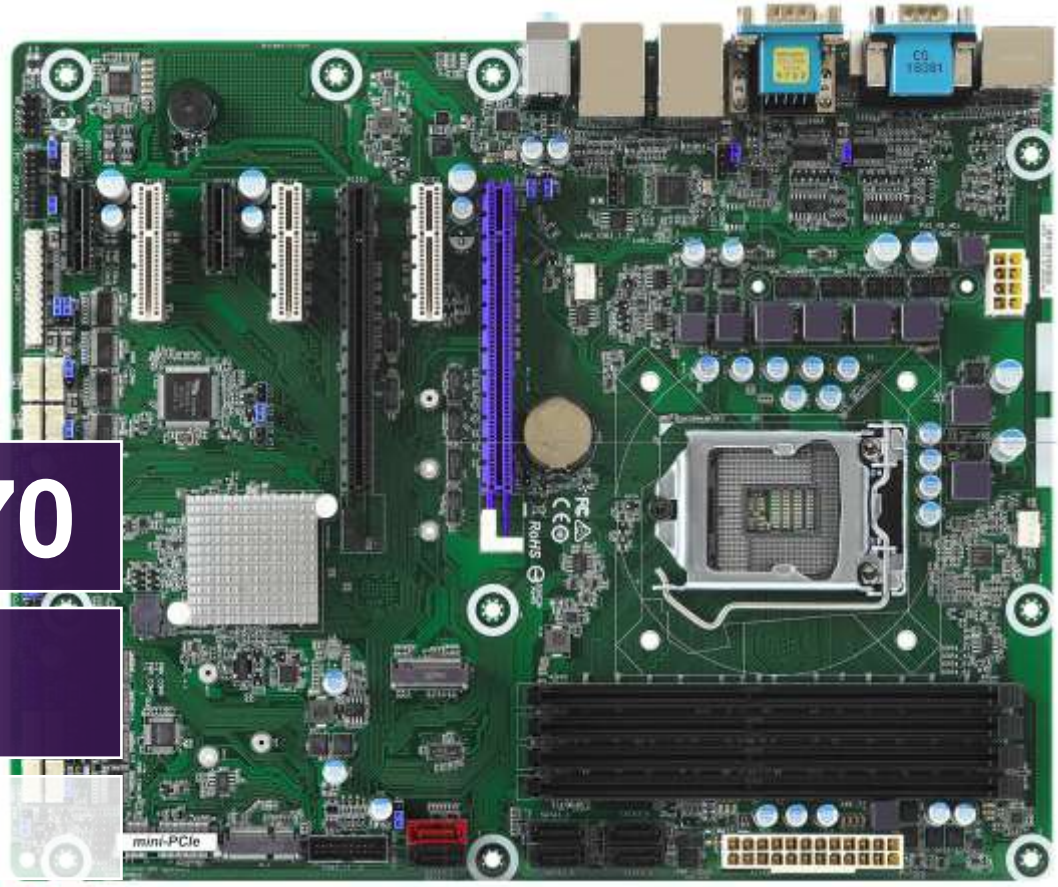
RUBY-D811-Q370

Industrial ATX Board

Version 1.6



Revision History

R1.0	Preliminary
R1.1	Update Mechanical Dimensions
R1.2	Update board size information
R1.3	Update jumper information
R1.4	Update USB3.0 information
R1.5	Update RAID Item of SATA Mode Selection & Super IO Configuration & TPM information
R1.6	Update memory support information

Contents

Preface

This user's guide provides information about the components, features, connectors and BIOS Setup menus available on the RUBY-D811. This document should be referred to when designing ATX IMB application. The other reference documents that should be used include the following:

- ✧Intel Coffee Lake Design Guide
- ✧Intel Coffee Lake Specification

Please contact Portwell Sales Representative for above documents.

Disclaimer

The information contained within this user's guide, including but not limited to any product specification, is subject to change without notice.
Portwell provides no warranty with regard to this user's guide or any other information contained herein and hereby expressly disclaims any implied warranties of merchantability or fitness for any particular purpose with regard to any of the foregoing. Portwell assumes no liability for any damages incurred directly or indirectly from any technical or typographical errors or omissions contained herein or for discrepancies between the product and the user's guide. In no event shall Portwell be liable for any incidental, consequential, special, or exemplary damages, whether based on tort, contract or otherwise, arising out of or in connection with this user's guide or any other information contained herein or the use thereof.

Trademarks

Product names, logos, brands, and other trademarks featured or referred to within this User's guide or the Portwell website, are the property of their respective trademark holders. These trademark holders are not affiliated with Portwell, our products, or our website.

Warranty

Portwell makes no representation, warranty or guaranty, express or implied regarding the products except its standard form of limited warranty ("Limited Warranty"). Portwell may in its sole discretion modify its Limited Warranty at any time and from time to time.

Beginning on the date of shipment to its direct customer and continuing for the published warranty period, Portwell represents that the products are new and warrants that each product failing to function properly under normal use, due to a defect in materials or workmanship or due to non conformance to the agreed upon specifications, will be repaired or exchanged, at Portwell's option and expense.

Certification

Portwell is certified to DIN EN ISO 9001:2000 standard.

Technical Support

Portwell technicians and engineers are committed to providing the best possible technical support for our customers so that our products can be easily used and implemented.

We request that you first visit our website at http://www.portwell.com.tw/support/ for the latest documentation, utilities and drivers, which have been made available to assist you. If you still require assistance after visiting our website then contact our technical support department by email at tsd@mail.portwell.com.twfor further assistance. Thank you!

1 Introduction

RUBY-D811-Q370 based on the Intel® Core™ Processor which offers 14nm Hi-K process technology with energy efficient architecture. RUBY-D811 support dual channels DDR4 Long - DIMM up to 64GB.

Desktop solution is still popular in the market of DVR and Factory Automation which can fulfill most of these applications; therefore, with high performance and high-end specifications, RUBY-D811 is our first generation Coffee -S chip architecture on ATX IMB line.

2 Specifications

Main Processor	◆ Intel® Coffee -S Core™ i Processors
System Chipset	◆ Intel®Q370 Express chipset
System BIOS	◆ AMI UEFI BIOS
Main Memory	◆ Up to 128GB in 4 slots DDR4 Long-DIMM sockets. Supports dual channel DDR4 2400/2666 MHz SDRAM
Graphics	◆ Controller: Intel® Gfx Gen 9, UHD 630 graphics ◆ VGA: Supports VGA up to resolution 1920 x 1200 ◆ Dual DP: Supports DP up to resolution 4096 x 2304 ◆ HDMI: Supports HDMI up to resolution 4096 x 2160
Expansion Interface	◆ One mini-PCIe socket ◆ One M.2 (Key E) for Wireless ◆ One M.2 (Key M) for SSD ◆ Two PCIe x16 slot ◆ Three PCIe x4 slot ◆ Two PCIe x1 slot
SATA Interface	◆ Six SATA ports(SATA 6Gb/s)
Input/Output	◆ COM Ports: 2x RS-232/422/485 on REAR I/O & 8x RS-232 on board header ◆ USB Port: 4x USB 3.0 on REAR I/O, 2x USB 3.0 on board header, 6x USB 2.0 on board header ◆ Audio Interface: Line-in / Line-Out / Mic-In
Ethernet	◆ Supports dual 10/100/1000 Mbps Ethernet port (s) via PCI Express x1 bus which provides 500 MB/s data transmission rate
High Drive GPIO	◆ One pin-header for GPIO(8bit GPIO)

Mechanical and environmental specifications	◆ Operating temperature: 0 ~ 60° C ◆ Storage temperature:-20 ~ 80° C ◆ Humidity: 5 ~ 90% non-condensing ◆ Power supply voltage: ATX ◆ Board size: 304.8mm x 243.8 mm

2.1 Supported Operating Systems

The RUBY-D811 supports the following operating systems.

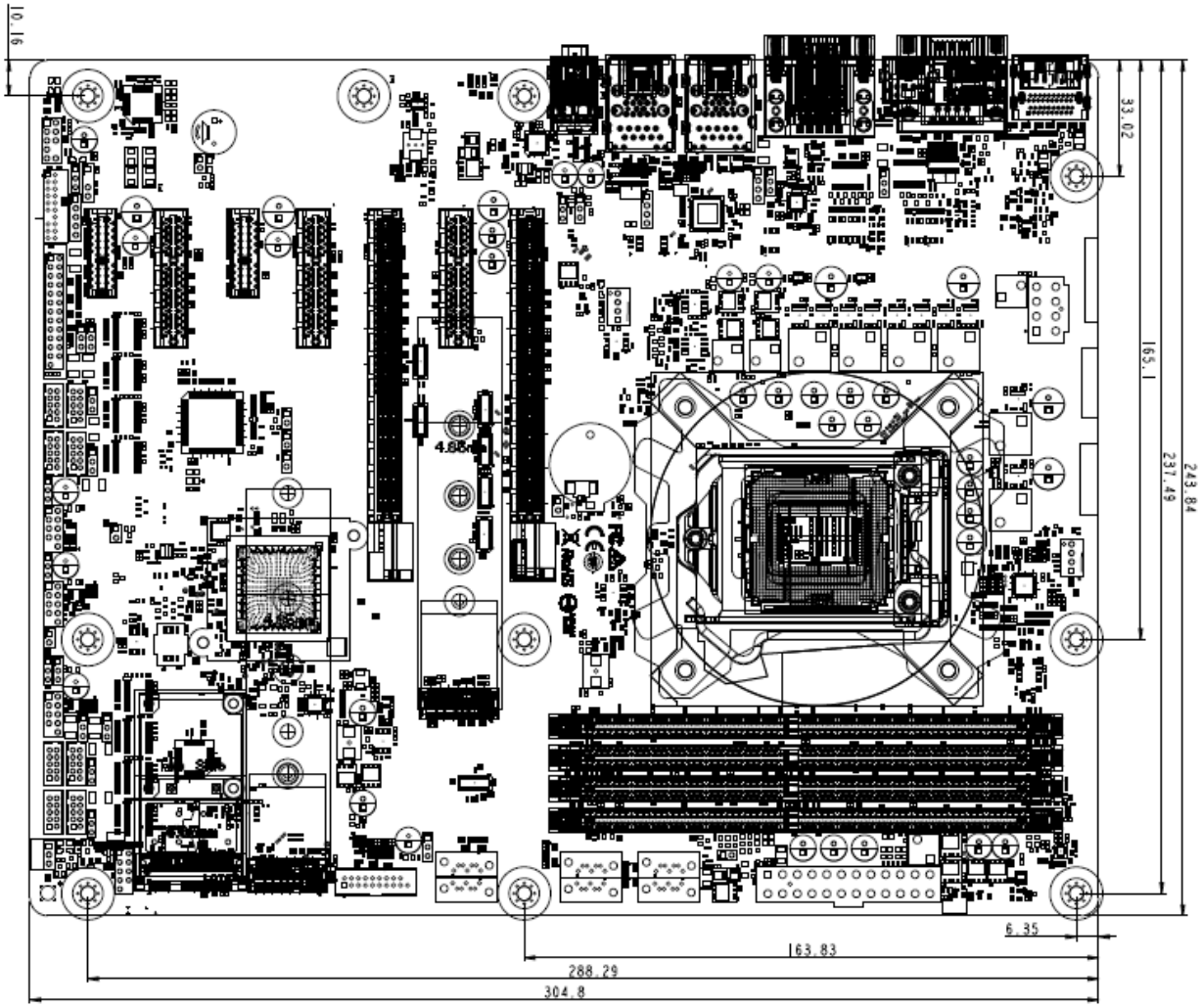
✧ Windows 10 Enterprise & IOT Enterprise (64b) RS3

✧ Ubuntu, SuSe, Redhat Enterprise 1,2 (Kernel 4.14)

✧ Yocto 2.4 Tool-based Embedded Linux Distribution

✧ Wind River VxWorks 7

2.2 Mechanical Dimensions



2.3 Power Consumption

Test Configuration	
CPU Type	Intel® Core™ i7-8700TE CPU @ 3.2GHz
SBC BIOS	Portwell, Inc. RUBY-D811 TEST BIOS (90418T00)
Memory	WARIS UB-DIMM DDR4 2133 16GB
VGA Card	Onboard Intel® UHD Graphics 630
VGA Driver	Intel® UHD Graphics 630 Version: 24.20.100.6286
LAN Card	Onboard Intel® Ethernet Connection(2) I219-LM
LAN Driver	Intel® Ethernet Connection(2) I219-LM Version: 12.13.17.7
LAN Card #2	Onboard Intel® I210AT Gigabit Network Connection
LAN Driver #2	Intel® I210AT Gigabit Network Connection Version: 12.14.7.0
Audio Card	Onboard Realtek ALC887 High Definition Audio
Audio Driver	Realtek ALC887 High Definition Audio Version: 6.0.1.8186
Chipset Driver	Intel® Coffee lake-S Chipset Device Software Version: 10.1.17541
USB 3.0 Driver	Intel® USB 3.0 eXtensible Host Controller Adaptation Driver Version: 6.3.9600.17393
SATA HDD	Intel SSD 256G
Power Supply	FSP460-60PFB 460W / GADIWA 5V/12V DC POWER

Power consumption

ATX:

Item	Power ON	Full Loading 10Min	Full Loading 30Min
CPU +12V	0.98A	2.05A	1.98A
System +12V	0.83A	1.48A	1.26A
System +3.3V	0.55A	0.65A	0.69A
System +5V	1.11A	1.37A	1.34A
System+ Device +12V	0.97A	1.81A	1.56A
System+ Device +5V	1.87A		
USB2.0 Loading Test	4.98 V/ 570 mA		

2.4 Environmental Specifications

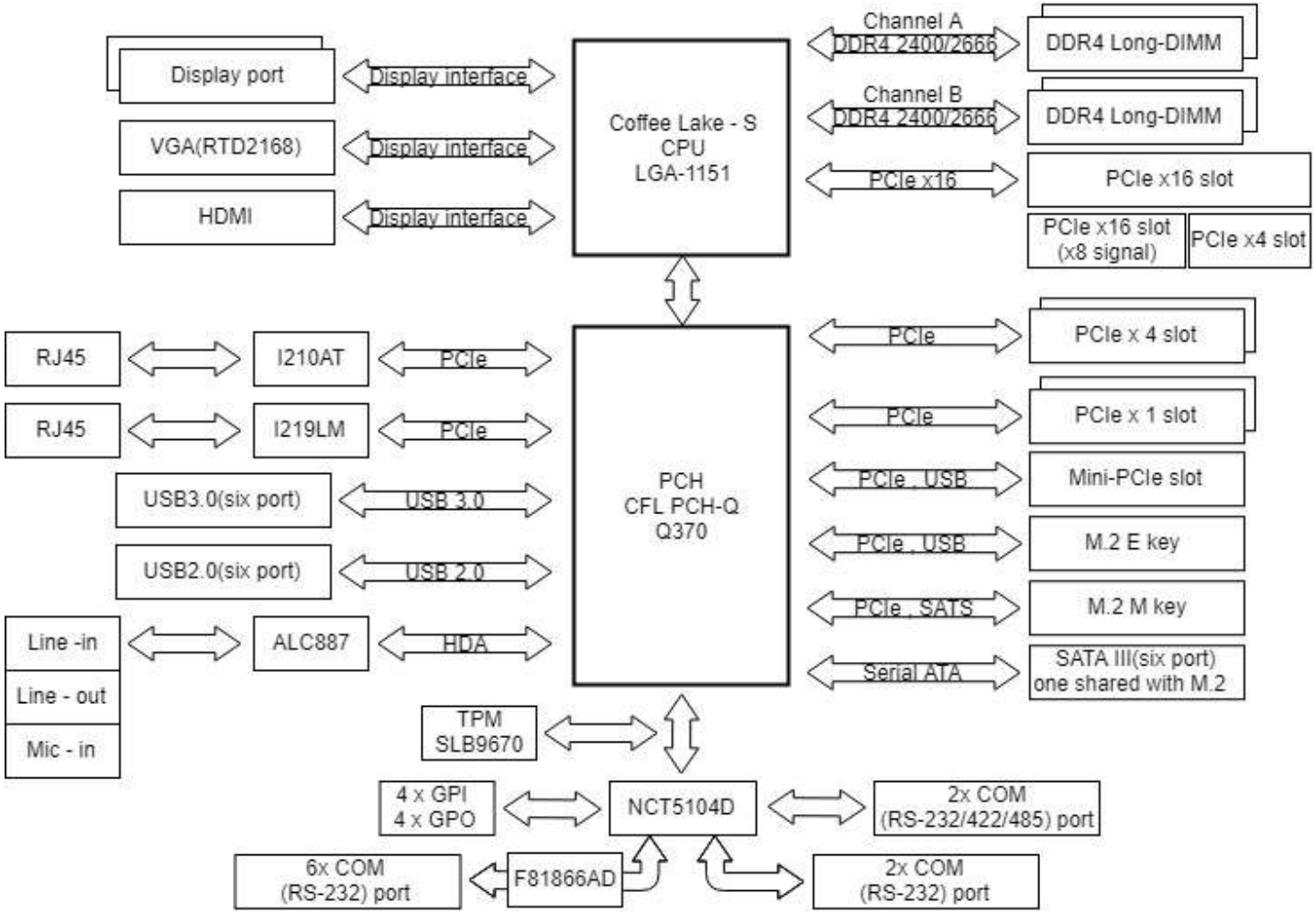
Storage Temperature : -20~80°C

Operation Temperature : 0~60°C

Storage Humidity : 5~90%

Operation Humidity: 10~90%

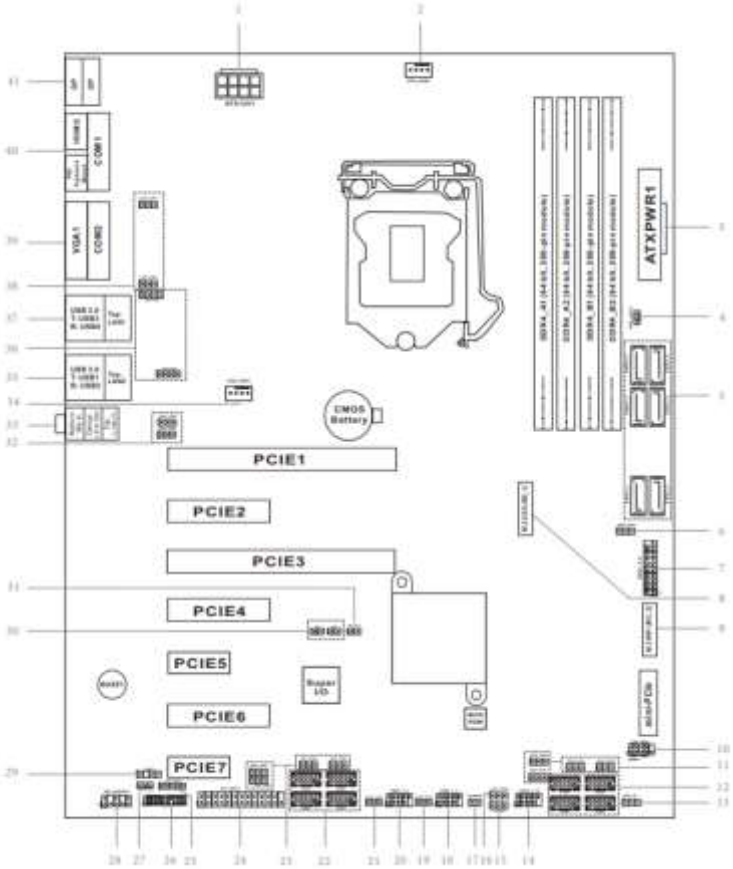
3 Block Diagram



4 Hardware Configuration

4.1 Jumpers and Connectors

This chapter indicates jumpers', headers' and connectors' locations. Users may find useful information related to hardware settings in this chapter.



4.2 Jumpers Settings

For users to customize RUBY-D811's features. In the following sections, Short means covering a jumper cap over jumper pins; Open or N/C (Not Connected) means removing a jumper cap from jumper pins. Users can refer to Figure 1 for the Jumper allocations.

Jumper Table

The jumper settings are schematically depicted in this manual as follows:

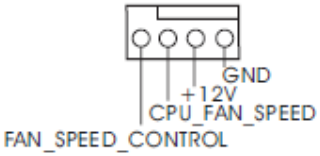
Jumper Function List	
1	ATX 12V Power Connector
2	CPU FAN Connector (+12V)
3	24-pin ATX Power Input Connector
4	PWR LOSS Header (PWR_LOSS1)
5	SATA3 Connectors (SATA3_1 ~ SATA3_6)
6	USB3_PWR3 (For USB3_5_6)
7	USB 3.0 Header (USB3_11_12)
8	M.2 Key-M Socket (M2_1)
9	M.2 Key-E Socket (M2_2)
10	System Panel Header
11	COM Port Pin9 PWR Setting Jumpers (For COM Port7~10)
12	COM7, 8, 9, 10 _COM Port Headers

13	ATX/AT Mode Jumper (PWR_JP1)
14	USB 2.0 Headers_ USB2_9_10
15	Clear CMOS Headers_ CLRMOS1
16	USB2_PWR3 (For USB2_9_10)
17	Clear CMOS Headers_ CLRMOS2
18	USB 2.0 Headers_ USB2_7_8
19	USB2_PWR2 (For USB2_7_8)
20	USB 2.0 Headers_ USB2_5_6
21	USB2_PWR1 (For USB2_5_6)
22	COM Port Headers _COM3, 4, 5, 6
23	COM Port Pin9 PWR Setting Jumpers (For COM Port3~6)
24	Printer Port / GPIO Header (LPT_GPIO1)
25	Digital Input / Output Power Select
26	LPC Header
27	Digital Input / Output Default Value Setting (JGPIO_SET1)
28	Front Panel Audio Header
29	SPDIF Header
30	Chassis Intrusion Headers (CI1, CI2)
31	ATX/AT Mode Jumper (SIO_AT1)
32	USB3_PWR1 (For USB3_1_2) USB3_PWR2 (For USB3_3_4)
33	Audio Jacks

34	Chassis FAN Connector (+12V)
35	Top : RJ45 LAN Port (LAN2) Bottom : USB 3.0 Ports (USB3_1_2)
36	LAN LED Headers
37	Top : RJ45 LAN Port (LAN1) Bottom : USB 3.0 Ports (USB3_3_4)
38	COM Port Pin9 PWR Setting Jumpers (For COM Port1~2)
39	Top : COM Port (COM2) (RS232/422/485)* Bottom : D-Sub Port (VGA1)
40	Top : COM Port (COM1) (RS232/422/485)* Bottom Right : PS/2 Keyboard/Mouse Port Bottom Left: HDMI Port (HDMI2)
41	Top : DisplayPort 1 Bottom : DisplayPort 2

1: ATX 12V Power Connector

2: CPU FAN Connector (+12V)

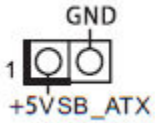


3: 24-pin ATX Power Input Connector

4: PWR LOSS Header (PWR_LOSS1)

Short： Power Loss

Open： no Power Loss



5: SATA3 Connectors (SATA3_1 ~ SATA3_6)

USB Power Setting Jumpers

6 : USB3_PWR3 (For USB3_5_6)

16 : USB2_PWR3 (For USB2_9_10)

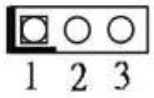
19 : USB2_PWR2 (For USB2_7_8)

21 : USB2_PWR1 (For USB2_5_6)

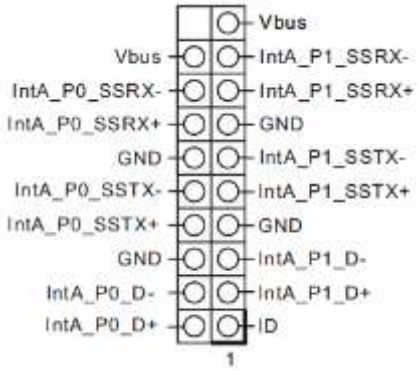
32 : USB3_PWR1 (For USB3_1_2)

1-2：+5V

2-3：+5VSB



7: USB 3.0 Header (USB3_11_12)



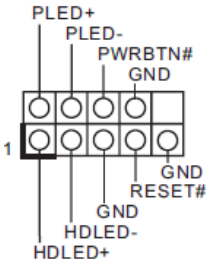
8: M.2 Key-M Socket (M2_1)

Pin	Signal	Signal	Pin
1	GND	+3.3V	2
3	GND	+3.3V	4
5	NA	NA	6
7	NA	NA	8
9	GND	SATA_LED	10
11	NA	+3.3V	12
13	NA	+3.3V	14
15	GND	+3.3V	16
17	NA	+3.3V	18
19	NA	NA	20
21	GND	NA	22
23	NA	NA	24
25	NA	NA	26
27	GND	NA	28
29	NA	NA	30
31	NA	NA	32
33	GND	NA	34
35	NA	NA	36
37	NA	DEVSLP	38
39	GND	SMB_CLK	40
41	SATA-B+	SMB_DATA	42
43	SATA-B-	NA	44
45	GND	NA	46
47	SATA-A-	NA	48
49	SATA-A+	PERST#	50
51	GND	CLKREQ#	52
53	PEFCLKn	WAKE#	54
55	PEFCLKp	NA	56
57	GND	NA	58
67	NA	NA	68
69	PEDET	+3.3V	70
71	GND	+3.3V	72
73	GND	+3.3V	74
75	GND		

9: M.2 Key-E Socket (M2_2)

Pin	Signal	Signal	Pin
1	GND	+3.3V	2
3	USB_D+	+3.3V	4
5	USB_D-	NA	6
7	GND	NA	8
9	CNV_WGR_D1-	CNV_RF_RESET	10
11	CNV_WGR_D1+	NA	12
13	GND	MODEM_CLKREQ	14
15	CNV_WGR_D0-	NA	16
17	CNV_WGR_D0+	GND	18
19	GND	NA	20
21	CNV_WGR_CLK-	CNV_BRI_RSP	22
23	CNV_WGR_CLK+		
33	GND	CNV_BGI_DT	32
35	PETp	CNV_RGI_RSP	34
37	PETn	CNV_BRI_DT	36
39	GND	NA	38
41	PERp	NA	40
43	PERn	NA	42
45	GND	NA	44
47	PEFCLKp	NA	46
49	PEFCLKn	NA	48
51	GND	SUSCLK	50
53	CLKREQ#	PERST0#	52
55	WAKE#	W_DISABLE1#	54
57	GND	W_DISABLE2#	56
59	CNV_WT_D1-	SMB_DATA	58
61	CNV_WT_D1+	SMB_CLK	60
63	GND	NA	62
65	CNV_WT_D0-	CLKIN_XTAL_LCP	64
67	CNV_WT_D0+	NA	66
69	GND	NA	68
71	CNV_WT_CLK-	NA	70
73	CNV_WT_CLK+	+3.3V	72
75	GND	+3.3V	74

10: System Panel Header



COM Port Pin9 PWR Setting Jumpers

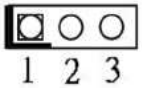
11 : PWR_COM7 (For COM Port7)
PWR_COM8 (For COM Port8)
PWR_COM9 (For COM Port9)
PWR_COM10 (For COM Port10)

23 : PWR_COM3 (For COM Port3)
PWR_COM4 (For COM Port4)
PWR_COM5 (For COM Port5)
PWR_COM6 (For COM Port6)

38 : PWR_COM1 (For COM Port1)
PWR_COM2 (For COM Port2)

1-2 : +5V

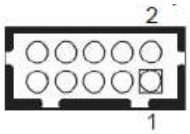
2-3 : +12V



COM Port Headers (COM3~10) (RS232)

12 : COM7, 8, 9, 10

22 : COM3, 4, 5, 6

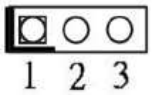


PIN	Signal Name	PIN	Signal Name	PIN	Signal Name	PIN	Signal Name	PIN	Signal Name
10	N/A	8	CCTS#	6	DDSR#	4	DDTR#	2	RRXD
9	+5V	7	RRTS#	5	GND	3	TTXD	1	DDCD#

13: ATX/AT Mode Jumper (PWR_JP1)

1-2： AT Mode

2-3： ATX Mode

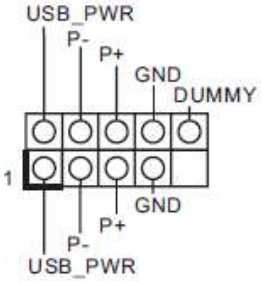


USB 2.0 Headers

14 : USB2_9_10

18 : USB2_7_8

20 : USB2_5_6

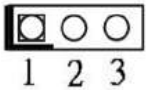


Clear CMOS Headers

15 : CLRMOS1 :

1-2： Normal

2-3： Clear CMOS



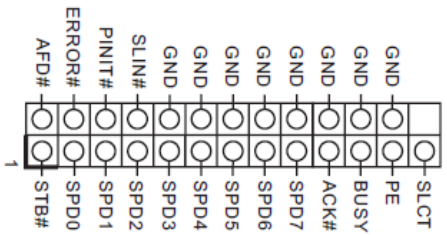
17 : CLRMOS2 :

Open： Normal

Short： Auto Clear CMOS (Power Off)

24: Printer Port / GPIO Header (LPT_GPIO1)

Printer Port：



GPIO：

PIN	Signal Name	PIN	Signal Name
26	NC	25	NA
24	GND	23	SIO_GP30
22	GND	21	SIO_GP31
20	GND	19	SIO_GP32
18	GND	17	SIO_GP33
16	GND	15	SIO_GP34
14	GND	13	SIO_GP35
12	JGPIOPWR	11	SIO_GP36
10	JGPIOPWR	9	SIO_GP37
8	SIO_GP43	7	SIO_GP40
6	SIO_GP44	5	SIO_GP41
4	SIO_GP45	3	SIO_GP42
2	SIO_GP46	1	SIO_GP47

* If you want to use the printer port function, please short pin4 and pin5 on Digital Input / Output Power Select (JGPIO_PWR1).

25: Digital Input / Output Power Select(JGPIOPWR) (JGPIO_PWR1)

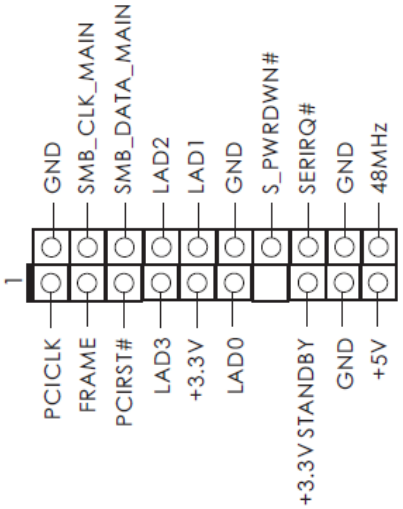
1-2 : +12V

2-3 : +5V

3-4 : +5V

4-5 : GND

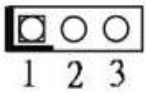
26: LPC Header



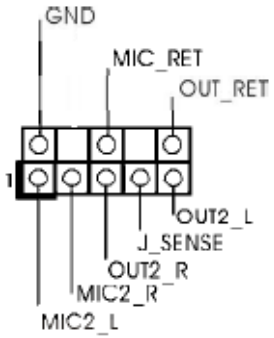
27: Digital Input / Output Default Value Setting (JGPIO_SET1)

1-2： Pull-High

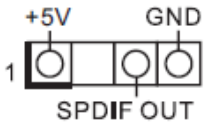
2-3： Pull-Low



28: Front Panel Audio Header



29: SPDIF Header



30: Chassis Intrusion Headers (CI1, CI2)

CI1：

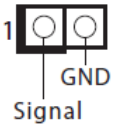
Close： Active Case Open

Open： Normal

CI2：

Close： Normal

Open： Active Case Open



31: ATX/AT Mode Jumper (SIO_AT1)

Open： ATX Mode

Short： AT Mode

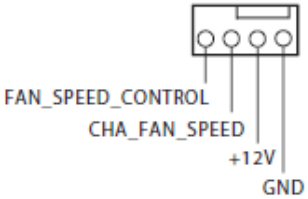
32: Audio Jacks

Blue - Line In

Green - Line Out

Pink - Mic In

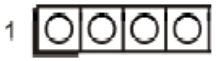
34: Chassis FAN Connector (+12V)



35: Top : RJ45 LAN Port (LAN2)
Bottom : USB 3.0 Ports (USB3_1_2)

36:LAN LED Headers :

LAN_LED1 (For LAN1 Port)
LAN_LED2 (For LAN2 Port)



37: Top : RJ45 LAN Port (LAN1)
Bottom : USB 3.0 Ports (USB3_3_4)

39: Top : COM Port (COM2) (RS232/422/485)*
Bottom : D-Sub Port (VGA1)

40: Top : COM Port (COM1) (RS232/422/485)*

Bottom Right : PS/2 Keyboard/Mouse Port

Bottom Left: HDMI Port (HDMI2)

* This motherboard supports RS232/422/485 on COM1, 2 ports. Please refer to below table for the pin definition. In addition, COM1, 2 ports (RS232/422/485) can be adjusted in BIOS setup utility > Advanced Screen > Super IO Configuration. You may refer to our user manual for details.

COM1, 2 Port Pin Definition

PIN	RS232	RS422	RS485
1	DCD, Data Carrier Detect	TX-	RTX-
2	RXD, Receive Data	RX+	N/A
3	TXD, Transmit Data	TX+	RTX+
4	DTR, Data Terminal Ready	RX-	N/A
5	GND	GND	GND
6	DSR, Data Set Ready	N/A	N/A
7	RTS, Request To Send	N/A	N/A
8	CTS, Clear To Send	N/A	N/A
9	No Power/5V/12V	N/A	N/A

41: Top : DisplayPort 1

Bottom : DisplayPort 2

5 Signal Descriptions

5.1 Watch Dog Signal

```
void WatchDogTest()
{
 bool bSuccess = false;
 int WDTimer = 30;
 printf("Please input WatchDog timer:");
 scanf("%d", &WDTimer);
 ShowError(bSuccess = AsrLibWDSetConfig(WDTimer));
 ShowError(bSuccess = AsrLibWDTrigger());
 char Key = 0;
 int CurrentTime = 0;
 int WaitSeconds = WDTimer;
 while (WaitSeconds) {
  ShowError(CurrentTime = AsrLibWDCounter());
  WaitSeconds++;
  while (1)
  {
   while (kbhit())
   {
    Key = _getch();
```

```
 if (Key == 'r')
 {
  AsrLibWDDisable();
  ShowError(bSuccess = AsrLibWDSetConfig(WDTimer));
  ShowError(bSuccess = AsrLibWDTrigger());
  WaitSeconds = WDTimer;
  ShowError(CurrentTime = AsrLibWDCounter());
  break;
 }
 else if (Key == 'c') {
  AsrLibWDDisable();
  WaitSeconds = 0;
  _tprintf(_T("\nWatchDog Disable"));
  break;
 }
}

if (WaitSeconds == 0)
 break;

if (CurrentTime != AsrLibWDCounter()) {
 WaitSeconds--;
 printf("\rWatchDog timer %d, press 'r' to reset timer, 'c' to disable WatchDog.", WaitSeconds);
```

```
      CurrentTime = AsrLibWDCounter();
     }
    }
    printf("\n");
  }
}
```

5.2 GPIO Signal

```
void GpioTest()
{
 bool bSuccess = false;
 _tprintf(_T("\n"));
 _tprintf(_T("Currnet state\n"));
 _tprintf(_T("--------------------------------------\n"));
 // Check AsrLibGetSioGpioGroup & AsrLibSetSioGpioGroup
 SSCORE_GPIO_VALUE GpioValue;
 ShowError(bSuccess = AsrLibGetSioGpioValue(0, &GpioValue));
 printf("PortPin=70\n");
 switch (GpioValue.Value)
 {
 case ESCORE_GPIO_NOT_GPIO:
  _tprintf(_T("ESCORE_GPIO_NOT_GPIO]\n"));

  break;
 case ESCORE_GPIO_DISABLED:
  _tprintf(_T("ESCORE_GPIO_DISABLED]\n"));
  break;
 case ESCORE_GPIO_INPUT_LOW:
  _tprintf(_T("ESCORE_GPIO_INPUT_LOW]\n"));
```

```
  break;
 case ESCORE_GPIO_INPUT_HIGH:
  _tprintf(_T("ESCORE_GPIO_INPUT_HIGH]\n"));
  break;
 case ESCORE_GPIO_OUTPUT_LOW:
  _tprintf(_T("ESCORE_GPIO_OUTPUT_LOW]\n"));
  break;
 case ESCORE_GPIO_OUTPUT_HIGH:
  _tprintf(_T("ESCORE_GPIO_OUTPUT_HIGH]\n"));
  break;
 default:
  _tprintf(_T("UNKNOWN VALUE[0x%02X]\n"), GpioValue.Value);
 }
 printf("\n");
 ShowError(bSuccess = AsrLibGetSioGpioValue(71, &GpioValue));
 printf("PortPin=71\n");
 switch (GpioValue.Value)
 {
 case ESCORE_GPIO_NOT_GPIO:
  _tprintf(_T("ESCORE_GPIO_NOT_GPIO]\n"));

  break;
 case ESCORE_GPIO_DISABLED:
```

```
  _tprintf(_T("ESCORE_GPIO_DISABLED]\n"));
  break;
 case ESCORE_GPIO_INPUT_LOW:
  _tprintf(_T("ESCORE_GPIO_INPUT_LOW]\n"));
  break;
 case ESCORE_GPIO_INPUT_HIGH:
  _tprintf(_T("ESCORE_GPIO_INPUT_HIGH]\n"));
  break;
 case ESCORE_GPIO_OUTPUT_LOW:
  _tprintf(_T("ESCORE_GPIO_OUTPUT_LOW]\n"));
  break;
 case ESCORE_GPIO_OUTPUT_HIGH:
  _tprintf(_T("ESCORE_GPIO_OUTPUT_HIGH]\n"));
  break;
 default:
  _tprintf(_T("UNKNOWN VALUE[0x%02X]\n"), GpioValue.Value);
 }
 printf("\n");
 ShowError(bSuccess = AsrLibGetSioGpioValue(72, &GpioValue));
 printf("PortPin=72\n");
 switch (GpioValue.Value)
 {
 case ESCORE_GPIO_NOT_GPIO:
```

```
  _tprintf(_T("ESCORE_GPIO_NOT_GPIO]\n"));

  break;
 case ESCORE_GPIO_DISABLED:
  _tprintf(_T("ESCORE_GPIO_DISABLED]\n"));
  break;
 case ESCORE_GPIO_INPUT_LOW:
  _tprintf(_T("ESCORE_GPIO_INPUT_LOW]\n"));
  break;
 case ESCORE_GPIO_INPUT_HIGH:
  _tprintf(_T("ESCORE_GPIO_INPUT_HIGH]\n"));
  break;
 case ESCORE_GPIO_OUTPUT_LOW:
  _tprintf(_T("ESCORE_GPIO_OUTPUT_LOW]\n"));
  break;
 case ESCORE_GPIO_OUTPUT_HIGH:
  _tprintf(_T("ESCORE_GPIO_OUTPUT_HIGH]\n"));
  break;
 default:
  _tprintf(_T("UNKNOWN VALUE[0x%02X]\n"), GpioValue.Value);
 }
 printf("\n");
 ShowError(bSuccess = AsrLibGetSioGpioValue(73, &GpioValue));
```

```
printf("PortPin=73\n");
switch (GpioValue.Value)
{
case ESCORE_GPIO_NOT_GPIO:
  _tprintf(_T("ESCORE_GPIO_NOT_GPIO]\n"));

  break;
case ESCORE_GPIO_DISABLED:
  _tprintf(_T("ESCORE_GPIO_DISABLED]\n"));
  break;
case ESCORE_GPIO_INPUT_LOW:
  _tprintf(_T("ESCORE_GPIO_INPUT_LOW]\n"));
  break;
case ESCORE_GPIO_INPUT_HIGH:
  _tprintf(_T("ESCORE_GPIO_INPUT_HIGH]\n"));
  break;
case ESCORE_GPIO_OUTPUT_LOW:
  _tprintf(_T("ESCORE_GPIO_OUTPUT_LOW]\n"));
  break;
case ESCORE_GPIO_OUTPUT_HIGH:
  _tprintf(_T("ESCORE_GPIO_OUTPUT_HIGH]\n"));
  break;
default:
```

```
  _tprintf(_T("UNKNOWN VALUE[0x%02X]\n"), GpioValue.Value);
 }
 printf("\n");
 ShowError(bSuccess = AsrLibGetSioGpioValue(74, &GpioValue));
 printf("PortPin=74\n");
 switch (GpioValue.Value)
 {
 case ESCORE_GPIO_NOT_GPIO:
  _tprintf(_T("ESCORE_GPIO_NOT_GPIO]\n"));

  break;
 case ESCORE_GPIO_DISABLED:
  _tprintf(_T("ESCORE_GPIO_DISABLED]\n"));
  break;
 case ESCORE_GPIO_INPUT_LOW:
  _tprintf(_T("ESCORE_GPIO_INPUT_LOW]\n"));
  break;
 case ESCORE_GPIO_INPUT_HIGH:
  _tprintf(_T("ESCORE_GPIO_INPUT_HIGH]\n"));
  break;
 case ESCORE_GPIO_OUTPUT_LOW:
  _tprintf(_T("ESCORE_GPIO_OUTPUT_LOW]\n"));
  break;
```

```
case ESCORE_GPIO_OUTPUT_HIGH:
  _tprintf(_T("ESCORE_GPIO_OUTPUT_HIGH]\n"));
  break;
default:
  _tprintf(_T("UNKNOWN VALUE[0x%02X]\n"), GpioValue.Value);
}
printf("\n");
ShowError(bSuccess = AsrLibGetSioGpioValue(75, &GpioValue));
printf("PortPin=75\n");
switch (GpioValue.Value)
{
case ESCORE_GPIO_NOT_GPIO:
  _tprintf(_T("ESCORE_GPIO_NOT_GPIO]\n"));

  break;
case ESCORE_GPIO_DISABLED:
  _tprintf(_T("ESCORE_GPIO_DISABLED]\n"));
  break;
case ESCORE_GPIO_INPUT_LOW:
  _tprintf(_T("ESCORE_GPIO_INPUT_LOW]\n"));
  break;
case ESCORE_GPIO_INPUT_HIGH:
  _tprintf(_T("ESCORE_GPIO_INPUT_HIGH]\n"));
```

```
  break;
 case ESCORE_GPIO_OUTPUT_LOW:
  _tprintf(_T("ESCORE_GPIO_OUTPUT_LOW]\n"));
  break;
 case ESCORE_GPIO_OUTPUT_HIGH:
  _tprintf(_T("ESCORE_GPIO_OUTPUT_HIGH]\n"));
  break;
 default:
  _tprintf(_T("UNKNOWN VALUE[0x%02X]\n"), GpioValue.Value);
 }
 printf("\n");
 ShowError(bSuccess = AsrLibGetSioGpioValue(76, &GpioValue));
 printf("PortPin=76\n");
 switch (GpioValue.Value)
 {
 case ESCORE_GPIO_NOT_GPIO:
  _tprintf(_T("ESCORE_GPIO_NOT_GPIO]\n"));

  break;
 case ESCORE_GPIO_DISABLED:
  _tprintf(_T("ESCORE_GPIO_DISABLED]\n"));
  break;
 case ESCORE_GPIO_INPUT_LOW:
```

```
  _tprintf(_T("ESCORE_GPIO_INPUT_LOW]\n"));
  break;
 case ESCORE_GPIO_INPUT_HIGH:
  _tprintf(_T("ESCORE_GPIO_INPUT_HIGH]\n"));
  break;
 case ESCORE_GPIO_OUTPUT_LOW:
  _tprintf(_T("ESCORE_GPIO_OUTPUT_LOW]\n"));
  break;
 case ESCORE_GPIO_OUTPUT_HIGH:
  _tprintf(_T("ESCORE_GPIO_OUTPUT_HIGH]\n"));
  break;
 default:
  _tprintf(_T("UNKNOWN VALUE[0x%02X]\n"), GpioValue.Value);
 }
 printf("\n");
 ShowError(bSuccess = AsrLibGetSioGpioValue(77, &GpioValue));
 printf("PortPin=77\n");
 switch (GpioValue.Value)
 {
 case ESCORE_GPIO_NOT_GPIO:
  _tprintf(_T("ESCORE_GPIO_NOT_GPIO]\n"));

  break;
```

```
case ESCORE_GPIO_DISABLED:
  _tprintf(_T("ESCORE_GPIO_DISABLED]\n"));
  break;
case ESCORE_GPIO_INPUT_LOW:
  _tprintf(_T("ESCORE_GPIO_INPUT_LOW]\n"));
  break;
case ESCORE_GPIO_INPUT_HIGH:
  _tprintf(_T("ESCORE_GPIO_INPUT_HIGH]\n"));
  break;
case ESCORE_GPIO_OUTPUT_LOW:
  _tprintf(_T("ESCORE_GPIO_OUTPUT_LOW]\n"));
  break;
case ESCORE_GPIO_OUTPUT_HIGH:
  _tprintf(_T("ESCORE_GPIO_OUTPUT_HIGH]\n"));
  break;
default:
  _tprintf(_T("UNKNOWN VALUE[0x%02X]\n"), GpioValue.Value);
}
printf("\n");
ShowError(bSuccess = AsrLibGetSioGpioValue(80, &GpioValue));
printf("PortPin=80\n");
switch (GpioValue.Value)
{
```

```
case ESCORE_GPIO_NOT_GPIO:
  _tprintf(_T("ESCORE_GPIO_NOT_GPIO]\n"));

  break;
case ESCORE_GPIO_DISABLED:
  _tprintf(_T("ESCORE_GPIO_DISABLED]\n"));
  break;
case ESCORE_GPIO_INPUT_LOW:
  _tprintf(_T("ESCORE_GPIO_INPUT_LOW]\n"));
  break;
case ESCORE_GPIO_INPUT_HIGH:
  _tprintf(_T("ESCORE_GPIO_INPUT_HIGH]\n"));
  break;
case ESCORE_GPIO_OUTPUT_LOW:
  _tprintf(_T("ESCORE_GPIO_OUTPUT_LOW]\n"));
  break;
case ESCORE_GPIO_OUTPUT_HIGH:
  _tprintf(_T("ESCORE_GPIO_OUTPUT_HIGH]\n"));
  break;
default:
  _tprintf(_T("UNKNOWN VALUE[0x%02X]\n"), GpioValue.Value);
}
printf("\n");
```

```
ShowError(bSuccess = AsrLibGetSioGpioValue(81, &GpioValue));
printf("PortPin=81\n");
switch (GpioValue.Value)
{
case ESCORE_GPIO_NOT_GPIO:
  _tprintf(_T("ESCORE_GPIO_NOT_GPIO]\n"));

  break;
case ESCORE_GPIO_DISABLED:
  _tprintf(_T("ESCORE_GPIO_DISABLED]\n"));
  break;
case ESCORE_GPIO_INPUT_LOW:
  _tprintf(_T("ESCORE_GPIO_INPUT_LOW]\n"));
  break;
case ESCORE_GPIO_INPUT_HIGH:
  _tprintf(_T("ESCORE_GPIO_INPUT_HIGH]\n"));
  break;
case ESCORE_GPIO_OUTPUT_LOW:
  _tprintf(_T("ESCORE_GPIO_OUTPUT_LOW]\n"));
  break;
case ESCORE_GPIO_OUTPUT_HIGH:
  _tprintf(_T("ESCORE_GPIO_OUTPUT_HIGH]\n"));
  break;
```

```
default:
  _tprintf(_T("UNKNOWN VALUE[0x%02X]\n"), GpioValue.Value);
}
printf("\n");
ShowError(bSuccess = AsrLibGetSioGpioValue(82, &GpioValue));
printf("PortPin=82\n");
switch (GpioValue.Value)
{
case ESCORE_GPIO_NOT_GPIO:
  _tprintf(_T("ESCORE_GPIO_NOT_GPIO]\n"));

  break;
case ESCORE_GPIO_DISABLED:
  _tprintf(_T("ESCORE_GPIO_DISABLED]\n"));
  break;
case ESCORE_GPIO_INPUT_LOW:
  _tprintf(_T("ESCORE_GPIO_INPUT_LOW]\n"));
  break;
case ESCORE_GPIO_INPUT_HIGH:
  _tprintf(_T("ESCORE_GPIO_INPUT_HIGH]\n"));
  break;
case ESCORE_GPIO_OUTPUT_LOW:
  _tprintf(_T("ESCORE_GPIO_OUTPUT_LOW]\n"));
```

```
  break;
 case ESCORE_GPIO_OUTPUT_HIGH:
  _tprintf(_T("ESCORE_GPIO_OUTPUT_HIGH]\n"));
  break;
 default:
  _tprintf(_T("UNKNOWN VALUE[0x%02X]\n"), GpioValue.Value);
 }
 printf("\n");
 ShowError(bSuccess = AsrLibGetSioGpioValue(83, &GpioValue));
 printf("PortPin=83\n");
 switch (GpioValue.Value)
 {
 case ESCORE_GPIO_NOT_GPIO:
  _tprintf(_T("ESCORE_GPIO_NOT_GPIO]\n"));

  break;
 case ESCORE_GPIO_DISABLED:
  _tprintf(_T("ESCORE_GPIO_DISABLED]\n"));
  break;
 case ESCORE_GPIO_INPUT_LOW:
  _tprintf(_T("ESCORE_GPIO_INPUT_LOW]\n"));
  break;
 case ESCORE_GPIO_INPUT_HIGH:
```

```
  _tprintf(_T("ESCORE_GPIO_INPUT_HIGH]\n"));
  break;
 case ESCORE_GPIO_OUTPUT_LOW:
  _tprintf(_T("ESCORE_GPIO_OUTPUT_LOW]\n"));
  break;
 case ESCORE_GPIO_OUTPUT_HIGH:
  _tprintf(_T("ESCORE_GPIO_OUTPUT_HIGH]\n"));
  break;
 default:
  _tprintf(_T("UNKNOWN VALUE[0x%02X]\n"), GpioValue.Value);
 }
 printf("\n");
 ShowError(bSuccess = AsrLibGetSioGpioValue(84, &GpioValue));
 printf("PortPin=84\n");
 switch (GpioValue.Value)
 {
 case ESCORE_GPIO_NOT_GPIO:
  _tprintf(_T("ESCORE_GPIO_NOT_GPIO]\n"));

  break;
 case ESCORE_GPIO_DISABLED:
  _tprintf(_T("ESCORE_GPIO_DISABLED]\n"));
  break;
```

```
case ESCORE_GPIO_INPUT_LOW:
  _tprintf(_T("ESCORE_GPIO_INPUT_LOW]\n"));
  break;
case ESCORE_GPIO_INPUT_HIGH:
  _tprintf(_T("ESCORE_GPIO_INPUT_HIGH]\n"));
  break;
case ESCORE_GPIO_OUTPUT_LOW:
  _tprintf(_T("ESCORE_GPIO_OUTPUT_LOW]\n"));
  break;
case ESCORE_GPIO_OUTPUT_HIGH:
  _tprintf(_T("ESCORE_GPIO_OUTPUT_HIGH]\n"));
  break;
default:
  _tprintf(_T("UNKNOWN VALUE[0x%02X]\n"), GpioValue.Value);
}
printf("\n");
ShowError(bSuccess = AsrLibGetSioGpioValue(85, &GpioValue));
printf("PortPin=85\n");
switch (GpioValue.Value)
{
case ESCORE_GPIO_NOT_GPIO:
  _tprintf(_T("ESCORE_GPIO_NOT_GPIO]\n"));
```

```
  break;
 case ESCORE_GPIO_DISABLED:
  _tprintf(_T("ESCORE_GPIO_DISABLED]\n"));
  break;
 case ESCORE_GPIO_INPUT_LOW:
  _tprintf(_T("ESCORE_GPIO_INPUT_LOW]\n"));
  break;
 case ESCORE_GPIO_INPUT_HIGH:
  _tprintf(_T("ESCORE_GPIO_INPUT_HIGH]\n"));
  break;
 case ESCORE_GPIO_OUTPUT_LOW:
  _tprintf(_T("ESCORE_GPIO_OUTPUT_LOW]\n"));
  break;
 case ESCORE_GPIO_OUTPUT_HIGH:
  _tprintf(_T("ESCORE_GPIO_OUTPUT_HIGH]\n"));
  break;
 default:
  _tprintf(_T("UNKNOWN VALUE[0x%02X]\n"), GpioValue.Value);
 }
 printf("\n");
 ShowError(bSuccess = AsrLibGetSioGpioValue(86, &GpioValue));
 printf("PortPin=86\n");
 switch (GpioValue.Value)
```

```
{
 case ESCORE_GPIO_NOT_GPIO:
  _tprintf(_T("ESCORE_GPIO_NOT_GPIO]\n"));

  break;
 case ESCORE_GPIO_DISABLED:
  _tprintf(_T("ESCORE_GPIO_DISABLED]\n"));
  break;
 case ESCORE_GPIO_INPUT_LOW:
  _tprintf(_T("ESCORE_GPIO_INPUT_LOW]\n"));
  break;
 case ESCORE_GPIO_INPUT_HIGH:
  _tprintf(_T("ESCORE_GPIO_INPUT_HIGH]\n"));
  break;
 case ESCORE_GPIO_OUTPUT_LOW:
  _tprintf(_T("ESCORE_GPIO_OUTPUT_LOW]\n"));
  break;
 case ESCORE_GPIO_OUTPUT_HIGH:
  _tprintf(_T("ESCORE_GPIO_OUTPUT_HIGH]\n"));
  break;
 default:
  _tprintf(_T("UNKNOWN VALUE[0x%02X]\n"), GpioValue.Value);
}
```

```
printf("\n");
ShowError(bSuccess = AsrLibGetSioGpioValue(87, &GpioValue));
printf("PortPin=87\n");
switch (GpioValue.Value)
{
case ESCORE_GPIO_NOT_GPIO:
  _tprintf(_T("ESCORE_GPIO_NOT_GPIO]\n"));

  break;
case ESCORE_GPIO_DISABLED:
  _tprintf(_T("ESCORE_GPIO_DISABLED]\n"));
  break;
case ESCORE_GPIO_INPUT_LOW:
  _tprintf(_T("ESCORE_GPIO_INPUT_LOW]\n"));
  break;
case ESCORE_GPIO_INPUT_HIGH:
  _tprintf(_T("ESCORE_GPIO_INPUT_HIGH]\n"));
  break;
case ESCORE_GPIO_OUTPUT_LOW:
  _tprintf(_T("ESCORE_GPIO_OUTPUT_LOW]\n"));
  break;
case ESCORE_GPIO_OUTPUT_HIGH:
  _tprintf(_T("ESCORE_GPIO_OUTPUT_HIGH]\n"));
```

```
  break;
 default:
  _tprintf(_T("UNKNOWN VALUE[0x%02X]\n"), GpioValue.Value);
 }
 printf("\n");

 int GPIOType = 0;
 printf("Input GPIOType:(1:GPIO_INPUT 2:GPIO_OUTPUT_LOW 3:GPIO_OUTPUT_HIGH)");
 scanf("%d", &GPIOType);

 if (GPIOType == 1) {
  ShowError(bSuccess = AsrLibSetSioGpioValue(70, ESCORE_GPIO_INPUT_LOW));
  ShowError(bSuccess = AsrLibSetSioGpioValue(71, ESCORE_GPIO_INPUT_LOW));
  ShowError(bSuccess = AsrLibSetSioGpioValue(72, ESCORE_GPIO_INPUT_LOW));
  ShowError(bSuccess = AsrLibSetSioGpioValue(73, ESCORE_GPIO_INPUT_LOW));
  ShowError(bSuccess = AsrLibSetSioGpioValue(74, ESCORE_GPIO_INPUT_LOW));
  ShowError(bSuccess = AsrLibSetSioGpioValue(75, ESCORE_GPIO_INPUT_LOW));
  ShowError(bSuccess = AsrLibSetSioGpioValue(76, ESCORE_GPIO_INPUT_LOW));
  ShowError(bSuccess = AsrLibSetSioGpioValue(77, ESCORE_GPIO_INPUT_LOW));
  ShowError(bSuccess = AsrLibSetSioGpioValue(80, ESCORE_GPIO_INPUT_LOW));
  ShowError(bSuccess = AsrLibSetSioGpioValue(81, ESCORE_GPIO_INPUT_LOW));
  ShowError(bSuccess = AsrLibSetSioGpioValue(82, ESCORE_GPIO_INPUT_LOW));
  ShowError(bSuccess = AsrLibSetSioGpioValue(83, ESCORE_GPIO_INPUT_LOW));
```

```
  ShowError(bSuccess = AsrLibSetSioGpioValue(84, ESCORE_GPIO_INPUT_LOW));
  ShowError(bSuccess = AsrLibSetSioGpioValue(85, ESCORE_GPIO_INPUT_LOW));
  ShowError(bSuccess = AsrLibSetSioGpioValue(86, ESCORE_GPIO_INPUT_LOW));
  ShowError(bSuccess = AsrLibSetSioGpioValue(87, ESCORE_GPIO_INPUT_LOW));
} else if (GPIOType == 2) {
  ShowError(bSuccess = AsrLibSetSioGpioValue(70, ESCORE_GPIO_OUTPUT_LOW));
  ShowError(bSuccess = AsrLibSetSioGpioValue(71, ESCORE_GPIO_OUTPUT_LOW));
  ShowError(bSuccess = AsrLibSetSioGpioValue(72, ESCORE_GPIO_OUTPUT_LOW));
  ShowError(bSuccess = AsrLibSetSioGpioValue(73, ESCORE_GPIO_OUTPUT_LOW));
  ShowError(bSuccess = AsrLibSetSioGpioValue(74, ESCORE_GPIO_OUTPUT_LOW));
  ShowError(bSuccess = AsrLibSetSioGpioValue(75, ESCORE_GPIO_OUTPUT_LOW));
  ShowError(bSuccess = AsrLibSetSioGpioValue(76, ESCORE_GPIO_OUTPUT_LOW));
  ShowError(bSuccess = AsrLibSetSioGpioValue(77, ESCORE_GPIO_OUTPUT_LOW));
  ShowError(bSuccess = AsrLibSetSioGpioValue(80, ESCORE_GPIO_OUTPUT_LOW));
  ShowError(bSuccess = AsrLibSetSioGpioValue(81, ESCORE_GPIO_OUTPUT_LOW));
  ShowError(bSuccess = AsrLibSetSioGpioValue(82, ESCORE_GPIO_OUTPUT_LOW));
  ShowError(bSuccess = AsrLibSetSioGpioValue(83, ESCORE_GPIO_OUTPUT_LOW));
  ShowError(bSuccess = AsrLibSetSioGpioValue(84, ESCORE_GPIO_OUTPUT_LOW));
  ShowError(bSuccess = AsrLibSetSioGpioValue(85, ESCORE_GPIO_OUTPUT_LOW));
  ShowError(bSuccess = AsrLibSetSioGpioValue(86, ESCORE_GPIO_OUTPUT_LOW));
  ShowError(bSuccess = AsrLibSetSioGpioValue(87, ESCORE_GPIO_OUTPUT_LOW));
} else if (GPIOType == 3) {
  ShowError(bSuccess = AsrLibSetSioGpioValue(70, ESCORE_GPIO_OUTPUT_HIGH));
```

```
  ShowError(bSuccess = AsrLibSetSioGpioValue(71, ESCORE_GPIO_OUTPUT_HIGH));
  ShowError(bSuccess = AsrLibSetSioGpioValue(72, ESCORE_GPIO_OUTPUT_HIGH));
  ShowError(bSuccess = AsrLibSetSioGpioValue(73, ESCORE_GPIO_OUTPUT_HIGH));
  ShowError(bSuccess = AsrLibSetSioGpioValue(74, ESCORE_GPIO_OUTPUT_HIGH));
  ShowError(bSuccess = AsrLibSetSioGpioValue(75, ESCORE_GPIO_OUTPUT_HIGH));
  ShowError(bSuccess = AsrLibSetSioGpioValue(76, ESCORE_GPIO_OUTPUT_HIGH));
  ShowError(bSuccess = AsrLibSetSioGpioValue(77, ESCORE_GPIO_OUTPUT_HIGH));
  ShowError(bSuccess = AsrLibSetSioGpioValue(80, ESCORE_GPIO_OUTPUT_HIGH));
  ShowError(bSuccess = AsrLibSetSioGpioValue(81, ESCORE_GPIO_OUTPUT_HIGH));
  ShowError(bSuccess = AsrLibSetSioGpioValue(82, ESCORE_GPIO_OUTPUT_HIGH));
  ShowError(bSuccess = AsrLibSetSioGpioValue(83, ESCORE_GPIO_OUTPUT_HIGH));
  ShowError(bSuccess = AsrLibSetSioGpioValue(84, ESCORE_GPIO_OUTPUT_HIGH));
  ShowError(bSuccess = AsrLibSetSioGpioValue(85, ESCORE_GPIO_OUTPUT_HIGH));
  ShowError(bSuccess = AsrLibSetSioGpioValue(86, ESCORE_GPIO_OUTPUT_HIGH));
  ShowError(bSuccess = AsrLibSetSioGpioValue(87, ESCORE_GPIO_OUTPUT_HIGH));
} else {
  ShowError(bSuccess = AsrLibSetSioGpioValue(70, ESCORE_GPIO_OUTPUT_HIGH));
  ShowError(bSuccess = AsrLibSetSioGpioValue(71, ESCORE_GPIO_OUTPUT_LOW));
  ShowError(bSuccess = AsrLibSetSioGpioValue(72, ESCORE_GPIO_OUTPUT_HIGH));
  ShowError(bSuccess = AsrLibSetSioGpioValue(73, ESCORE_GPIO_OUTPUT_LOW));
  ShowError(bSuccess = AsrLibSetSioGpioValue(74, ESCORE_GPIO_OUTPUT_HIGH));
  ShowError(bSuccess = AsrLibSetSioGpioValue(75, ESCORE_GPIO_OUTPUT_LOW));
  ShowError(bSuccess = AsrLibSetSioGpioValue(76, ESCORE_GPIO_OUTPUT_HIGH));
```

```
  ShowError(bSuccess = AsrLibSetSioGpioValue(77, ESCORE_GPIO_OUTPUT_LOW));
  ShowError(bSuccess = AsrLibSetSioGpioValue(80, ESCORE_GPIO_OUTPUT_HIGH));
  ShowError(bSuccess = AsrLibSetSioGpioValue(81, ESCORE_GPIO_OUTPUT_LOW));
  ShowError(bSuccess = AsrLibSetSioGpioValue(82, ESCORE_GPIO_OUTPUT_HIGH));
  ShowError(bSuccess = AsrLibSetSioGpioValue(83, ESCORE_GPIO_OUTPUT_LOW));
  ShowError(bSuccess = AsrLibSetSioGpioValue(84, ESCORE_GPIO_OUTPUT_HIGH));
  ShowError(bSuccess = AsrLibSetSioGpioValue(85, ESCORE_GPIO_OUTPUT_LOW));
  ShowError(bSuccess = AsrLibSetSioGpioValue(86, ESCORE_GPIO_OUTPUT_HIGH));
  ShowError(bSuccess = AsrLibSetSioGpioValue(87, ESCORE_GPIO_OUTPUT_LOW));
}

_tprintf(_T("New state\n"));
_tprintf(_T("-------------------------------------\n"));

ShowError(bSuccess = AsrLibGetSioGpioValue(70, &GpioValue));
printf("PortPin=70\n");
switch (GpioValue.Value)
{
case ESCORE_GPIO_NOT_GPIO:
  _tprintf(_T("ESCORE_GPIO_NOT_GPIO]\n"));

  break;
case ESCORE_GPIO_DISABLED:
```

```
  _tprintf(_T("ESCORE_GPIO_DISABLED]\n"));
  break;
 case ESCORE_GPIO_INPUT_LOW:
  _tprintf(_T("ESCORE_GPIO_INPUT_LOW]\n"));
  break;
 case ESCORE_GPIO_INPUT_HIGH:
  _tprintf(_T("ESCORE_GPIO_INPUT_HIGH]\n"));
  break;
 case ESCORE_GPIO_OUTPUT_LOW:
  _tprintf(_T("ESCORE_GPIO_OUTPUT_LOW]\n"));
  break;
 case ESCORE_GPIO_OUTPUT_HIGH:
  _tprintf(_T("ESCORE_GPIO_OUTPUT_HIGH]\n"));
  break;
 default:
  _tprintf(_T("UNKNOWN VALUE[0x%02X]\n"), GpioValue.Value);
 }
 printf("\n");
 ShowError(bSuccess = AsrLibGetSioGpioValue(71, &GpioValue));
 printf("PortPin=71\n");
 switch (GpioValue.Value)
 {
 case ESCORE_GPIO_NOT_GPIO:
```

```
  _tprintf(_T("ESCORE_GPIO_NOT_GPIO]\n"));

  break;
 case ESCORE_GPIO_DISABLED:
  _tprintf(_T("ESCORE_GPIO_DISABLED]\n"));
  break;
 case ESCORE_GPIO_INPUT_LOW:
  _tprintf(_T("ESCORE_GPIO_INPUT_LOW]\n"));
  break;
 case ESCORE_GPIO_INPUT_HIGH:
  _tprintf(_T("ESCORE_GPIO_INPUT_HIGH]\n"));
  break;
 case ESCORE_GPIO_OUTPUT_LOW:
  _tprintf(_T("ESCORE_GPIO_OUTPUT_LOW]\n"));
  break;
 case ESCORE_GPIO_OUTPUT_HIGH:
  _tprintf(_T("ESCORE_GPIO_OUTPUT_HIGH]\n"));
  break;
 default:
  _tprintf(_T("UNKNOWN VALUE[0x%02X]\n"), GpioValue.Value);
 }
 printf("\n");
 ShowError(bSuccess = AsrLibGetSioGpioValue(72, &GpioValue));
```

```
printf("PortPin=72\n");
switch (GpioValue.Value)
{
case ESCORE_GPIO_NOT_GPIO:
  _tprintf(_T("ESCORE_GPIO_NOT_GPIO]\n"));

  break;
case ESCORE_GPIO_DISABLED:
  _tprintf(_T("ESCORE_GPIO_DISABLED]\n"));
  break;
case ESCORE_GPIO_INPUT_LOW:
  _tprintf(_T("ESCORE_GPIO_INPUT_LOW]\n"));
  break;
case ESCORE_GPIO_INPUT_HIGH:
  _tprintf(_T("ESCORE_GPIO_INPUT_HIGH]\n"));
  break;
case ESCORE_GPIO_OUTPUT_LOW:
  _tprintf(_T("ESCORE_GPIO_OUTPUT_LOW]\n"));
  break;
case ESCORE_GPIO_OUTPUT_HIGH:
  _tprintf(_T("ESCORE_GPIO_OUTPUT_HIGH]\n"));
  break;
default:
```

```
  _tprintf(_T("UNKNOWN VALUE[0x%02X]\n"), GpioValue.Value);
 }
 printf("\n");
 ShowError(bSuccess = AsrLibGetSioGpioValue(73, &GpioValue));
 printf("PortPin=73\n");
 switch (GpioValue.Value)
 {
 case ESCORE_GPIO_NOT_GPIO:
  _tprintf(_T("ESCORE_GPIO_NOT_GPIO]\n"));

  break;
 case ESCORE_GPIO_DISABLED:
  _tprintf(_T("ESCORE_GPIO_DISABLED]\n"));
  break;
 case ESCORE_GPIO_INPUT_LOW:
  _tprintf(_T("ESCORE_GPIO_INPUT_LOW]\n"));
  break;
 case ESCORE_GPIO_INPUT_HIGH:
  _tprintf(_T("ESCORE_GPIO_INPUT_HIGH]\n"));
  break;
 case ESCORE_GPIO_OUTPUT_LOW:
  _tprintf(_T("ESCORE_GPIO_OUTPUT_LOW]\n"));
  break;
```

```
case ESCORE_GPIO_OUTPUT_HIGH:
  _tprintf(_T("ESCORE_GPIO_OUTPUT_HIGH]\n"));
  break;
default:
  _tprintf(_T("UNKNOWN VALUE[0x%02X]\n"), GpioValue.Value);
}
printf("\n");
ShowError(bSuccess = AsrLibGetSioGpioValue(74, &GpioValue));
printf("PortPin=74\n");
switch (GpioValue.Value)
{
case ESCORE_GPIO_NOT_GPIO:
  _tprintf(_T("ESCORE_GPIO_NOT_GPIO]\n"));

  break;
case ESCORE_GPIO_DISABLED:
  _tprintf(_T("ESCORE_GPIO_DISABLED]\n"));
  break;
case ESCORE_GPIO_INPUT_LOW:
  _tprintf(_T("ESCORE_GPIO_INPUT_LOW]\n"));
  break;
case ESCORE_GPIO_INPUT_HIGH:
  _tprintf(_T("ESCORE_GPIO_INPUT_HIGH]\n"));
```

```
  break;
 case ESCORE_GPIO_OUTPUT_LOW:
  _tprintf(_T("ESCORE_GPIO_OUTPUT_LOW]\n"));
  break;
 case ESCORE_GPIO_OUTPUT_HIGH:
  _tprintf(_T("ESCORE_GPIO_OUTPUT_HIGH]\n"));
  break;
 default:
  _tprintf(_T("UNKNOWN VALUE[0x%02X]\n"), GpioValue.Value);
 }
 printf("\n");
 ShowError(bSuccess = AsrLibGetSioGpioValue(75, &GpioValue));
 printf("PortPin=75\n");
 switch (GpioValue.Value)
 {
 case ESCORE_GPIO_NOT_GPIO:
  _tprintf(_T("ESCORE_GPIO_NOT_GPIO]\n"));

  break;
 case ESCORE_GPIO_DISABLED:
  _tprintf(_T("ESCORE_GPIO_DISABLED]\n"));
  break;
 case ESCORE_GPIO_INPUT_LOW:
```

```
  _tprintf(_T("ESCORE_GPIO_INPUT_LOW]\n"));
  break;
 case ESCORE_GPIO_INPUT_HIGH:
  _tprintf(_T("ESCORE_GPIO_INPUT_HIGH]\n"));
  break;
 case ESCORE_GPIO_OUTPUT_LOW:
  _tprintf(_T("ESCORE_GPIO_OUTPUT_LOW]\n"));
  break;
 case ESCORE_GPIO_OUTPUT_HIGH:
  _tprintf(_T("ESCORE_GPIO_OUTPUT_HIGH]\n"));
  break;
 default:
  _tprintf(_T("UNKNOWN VALUE[0x%02X]\n"), GpioValue.Value);
 }
 printf("\n");
 ShowError(bSuccess = AsrLibGetSioGpioValue(76, &GpioValue));
 printf("PortPin=76\n");
 switch (GpioValue.Value)
 {
 case ESCORE_GPIO_NOT_GPIO:
  _tprintf(_T("ESCORE_GPIO_NOT_GPIO]\n"));

  break;
```

```
case ESCORE_GPIO_DISABLED:
  _tprintf(_T("ESCORE_GPIO_DISABLED]\n"));
  break;
case ESCORE_GPIO_INPUT_LOW:
  _tprintf(_T("ESCORE_GPIO_INPUT_LOW]\n"));
  break;
case ESCORE_GPIO_INPUT_HIGH:
  _tprintf(_T("ESCORE_GPIO_INPUT_HIGH]\n"));
  break;
case ESCORE_GPIO_OUTPUT_LOW:
  _tprintf(_T("ESCORE_GPIO_OUTPUT_LOW]\n"));
  break;
case ESCORE_GPIO_OUTPUT_HIGH:
  _tprintf(_T("ESCORE_GPIO_OUTPUT_HIGH]\n"));
  break;
default:
  _tprintf(_T("UNKNOWN VALUE[0x%02X]\n"), GpioValue.Value);
}
printf("\n");
ShowError(bSuccess = AsrLibGetSioGpioValue(77, &GpioValue));
printf("PortPin=77\n");
switch (GpioValue.Value)
{
```

```
case ESCORE_GPIO_NOT_GPIO:
  _tprintf(_T("ESCORE_GPIO_NOT_GPIO]\n"));

  break;
case ESCORE_GPIO_DISABLED:
  _tprintf(_T("ESCORE_GPIO_DISABLED]\n"));
  break;
case ESCORE_GPIO_INPUT_LOW:
  _tprintf(_T("ESCORE_GPIO_INPUT_LOW]\n"));
  break;
case ESCORE_GPIO_INPUT_HIGH:
  _tprintf(_T("ESCORE_GPIO_INPUT_HIGH]\n"));
  break;
case ESCORE_GPIO_OUTPUT_LOW:
  _tprintf(_T("ESCORE_GPIO_OUTPUT_LOW]\n"));
  break;
case ESCORE_GPIO_OUTPUT_HIGH:
  _tprintf(_T("ESCORE_GPIO_OUTPUT_HIGH]\n"));
  break;
default:
  _tprintf(_T("UNKNOWN VALUE[0x%02X]\n"), GpioValue.Value);
}
printf("\n");
```

```
ShowError(bSuccess = AsrLibGetSioGpioValue(80, &GpioValue));
printf("PortPin=80\n");
switch (GpioValue.Value)
{
case ESCORE_GPIO_NOT_GPIO:
  _tprintf(_T("ESCORE_GPIO_NOT_GPIO]\n"));

  break;
case ESCORE_GPIO_DISABLED:
  _tprintf(_T("ESCORE_GPIO_DISABLED]\n"));
  break;
case ESCORE_GPIO_INPUT_LOW:
  _tprintf(_T("ESCORE_GPIO_INPUT_LOW]\n"));
  break;
case ESCORE_GPIO_INPUT_HIGH:
  _tprintf(_T("ESCORE_GPIO_INPUT_HIGH]\n"));
  break;
case ESCORE_GPIO_OUTPUT_LOW:
  _tprintf(_T("ESCORE_GPIO_OUTPUT_LOW]\n"));
  break;
case ESCORE_GPIO_OUTPUT_HIGH:
  _tprintf(_T("ESCORE_GPIO_OUTPUT_HIGH]\n"));
  break;
```

```
default:
  _tprintf(_T("UNKNOWN VALUE[0x%02X]\n"), GpioValue.Value);
}
printf("\n");
ShowError(bSuccess = AsrLibGetSioGpioValue(81, &GpioValue));
printf("PortPin=81\n");
switch (GpioValue.Value)
{
case ESCORE_GPIO_NOT_GPIO:
  _tprintf(_T("ESCORE_GPIO_NOT_GPIO]\n"));

  break;
case ESCORE_GPIO_DISABLED:
  _tprintf(_T("ESCORE_GPIO_DISABLED]\n"));
  break;
case ESCORE_GPIO_INPUT_LOW:
  _tprintf(_T("ESCORE_GPIO_INPUT_LOW]\n"));
  break;
case ESCORE_GPIO_INPUT_HIGH:
  _tprintf(_T("ESCORE_GPIO_INPUT_HIGH]\n"));
  break;
case ESCORE_GPIO_OUTPUT_LOW:
  _tprintf(_T("ESCORE_GPIO_OUTPUT_LOW]\n"));
```

```
  break;
 case ESCORE_GPIO_OUTPUT_HIGH:
  _tprintf(_T("ESCORE_GPIO_OUTPUT_HIGH]\n"));
  break;
 default:
  _tprintf(_T("UNKNOWN VALUE[0x%02X]\n"), GpioValue.Value);
 }
 printf("\n");
 ShowError(bSuccess = AsrLibGetSioGpioValue(82, &GpioValue));
 printf("PortPin=82\n");
 switch (GpioValue.Value)
 {
 case ESCORE_GPIO_NOT_GPIO:
  _tprintf(_T("ESCORE_GPIO_NOT_GPIO]\n"));

  break;
 case ESCORE_GPIO_DISABLED:
  _tprintf(_T("ESCORE_GPIO_DISABLED]\n"));
  break;
 case ESCORE_GPIO_INPUT_LOW:
  _tprintf(_T("ESCORE_GPIO_INPUT_LOW]\n"));
  break;
 case ESCORE_GPIO_INPUT_HIGH:
```

```
  _tprintf(_T("ESCORE_GPIO_INPUT_HIGH]\n"));
  break;
 case ESCORE_GPIO_OUTPUT_LOW:
  _tprintf(_T("ESCORE_GPIO_OUTPUT_LOW]\n"));
  break;
 case ESCORE_GPIO_OUTPUT_HIGH:
  _tprintf(_T("ESCORE_GPIO_OUTPUT_HIGH]\n"));
  break;
 default:
  _tprintf(_T("UNKNOWN VALUE[0x%02X]\n"), GpioValue.Value);
 }
 printf("\n");
 ShowError(bSuccess = AsrLibGetSioGpioValue(83, &GpioValue));
 printf("PortPin=83\n");
 switch (GpioValue.Value)
 {
 case ESCORE_GPIO_NOT_GPIO:
  _tprintf(_T("ESCORE_GPIO_NOT_GPIO]\n"));

  break;
 case ESCORE_GPIO_DISABLED:
  _tprintf(_T("ESCORE_GPIO_DISABLED]\n"));
  break;
```

```
case ESCORE_GPIO_INPUT_LOW:
  _tprintf(_T("ESCORE_GPIO_INPUT_LOW]\n"));
  break;
case ESCORE_GPIO_INPUT_HIGH:
  _tprintf(_T("ESCORE_GPIO_INPUT_HIGH]\n"));
  break;
case ESCORE_GPIO_OUTPUT_LOW:
  _tprintf(_T("ESCORE_GPIO_OUTPUT_LOW]\n"));
  break;
case ESCORE_GPIO_OUTPUT_HIGH:
  _tprintf(_T("ESCORE_GPIO_OUTPUT_HIGH]\n"));
  break;
default:
  _tprintf(_T("UNKNOWN VALUE[0x%02X]\n"), GpioValue.Value);
}
printf("\n");
ShowError(bSuccess = AsrLibGetSioGpioValue(84, &GpioValue));
printf("PortPin=84\n");
switch (GpioValue.Value)
{
case ESCORE_GPIO_NOT_GPIO:
  _tprintf(_T("ESCORE_GPIO_NOT_GPIO]\n"));
```

```
  break;
 case ESCORE_GPIO_DISABLED:
  _tprintf(_T("ESCORE_GPIO_DISABLED]\n"));
  break;
 case ESCORE_GPIO_INPUT_LOW:
  _tprintf(_T("ESCORE_GPIO_INPUT_LOW]\n"));
  break;
 case ESCORE_GPIO_INPUT_HIGH:
  _tprintf(_T("ESCORE_GPIO_INPUT_HIGH]\n"));
  break;
 case ESCORE_GPIO_OUTPUT_LOW:
  _tprintf(_T("ESCORE_GPIO_OUTPUT_LOW]\n"));
  break;
 case ESCORE_GPIO_OUTPUT_HIGH:
  _tprintf(_T("ESCORE_GPIO_OUTPUT_HIGH]\n"));
  break;
 default:
  _tprintf(_T("UNKNOWN VALUE[0x%02X]\n"), GpioValue.Value);
 }
 printf("\n");
 ShowError(bSuccess = AsrLibGetSioGpioValue(85, &GpioValue));
 printf("PortPin=85\n");
 switch (GpioValue.Value)
```

```
{
 case ESCORE_GPIO_NOT_GPIO:
  _tprintf(_T("ESCORE_GPIO_NOT_GPIO]\n"));

  break;
 case ESCORE_GPIO_DISABLED:
  _tprintf(_T("ESCORE_GPIO_DISABLED]\n"));
  break;
 case ESCORE_GPIO_INPUT_LOW:
  _tprintf(_T("ESCORE_GPIO_INPUT_LOW]\n"));
  break;
 case ESCORE_GPIO_INPUT_HIGH:
  _tprintf(_T("ESCORE_GPIO_INPUT_HIGH]\n"));
  break;
 case ESCORE_GPIO_OUTPUT_LOW:
  _tprintf(_T("ESCORE_GPIO_OUTPUT_LOW]\n"));
  break;
 case ESCORE_GPIO_OUTPUT_HIGH:
  _tprintf(_T("ESCORE_GPIO_OUTPUT_HIGH]\n"));
  break;
 default:
  _tprintf(_T("UNKNOWN VALUE[0x%02X]\n"), GpioValue.Value);
}
```

```
printf("\n");
ShowError(bSuccess = AsrLibGetSioGpioValue(86, &GpioValue));
printf("PortPin=86\n");
switch (GpioValue.Value)
{
case ESCORE_GPIO_NOT_GPIO:
  _tprintf(_T("ESCORE_GPIO_NOT_GPIO]\n"));

  break;
case ESCORE_GPIO_DISABLED:
  _tprintf(_T("ESCORE_GPIO_DISABLED]\n"));
  break;
case ESCORE_GPIO_INPUT_LOW:
  _tprintf(_T("ESCORE_GPIO_INPUT_LOW]\n"));
  break;
case ESCORE_GPIO_INPUT_HIGH:
  _tprintf(_T("ESCORE_GPIO_INPUT_HIGH]\n"));
  break;
case ESCORE_GPIO_OUTPUT_LOW:
  _tprintf(_T("ESCORE_GPIO_OUTPUT_LOW]\n"));
  break;
case ESCORE_GPIO_OUTPUT_HIGH:
  _tprintf(_T("ESCORE_GPIO_OUTPUT_HIGH]\n"));
```

```
  break;
 default:
  _tprintf(_T("UNKNOWN VALUE[0x%02X]\n"), GpioValue.Value);
 }
 printf("\n");
 ShowError(bSuccess = AsrLibGetSioGpioValue(87, &GpioValue));
 printf("PortPin=87\n");
 switch (GpioValue.Value)
 {
 case ESCORE_GPIO_NOT_GPIO:
  _tprintf(_T("ESCORE_GPIO_NOT_GPIO]\n"));

  break;
 case ESCORE_GPIO_DISABLED:
  _tprintf(_T("ESCORE_GPIO_DISABLED]\n"));
  break;
 case ESCORE_GPIO_INPUT_LOW:
  _tprintf(_T("ESCORE_GPIO_INPUT_LOW]\n"));
  break;
 case ESCORE_GPIO_INPUT_HIGH:
  _tprintf(_T("ESCORE_GPIO_INPUT_HIGH]\n"));
  break;
 case ESCORE_GPIO_OUTPUT_LOW:
```

```
  _tprintf(_T("ESCORE_GPIO_OUTPUT_LOW]\n"));
  break;
 case ESCORE_GPIO_OUTPUT_HIGH:
  _tprintf(_T("ESCORE_GPIO_OUTPUT_HIGH]\n"));
  break;
 default:
  _tprintf(_T("UNKNOWN VALUE[0x%02X]\n"), GpioValue.Value);
 }
 printf("\n");

      /*_tprintf(_T("\n"));
      _tprintf(_T("Currnet state\n"));
      _tprintf(_T("-------------------------------------\n"));
      int GP2x = 5;
      int PinCount = 8;
      int n = 0;
      SSCORE_GPIO_VALUE Values[8];

      ::AsrLibGetSioGpioGroup(GP2x, Values, &PinCount);
      for (n = 0; n < PinCount; n++) {
            DisplayGpioString(&Values[n]);
      }
```

```
    _tprintf(_T("\n"));

    ::AsrLibSetSioGpioValue(50, ESCORE_GPIO_OUTPUT_LOW);
    ::AsrLibSetSioGpioValue(51, ESCORE_GPIO_INPUT);
    ::AsrLibSetSioGpioValue(52, ESCORE_GPIO_OUTPUT_LOW);
    ::AsrLibSetSioGpioValue(53, ESCORE_GPIO_INPUT);
    ::AsrLibSetSioGpioValue(54, ESCORE_GPIO_OUTPUT_HIGH);
    ::AsrLibSetSioGpioValue(55, ESCORE_GPIO_INPUT);
    ::AsrLibSetSioGpioValue(56, ESCORE_GPIO_OUTPUT_HIGH);
    ::AsrLibSetSioGpioValue(57, ESCORE_GPIO_INPUT);

    _tprintf(_T("New state\n"));
    _tprintf(_T("-------------------------------------\n"));
    ::AsrLibGetSioGpioGroup(GP2x, Values, &PinCount);
    for (n = 0; n < PinCount; n++) {
        DisplayGpioString(&Values[n]);
    }*/
}
```

6 System Resources

6.1 Intel® Coffee Lake -S PCH

Intel® Q370 Chipset (Intel® GL82Q370 PCH)

6.2 Main Memory

RUBY-D811 provides 4x Long-DIMM sockets. The maximum memory can be up to 128GB. Memory clock and related settings can be detected by BIOS via SPD interface.

Watch out the contact and lock integrity of memory module with socket, it will impact on the system reliability. Follow normal procedures to install memory module into memory socket. Before locking, make sure that all modules have been fully inserted into the card slots.

6.3 Installing the Single Board Computer

To install your RUBY-D811 into standard chassis or proprietary environment, please perform the following:
Step 1 : Check all jumpers setting on proper position
Step 2 : Install and configure CPU,CPU cooling and memory module on right position
Step 3 : Place RUBY-D811 into the dedicated position in the system
Step 4 : Attach cables to existing peripheral devices and secure it

WARNING

Please ensure that mother board is properly inserted and fixed by mechanism.

Note:

Please refer to section 6.3.1 to 6.3.4 to install INF/Graphic/LAN

6.3.1 Chipset Component Driver

RUBY-D811 is based on Intel® Q370 chipset and desktop processors including Core™ i7 / i5 / i3 sku . It's a new chipset that some old operating systems might not be able to recognize. To overcome this compatibility issue, for Windows Operating Systems such as Windows 8, please install its INF before any of other Drivers are installed. You can find very easily this chipset component driver in RUBY-D811 CD-title

6.3.2 Intel® UHD Graphics 630

RUBY-D811 has integrated Intel® UHD Graphics 630 which supports DirectX 12、OpenGL 4.4. It is the most advanced design to gain an outstanding graphic performance. RUBY-D811 supports VGA, DP,LVDS display output. This combination makes RUBY-D811 an excellent performance hardware.

Drivers Support

Please find the Graphic driver in the RUBY-D811 CD-title. The driver supports Windows 10.

6.3.3 Intel LAN I210AT/I219LM Gigabit Ethernet Controller

- Intel I210AT Gigabit Ethernet controller and 1x RJ45 connectors on rear I/O
- Intel I219LM Gigabit Ethernet controller and 1x RJ45 connectors on rear I/O

Drivers Support

Please find Intel I210AT/I219LM LAN driver in /Ethernet directory of RUBY-D811 CD-title. The driver supports Windows 10.

7 BIOS Setup Items

7.1 Introduction

The following section describes the BIOS setup program. The BIOS setup program can be used to view and change the BIOS settings for the module. Only experienced users should change the default BIOS settings.

7.2 BIOS Setup

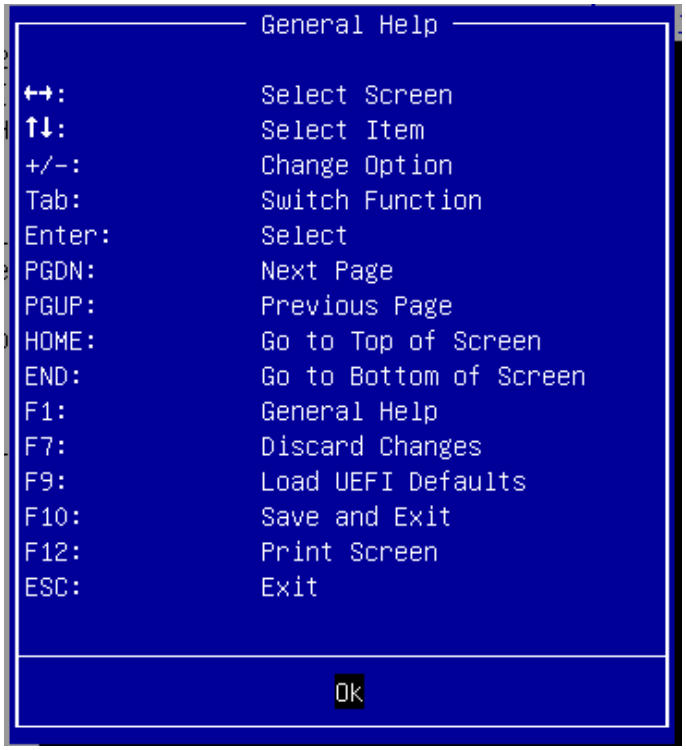
Power on the computer and the system will start POST (Power on Self Test) process. When the message below appears on the screen, press <Delete> or <ESC> key will enter BIOS setup screen.

Press <F2 > or <Delete> to enter SETUP

If the message disappears before responding and still wish to enter Setup, please restart the system by turning it OFF and On or pressing the RESET button. It can be also restarted by pressing <Ctrl>, <Alt>, and <Delete> keys on keyboard simultaneously.

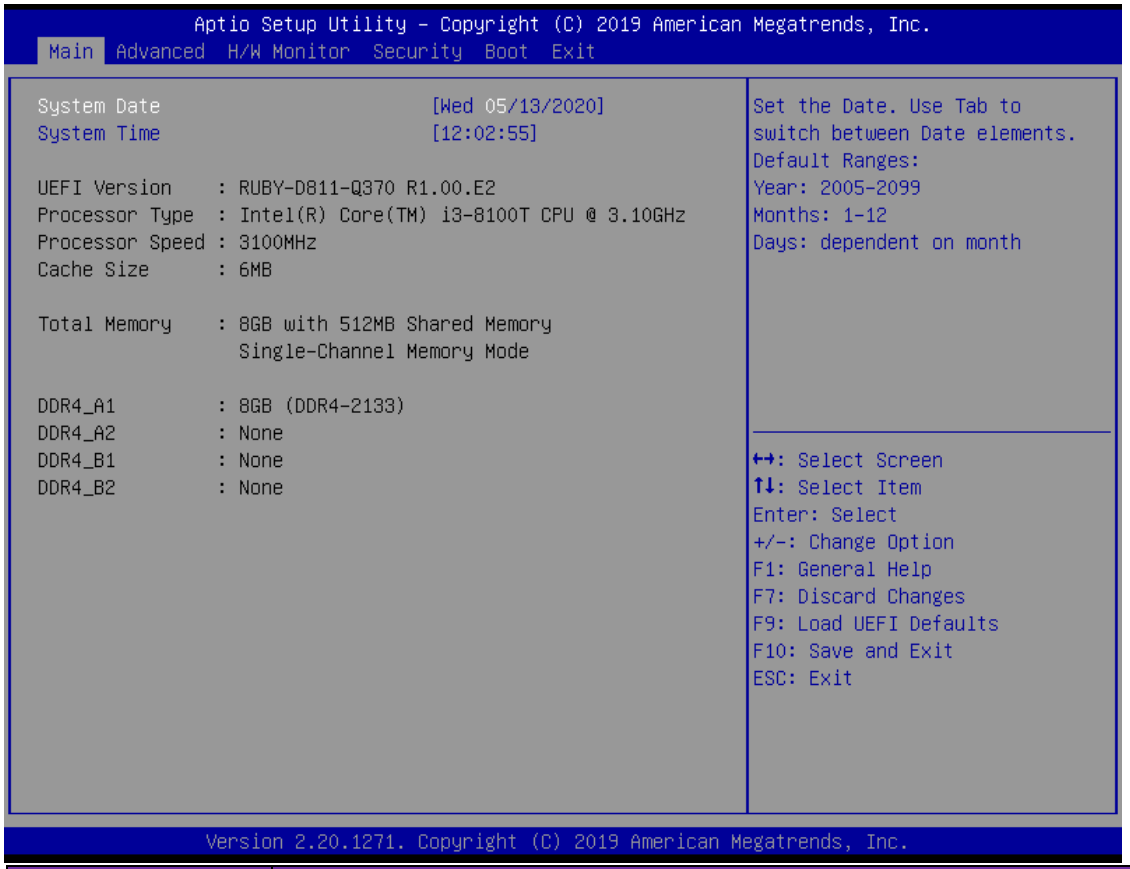
Press <F1> to Run General Help or Resume

The BIOS setup program provides a General Help screen. The menu can be easily called up from any menu by pressing <F1>. The Help screen lists all the possible keys to use and the selections for the highlighted item. Press <Esc> to exit the Help Screen.


General Help
←→: Select Screen
↑↓: Select Item
+/-: Change Option
Tab: Switch Function
Enter: Select
PGDN: Next Page
PGUP: Previous Page
HOME: Go to Top of Screen
END: Go to Bottom of Screen
F1: General Help
F7: Discard Changes
F9: Load UEFI Defaults
F10: Save and Exit
F12: Print Screen
ESC: Exit
Ok

7.2.1 Main

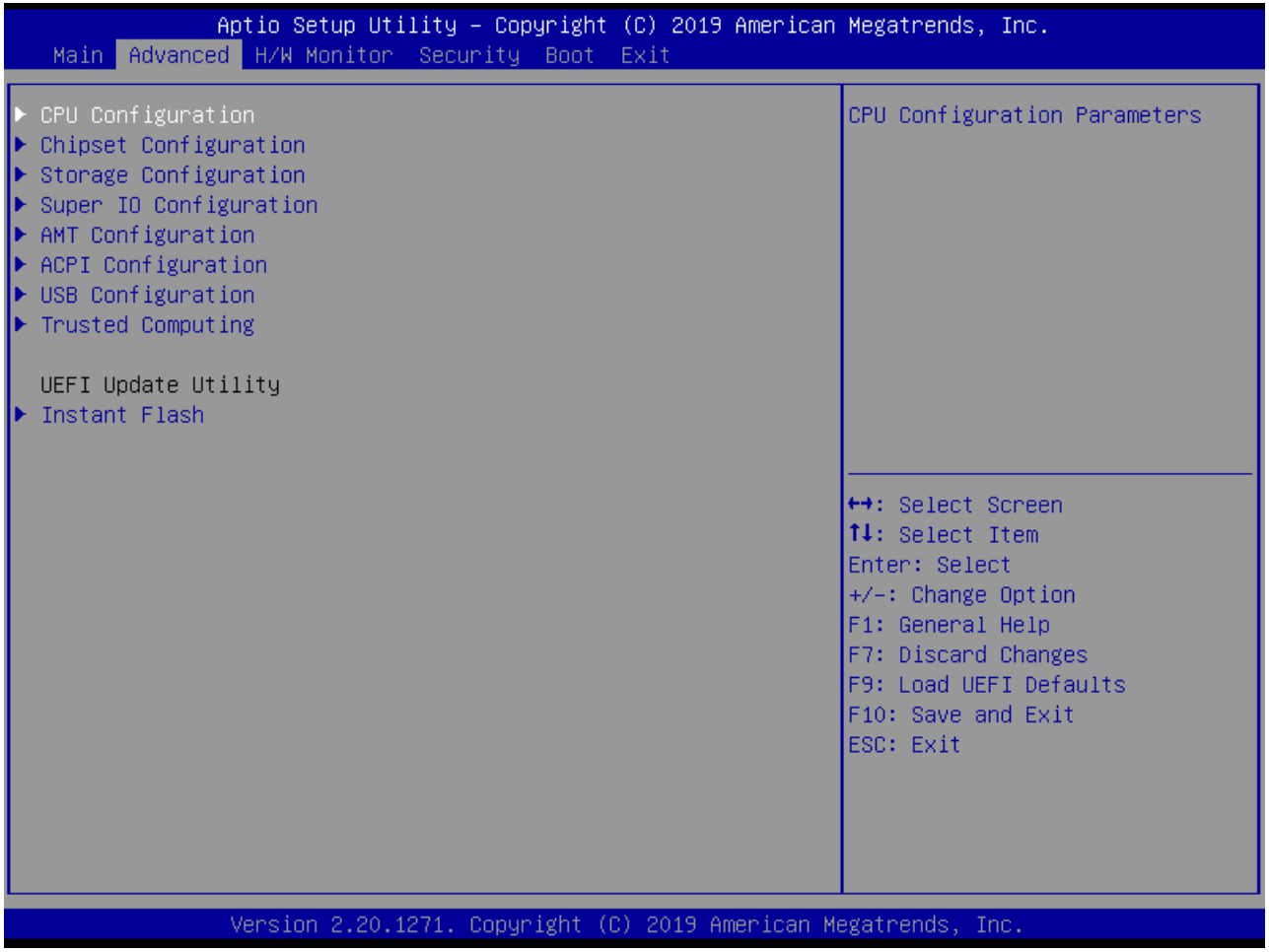
Use this menu for basic system configurations, such as time, date etc.



Feature	Description	Options
System Date	The date format is <Day>, <Month> <Date> <Year>. Use [+] or [—] to configure system Date.	
System Time	The time format is <Hour> <Minute> <Second>. Use [+] or [—] to configure system Time.	

7.2.2 Advanced

Use this menu to set up the items of special enhanced features



CPU Configuration

CPU Configuration Parameters

```
Aptio Setup Utility - Copyright (C) 2019 American Megatrends, Inc.
Advanced

Intel(R) Core(TM) i3-8100T CPU @ 3.10GHz          Select the number of cores to
Microcode Revision                906EB B4        enable in each processor
Max CPU Speed                     3100 MHz        package.
Min CPU Speed                     800 MHz
Processor Cores                   4

Active Processor Cores            [All]
CPU C States Support              [Enabled]
 Enhanced Halt State(C1E)         [Auto]
 Package C State Support          [Disabled]
 CFG Lock                         [Disabled]

Intel Virtualization Technology   [Enabled]        ←→: Select Screen
Intel SpeedStep Technology        [Enabled]        ↑↓: Select Item
                                                   Enter: Select
CPU Thermal Throttling            [Enabled]        +/-: Change Option
                                                   F1: General Help
                                                   F7: Discard Changes
                                                   F9: Load UEFI Defaults
                                                   F10: Save and Exit
                                                   ESC: Exit

Version 2.20.1271. Copyright (C) 2019 American Megatrends, Inc.
```

Feature	Description	Options
Active Processor Cores	Select the number of cores to enable in each processor package.	★All, 1, 2, 3
CPU C states Support	Enable CPU C states Support for power saving. It is recommended to keep C3, C6 and C7 all enabled for better power saving.	★Disabled, Enabled
CPU C states Support[Enabled]		
Enhanced Halt State(C1E)	Enable Enhanced Halt State (C1E) for lower power consumption.	★Auto ,Disabled, Enabled
Package C State Support	Maximum Package C State Limit Setting. CPU Default: Leaves to Factory default value. Auto: Initializes to deepest available Package C State Limit.	★Disabled, Auto, Enabled
CFG Lock	This item allows you to disable or enable the CFG Lock.	★Disabled, Enabled
Intel Virtualization Technology	Intel Virtualization Technology allows a platform to run multiple operating systems and application in independent partitions, so that one computer system can function as multiple virtual systems.	★Enabled, Disabled
Intel Speed Step Technology	Allows more than two frequency ranges to be supported.	★Enabled, Disabled
CPU Thermal Throttling	Enable CPU internal thermal control mechanisms to keep the CPU from overheating.	★Enabled, Disabled

Chipset Configuration

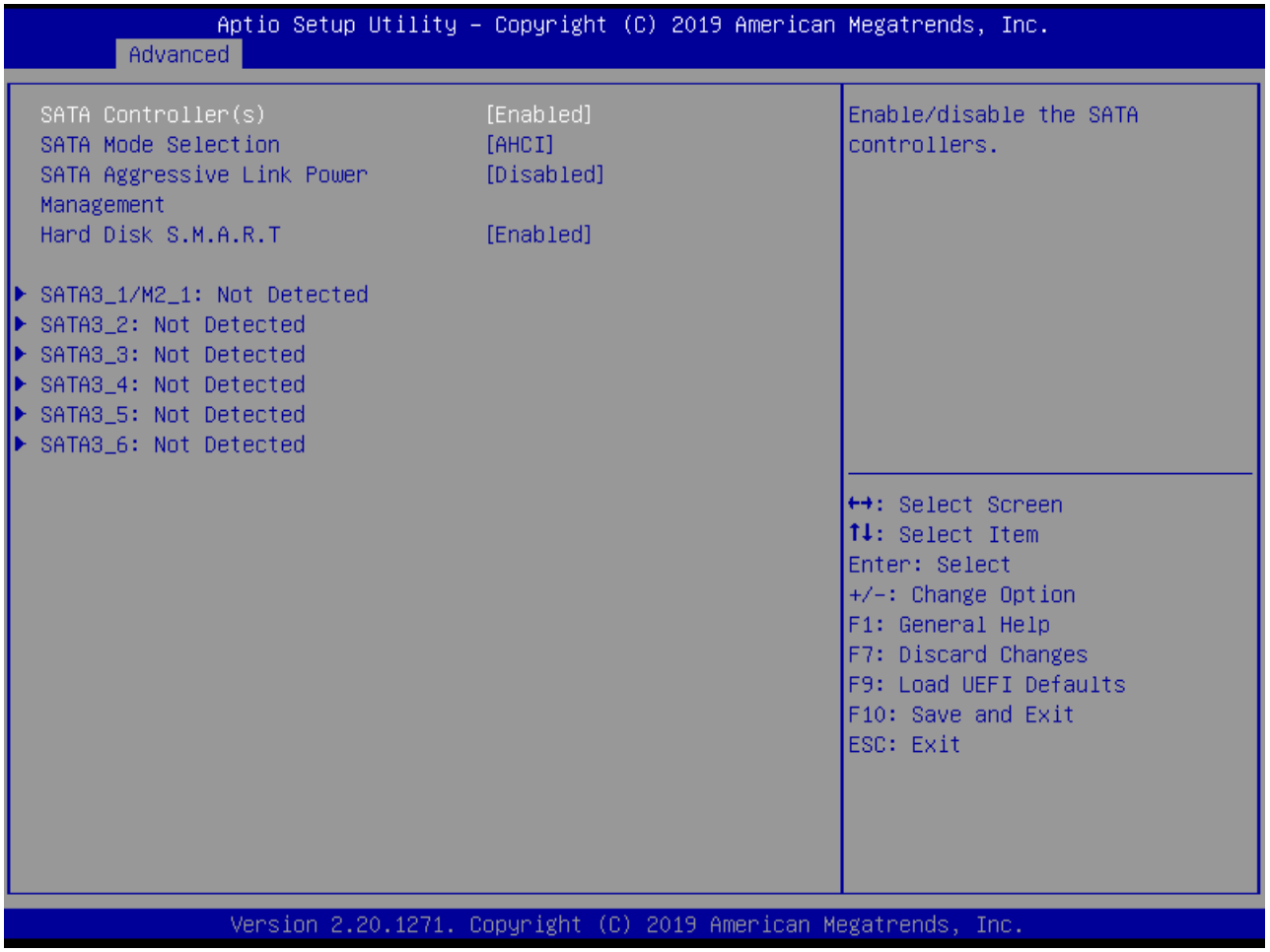
Configuration Chipset Settings

```
Aptio Setup Utility - Copyright (C) 2019 American Megatrends, Inc.
Advanced

ME Firmware Version                12.0.34.1425
VT-d Capability                    Supported

Primary Graphics Adapter           [PCI Express]
Above 4G Decoding                  [Disabled]
VT-d                               [Enabled]

PCIE1 Link Speed                   [Auto]
PCIE2 Link Speed                   [Auto]
PCIE3 Link Speed                   [Auto]
PCIE4 Link Speed                   [Auto]
PCIE5 Link Speed                   [Auto]
PCIE6 Link Speed                   [Auto]
PCIE7 Link Speed                   [Auto]

PCIE2 Slot                         [Auto detection]

Share Memory                       [Auto]
IGPU Multi-Monitor                 [Disabled]

Onboard LAN1                       [Enabled]
Onboard LAN2                       [Enabled]

Onboard HD Audio                   [Enabled]

Deep Sleep                         [Disabled]
Restore on AC/Power Loss           [Power Off]

Select a primary VGA.

←→: Select Screen
↑↓: Select Item
Enter: Select
+/-: Change Option
F1: General Help
F7: Discard Changes
F9: Load UEFI Defaults
F10: Save and Exit
ESC: Exit

Version 2.20.1271. Copyright (C) 2019 American Megatrends, Inc.
```

Feature	Description	Options
Primary Graphics Adapter	Select a primary VGA.	★PCI Express ,Onboard
Above 4GB Decoding	Enable/Disable above 4G Memory Mapped IO decoding .This is disabled automatically when Aperture Size is set to 2048MB.	★Disabled, Enabled
VT-d	VT-d Capability	★Enabled ,Disabled
PCIE1~7 Link Speed	Configure PCIE1~7 Slot Link Speed. Auto mode is optimizing for over clocking.	★Auto,Gen1, Gen2, Gen3
PCIE2 Slot	Configure PCIE2 Slot	★Auto detection, Fix PCIE x4 mode
Share Memory	Configure the size of memory that is allocated to the integrated graphics processor when the system boots up.	★Auto,32M,64M,128M,256M, 512M,1024M
IGPU Multi-Monitor	Select disable to disable the integrated graphics when an external graphics card is installed. Select enable to keep the integrated graphics enable at all times.	★Disabled, Enabled
Onboard LAN1	Enable or disable the onboard LAN1 network interface controller.	★Enabled ,Disabled
Onboard LAN2	Enable or disable the onboard LAN2 network interface controller.	★Enabled ,Disabled
Onboard HD Audio	Enable/disable onboard HD audio.	★Enabled ,Disabled
Deep Sleep	Configure deep sleep mode for power saving when the computer is shut down. We recommend disabling Deep Sleep for better system compatibility and stability.	★Disabled, Enabled in S5, Enabled in S4-S5
Restore on AC/Power Loss	Select the power state after a power failure. If [Power Off] is selected, the power will remain off when the power recovers. If [Power On] is selected, the system will start to boot up when the power recovers.	★Power Off ,Power On

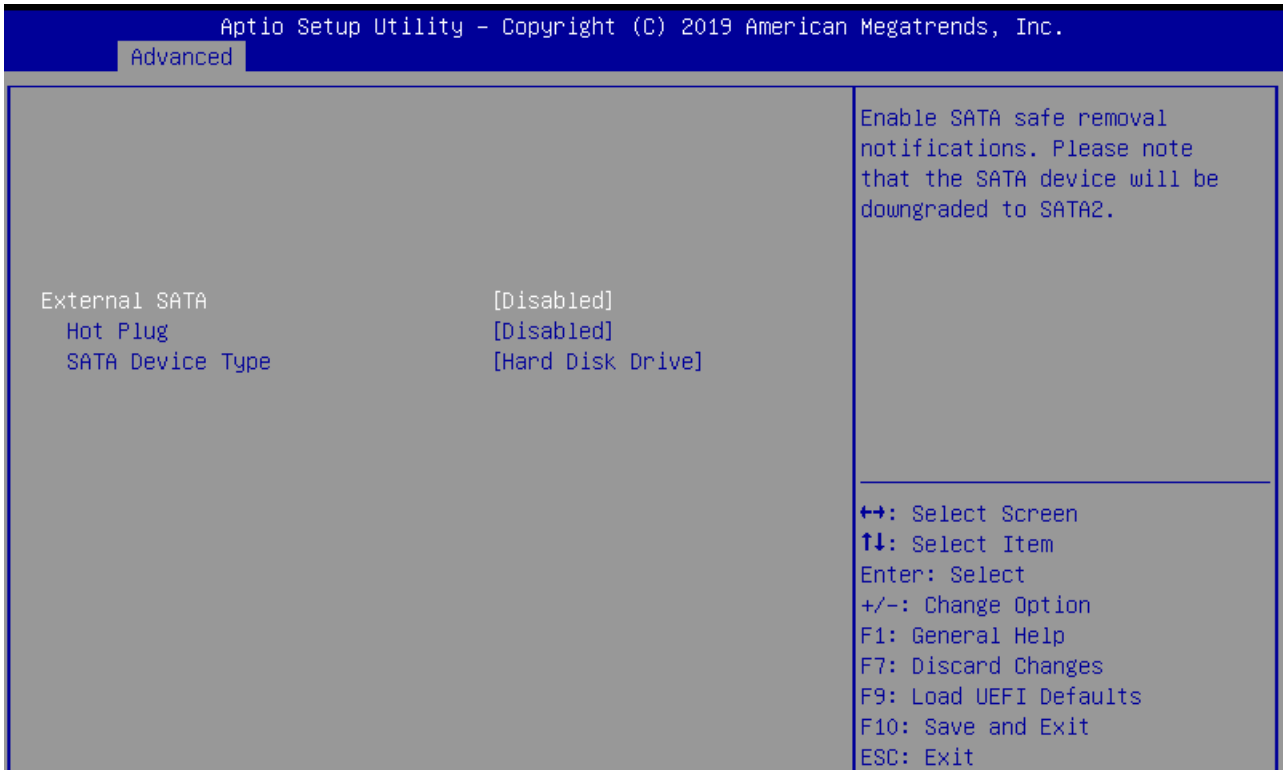
Storage Configuration

Configure storage devices



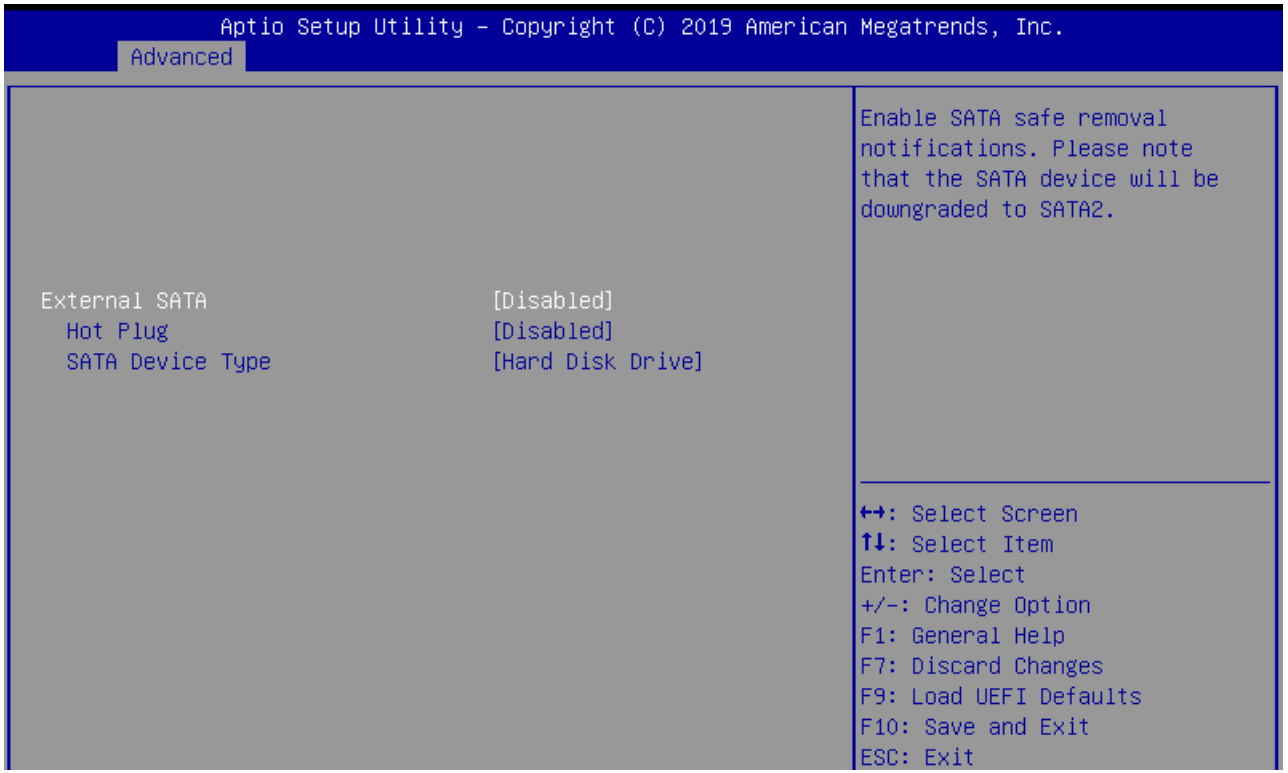
Feature	Description	Options
SATA Controller(s)	Enable/disable the SATA controllers.	★Enabled , Disabled
SATA Mode Selection	AHCI: Supports new features that improve performance. Intel RST Premium (RAID): Combine multiple disk drives into a logical unit. Please press <CTRL - I> to enter RAID ROM during UEFI POST process.	★AHCI, RAID
SATA Aggressive Link Power Management	SATA Aggressive Link Power Management allows SATA devices to enter a low power state during periods of inactivity to save power. It is only supported by AHCI mode.	★Disabled, Enabled
Hard Disk S.M.A.R.T	S.M.A.R.T stands for Self-Monitoring, Analysis, and Reporting system for computer hard disk drives to detect and report on various indicators of reliability.	★Enabled , Disabled

SATA3_1/M2_1: Not Detected Configuration



Feature	Description	Options
External SATA	Enable SATA safe removal notifications. Please note that the SATA device will be downgraded to SATA2.	★Disabled ,Enabled
Hot Plug	Enable or disable Hot Plug for this port.	★Disabled ,Enabled
SATA Device Type	Identify the SATA port is connected to Solid State Drive or Hard Disk Drive.	★Hard Disk Drive, Solid State Drive

SATA3_2,3,4,5,6: Not Detected Configuration



Feature	Description	Options
External SATA	Enable SATA safe removal notifications. Please note that the SATA device will be downgraded to SATA2.	★Disabled ,Enabled
Hot Plug	Enable or disable Hot Plug for this port.	★Disabled ,Enabled
SATA Device Type	Identify the SATA port is connected to Solid State Drive or Hard Disk Drive.	★Hard Disk Drive, Solid State Drive

Super IO Configuration

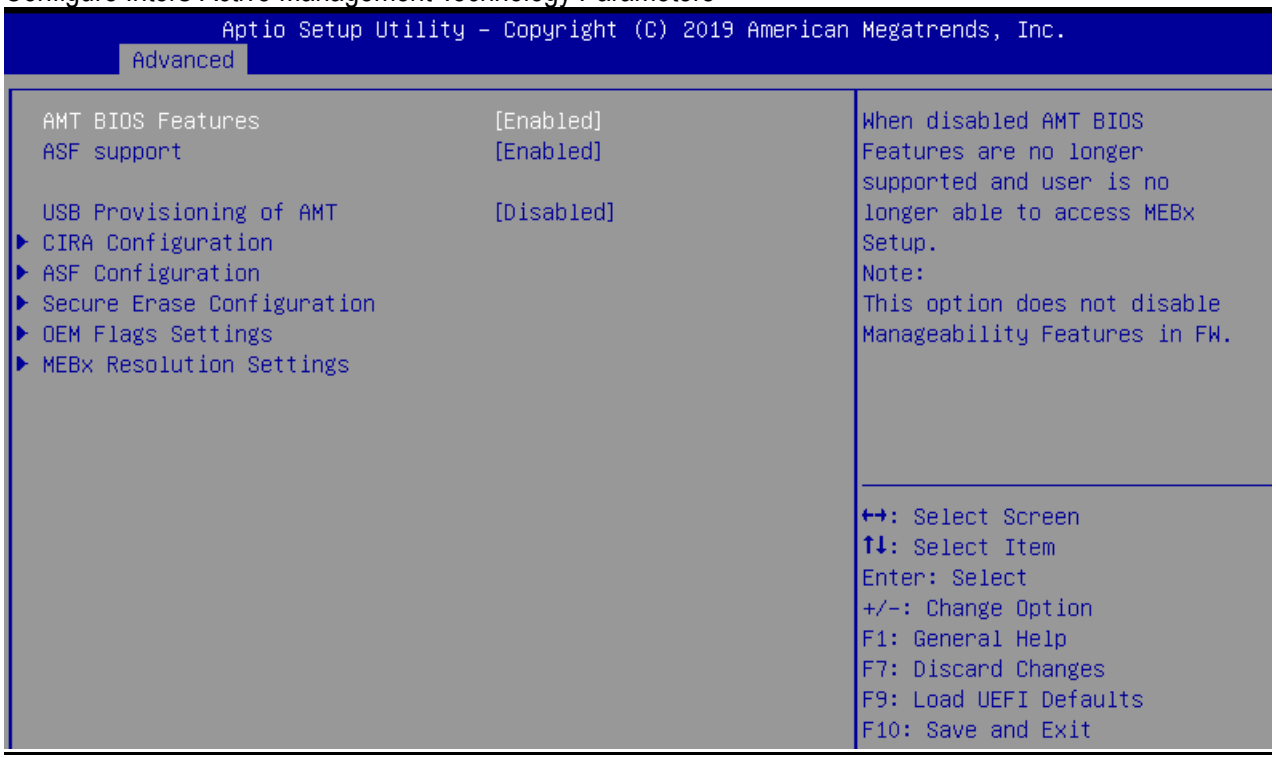
Configure Super IO Settings

```
Aptio Setup Utility - Copyright (C) 2019 American Megatrends, Inc.
Advanced

COM1                                [Enabled]          Enable or Disable COM1
    Type Select                     [RS232]            IO=3F8h; IRQ=4;
COM2                                [Enabled]
    Type Select                     [RS232]
COM3                                [Enabled]
COM4                                [Enabled]
COM5                                [Enabled]
COM6                                [Enabled]
COM7                                [Enabled]
COM8                                [Enabled]
COM9                                [Enabled]
COM10                               [Enabled]

Parallel Port                       [Enabled]          ←→: Select Screen
    Device Mode                     [ECP and EPP 1.9 Mode]  ↑↓: Select Item
    Change Settings                 [Auto]             Enter: Select
                                                       +/-: Change Option
WDT Timeout Reset                   [Enabled]          F1: General Help
WDT Initial Value (Sec.)            255                F7: Discard Changes
                                                       F9: Load UEFI Defaults
                                                       F10: Save and Exit
                                                       ESC: Exit

Version 2.20.1271. Copyright (C) 2019 American Megatrends, Inc.
```

Feature	Description	Options
COM1	Enable or Disable COM1 , IO=3F8h; IRQ=4;	★Enabled ,Disabled
Type Select	Set COM Type	★RS232,RS422,RS485
COM2	Enable or Disable COM2 , IO=2F8h; IRQ=3;	★Enabled ,Disabled
Type Select	Set COM Type	★RS232,RS422,RS485
COM3	Enable or Disable COM3 , IO=3E8h; IRQ=7;	★Enabled ,Disabled
COM4	Enable or Disable COM4, IO=2E8h; IRQ=7;	★Enabled ,Disabled
COM5	Enable or Disable COM5 , IO=2E0h; IRQ=10;	★Enabled ,Disabled
COM6	Enable or Disable COM6 , IO=2F0h; IRQ=10;	★Enabled ,Disabled
COM7	Enable or Disable COM7 , IO=240h; IRQ=11;	★Enabled ,Disabled
COM8	Enable or Disable COM8 , IO=248h; IRQ=11;	★Enabled ,Disabled
COM9	Enable or Disable COM9 , IO=250h; IRQ=11;	★Enabled ,Disabled
COM10	Enable or Disable COM10 , IO=258h; IRQ=11;	★Enabled ,Disabled
Parallel Port	Select Parallel Port (LPT/LPTE) or GPIO. Enabled: LPT. Disabled: GPIO	★Enabled ,Disabled
Device Mode	Change the Printer Port mode.	★ECP and EPP 1.9 Mode, Normal, Bi-Directional, ECP and EPP 1.7 Mode
Change Settings	Select an optimal settings for Super IO Device	★Auto, IO=378h; IRQ=5; DMA=3 IO=378h; IRQ=5,6,7,9,10,11,12; DMA=1,3 IO=278h; IRQ=5,6,7,9,10,11,12; DMA=1,3
WDT Timeout Reset	Enable/Disable Watch Dog Timer timeout to reset system.	★Disabled , Enabled
WDT Timeout Reset [Enabled]		
WDT Initial Value (Sec.)	Watch Dog Timer Initial Value to count down.	★255

AMT Configuration

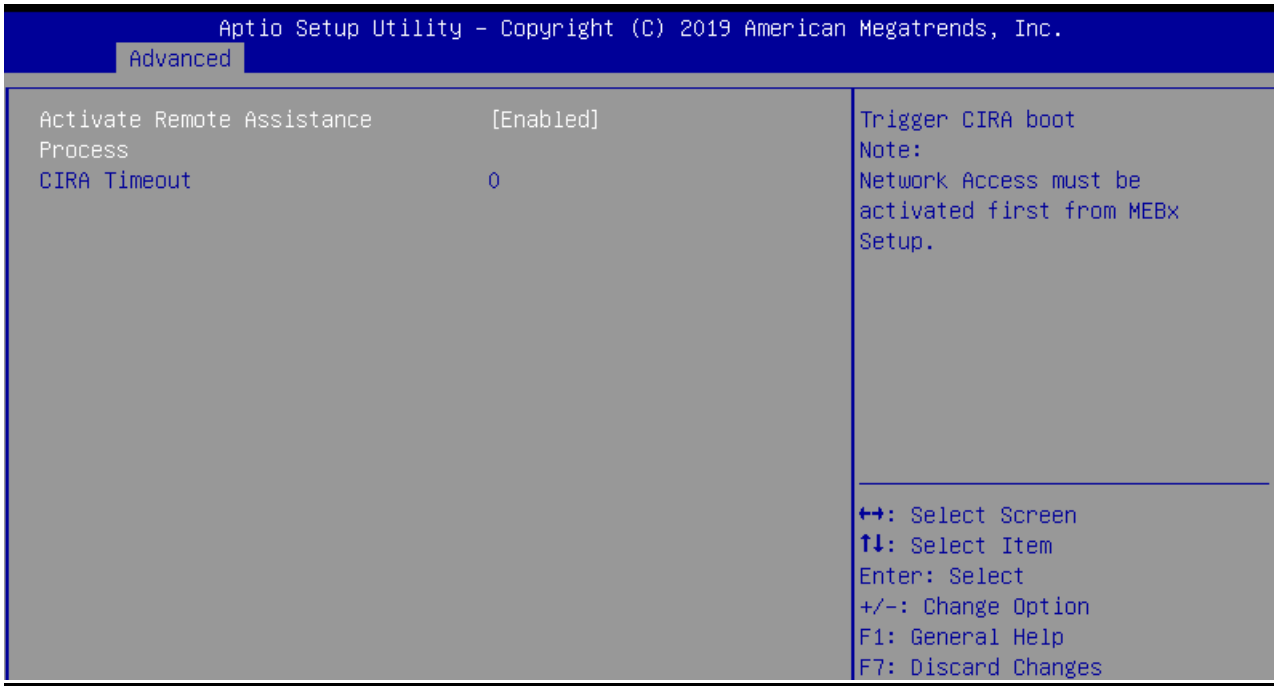
Configure Intel® Active Management Technology Parameters



Feature	Description	Options
AMT BIOS Features	When disabled AMT BIOS Features are no longer supported and user us no longer able to access MEBx Setup. Note: This option does not disable Manageability Features in FW	★Enable ,Disabled,
ASF support	Enable/Disable Alert Standard Format support.	★Enable ,Disabled,
USB Provisioning of AMT	Enable/Disable of AMT USB Provisioning.	★Disabled, Enable

CIRA Configuration

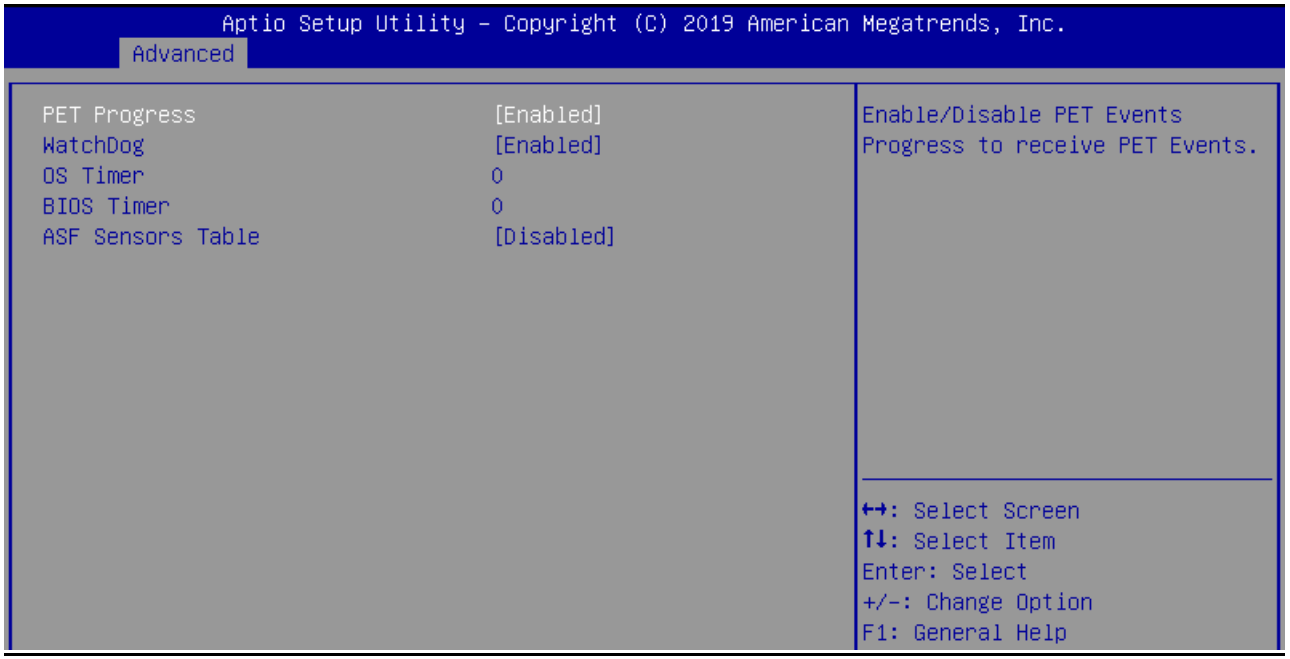
Configure Remote Assistance Process parameters



Feature	Description	Options
Activate Remote Assistance Process	Trigger CIRA boot. Note: Network Access must be activated first from MEBx Setup.	★Disabled, Enable
Activate Remote Assistance Process[Enabled]		
CIRA Timeout	OEM defined timeout for MPS connection to be established. 0-Use the default timeout value of 60 seconds. 255- MEBx waits until the connection succeeds.	★0

ASF Configuration

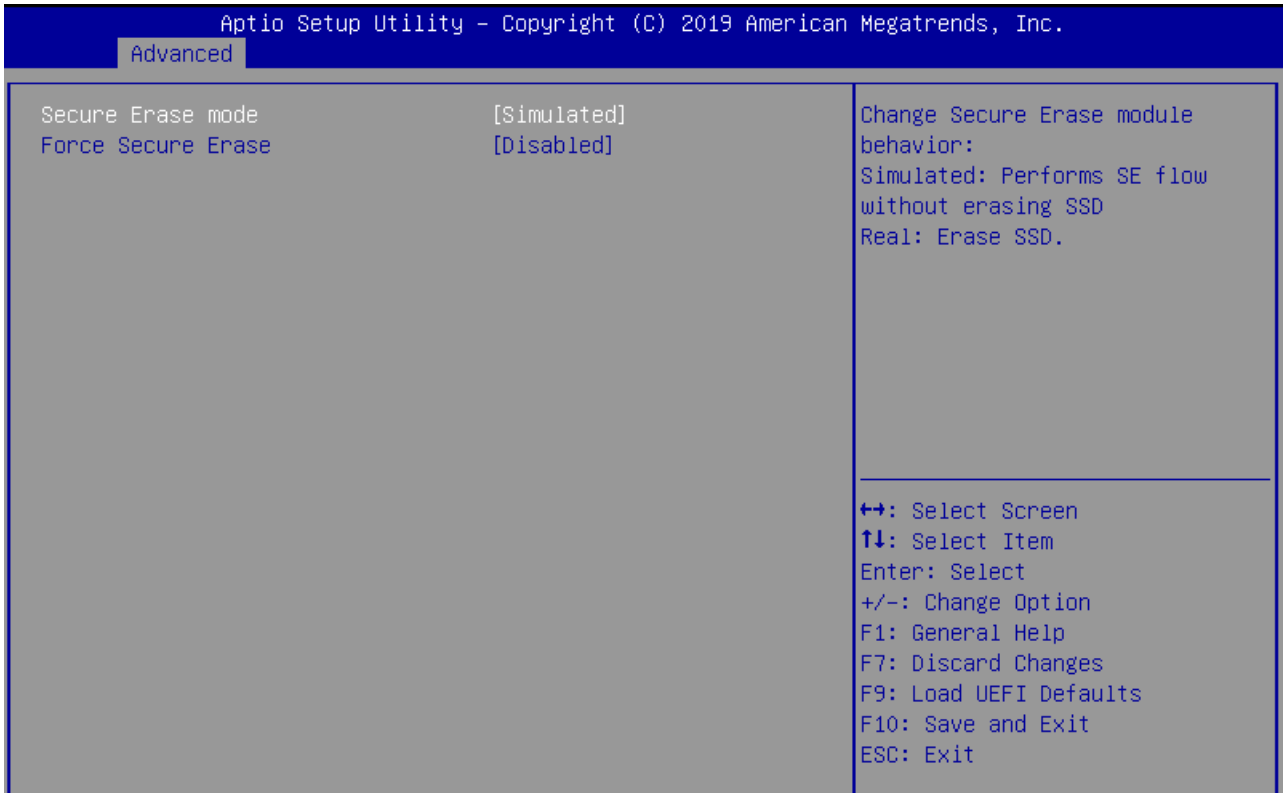
Configure Alert Standard Format parameters



Feature	Description	Options
PET Process	Enable/Disable PET Events Progress to receive PET Events.	★Enable, Disabled
WatchDog	Enable/Disable WatchDog Timer.	★Disabled, Enable
WatchDog[Enabled]		
OS Timer	Set OS watchdog timer	★0
BIOS Timer	Set BIOS watchdog timer	★0
ASF Sensors Table	Adds ASF Sensor Table into ASF! ACPI Table	★Disabled, Enable

Secure Erase Configuration

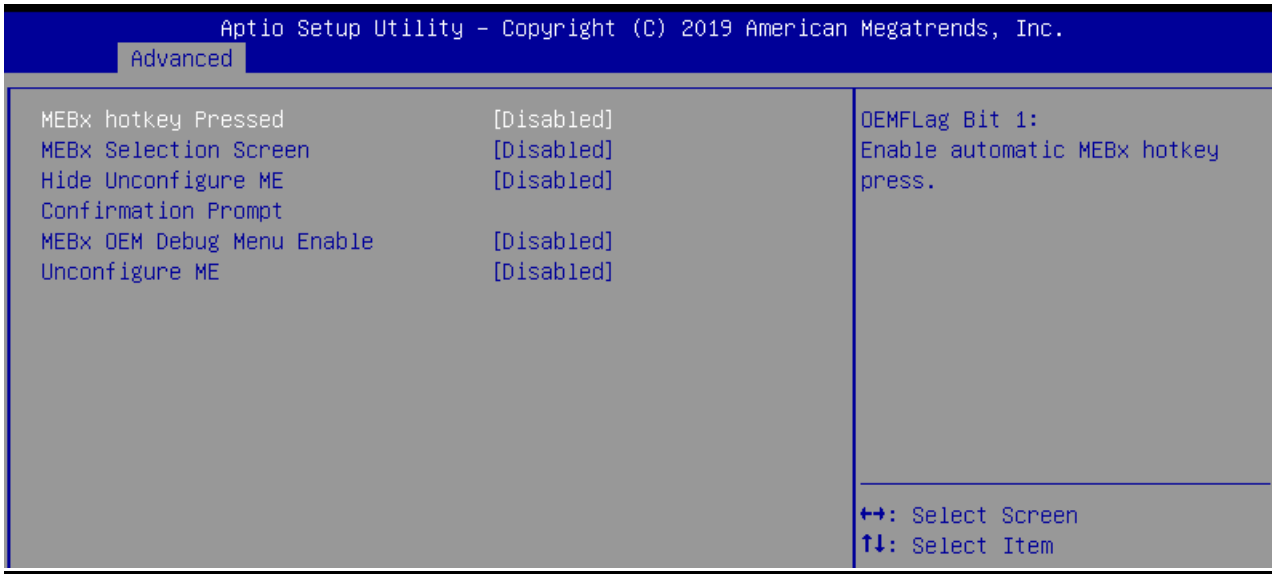
Secure Erase Configuration menu



Feature	Description	Options
Secure Erase mode	Change Secure Erase module behavior: Simulated: Performs SE flow without erasing SSD Real: Erase SSD.	★Simulation, Real
Force Secure Erase	Force Secure Erase on next boot	★Disabled, Enable

OEM Flags Settings

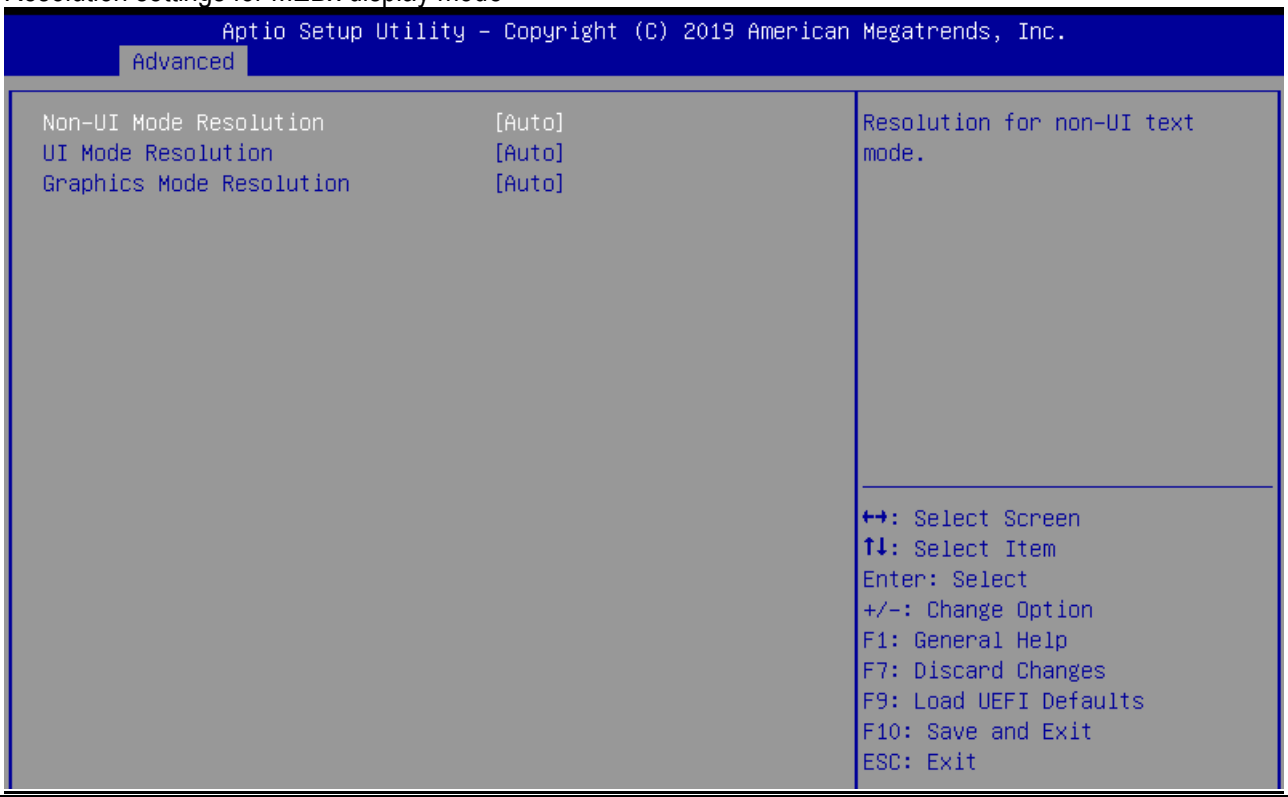
Configure OEM Flags



Feature	Description	Options
MEBx hotkey Pressed	OEMFLag Bit 1: Enable automatic MEBx hotkey press.	★Disabled, Enable
MEBx Selection Screen	OEMFLag Bit 2: Enable MEBx selection screen with 2 options: Press 1 to enter ME Configuration Screens Press 2 to initiate a remote connection Note: Network Access must be activated from MEBx Setup for this screen to be displayed.	★Disabled, Enable
Hide Unconfigure ME Confirmation Prompt	OEMFlag Bit 6: Hide Unconfigure ME confirmation prompt when attempting ME unconfiguration.	★Disabled, Enable
MEBx OEM Debug Menu Enable	OEMFlag Bit 14: Enable OEM debug menu in MEBx.	★Disabled, Enable
Unconfigure ME	OEMFlag Bit 15: Unconfigure ME with resetting MEBx password to default.	★Disabled, Enable

MEBx Resolution Settings

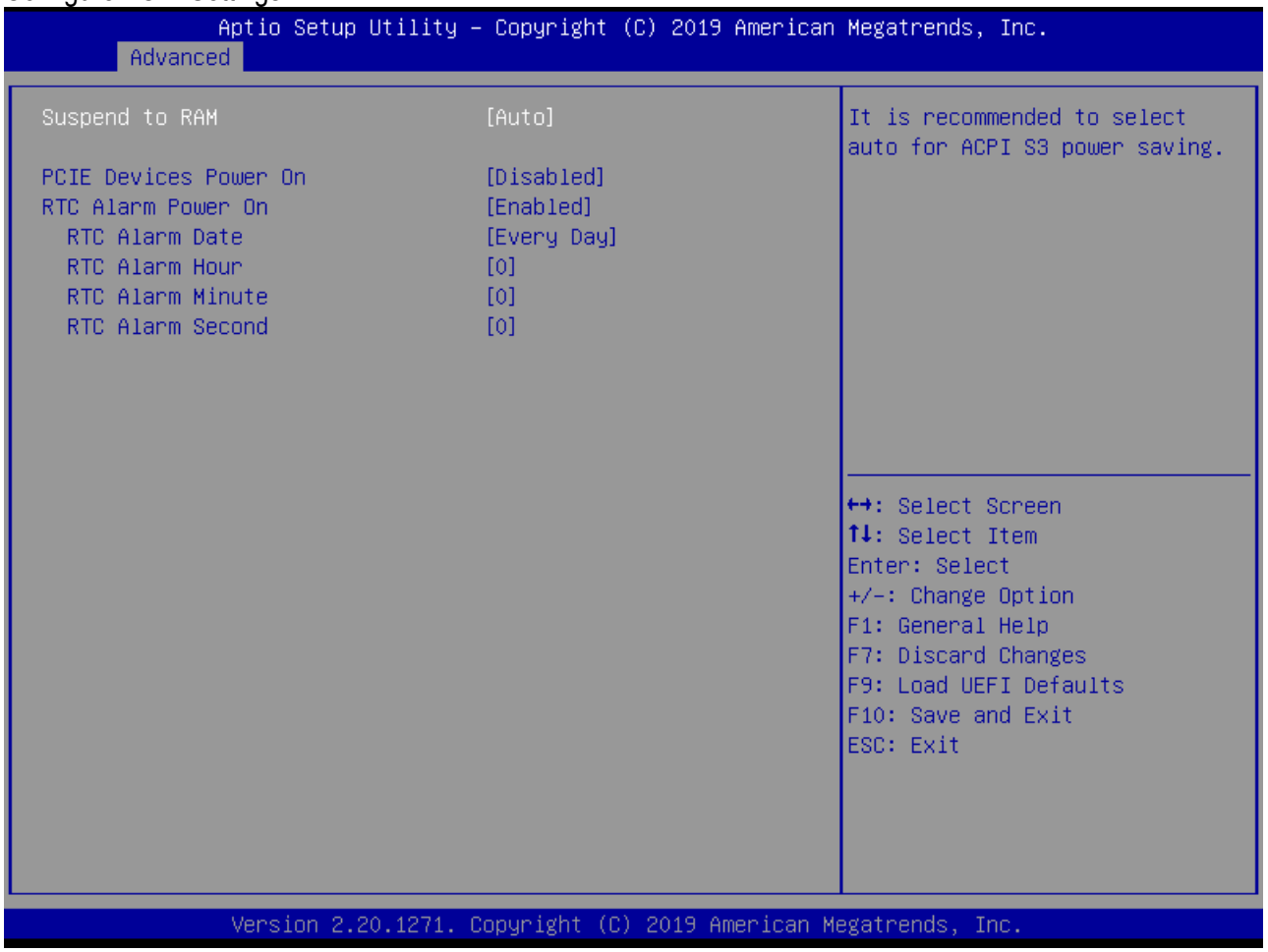
Resolution settings for MEBx display mode



Feature	Description	Options
Non-UI Mode Resolution	Resolution for non-UI text mode.	★Auto, 80x25,100x31
UI Mode Resolution	Resolution for UI text mode.	★Auto, 80x25,100x31
Graphics Mode Resolution	Resolution for graphic mode.	★Auto, 640x480,800x600, 1024x768

ACPI Configuration

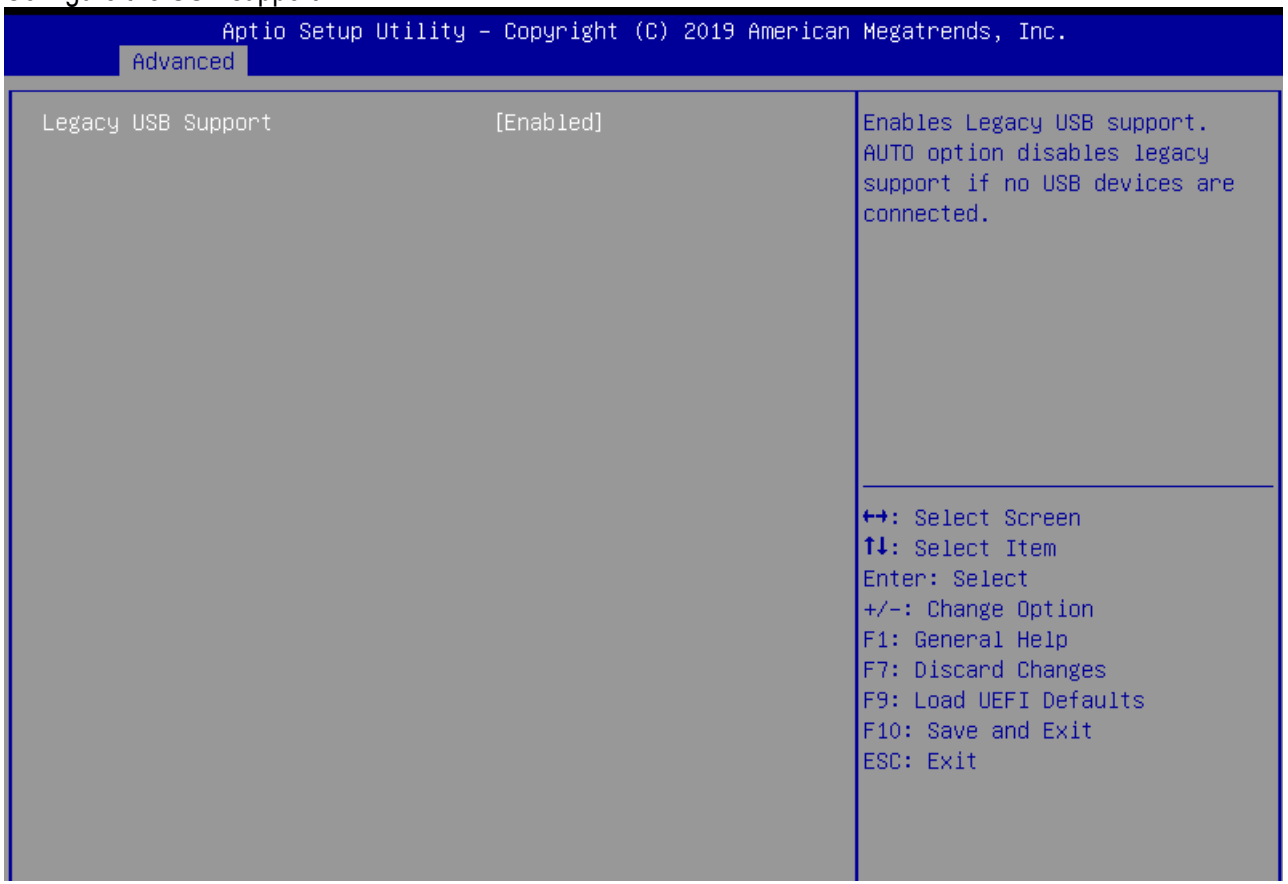
Configure ACPI Settings



Feature	Description	Options
Suspend to RAM	It is recommended to select auto for ACPI S3 power saving.	★Auto ,Disabled
PCIE Devices Power On	Allow the system to be waked up by a PCIE device and enable wake on LAN.	★Disabled, Enable
RTC Alarm Power On	Allow the system to be waked up by the real time clock alarm. Set it to By OS to let it be handled by your operating system.	★By OS ,Disabled, Enable
RTC Alarm Power On[Enabled]		
RTC Alarm Date	Set Date of RTC power on feature.	★EveryDay,1,2,3,4,5,6,7,8,9, 10,11,12,13,14,15,16,17,18, 19,20,21,22,23,24,25,26,27, 28,29,30,31
RTC Alarm Hour	Set Hour of RTC power on feature.	★0,1,2,3,4,5,6,7,8,9,10,11, 12,13,14,15,16,17,18,19,20, 21,22,23,
RTC Alarm Minute	Set Minute of RTC power on feature.	★0,1,2,3,4,5,6,7,8,9,10,11, 12,13,14,15,16,17,18,19,20, 21,22,23,24,25,26,27,28,20, 30,31,32,33,34,35,36,37,38, 39,40,41,42,43,44,45,46,47, 48,49,50,51,52,53,54,55,56, 57,58,59
RTC Alarm Second	Set Second of RTC power on feature.	★0,1,2,3,4,5,6,7,8,9,10,11, 12,13,14,15,16,17,18,19,20, 21,22,23,24,25,26,27,28,20, 30,31,32,33,34,35,36,37,38, 39,40,41,42,43,44,45,46,47, 48,49,50,51,52,53,54,55,56, 57,58,59

USB Configuration

Configure the USB support



Feature	Description	Options
Legacy USB Support	Enables Legacy USB support. AUTO option disables legacy support if no USB devices are connected.	★Enabled, UEFI Setup Only

Trusted Computing

Trusted Computing Settings

```
Aptio Setup Utility - Copyright (C) 2019 American Megatrends, Inc.
Advanced

TPM20 Device Found
Firmware Version:                 7.62
Vendor:                           IFX

Security Device Support           [Enable]
Active PCR banks                  SHA-1,SHA256
Available PCR banks               SHA-1,SHA256


Pending operation                 [None]
Platform Hierarchy                [Enabled]
Storage Hierarchy                 [Enabled]
Endorsement Hierarchy             [Enabled]
TPM2.0 UEFI Spec Version          [TCG_2]
Physical Presence Spec Version    [1.3]
TPM 20 InterfaceType              [TIS]
Device Select                     [Auto]

Onboard TPM                       [Enabled]

Enables or Disables BIOS
support for security device.
O.S. will not show Security
Device. TCG EFI protocol and
INT1A interface will not be
available.

←→: Select Screen
↑↓: Select Item
Enter: Select
+/-: Change Option
F1: General Help
F7: Discard Changes
F9: Load UEFI Defaults
F10: Save and Exit
ESC: Exit

Version 2.20.1271. Copyright (C) 2019 American Megatrends, Inc.
```

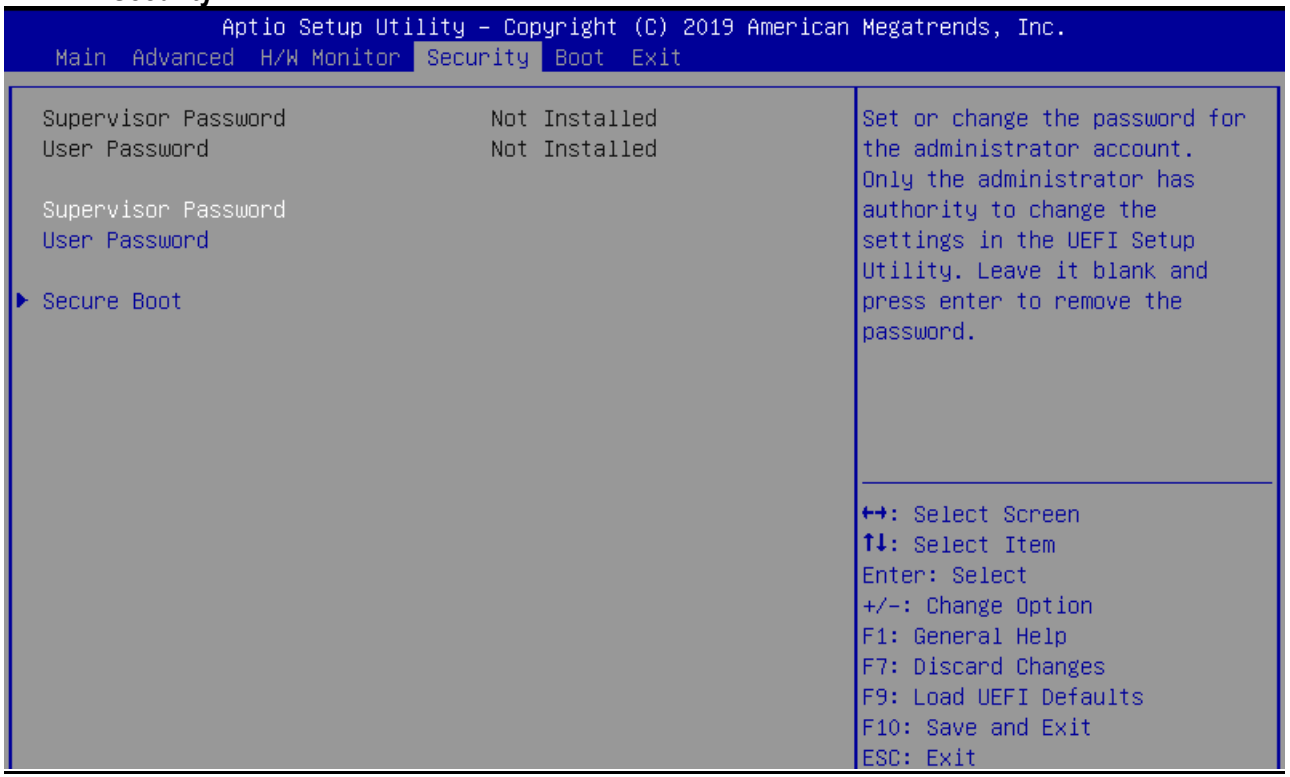
Feature	Description	Options
Security Device Support	Enables or Disables BIOS support for security device. O.S. will not show Security Device. TCG EFI protocol and INT1A interface will not be available.	★Enable ,Disable
Pending operation	Schedule an Operation for the Security Device. NOTE: Your Computer will reboot during restart in order to change State of Security Device.	★None, TPM Clear
Platform Hierarchy	Enable or Disable Platform Hierarchy	★Enable ,Disable
Storage Hierarchy	Enable or Disable Storage Hierarchy	★Enable ,Disable
Endorsement Hierarchy	Enable or Disable Endorsement Hierarchy	★Enable ,Disable
TPM2.0 UEFI Spec Version	Select the TCG2 Spec Version Support, TCG_1_2: the Compatible mode for Win8/Win10, TCG_2: Support new TCG2 protocol and event format for Win10 or later	★TCG_2,TCG_1_2,
Physical Presence Spec Version	Select to Tell O.S.to support PPI Spec Version 1.2 or 1.3. Note some HCK tests might not support 1.3.	★1.3,1.2
Device Select	TPM1.2 will restrict support to TPM1.2 devices,TPM2.0 will restrict support to TPM2.0 devices, Auto will support both with the default set to TPM2.0 devices if not found, TPM1.2 devices will be enumerated	★Auto ,TPM1.2,TPM2.0
Onboard TPM	Enable or disable the onboard TPM interface controller.	★Enable ,Disable

7.2.3 H/W Monitor

```
Aptio Setup Utility - Copyright (C) 2019 American Megatrends, Inc.
Main  Advanced  H/W Monitor  Security  Boot  Exit

Hardware Health Event Monitoring                    Quiet Fan Function Control

CPU Temperature                  : +56 °C
M/B Temperature                  : +28 °C

CPU_FAN1 Speed                   : 20547 RPM
CHA_FAN1 Speed                   : N/A

+3V                              : +3.376 V
+3VSB                            : +3.376 V
+5V                              : +5.112 V
VCORE                            : +0.920 V
VCCM                             : +1.232 V
+12V                             : +12.160 V

CPU_FAN1 Setting                 [Automatic mode]      ↔: Select Screen
  Target CPU Temperature         [50 °C/122 °F]        ↑↓: Select Item
  Target Fan Speed               [Level 9]             Enter: Select
CHA_FAN1 Setting                 [Automatic mode]      +/-: Change Option
  Target CPU Temperature         [50 °C/122 °F]        F1: General Help
  Target Fan Speed               [Level 9]             F7: Discard Changes
Case Open Feature                [Disabled]            F9: Load UEFI Defaults
OTP                              [Enabled]             F10: Save and Exit
                                                       ESC: Exit

Version 2.20.1271. Copyright (C) 2019 American Megatrends, Inc.
```

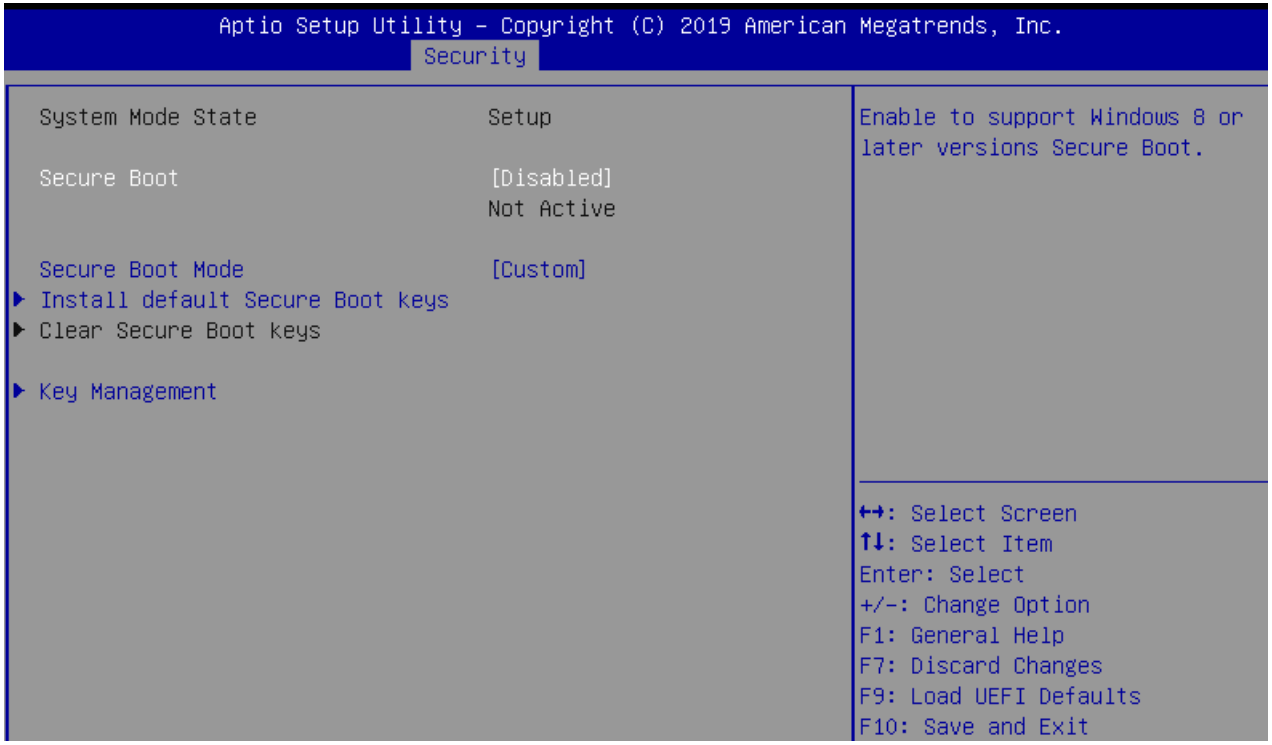
Feature	Description	Options
CPU_FAN1 Setting	Quiet Fan Function Control	★Full On, Automatic Mode
CPU_FAN1 Setting[Automatic Mode]		
Target CPU Temperature	Target CPU Temperature Value.	★50℃/122℉,45℃/113℉, 46℃/114℉, 47℃/116℉, 48℃/118℉, 49℃/120℉, 51℃/123℉, 52℃/125℉, 53℃/127℉, 54℃/129℉, 55℃/131℉, 56℃/132℉,57℃/134℉, 58℃/136℉, 59℃/138℉, 60℃/140℉, 61℃/141℉, 62℃/143℉, 63℃/145℉, 64℃/147℉, 65℃/149℉
Target Fan Speed	The higher the level, the higher the fan speed.	★Level 9,Level 1, Level 2, Level 3, Level 4, Level 5, Level 6, Level 7, Level 8,
CHA_FAN1 Setting	Quiet Fan Function Control	★Full On, Automatic Mode
CHA_FAN1 Setting[Automatic Mode]		
Target CPU Temperature	Target CPU Temperature Value.	★50℃/122℉,45℃/113℉, 46℃/114℉, 47℃/116℉, 48℃/118℉, 49℃/120℉, 51℃/123℉, 52℃/125℉, 53℃/127℉, 54℃/129℉, 55℃/131℉, 56℃/132℉,57℃/134℉, 58℃/136℉, 59℃/138℉, 60℃/140℉, 61℃/141℉, 62℃/143℉, 63℃/145℉, 64℃/147℉, 65℃/149℉
Target Fan Speed	The higher the level, the higher the fan speed.	★Level 9,Level 1, Level 2, Level 3, Level 4, Level 5, Level 6, Level 7, Level 8,
Case Open Feature	Enable or disable the feature of Case Open.	★Disabled, Enabled
OTP	Enable or disable OTP	★Enabled ,Disabled,

7.2.4 Security



Feature	Description	Options
Supervisor Password	Set or change the password for the administrator account. Only the administrator has authority to change the settings in the UEFI Setup Utility. Leave it blank and press enter to remove the password.	
User Password	Set or change the password for the user account. Users are unable to change the settings in the UEFI Setup Utility. Leave it blank and press enter to remove the password.	

Secure Boot



Feature	Description	Options
Secure Boot	Enable to support Windows 8 or later versions Secure Boot.	★Disabled, Enabled
Secure Boot Mode	Secure Boot Mode options: Standard or Custom. In Custom mode, Secure Boot Policy variables can be configured by a physically present user without full authentication	★Custom ,Standard
Install default Secure Boot keys	Please install default secure boot keys if it's the first time you use secure boot.	

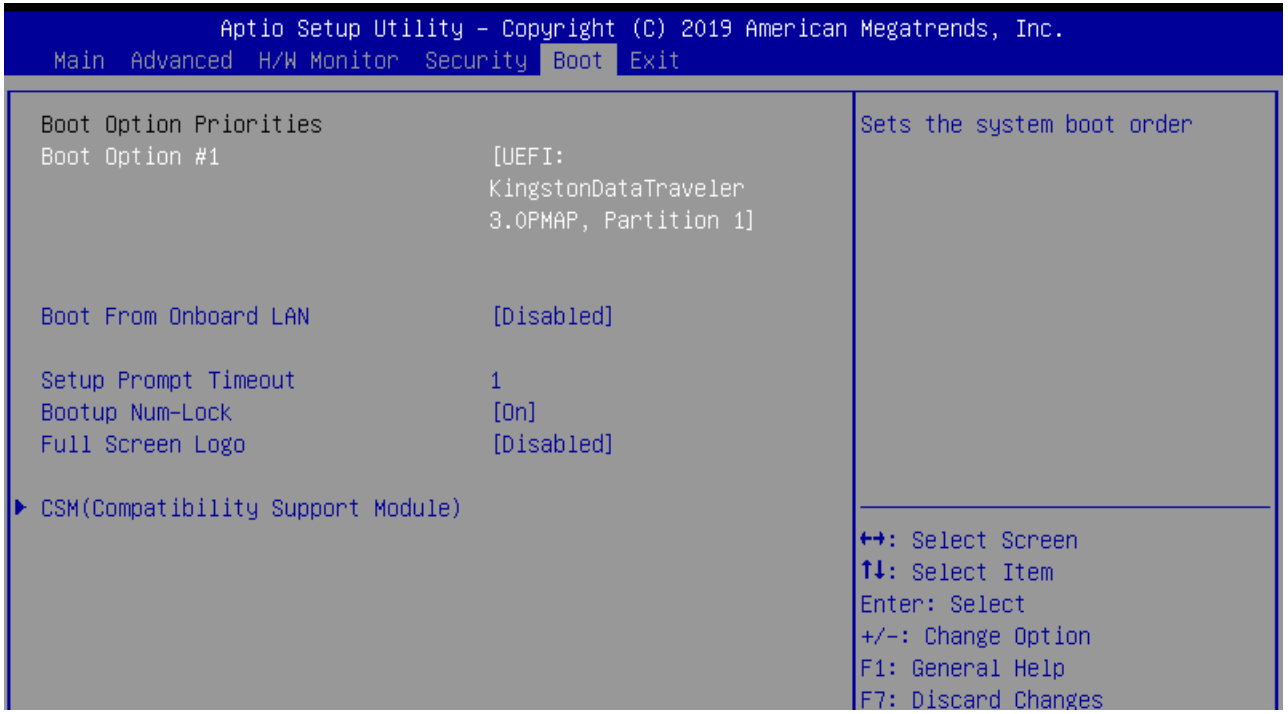
Key Management

```
Aptio Setup Utility - Copyright (C) 2019 American Megatrends, Inc.
                          Security

  Vendor Keys                          Valid

  Factory Key Provision                [Enabled]
▶ Install default Secure Boot keys
▶ Clear Secure Boot keys
▶ Export Secure Boot variables
▶ Enroll Efi Image

  Device Guard Ready
▶ Remove 'UEFI CA' from DB
▶ Restore DB defaults

  Secure Boot variable | Size| Keys| Key Source
▶ Platform Key(PK)     |    0|    0| No Keys
▶ Key Exchange Keys    |    0|    0| No Keys
▶ Authorized Signatures|    0|    0| No Keys
▶ Forbidden  Signatures| 3724|   77| Factory
▶ Authorized TimeStamps|    0|    0| No Keys
▶ OsRecovery Signatures|    0|    0| No Keys

Install factory default Secure
Boot keys after the platform
reset and while the System is
in Setup mode

←→: Select Screen
↑↓: Select Item
Enter: Select
+/-: Change Option
F1: General Help
F7: Discard Changes
F9: Load UEFI Defaults
F10: Save and Exit
ESC: Exit

Version 2.20.1271. Copyright (C) 2019 American Megatrends, Inc.
```

Feature	Description	Options
Factory Key Provision	Install factory default Secure Boot keys after the platform reset and while the System is in Setup mode.	★Disabled, Enabled
Install default Secure Boot keys	Please install default secure boot keys if it's the first time you use secure boot.	
Export Secure Boot variables	Copy NVRAM content of Secure Boot variables to files in a root folder on a file system device	
Enroll Efi Image	Allow the image to run in Secure Boot mode. Enroll SHA256 Hash certificate of a PE image into Authorized Signature Database (db)	
Restore DB defaults	Restore DB variable to factory defaults	
Platform Key(PK)	Enroll Factory Defaults or load certificates from a file: 1.Publuc Key Certificate: a)EFI_SIGNATURE_LIST b) EFI_CERT_X509 (DER) c) EFI_CERT_RSA2048 (bin) d)EFI_CERT_SHAXXX 2.Authenticated UEFI Variable 3.EFI PE/COFF Image(SHA256) Key Source: Factory, External, Mixed	
Key Exchange Keys		
Authorized Signatures		
Forbidden Signatures		
Authorized TimeStamps		
OsRecovery Signatures		

7.2.5 Boot

Use this menu to specify the priority of boot devices.

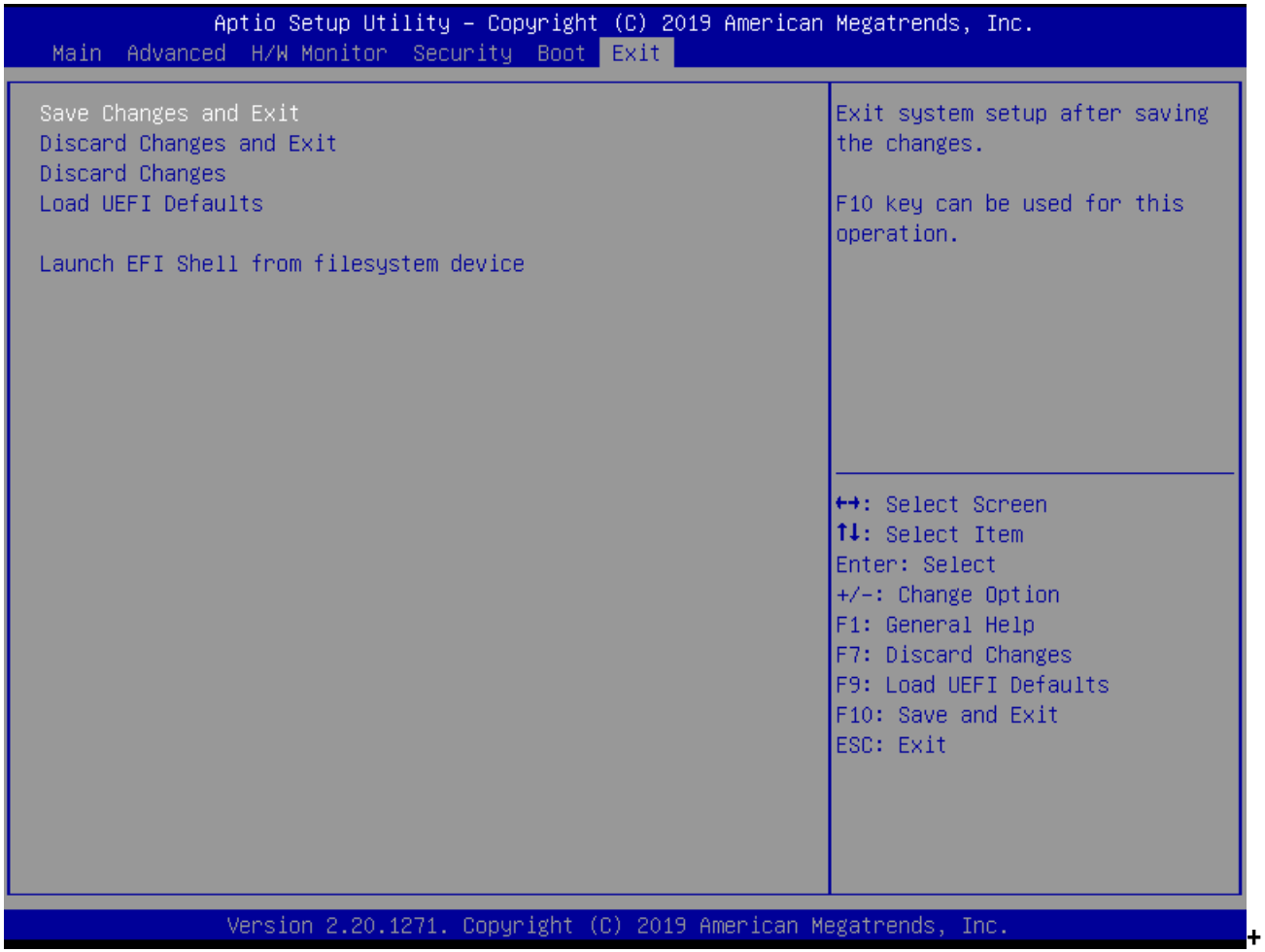


Feature	Description	Options
Boot Option #1	Sets the system boot order	★Disable, Enabled
Boot From Onboard LAN	Boot From Onboard LAN	★Disabled, Enabled
Setup Prompt Timeout	Configure the number of seconds to wait for the UEFI setup utility.	★1
Bootup Num-Lock	Select whether Num Lock should be turned on or off when the system boots up.	★On, Off
Full Screen Logo	Enable to display the boot logo or disable to show normal POST messages.	★Disabled, Enabled

CSM(Compatibility Support Module)

Feature	Description	Options
CSM	Enable to launch the Compatibility Support Module. If you are using Windows 8 or later versions 64-bit UEFI and all of your devices support UEFI, you may also disable CSM for faster boot speed.	★Disable, Enabled
CSM[Enabled]		
Launch PXE OpROM Policy	Select UEFI only to run those that support UEFI option ROM only. Select Legacy only to run those that support legacy option ROM only. Select Do not launch to not execute both legacy and UEFI option ROM.	★Legacy only ,Do not launch, UEFI only,
Launch Storage OpROM Policy	Select UEFI only to run those that support UEFI option ROM only. Select Legacy only to run those that support legacy option ROM only. Select Do not launch to not execute both legacy and UEFI option ROM.	★Legacy only ,Do not launch, UEFI only,

7.2.6 Exit


Aptio Setup Utility - Copyright (C) 2019 American Megatrends, Inc.
Main Advanced H/W Monitor Security Boot Exit
Save Changes and Exit
Discard Changes and Exit
Discard Changes
Load UEFI Defaults
Launch EFI Shell from filesystem device
Exit system setup after saving the changes.
F10 key can be used for this operation.
←→: Select Screen
↑↓: Select Item
Enter: Select
+/-: Change Option
F1: General Help
F7: Discard Changes
F9: Load UEFI Defaults
F10: Save and Exit
ESC: Exit
Version 2.20.1271. Copyright (C) 2019 American Megatrends, Inc.

Feature	Description	Options
Save Changes and Exit	Exit system setup after saving the changes. F10 key can be used for this operation.	
Discard Changes and Exit	Exit system setup without saving any changes. Esc key can be used for this operation.	
Discard Changes	Discard Changes done so far to any of setup options. F7 key can be used for this operation.	
Load UEFI Defaults	Load UEFI Default values for all the setup question. F9 key can be used for this operation.	
Launch EFI Shell from filesystem device	ASRGUID:85D2220D-BE1B-428c-8189-02549689EC42#Copy shellx64.efi to the root directory to launch EFI Shell.	

8 Troubleshooting

This section provides a few useful tips to quickly get RUBY-D811-Q370 running with success. This section will primarily focus on system integration issues, in terms of BIOS setting, and OS diagnostics.

8.1 Hardware Quick Installation

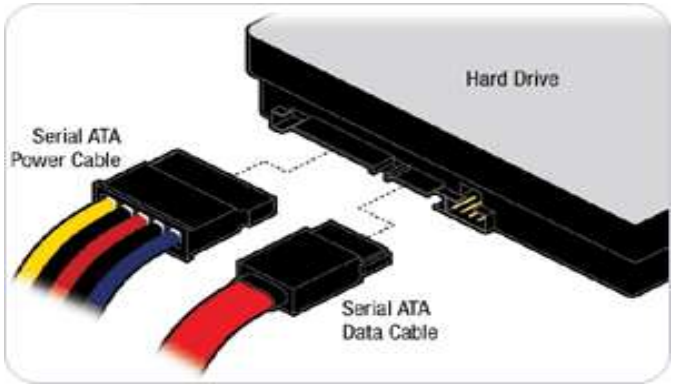
ATX Power Setting

Unlike other Single board computer, RUBY-D811-Q370 supports ATX only. Therefore, there is no other setting that needs to be set up. However, there are only two connectors that must be connected—ATX12V1 (4 pins ATX power connector) & ATXPWR1 (20 pins ATX Power Connector) on the RUBY-D811-Q370 board.

Serial ATA

Unlike IDE bus, each Serial ATA channel can only connect to one SATA hard disk at a time;

The installation of Serial ATA is simpler and easier than IDE, because SATA hard disk doesn't require setting up Master and Slave, which can reduce mistake of hardware installation.



RUBY-D811-Q370 can support four SATA interface (SATAIII, 6.0Gb/s) on board. It has SATA ports on board.

8.2 BIOS Setting

It is assumed that users have correctly adopted modules and connected all the devices cables required before turning on ATX power. DDR4 Long DIMM Memory, keyboard, mouse, SATA hard disk, VGA connector, power cable of the device, ATX accessories are good examples that deserve attention. With no assurance of properly and correctly accommodating these modules and devices, it is very possible to encounter system failures that result in malfunction of any device.

To make sure that you have a successful start with RUBY-D811-Q370, it is recommended, when going with the boot-up sequence, to hit "delete" or "F2" key and enter the BIOS setup menu to tune up a stable BIOS configuration so that you can wake up your system far well.

Loading the default optimal setting

When prompted with the main setup menu, please scroll down to “Restore Defaults”, press “Enter” and select “Yes” to load default optimal BIOS setup. This will force your BIOS setting back to the initial factory configurations. It is recommended to do this so you can be sure the system is running with the BIOS setting that Portwell has highly endorsed. As a matter of fact, users can load the default BIOS setting at any time when system appears to be unstable in boot up sequence.

8.3 FAQ

Information & Support

Question: I forgot my password of system BIOS, what am I supposed to do?

Answer: You can switch off your power supply then find the JP24 on the RUBY-D811-Q370 board to set it from 1-2 short to 2-3 short and wait 5 seconds to clean your password then set it back to 1-2 short to switch on your power supply.

CLRMOS_1：CMOS Setting

	Jumper Setting Describe
*1-2	Normal
2-3	Clean CMOS

Question: How to update the BIOS file of RUBY-D811-Q370

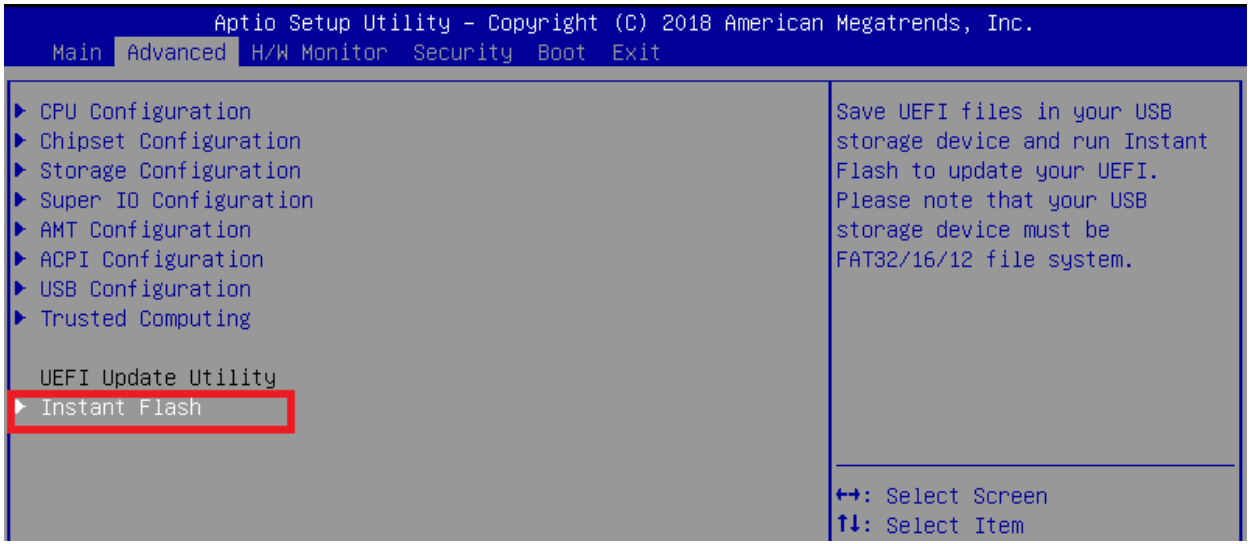
Answer: 1. Please visit web site of Portwell download center as below hyperlink

http://www.portwell.com.tw/support/download_center.php

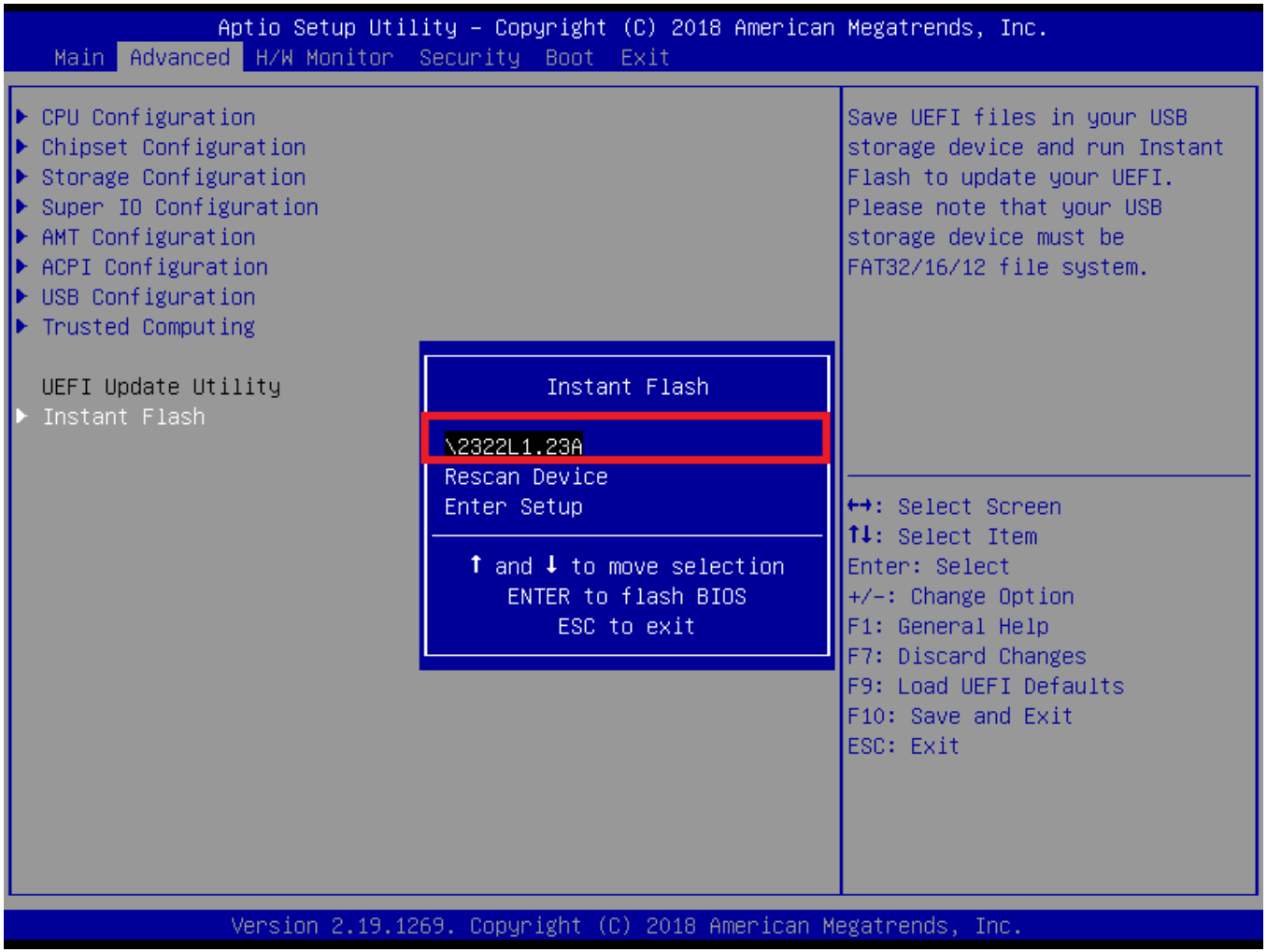
Registering an account in advance is a must. (The E-Mail box should be an existing Company email address that you check regularly.)

http://www.portwell.com.tw/member/newmember.php

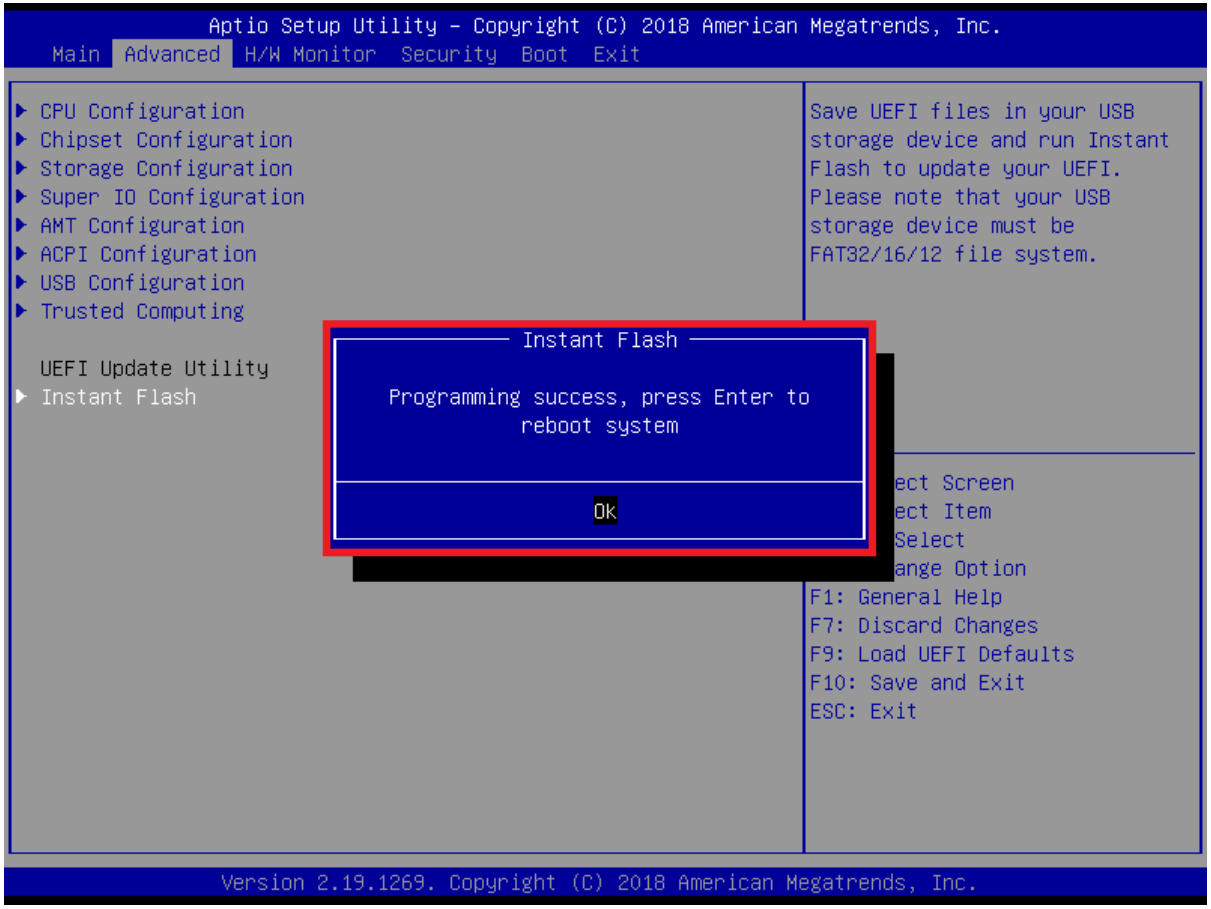
2. Type in your User name and password and log in the download center.
3. Select "Search download" and type the keyword "RUBY-D811-Q370".
4. Find the "BIOS "page and download the ROM file and unzip file to USB flash drive (FAT 32 / 16 format).
5. Boot into BIOS and switch to "Advanced" page then select" Instant Flash".



6. .Select “xxxx.23A” file then start updating BIOS.



7. When you see the "Programming success" message, which means the BIOS update processes finished. Please cut the AC power off and **wait for 10 seconds** before powering on.



Question: What are the display options while using RUBY-D811-Q370 board?

Answer: - The RUBY-D811-Q370 supports VGA、HDMI and DP display output.

Note:

Please visit our Download Center to get the Catalog, User manual, BIOS, and driver files.

http://www.portwell.com.tw/support/download_center.php

If you have other additional technical information or request which is not covered in this manual, please fill in the technical request form as below hyperlink.

http://www.portwell.com.tw/support/problem_report.php

We will do our best to provide a suggestion or solution for you.

Thanks

9 Portwell Software Service

1. If you have customized requirements of BIOS, you can contact person of our company or branch.
2. If you have requirements of WDT、GPIO APP, you can contact our headquarter or branch, and we can render you assistance on developing.

Portwell Worldwide:	
Portwell, Inc.	E-mail: info@portwell.com.tw
Shanghai Portwell	E-mail: info@portwell.com.cn
Portwell Japan, Inc	E-mail: info@portwell.co.jp
American Portwell Technology	E-mail: info@portwell.com
European Portwell Technology	E-mail: info@portwell.eu
Portwell UK Ltd.	E-mail: info@portwell.co.uk
Portwell Deutschland GmbH	E-mail: info@portwell.eu
Portwell India Technology	E-mail: info@portwell.in
Portwell Korea, Inc.	E-mail: info@portwell.co.kr
Portwell Latin America	E-mail: vendas@portwell.com.br

10 Industry Specifications

10.1 Industry Specifications

The list below provides links to industry specifications that apply to Portwell modules.

Low Pin Count Interface Specification, Revision 1.0 (LPC) http://www.intel.com/design/chipsets/industry/lpc.htm
Universal Serial Bus (USB) Specification, Revision 2.0 http://www.usb.org/home
PCI Specification, Revision 2.3 https://www.pcisig.com/specifications
Serial ATA Specification, Revision 3.0 http://www.serialata.org/
PCI Express Base Specification, Revision 2.0 https://www.pcisig.com/specifications