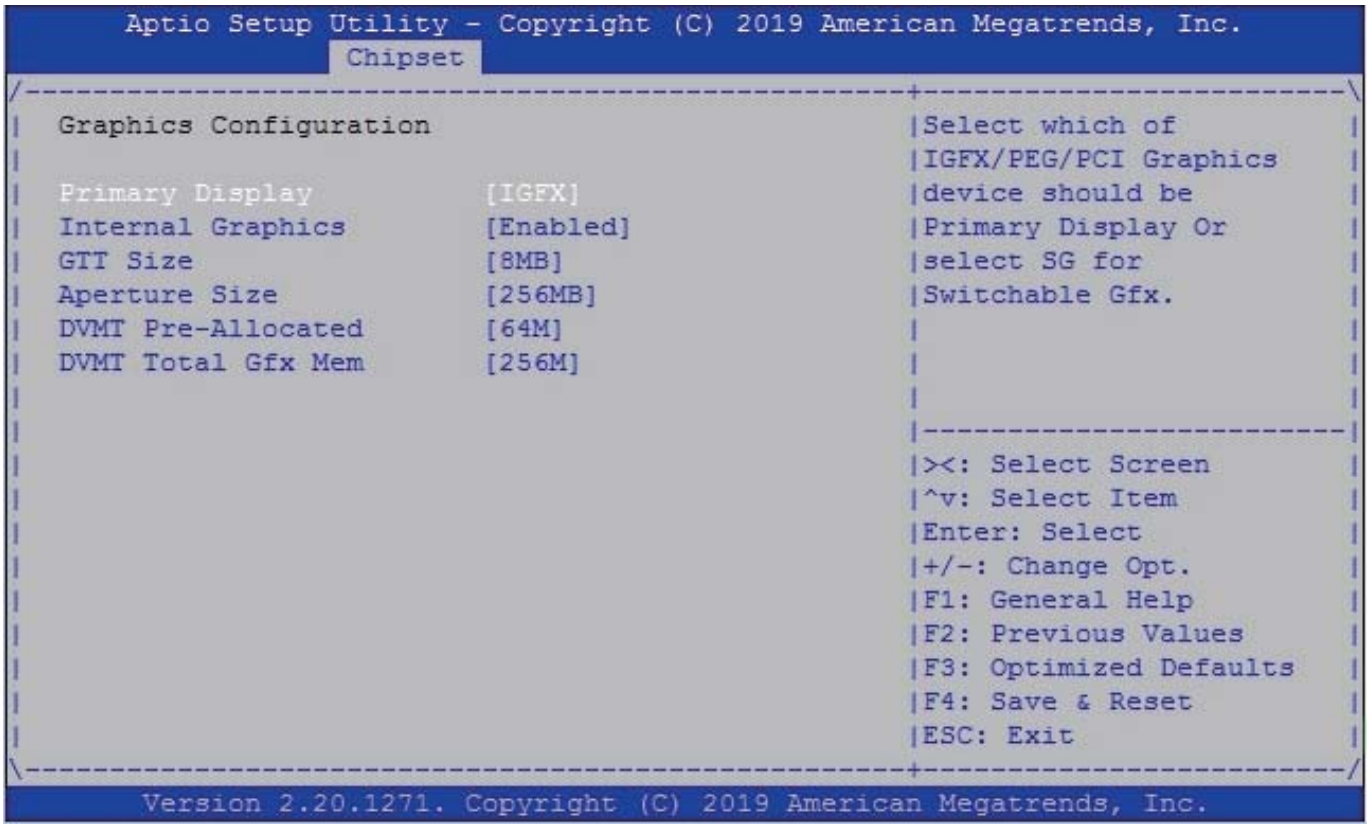
Graphics Configuration.

Please refer section 4-7-1-1

PEG Port Configuration.

Please refer section 4-7-1-2

4-7-1-1 ► Graphics Configuration



Primary Display

To select which of IGFX/PEG Graphics device should be Primary Display Or select SG for Switchable Gfx. The optional settings are: Auto, IGFX (default), PEG, PCI.

Internal Graphics

Keep IGFX enabled based on the setup options. The optional settings are: Auto, Enabled (default), Disabled.

GTT Size

Graphics Translation Table Size. The optional settings are: 2MB, 4MB, 8MB (default)

Aperture Size

The optional settings are: 128MB, 256MB (default), 512MB, 1024MB, 2048MB

DVMT Pre-Allocated

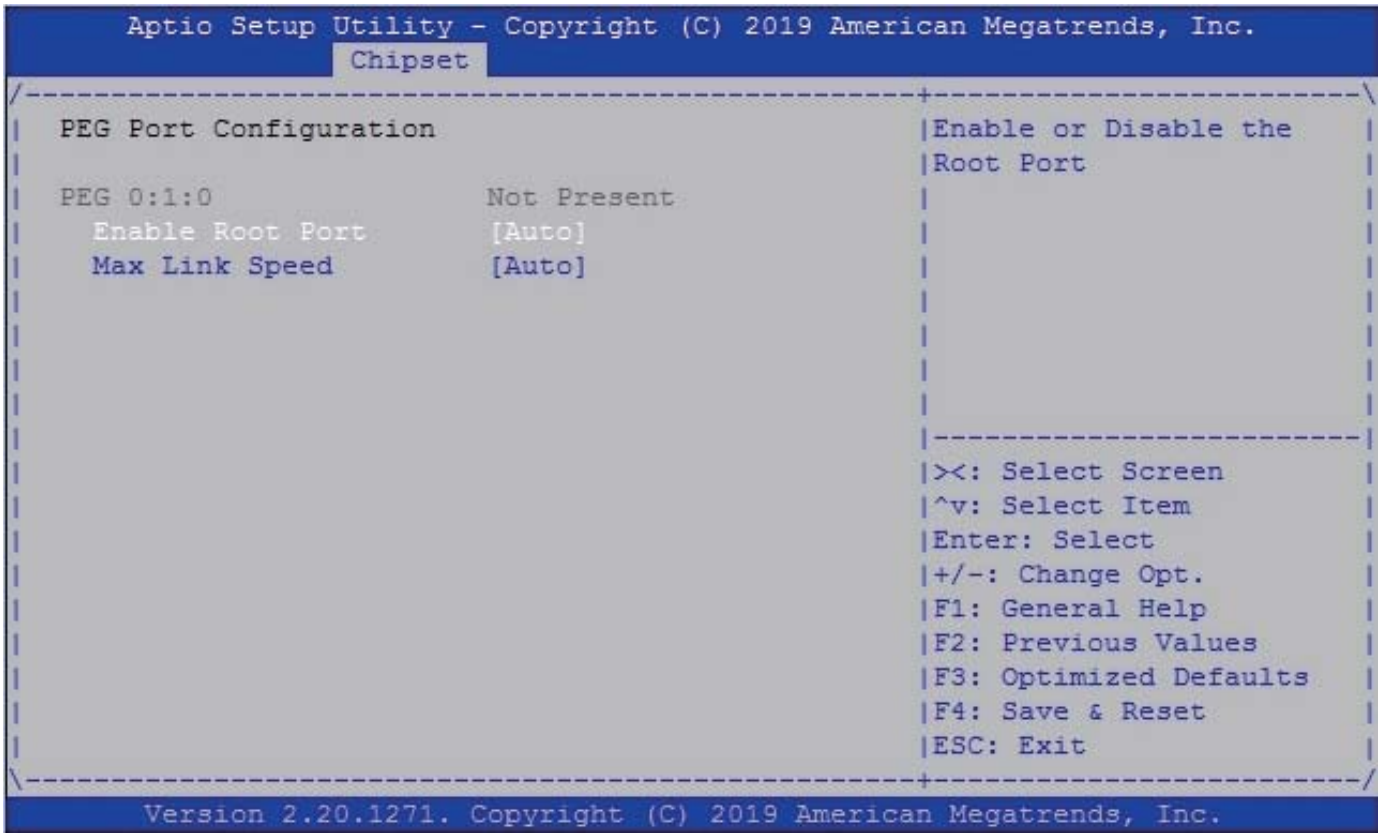
Use this item to select DVMT 5.0 pre-allocated (fixed) graphics memory size used by the internal graphics device. The optional settings are: 16MB, 32MB, 64MB (default)

DVMT Total Gfx Mem

Use this item to select DVMT 5.0 total graphics memory size used by the internal graphics device

The optional settings are: 128MB, 256MB (default), MAX.

4-7-1-2 ► PEG Port Configuration



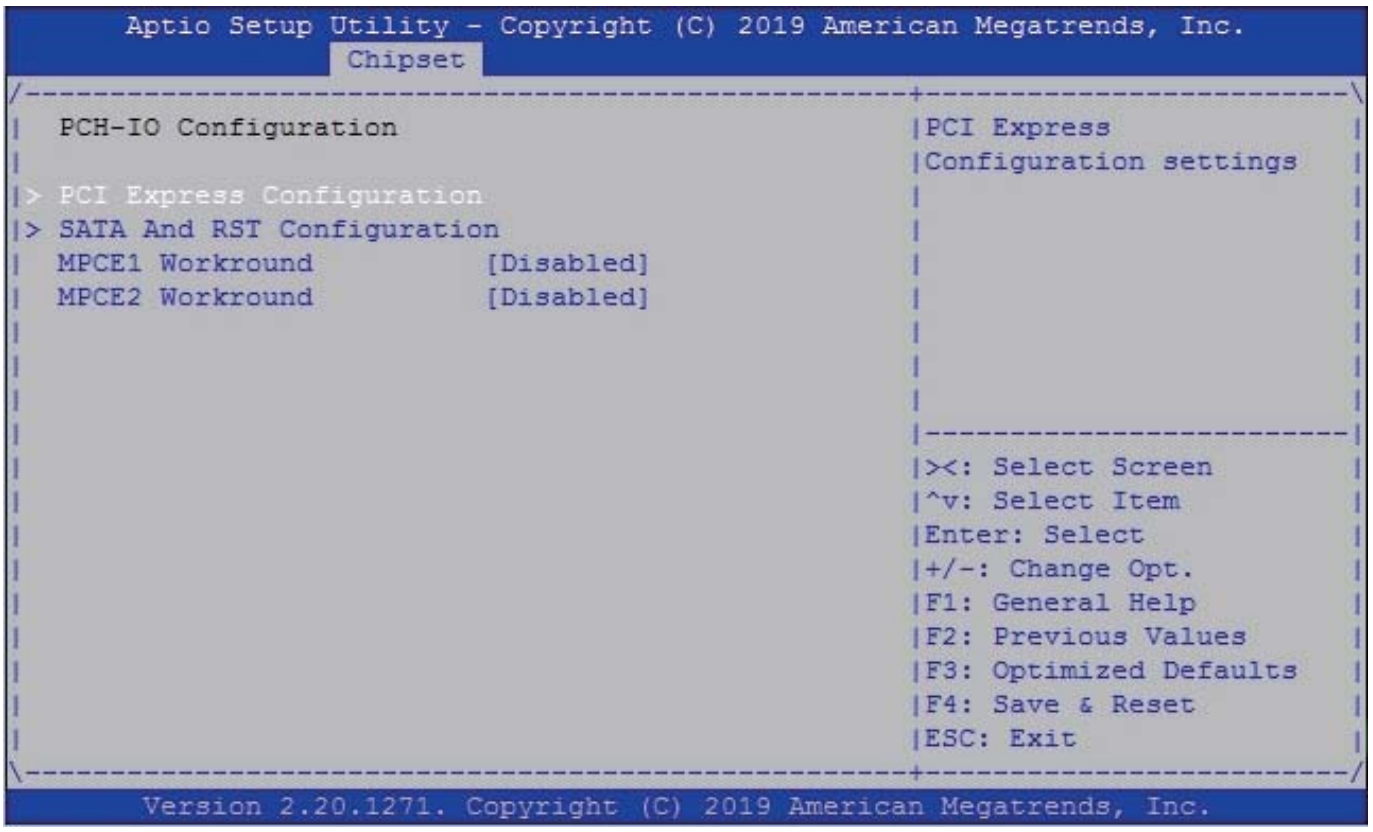
Enable Root Port

To enable the Root Port or not. The optional settings are: Auto (default), Enabled, Disabled.

Max Link Speed

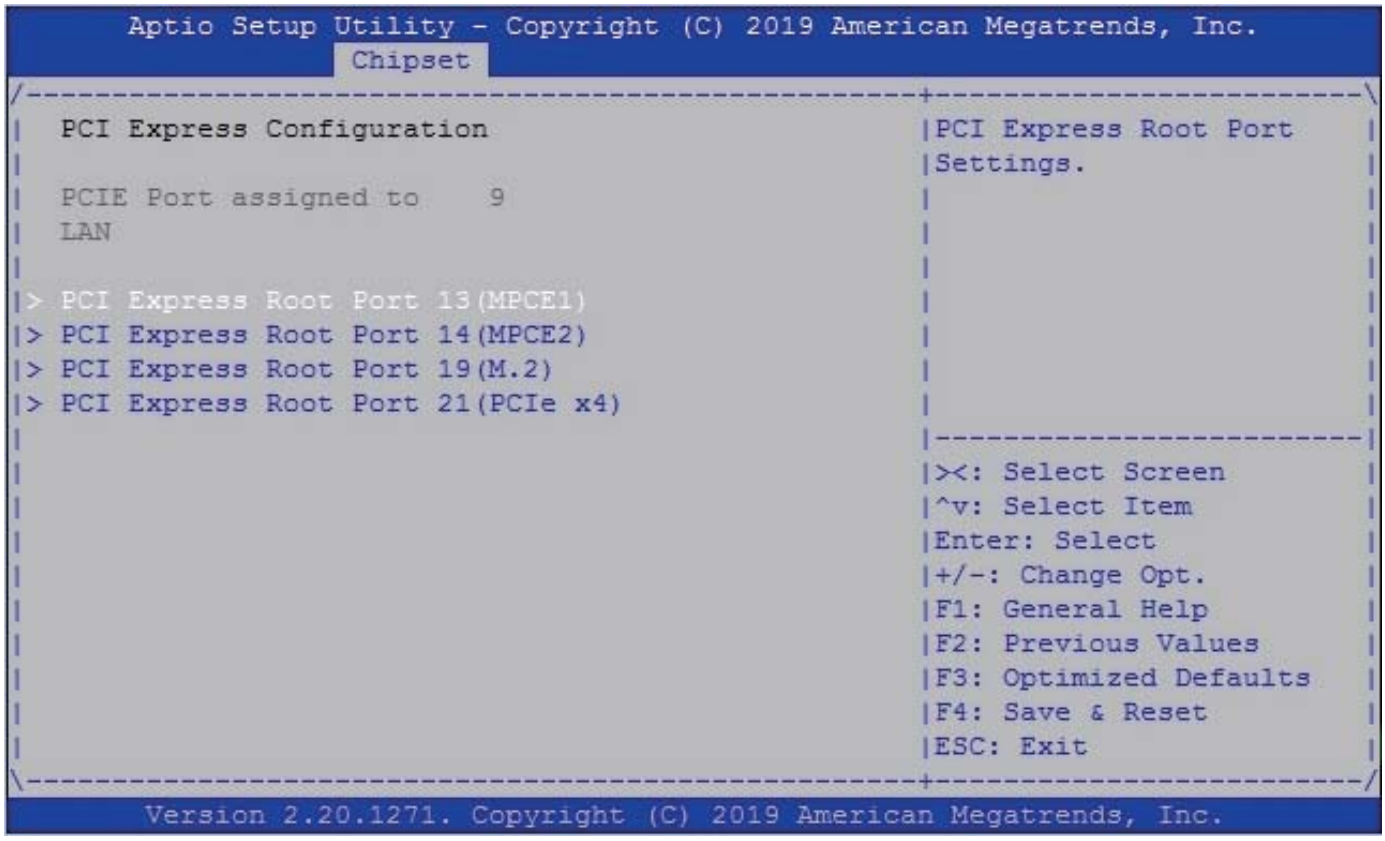
Configure PEG Port Max Speed. The optional settings are: Auto (default), Gen1, Gen2, Gne3.

4-7-2 PCH-IO Configuration



PCI Express Configuration.
Please refer section 4-7-2-1
SATA And RST Configuration.
Please refer section 4-7-2-2

4-7-2-1 ► PCI Express Configuration



PCI Express Root Port 13 (MPCE1)

Please refer section 4-7-2-1-1

PCI Express Root Port 14 (MPCE2)

Please refer section 4-7-2-1-2

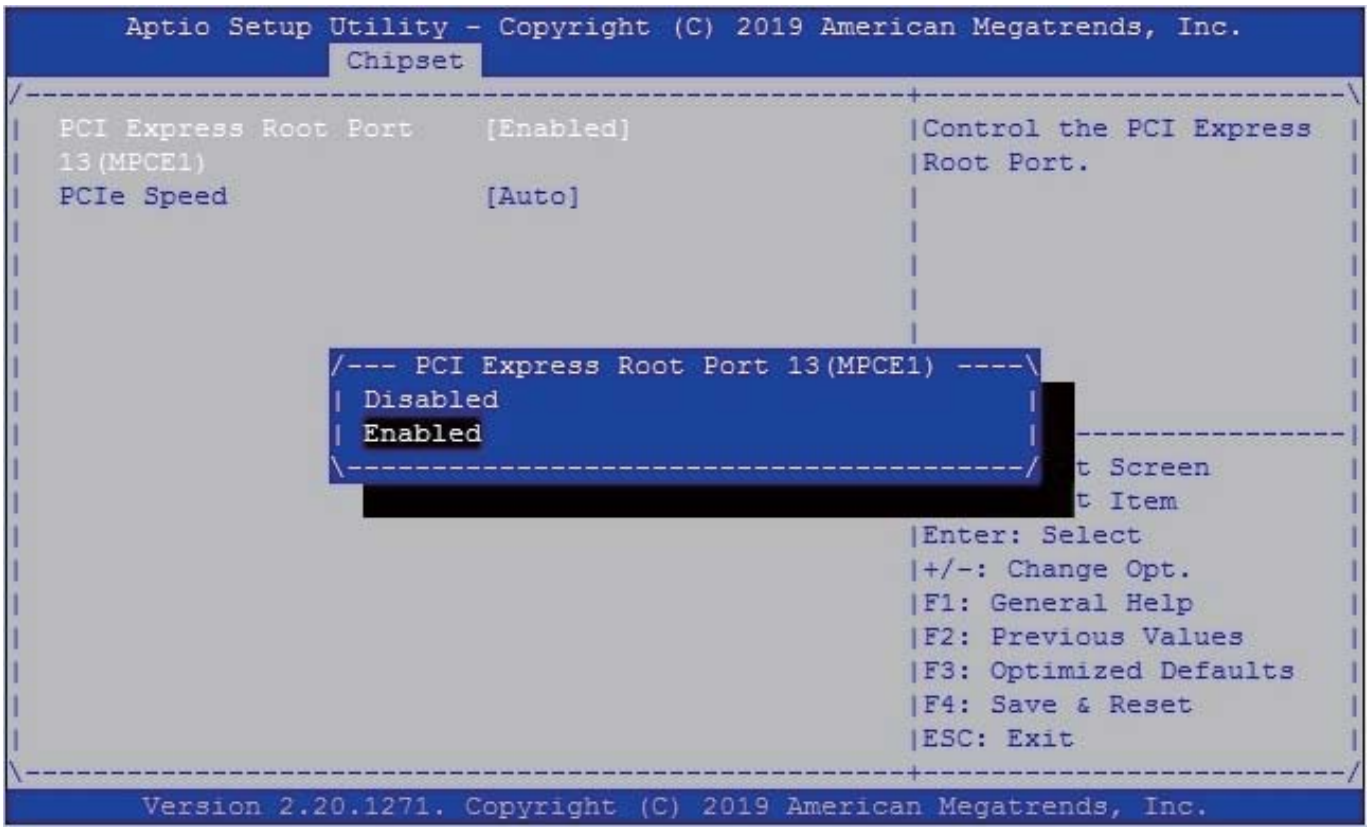
PCI Express Root Port 19 (M.2)

Please refer section 4-7-2-1-3

PCI Express Root Port 21 (PCIe x 4)

Please refer section 4-7-2-1-4

4-7-2-1-1 ► PCI Express Root Port 13 (MPCE1)



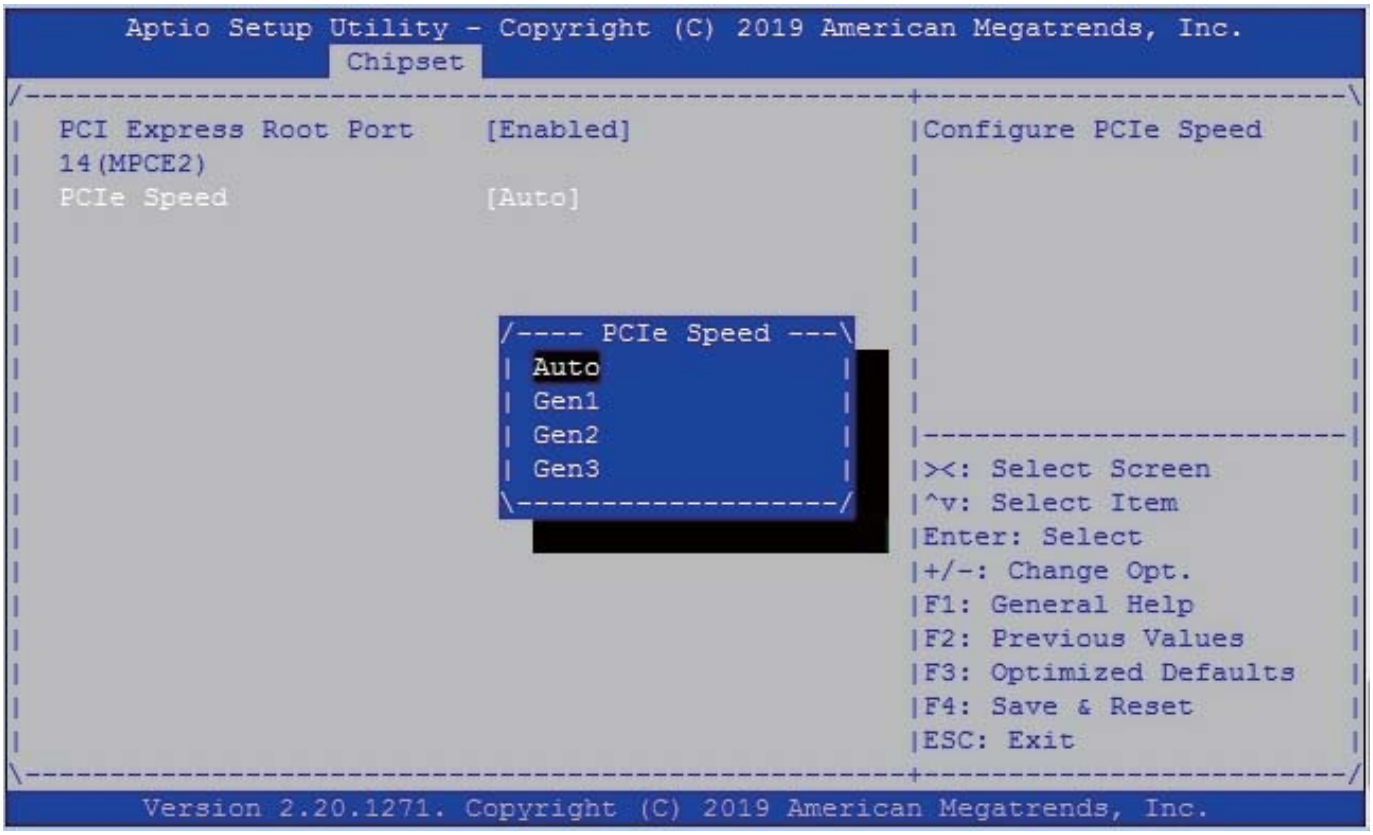
PCI Express Root Port 13 (MPCE1)

The optional settings are: Enabled (default), Disabled.

PCI Speed

To select PCI Express port speed. The optional settings are: Auto (default), Gen1, Gen2, Gen3

4-7-2-1-2 ► PCI Express Root Port 14 (MPCE2)



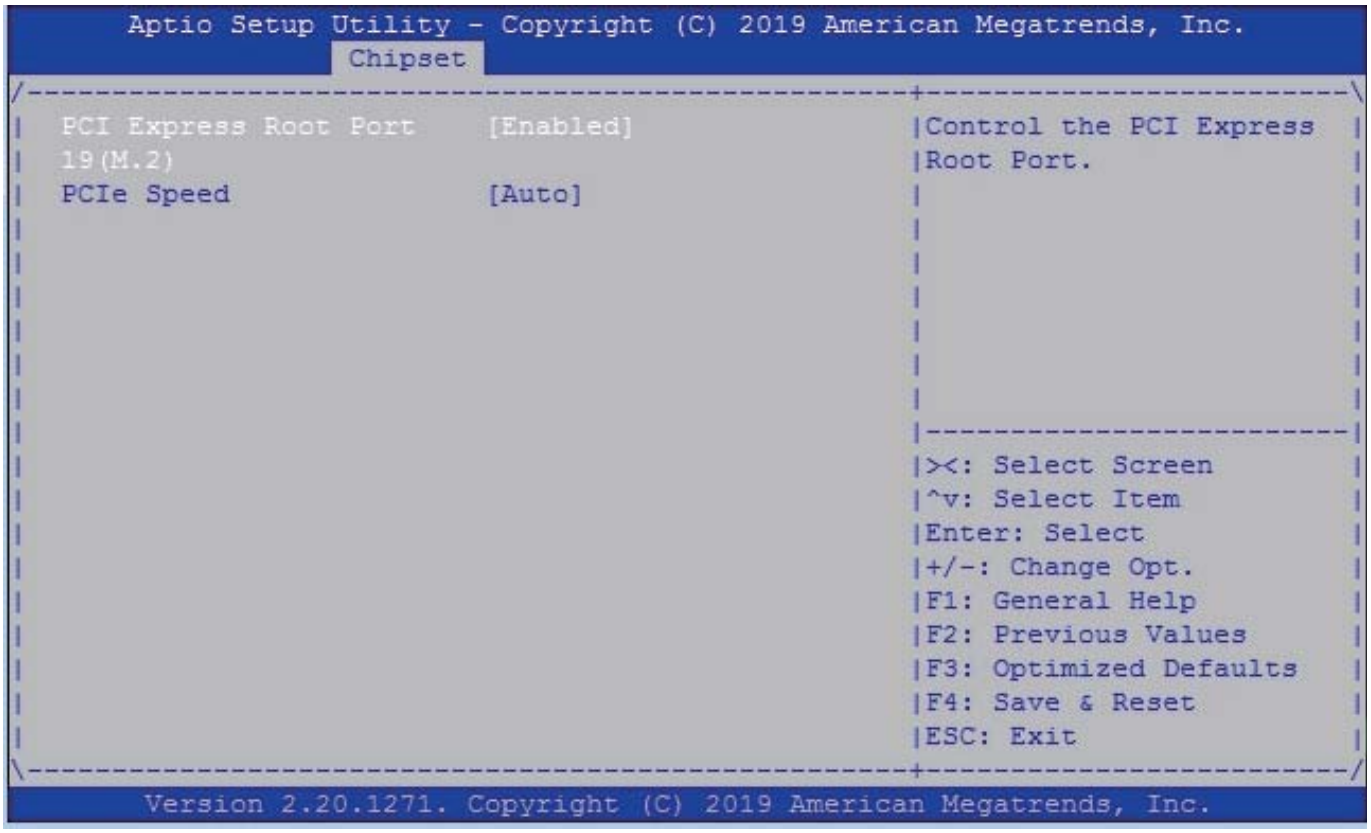
PCI Express Root Port 12 (MPCE2)

The optional settings are: Enabled (default), Disabled.

PCI Speed

To select PCI Express port speed. The optional settings are: Auto (default), Gen1, Gen2, Gen3

4-7-2-1-3 ► PCI Express Root Port 19 (M.2)



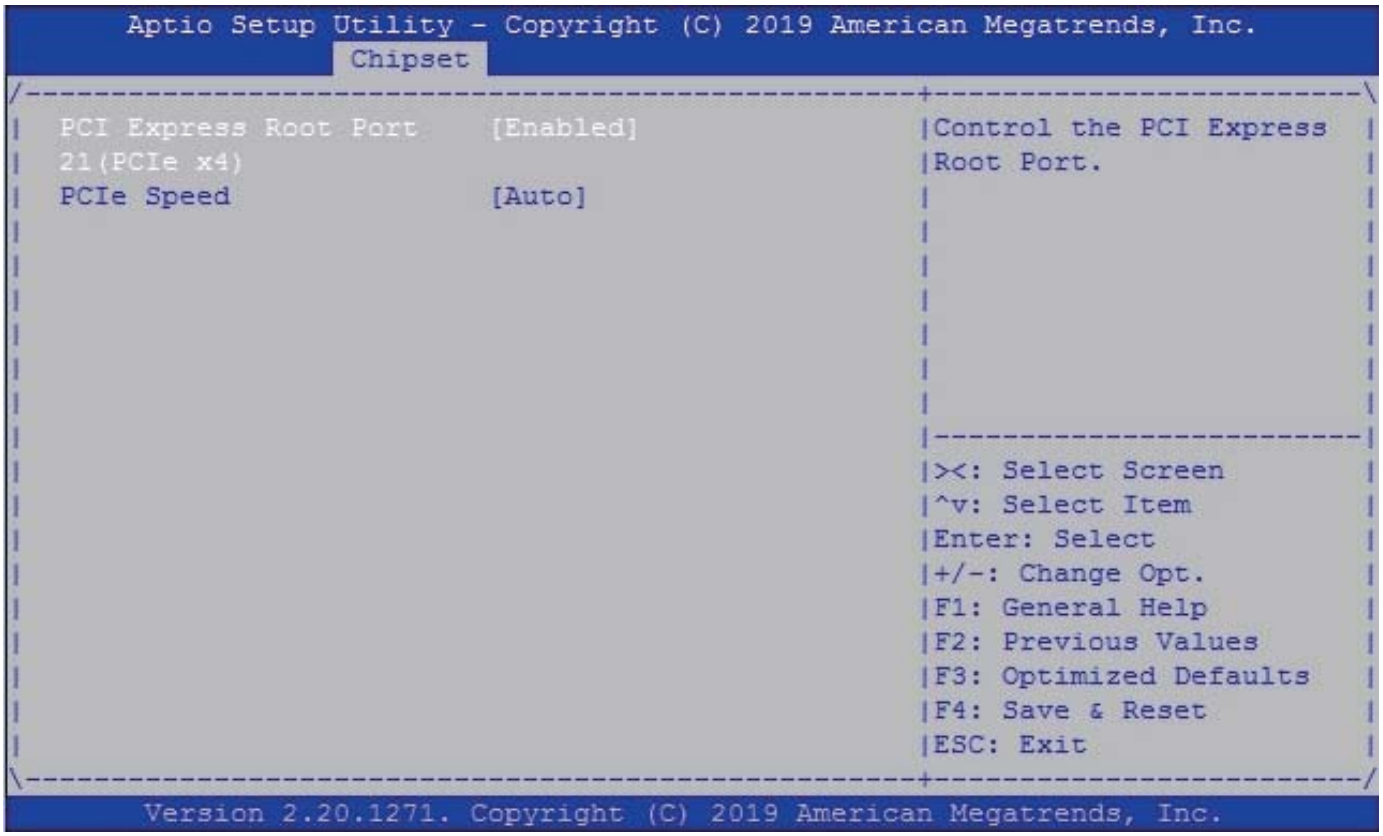
PCI Express Root Port 19 (M.2)

The optional settings are: Enabled (default), Disabled.

PCI Speed

To select PCI Express port speed. The optional settings are: Auto (default), Gen1, Gen2, Gen3

4-7-2-1-4 ► PCI Express Root Port 21 (PCIe x 4)



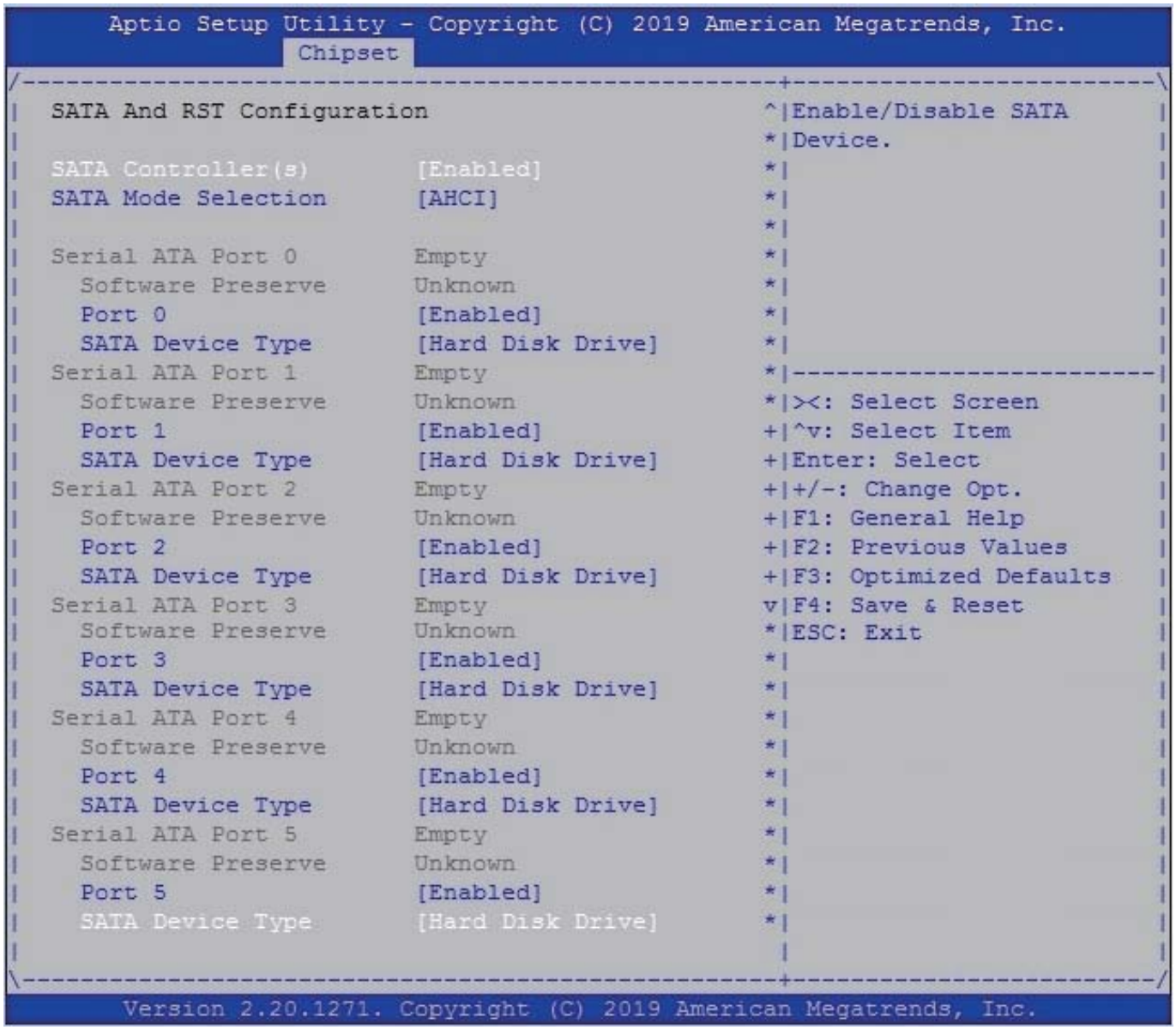
PCI Express Root Port 21 (PCIe x 4)

The optional settings are: Enabled (default), Disabled.

PCI Speed

To select PCI Express port speed. The optional settings are: Auto (default), Gen1, Gen2, Gen3

4-7-2-2 ► SATA And RST Configuration



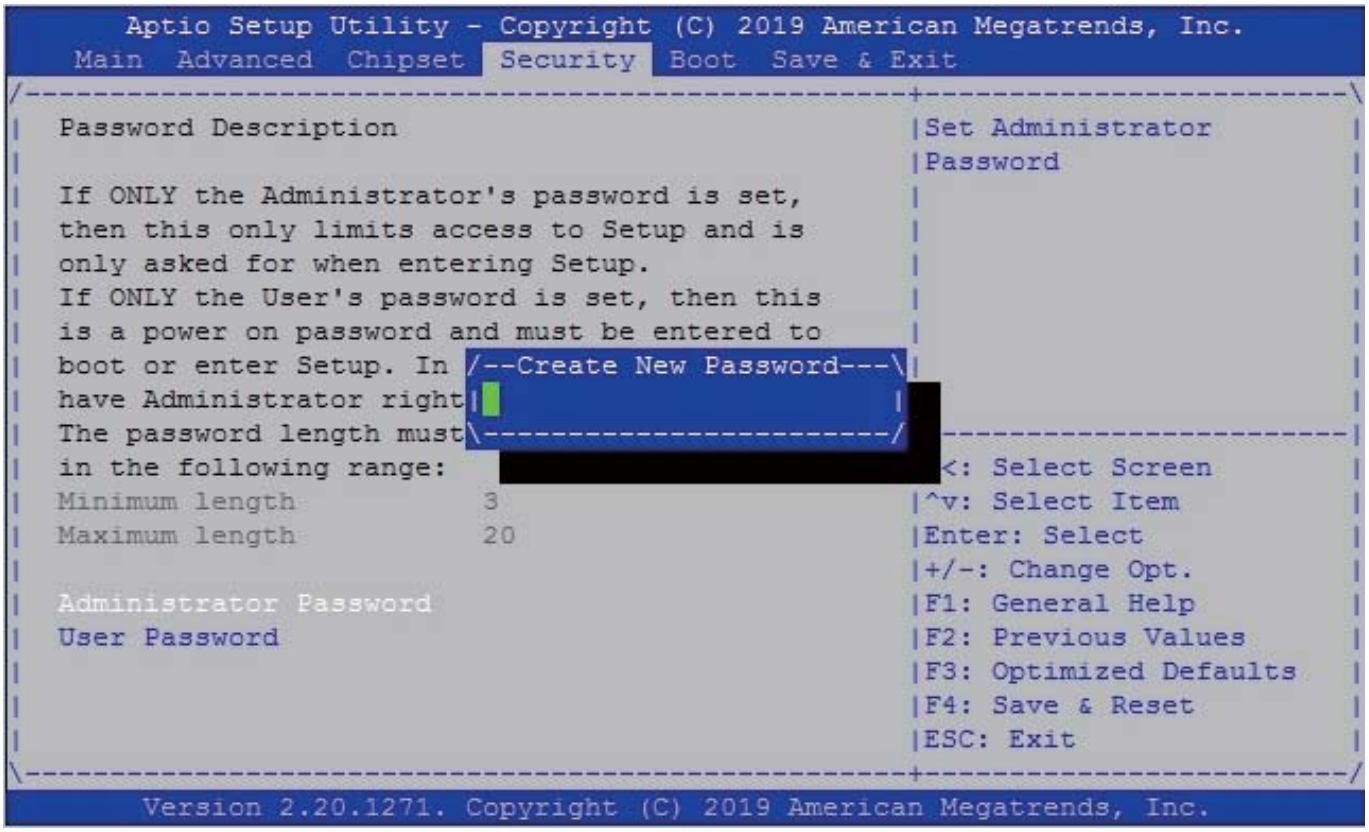
SATA Controller

Use this item to Enable or Disable SATA Device.

SATA Mode Selection

Determines how SATA controller(s) operate. The optional settings are: AHCI, Intel RST Premium.

4-8 Security

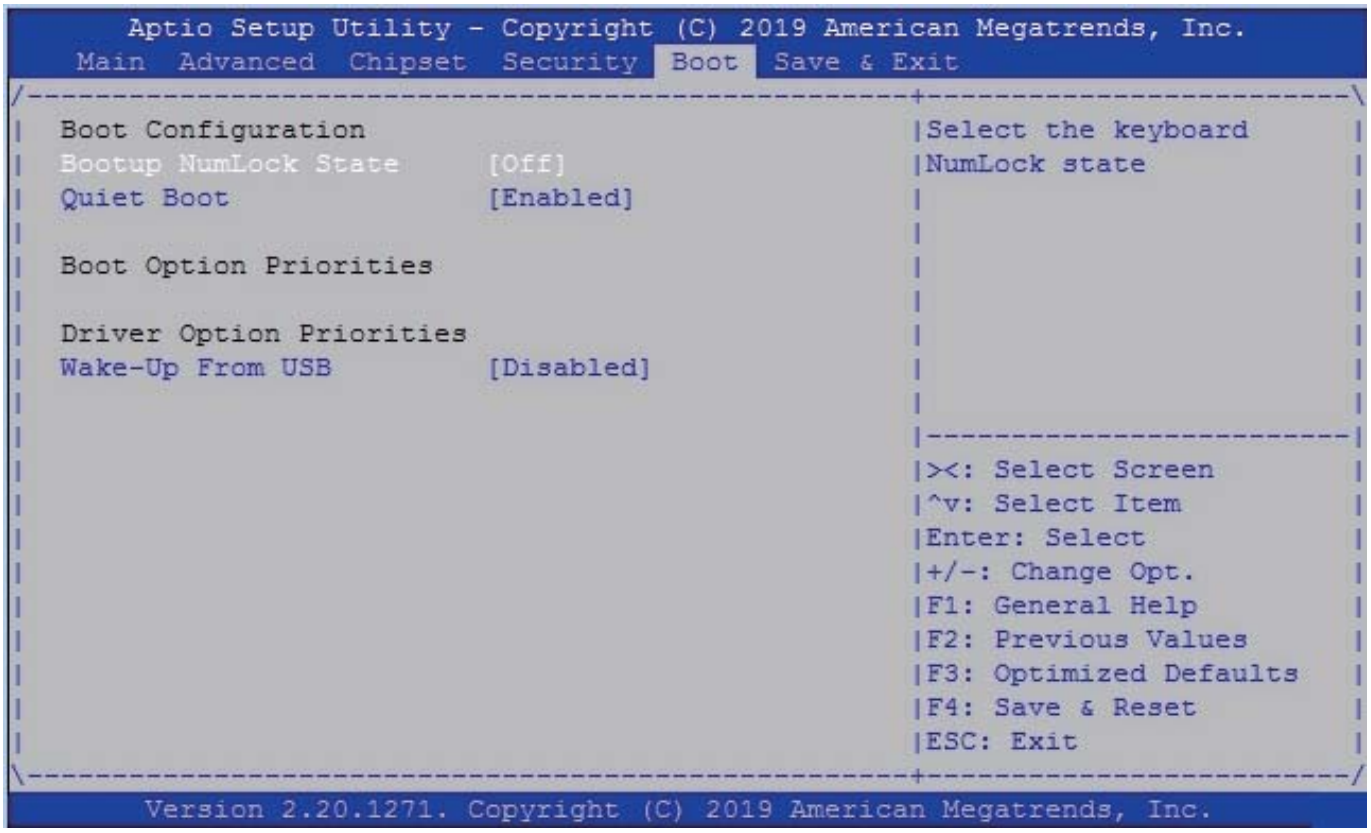


Administrator Password

User Password

To set up an Administrator or an User password

4-9 Boot



Bootup NumLock State

To select Power-on state for NumLock, default is <On>

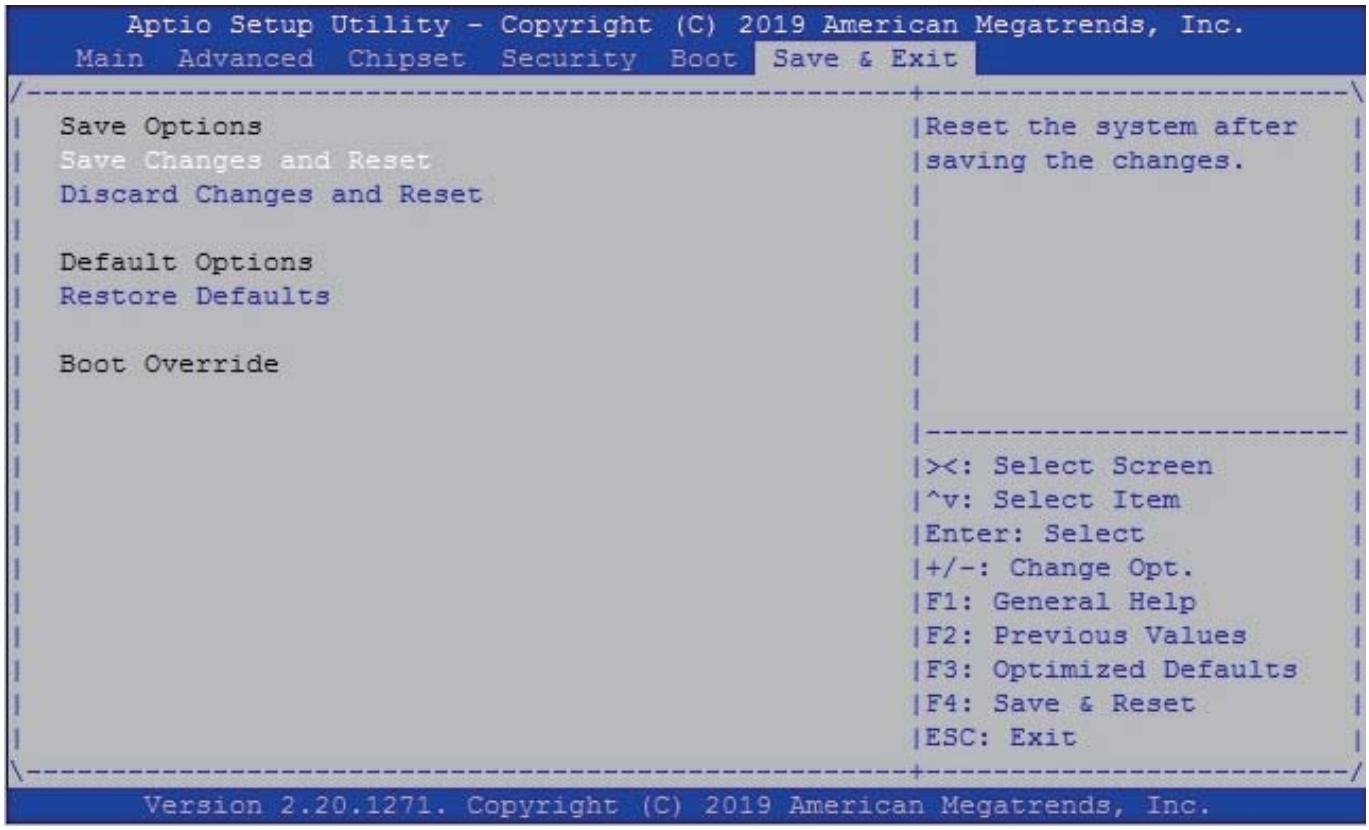
Quiet Boot

The optional settings are: Enabled (default), Disabled.

Wake-Up From USB

The optional settings are: Enabled, Disabled (default).

4-10 Save & Exit



Save Change and Reset

Save configuration and reset

Discard Changes and Reset

Reset without saving the changes

Restore Defaults

To restore the optimal default for all the setup options

4-11 How to update AMI BIOS

STEP 1. Prepare a bootable disc.
(Storage device could be USB pen drive.)
STEP 2. Copy utility program and latest BIOS to your bootable disc.
You may download it from our website.
STEP 3. Here take CI370D as an example, insert your bootable disc into X: (X could be C:, A: or others.
It depends on which type of storage device you use.)

Start the computer and type
For legacy mode,
X:\>afudos.exe CI370DA1.bin /p /b /n /x
For UEFI mode,
X:\>AfuEfix64.efi CI370DA1.bin /p /b /n /x

Appendix B: Resolution list

640 x 480 x (256 / 16bit / 32bit)
800 x 600 x (256 / 16bit / 32bit)
1024 x 768 x (256 / 16bit / 32bit)
1152 x 864 x (256 / 16bit / 32bit)
1280 x 600 x (256 / 16bit / 32bit)
1280 x 720 x (256 / 16bit / 32bit)
1280 x 768 x (256 / 16bit / 32bit)
1280 x 800 x (256 / 16bit / 32bit)
1280 x 960 x (256 / 16bit / 32bit)
1280 x 1024 x (256 / 16bit / 32bit)
1400 x 1050 x (256 / 16bit / 32bit)
1440 x 900 x (256 / 16bit / 32bit)
1600 x 900 x (256 / 16bit / 32bit)
1600 x 1200 x (256 / 16bit / 32bit)
1680 x 1050 x (256 / 16bit / 32bit)
1920 x 1080 x (256 / 16bit / 32bit)
1920 x 1200 x (256 / 16bit / 32bit)