AICOTEK

Empowering the world with AI solutions

BPC-221
User Manual



www.aicotek.com
info@aicotek.com

Safety Precautions 1

Chapter 1 General Introduction 2

1.1 BPC-221 Introduction 3

1.1.1 Product Specifications 3

1.1.2 Dimensions 4

1.1.3 I/O Introduction 5

1.1.4 External I/O 6

1.1.5 CMOS Clear Select 12

Chapter 2 H/W Installation 13

2.4 Installation 14

2.4.1 Memory Installation 14

2.4.2 M.2 B Key 2280 SSD Installation 15

2.4.3 mini PCIE Installation 16

2.4.4 M.2 E Key 2230 Installation 16

2.4.5 Wall Mount Installation (Optional A) 17

2.4.6 Wall Mount Installation (Optional B) 18

2.4.7 Din Rail Installation (Optional C) 19

2.4.8 Nano SIM card Installation 19

Chapter 3 BIOS Settings 20

3.1 Introduction 21

3.2 BIOS Update 21

3.2.1 General BIOS Update 21

3.2.2 Enter BIOS Setup Utility and Basic BIOS Operation 24

3.2.3 Save Changes 25

3.2.4 Setting Date & Time 26

3.2.5 Set BIOS password 26

3.2.6 Set Up Power-on Self & Timed Power On 27

3.2.7 Set Up Fan Control & Hardware Monitoring 27

3.2.8 Setting Boot Order of Startup Disk 28

3.2.9 Setting PXE Boot 28

3.3 Description of the BIOS Options 28
3.3.1 Main 29
3.3.2 Advanced 30
3.3.3 Security 60
3.3.4 Boot 62
3.3.5 Save & Exit 63

Chapter 4 PL1 & PL2 & PL3 Sample Code 65
4.1 Panel Light Control 66
4.2 Switch the Panel Lights 67
4.3 Read and Confirm Light State 68
4.4 Functions Used for Light Control 69
4.5 Build and Troubleshooting 70
4.6 Application Source 72

Chapter 5 Watchdog Control Sample Code 74
5.1 Watchdog Control 75
5.2 Operating the Watchdog 76
5.3 Function Reference 77
5.4 Feeding Sequence and Source Project 78
5.5 Application Source 80

Appendix A BIOS Hidden Menu Access Tutorial 82
A.1 Enter BIOS Setup 83
A.2 Reveal the Debug Setup Option 84
A.3 Enable Access to Hidden Menus 85
A.4 Open the Advanced Tab 86

Appendix B Alarm and Protection Functions 87
B.1 Buzzer Alarm Indications 88
B.2 Power Protection Specifications 88

Safety Precautions

1. Please read and follow the safety precautions before you are going to use it.
2. Pay attention to the labels on the product.
3. Make sure to use in an environment that meets the design specifications, otherwise, malfunction or partial damage caused by non-compliance with relevant regulations is not covered under the product quality guarantee.
4. Please unplug the power cord and do not use liquids to clean the PC.
5. Please keep the PC in a safe space to prevent it from falling and damaging its components.
6. Please keep the power cord in a safe location to avoid causing personal injury.
7. Please do not bundle control wires, communication cables and power wires together, it would be better to keep a distance of at least 100mm between them to avoid mutual interference.
8. It is recommended to use wires with isolation, especially in environments with severe electromagnetic interference.
9. Please disconnect it from the power socket if the PC is not used for a long time.
10. Please make sure that no liquids enter the device to avoid the risk of fire or short circuit.
11. Please disconnect the power cord before opening the computer case.
12. Please clean the dust regularly.
13. Please ask for technical support and return the PC to RMA:
    The power cord or plug is damaged;
    Liquid has entered the interior of the PC;
    PC doesn't work;
    PC is damaged;
    Physical damage on the PC.

Chapter 1

Genernal Introduction

This chapter gives background information on the BPC-221.

1.1 BPC-221 Introduction

The BPC-221 is a compact fanless Box PC featuring a wide range of I/O interfaces, including 4 × USB 3.2 Gen1, 4 × Gigabit LAN, 1 × HDMI supporting 4K@30Hz, 1 × RS232, 1 × RS485, 1 × Audio Combo Jack, and 1 × SIM card slot. It supports DC 12–24V wide voltage input, as well as optional Wi-Fi, Bluetooth, and 4G/5G connectivity, providing flexible connectivity for industrial automation, data acquisition, and edge computing applications.

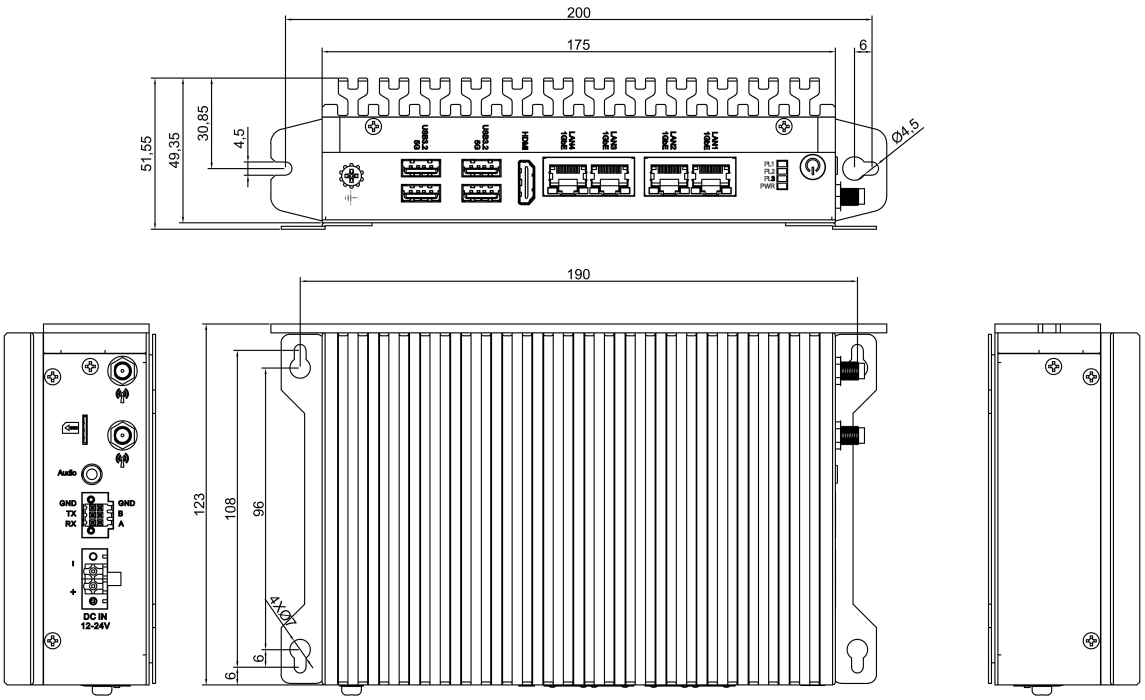
1.1.1 Product Specifications

Model Name		BPC-221
System	CPU	Intel Meteor Lake-U Processor 125U
	Memory	1 x DDR5 SO-DIMM 5600MT/s, Max. 32 GB
	Storage	1 x M.2 M Key 2280 PCIe4.0 x4 NVMe SSD,
	Display	1 x HDMI, support 4K@30Hz
	USB	4 x USB3.2 Gen1 Type-A
	LAN	4 x RJ45, 10/100/1000Mbps
	Expansion	1 x M.2 E Key 2230 for WiFi & Bluetooth, 1 x Mini PCIe for 4G/5G
	Audio	1 x Audio Combo Jack(CTIA)
	COM	1 x RS232, 1 x RS485
	SIM Card	1 x Nano SIM Card slot
	TPM	TPM2.0
OS	OS Support	Windows10/11, Linux
Power	Input Voltage	DC IN, 12-24V
Mechanical	Dimensions	175x123x48.6mm (6.9x4.8x1.9 in)
	Weight	1.3 kg

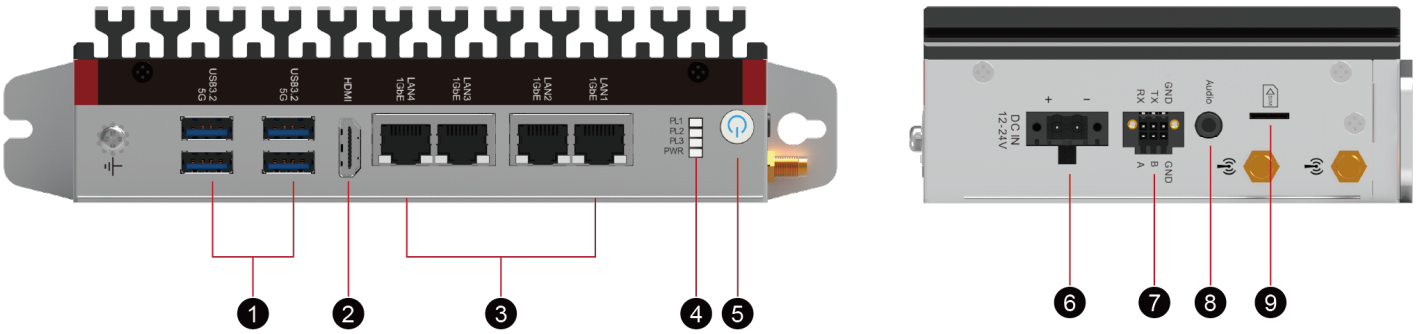
Model Name		BPC-221
Environment	Operating Temperature	0-60℃ (32-140°F)
	Storage Temperature	-20-75℃ (-4-167°F)
	Relative Humidity	10%-90% (non-condensing)
	Vibration	Operating random vibration test 5~500Hz, 1Grms @with SSD, follows IEC 60068-2-64
	Shock	Operating 10G peak acceleration (11ms duration), follows IEC 60068-2-27
	EMC	CE FCC

1.1.2 Dimensions

175 x 123 x 49.5mm



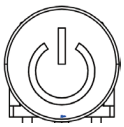
1.1.3 I/O Introduction



1	4 x USB3.2 Gen1	6	DC IN, 12V-24V
2	1 x HDMI, 4K@30Hz	7	1 x RS232, 1 x RS485
3	4 x RJ45, 1000Mbps	8	Audio Combo Jack
4	PWR & PL1 & PL2 & PL3 LED	9	Nano SIM Card slot
5	Power Button		

1.1.4 External I/O

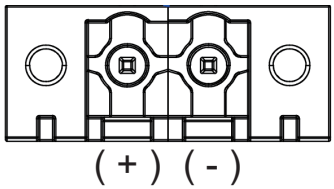
1 Power Button



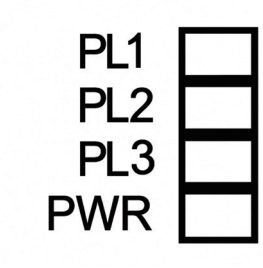
Operation	Function
Press One Times	Turn on or off
Long press for 4 seconds	Forced shutdown

2 Power Input Connector

BPC-221 is designed with the phoenix connector and 12-24V DC input. (-) stands for GND; (+) stands for +12-24V.



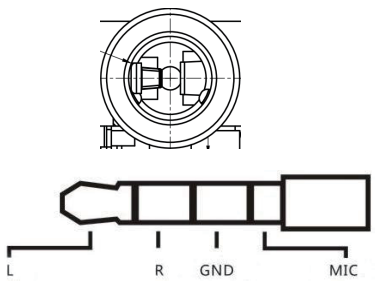
3 LED



PIN	LED	Status	Description
1	Power Led	off	The product is power off
		on(Yellow)	The product is power on
2	PL3	(Yellow)	User programmable led by setting the GPO
3	PL2	(Yellow)	User programmable led by setting the GPO
4	PL1	(Yellow)	User programmable led by setting the GPO

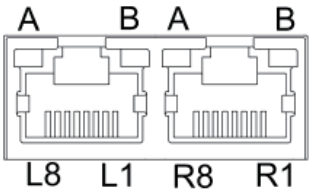
4 Audio

3.5mm audio combo jack, supporting left and right stereo audio output and microphone input.



5 LAN

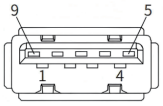
BPC-221 is equipped with four Ethernet controllers that are fully compliant with IEEE 802.3u 10/100/1000 Mbps CSMA/CD standards. LAN1, LAN2 , LAN3 , LAN4 are all equipped with i210 Ethernet controller. The Ethernet port provides a standard RJ-45 jack connector with LED indicators on the front side to show its Active/Link status (Green LED) and Speed status (Yellow LED).



PIN	Definition		PIN	Definition	
L1	MDI0_1+		R1	MDI0_2+	
L2	MDI0_1-		R2	MDI0_2-	
L3	MDI1_1+		R3	MDI1_2+	
L4	MDI2_1+		R4	MDI2_2+	
L5	MDI2_1-		R5	MDI2_2-	
L6	MDI1_1-		R6	MDI1_2-	
L7	MDI3_1+		R7	MDI3_2+	
L8	MDI3_1-		R8	MDI3_2-	
A	Active LED	ACT:Twinkling Yellow	B	Speed LED	1000M: Turn Yellow
		Only LINK: Lights On			100M: Turn Green
		NO LINK: Lights Off			10M: Lights Off

6 USB

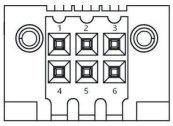
BPC-221 supports four USB3.2 Gen2 interfaces.



PIN	Definition	PIN	Definition
1	VBUS	6	SSRX+
2	DATA-	7	GND-DRAIN
3	DATA+	8	SSTX-
4	GND	9	SSTX+
5	SSRX-		

7 COM

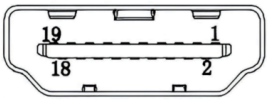
RX/TX are used for RS-232(COM2) communication; A/B are used for RS-485(COM1) communication.



PIN	Definition	PIN	Definition
1	RX	4	A
2	TX	5	B
3	GND	6	GND

8 HDMI

HDMI support 4K@60Hz.



PIN	Definition	PIN	Definition
1	TMDS DATA2+	11	TMDS Clock Shield
2	TMDS DATA2 Shield	12	TMDS Clock-
3	TMDS DATA2-	13	NC
4	TMDS DATA1+	14	NC
5	TMDS DATA1 Shield	15	SCL
6	TMDS DATA1-	16	SDA
7	TMDS DATA0+	17	GND
8	TMDS DATA0 Shield	18	+5V Power
9	TMDS DATA0-	19	Hot Plug Detect
10	TMDS Clock+		

9 M.2 E key 2230

M.2 E Key 2230 for WiFi & Bluetooth.



PIN	Definition	PIN	Definition
1	GND	2	3.3V
3	USB_D+	4	3.3V
5	USB_D-	6	/
7	GND	8	/
9	/	10	/
11	/	12	/
13	GND	14	/
15	/	16	/
17	/	18	GND
19	GND	20	/
21	/	22	/
23	/	24	/
25	/	26	/
27	/	28	/
29	/	30	/
31	/	32	/
33	GND	34	/
35	PETP0	36	/
37	PETN0	38	/
39	GND	40	/
41	PERP0	42	/
43	PERN0	44	/
45	GND	46	/
47	REFCLKP0	48	/
49	REFCLKN0	50	/
51	GND	52	PERST0*
53	CLKREQ0*	54	BT_RF_KILL_N
55	PEWAKE0*	56	WIFI_RF_KILL_N
57	GND	58	/
59	/	60	/
61	/	62	/
63	GND	64	/
65	/	66	/
67	/	68	/
69	GND	70	/
71	/	72	3.3V
73	/	74	3.3V
75	GND		

10 M.2 M key 2280

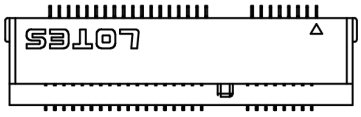
M.2 M Key 2280 PCIe4.0 x4 NVMe SSD.



PIN	Definition	PIN	Definition
1	GND	2	3.3V
3	GND	4	3.3V
5	PERN3	6	/
7	PERP3	8	/
9	GND	10	/
11	PETN3	12	3.3V
13	PETP3	14	3.3V
15	GND	16	3.3V
17	PERN2	18	3.3V
19	PERP2	20	/
21	GND	22	/
23	PETN2	24	/
25	PETP2	26	/
27	GND	28	/
29	PERN1	30	/
31	PERP1	32	/
33	GND	34	/
35	PETN0	36	/
37	PETP0	38	/
39	GND	40	/
41	PERN0	42	/
43	PERP0	44	/
45	GND	46	/
47	PETN0	48	/
49	PETP0	50	PERST#
51	GND	52	CLKREQ#
53	REFCLKN	54	PEWake#
55	REFCLKP	56	/
57	GND	58	/
59	/	60	/
61	/	62	/
63	/	64	/
65	/	66	/
67	/	68	/
69	/	70	3.3V
71	GND	72	3.3V
73	GND	74	3.3V
75	GND		

11 Mini PCIe

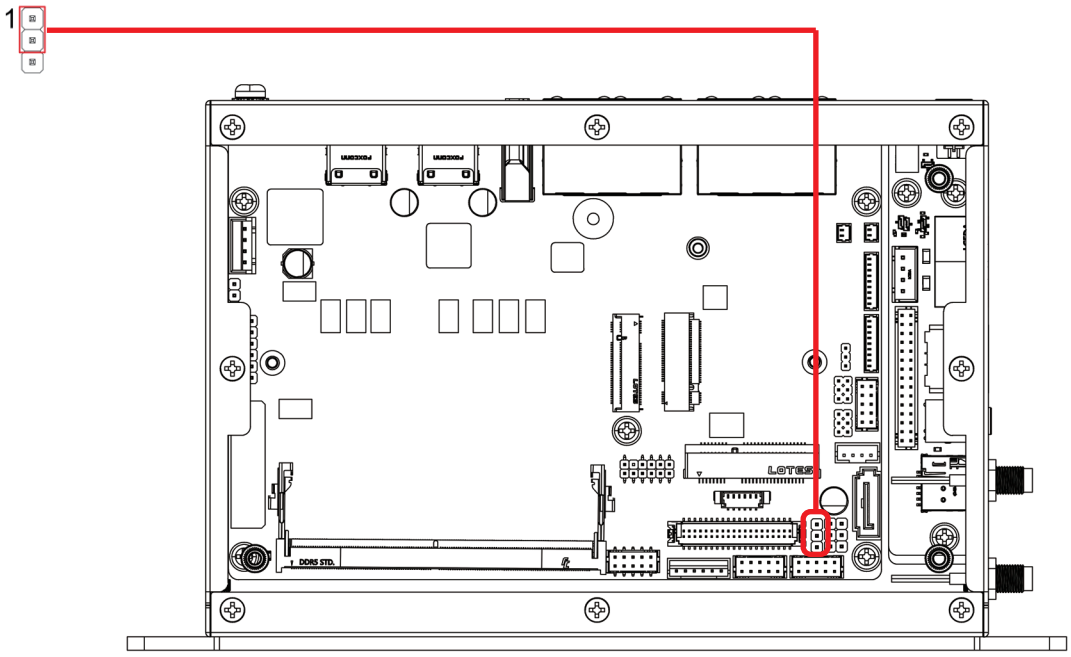
Mini PCIe for 4G/5G.



PIN	Definition	PIN	Definition
1	/	2	3.3V
3	/	4	GND
5	/	6	1.5V
7	CLKREQ#	8	UIM1_PWR
9	GND	10	UIM1_DAT
11	REFCLK-	12	UIM1_CLK
13	REFCLK+	14	UIM1_RST
15	GND	16	UIM1_VPP
17	/	18	GND
19	/	20	/
21	GND	22	PERST#
23	PERN0	24	3.3V
25	PERP0	26	GND
27	GND	28	1.5V
29	GND	30	SMB_CLK
31	PETN0	32	SMB_DAT
33	PETP0	34	GND
35	GND	36	USB_D-
37	GND	38	USB_D+
39	3.3V	40	GND
41	3.3V	42	/
43	GND	44	/
45	/	46	/
47	/	48	1.5V
49	/	50	GND
51	/	52	3.3V
53	GND	54	GND
55	/	56	/

1.1.5 CMOS Clear Select

Setting	Function
1-2(Default)	Normal
2-3	Clear CMOS



Chapter 2

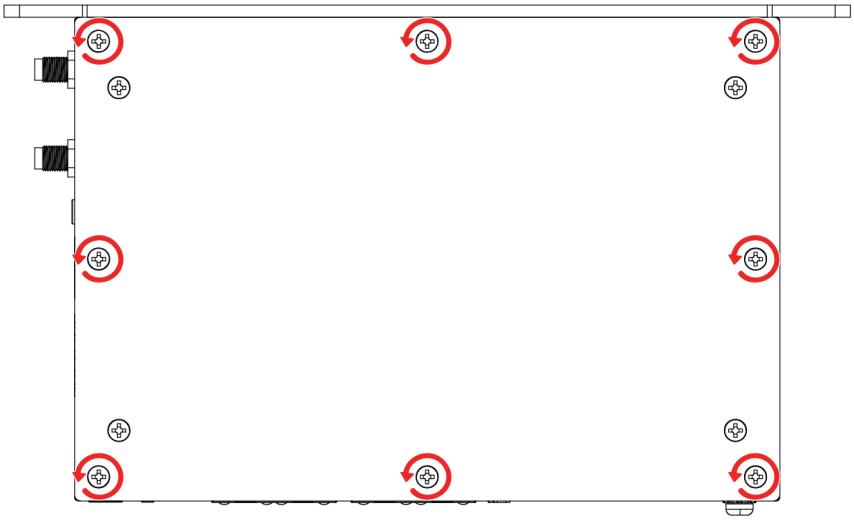
H/W Installation

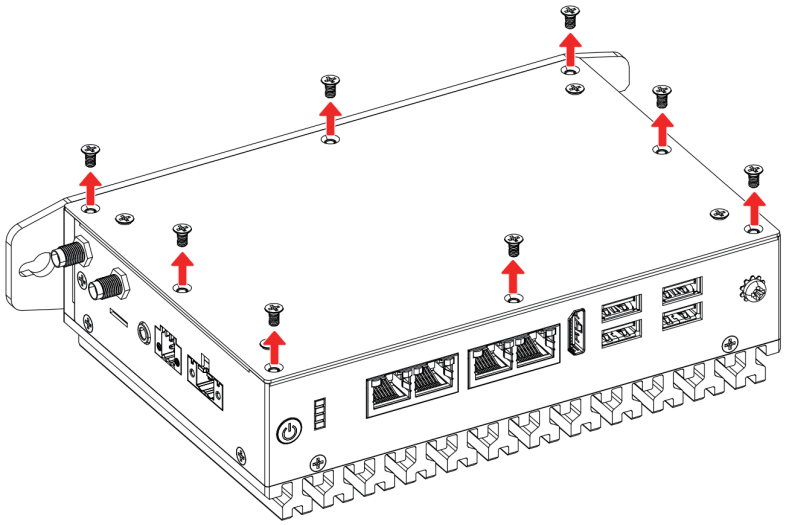
This chapter introduces the external I/O and the installation of BPC-221 hardware.

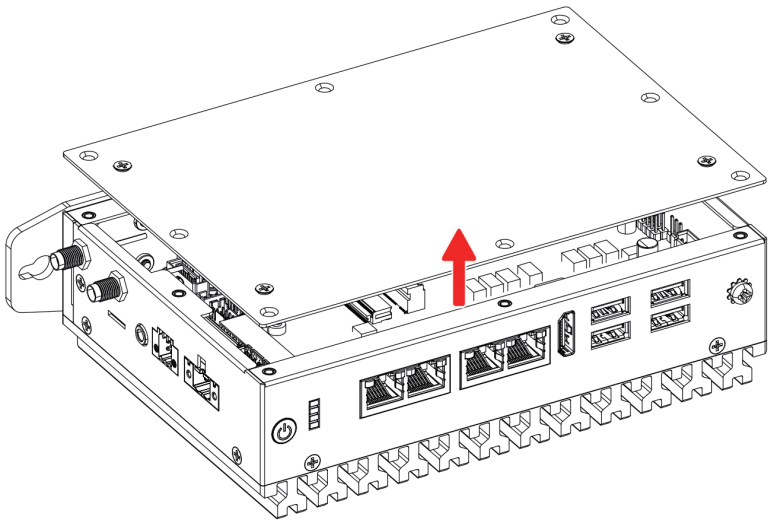
2.4 Installation

2.4.1 Memory Installation

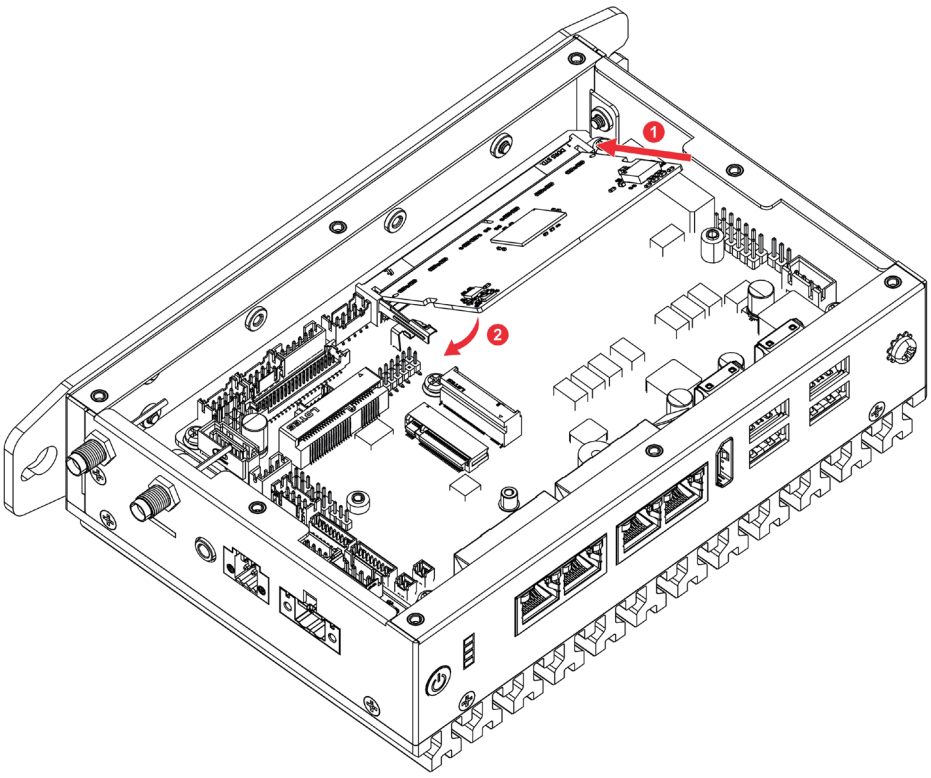
1. Loosen screws, open bottom case





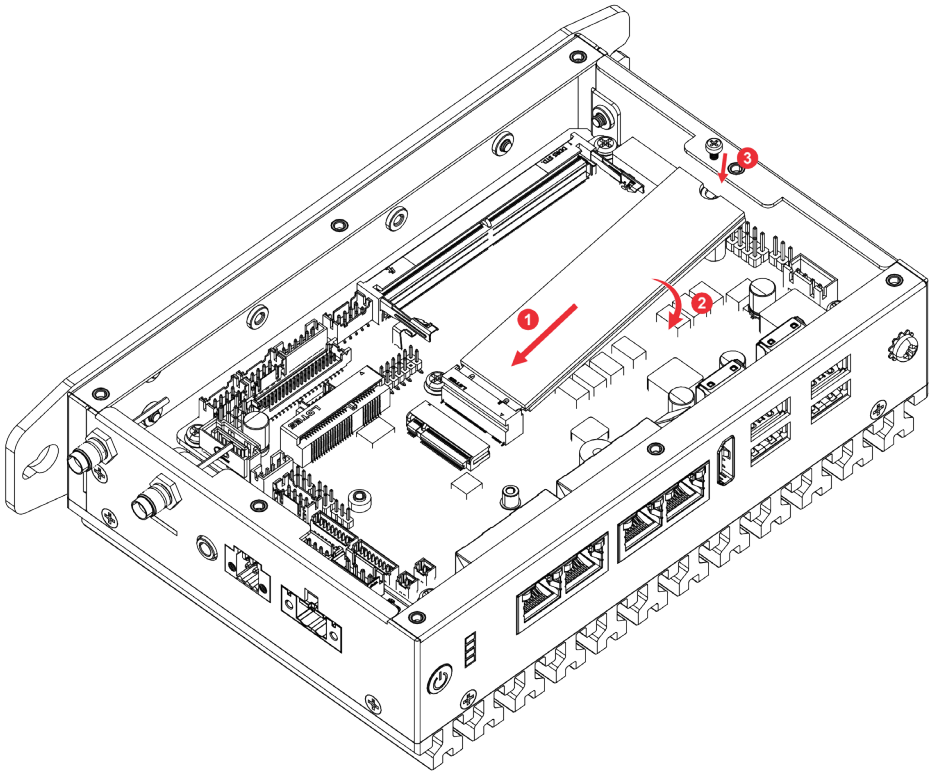


2. Install memory and tighten up the screws again.



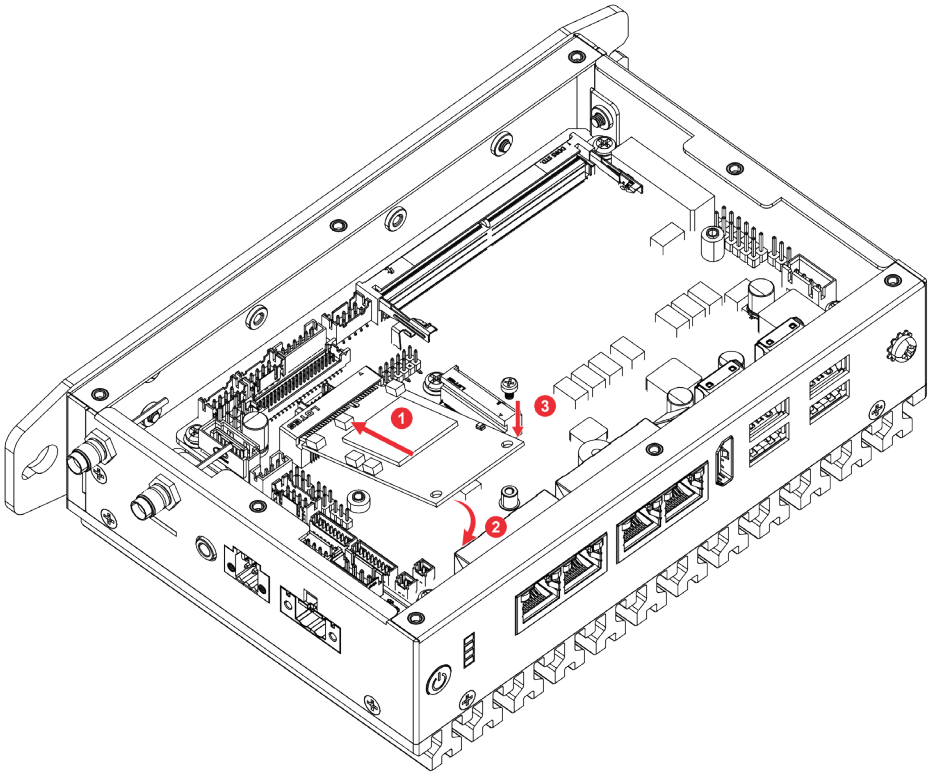
2.4.2 M.2 B Key 2280 SSD Installation

1. Repeat step 1 of Chapter 2.4.1 to open the back cover.Install M.2 Bkey 2280 SSD(NVME) and tighten up the screws again.



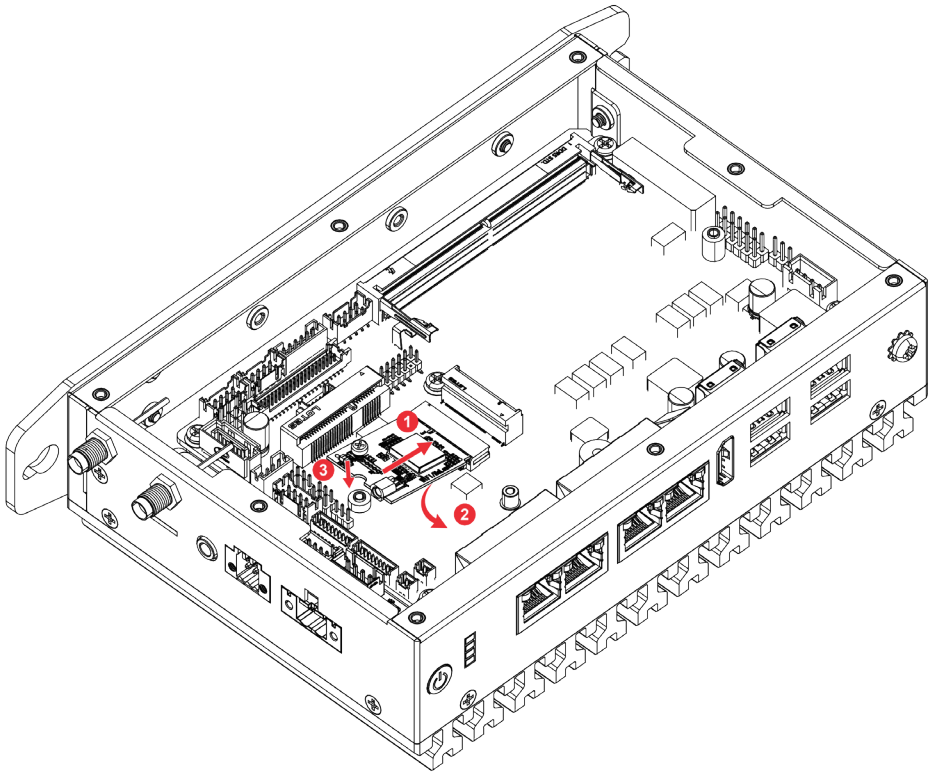
2.4.3 mini PCIE Installation

1. Repeat step 1 of Chapter 2.4.1 to open the back cover.Install mini PCIE and tighten up the screws again.



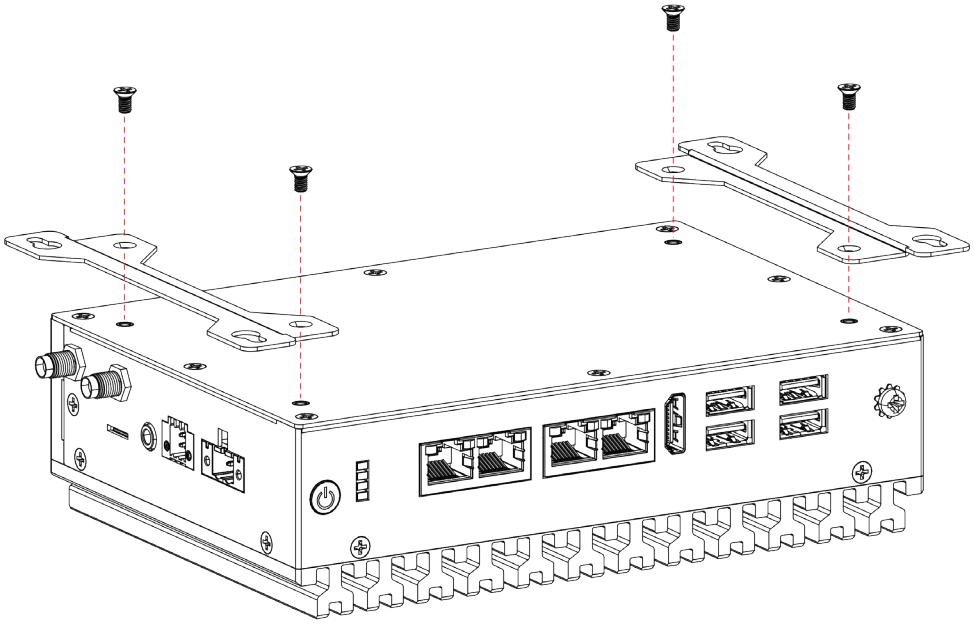
2.4.4 M.2 E Key 2230 Installation

1. Repeat step 1 of Chapter 2.4.1 to open the back cover.Install M.2 Ekey 2230 card and tighten up the screws again.

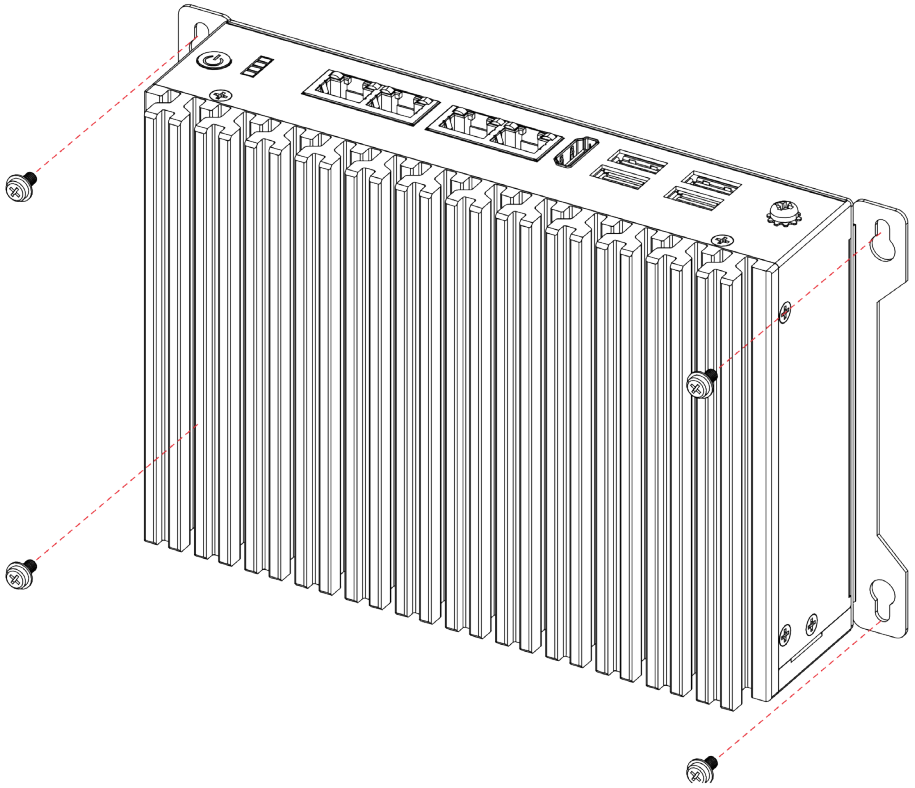


2.4.5 Wall Mount Installation (Optional A)

1. Loosen four side screws of the top case and screw back with the wall mount bracket.

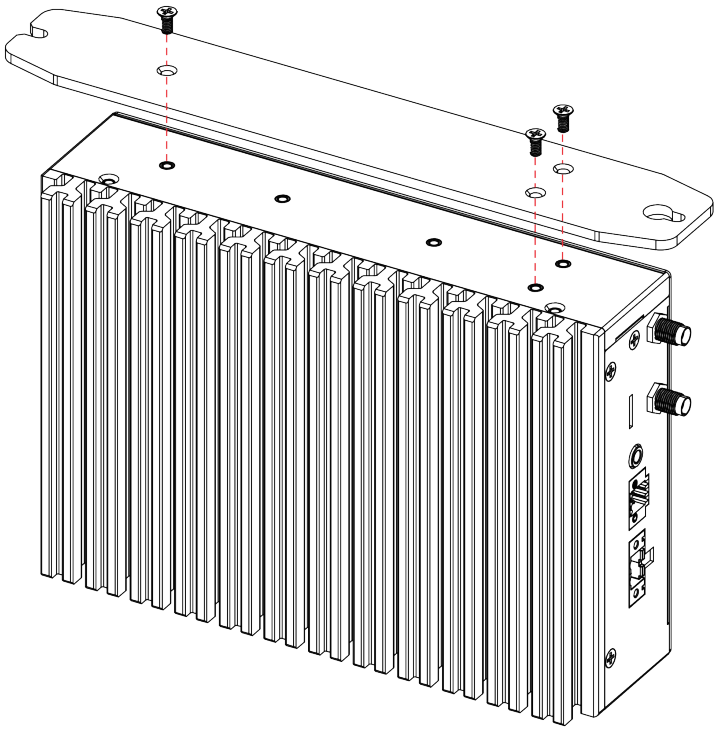


2. Screw down the four wall mount screws to the table.

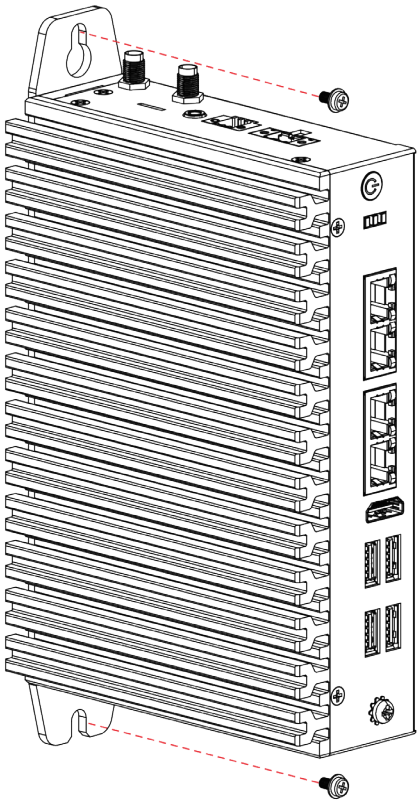


2.4.6 Wall Mount Installation (Optional B)

1. Loosen three side screws of the top case and screw back with the wall mount bracket.

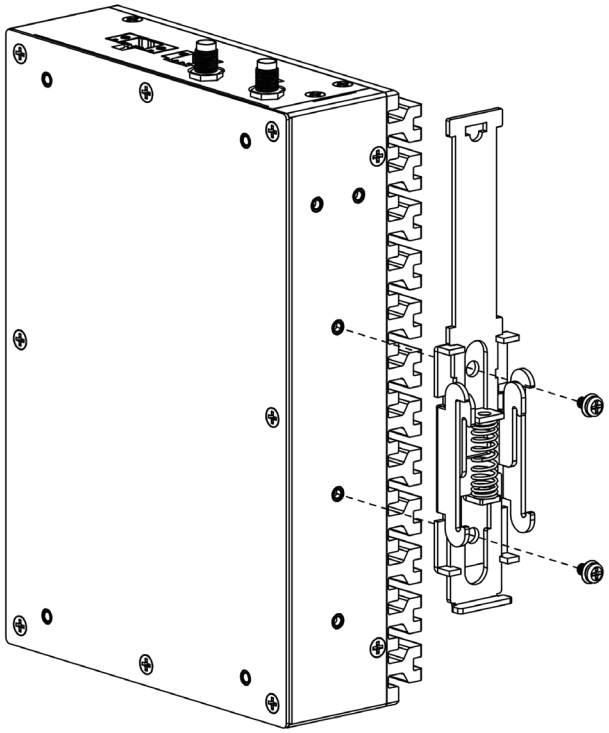


2. Screw down the two wall mount screws to the table.



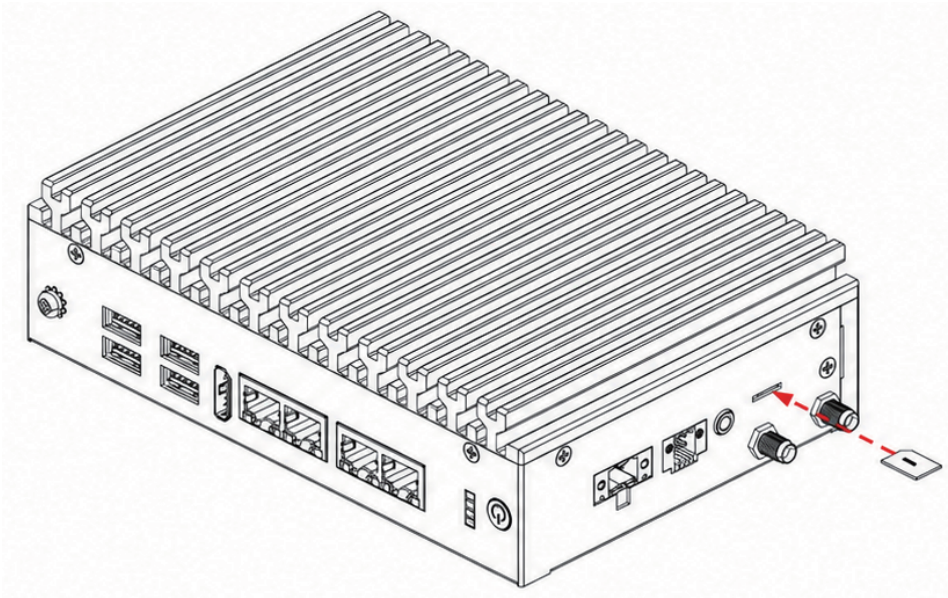
2.4.7 Din Rail Installation (Optional C)

1. Loosen two pcs Din rail screws on back cover.Use Din rail screws to mount Din rail.



2.4.8 Nano SIM card Installation

1. Insert the Nano SIM card in the direction shown in the illustration.



Chapter 3

BIOS Settings

3.1 Introduction

BIOS (Basic Input Output System) is the basic program responsible for mother-board POST (Power-OnSelf-Test), initializing the central processing unit, memory and other hardware, detecting input/output devices and booting the system.
BIOS of this platform is developed based on AMI UEFI (Unified Extensible Firm-ware Interface) BIOS, which is safe, stable, reliable and provides rich functions.

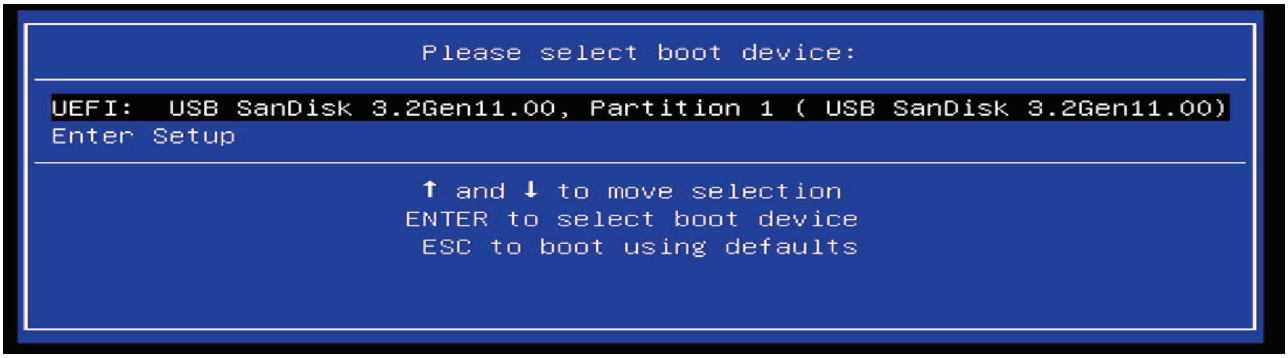
3.2 BIOS Update

It supports to update BIOS under UEFI Shell and Windows 10/11 environment.

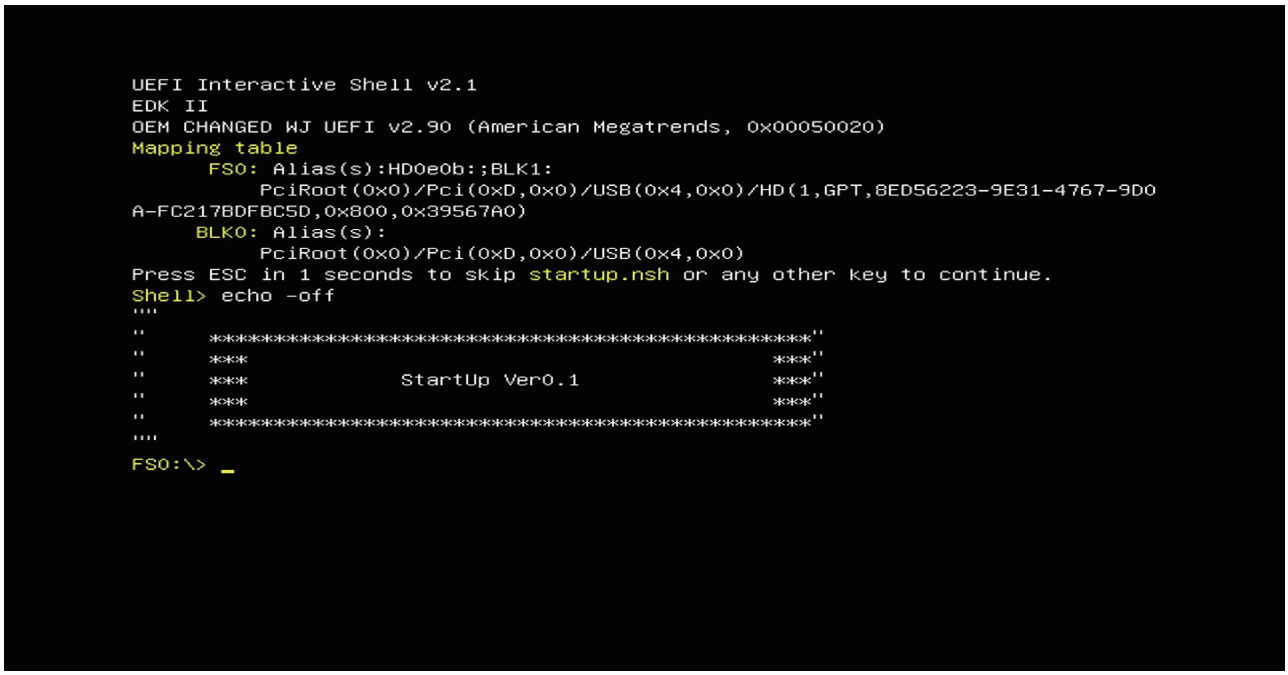
3.2.1 General BIOS Update

3.2.1.1 Updating BIOS under UEFI Shell Environment

1)Connect the U disk which stores BIOS update tool and BIOS file to the mother-board. Turn on the board, while screen displays POST interface, press “F7” key. After this, screen shows boot device selection menu, select “UEFI: U disk name” to boot.



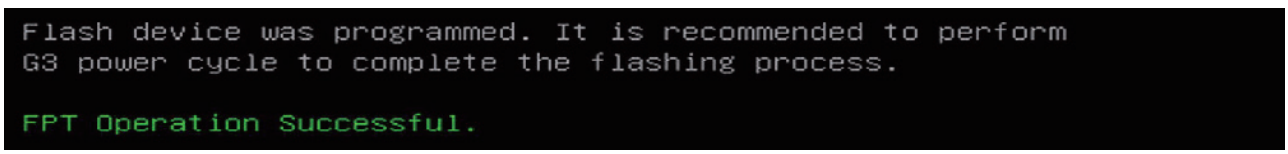
2)After entering into UEFI Shell, Find USB disk number under Device mapping table as shown in the figure below. In the figure, USB is mapped to fs0:(while there are more than one storage devices, the mapping number may not be [0]), and then type “fs0:” and press “Enter” key. If startup.nsh exists, the system will automatically switch to the USB mapped disk.



3)Use the “cd [folder path]” command to enter the directory where the update tool and BIOS files are store. After this, type the command “fpt.efi /f [BIOS File Name]” or “up.nsh” if you have flashing package to update BIOS, where “/f” means flash the ROM, “/bios” means flash the BIOS region only. Some “up.nsh” contains other flashing commands, for example, “TR5511Tool_2.1.efi /f [5511 File Name] /check” to update 5511, “AMIDEEFIx64.efi /DMS” to download the original DMI information of the motherboard, and then use AMIDEEFIx64.efi to write it back in order to retain the DMI information when updating the BIOS.



4)After the step 3), it goes to update BIOS. Do not reboot, turn off and disconnect power until screen shows green strings” FPT Operation Successful”.



5)Shut down and disconnect power source for several seconds, and restart the motherboard. The first-time power on, it will reboot for loading defaults and others normally. It is recommended that press “F3-Enter-F4-Enter” in BIOS Setup Utility to load defaults again.

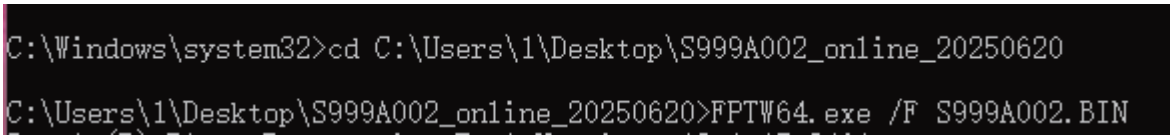
⚠ Do not reboot, turn off, disconnect power or perform other actions while updating BIOS.

3.2.1.2 Updating BIOS under Windows Environment

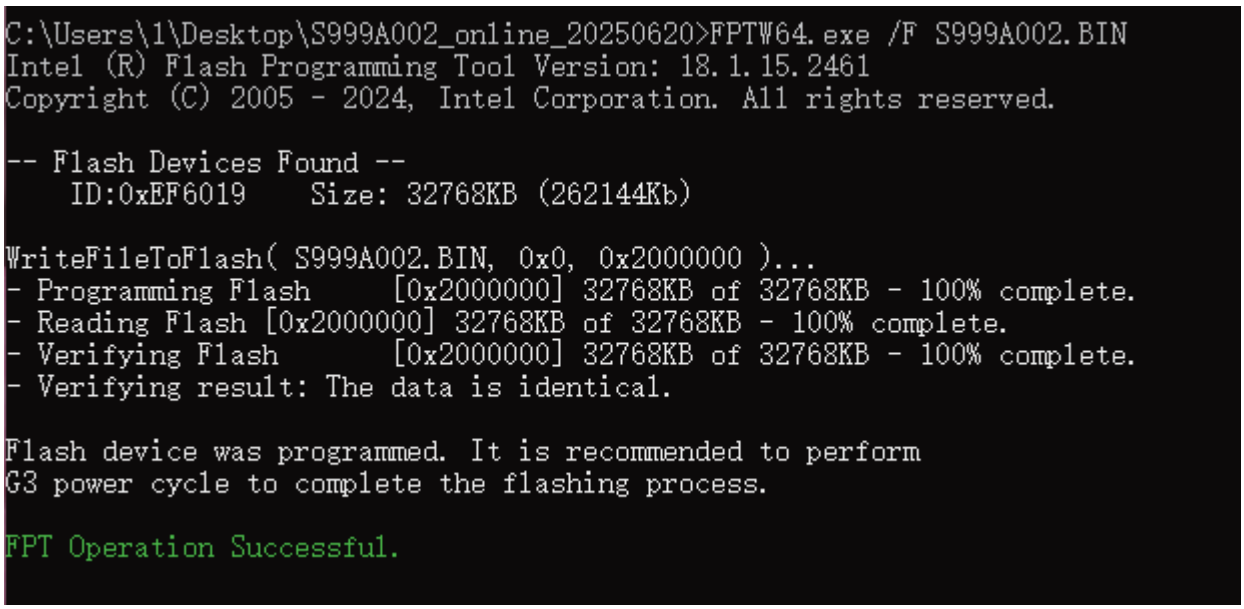
1)Run Windows command prompt as administrator.

2)Type “driver letter:”, for example “C:”, enter to the partition where stores update tools and BIOS file. Use the “cd [folder path]” command to enter the directory.

3)Type the command “FPTW64.exe /f [BIOS File Name] /BIOS” or “up.bat” if you have flashing package to update BIOS， where “/f” means flash the ROM, “/bios” means flash the BIOS region only. Some “up.bat” contains other flashing commands similar to shell environment above.



4)After the step 3), it goes to update BIOS. Do not reboot, turn off and disconnect power until screen shows strings” Process completed”.



5)Shut down and disconnect power source for several seconds, and restart the motherboard. The first-time power on, it will reboot for loading defaults and others normally. It is recommended that press “F3-Enter-F4-Enter” in BIOS Setup Utility to load defaults again.

⚠ Do not reboot, turn off, disconnect power or perform other actions while updating BIOS.

3.2.1.3 Updating BIOS under Linux Environment

Please refer to the update method under Windows. Administrator under Windows means that Linux Root privilege, command prompt corresponds to Linux Shell.

Key Notes

1)For NICs with a PHY (Physical Layer), When flashing firmware, ensure that the flashing command includes address retention. If the MAC address is not preserved, you must re-flash the MAC address or contact technical support to obtain a MAC-retained flashing package.

2)If the motherboard has pre-programmed DMI information (e.g., Serial Number, UUID) that needs to be preserved, confirm whether the flashing command retains this data. If not, re-flash the DMI information or request a DMI-retained flashing package from technical support.

3.2.2 Enter BIOS Setup Utility and Basic BIOS Operation

AMI UEFI BIOS supports Multiple function settings with its BIOS Setup Utility. Press “Delete” to enter BIOS Setup Utility while POST screen shows.

In the BIOS Setup Utility, the meaning of different font colors and signs is as follow：

Color/Sign	Meaning
Blue	Items allowed to be set or changed
Black	Items that cannot be set or changed
White	Items currently selected
►	There are submenu pages for this option

The operation keys are described as follows:

Control button	
→←	Select Screen
↑↓	Select Item
<Enter>	Select
+/-	Change Opt.
F1	General Help
F2	Previous Values
F3	Optimized Defaults
F4	Save & Exit
Pause	Print Screen
<Esc>	Exit

3.2.3 Save Changes

Operation steps:

1) Press F4 to save the settings and exit in the BIOS Setup Utility.

3.2.4 Setting Date & Time

Operation steps:

1) Enter BIOS, and in the main menu page. Select system date and system time by "↑↓" key.

2) On the selected item, change the selected date or time by entering a numeric value or pressing "+ / -". Date format is "month / day / year", time format is "hour / minute / second".

3.2.5 Set BIOS password

When you set up an administrator password, it is recommended that you enter the administrator password when you enter the BIOS setup program to avoid some BIOS information or settings that are unavailable because user rights are insufficient (no password is entered as a user).

When you set a user password, you must use the user password to access the BIOS setup program, which means that you need to enter the user password when booting the operating system (using the BIOS boot feature).

Set/Change the Administrator Password Procedure:

1) Select "Security - Administrator Password";

2) Enter the password you want to set in the "Create New Password" window and press enter. You will need to enter your current password to confirm when changing your administrator password before the "Create New Password" window;

3) Enter the password you want to set again to make sure it is entered correctly.

When you want to delete an administrator password, you should change the password by pressing carriage return directly in the Create New Password window to clear the password.

Set/Change the User Password Procedure:

1) Select "Security - User Password";

2) Enter the password you want to set in the “Create New Password” window that appears and press enter. You will need to enter the current password to confirm when changing the user's password before the “Create New Password” window;

3) Enter the password you want to set again to make sure it is entered correctly.

When you want to delete a user password, you should change the password by pressing carriage return directly in the Create New Password window to clear the password.

3.2.6 Set Up Power-on Self & Timed Power On

Set Up Power-on Self:

1) Select “Advanced - Power Control – Power On after Power Fail”;

2) Change “Power On after Power Fail” value: Power On, Power Off, Last State.

Set Up Timed Power On:

1) Select “Advanced - S5 RTC Wake Settings – Wake system from S5”;

2) Change it from Disabled to Daily Time/Weekly Time/Monthly Time;

3) Set the point in time to customize the boot settings in the “Wake up day of month, Wake up Day of week, Wake up day, Wake up hour, Wake up minute, wake up second” that appears. Such as Daily Time, hour: 13, minute:0, second:0 means each day at 13 o'clock on the boot.

3.2.7 Set Up Fan Control & Hardware Monitoring

Set Up Fan Control:

1) Select “Advanced – Hardware Monitor – Fan Function” item (This menu shows only this function is supported);

2) Select the fan to set up, such as “CPU_FAN1 Mode”. Select the operating mode for this fan: Full on Mode，Automatic Mod，Manual Mode；

3) After selecting manual Mode manual speed mode, the Manual PWM Setting input PWM (set range 0-255) value specifies the fan speed;

4) After selecting Automatic Mode, there will be Fan off temperature limit, Fan start temperature limit, Fan start PWM, PWM SLOPE SETTING options to set up.

Generally due to differences in fan performance, the actual performance of smart fans will differ from the set value, it is generally recommended to set THE PWM SLOPE SETTING to 8PWM to ensure that the fan speed with the temperature rise speed.

3.2.8 Setting Boot Order of Startup Disk

Set Boot Sequence:

1) Select “Boot” menu page；

2) Change “Boot Option #1” to the desired device；

3) If there are more than one similar device, such as more than one hard drive. Enter “Hard Disk Drive BBS Priorities” and change the boot order priority between multiple drives first.

When the Boot page has the Boot mode select option, it is recommended that the value of dual so that both Legacy and UEFI boot projects can be booted. UEFI boot items will be filtered when selected for Legacy, and legacy boot items will be filtered when selected as UEFI.

3.2.9 Setting PXE Boot

Open UEFI PXE:

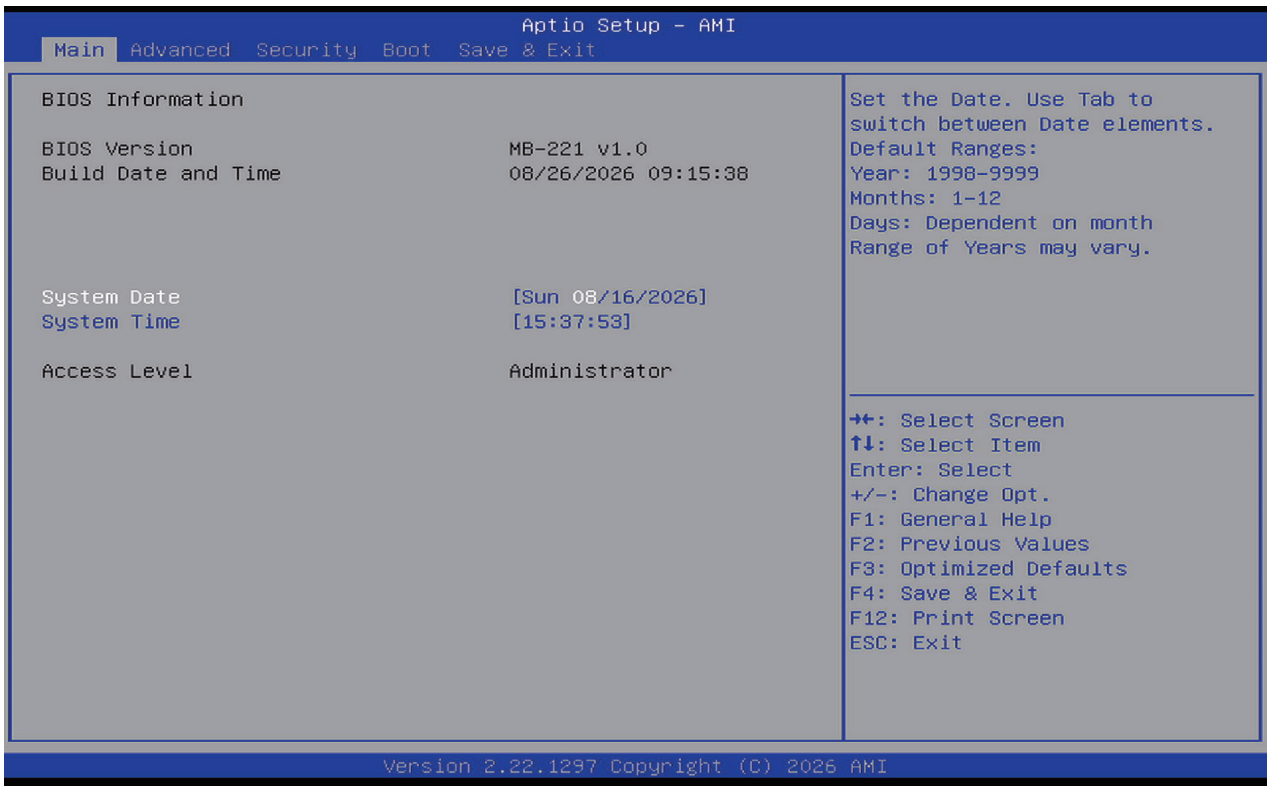
1) Please select the “Advanced - Network Stack Configuration” item;

2) Set Network Stack to Enabled.

3.3 Description of the BIOS Options

Due to the differences in the specific model of the motherboard and the update of the BIOS version, there may be a few menus that do not match this manual, please take the actual basis.

3.3.1 Main



This menu contains the following information:

- **BIOS Version**

  Display the BIOS version information.

- **Build Date and Time**

  Display the BIOS Build date and time.

- **System Date**

  Set the date. Use Tab to switch between date elements. Default ranges:

  Year: 1998-9999

  Months: 1-12

  Days: Dependent on month Range of Years may vary.

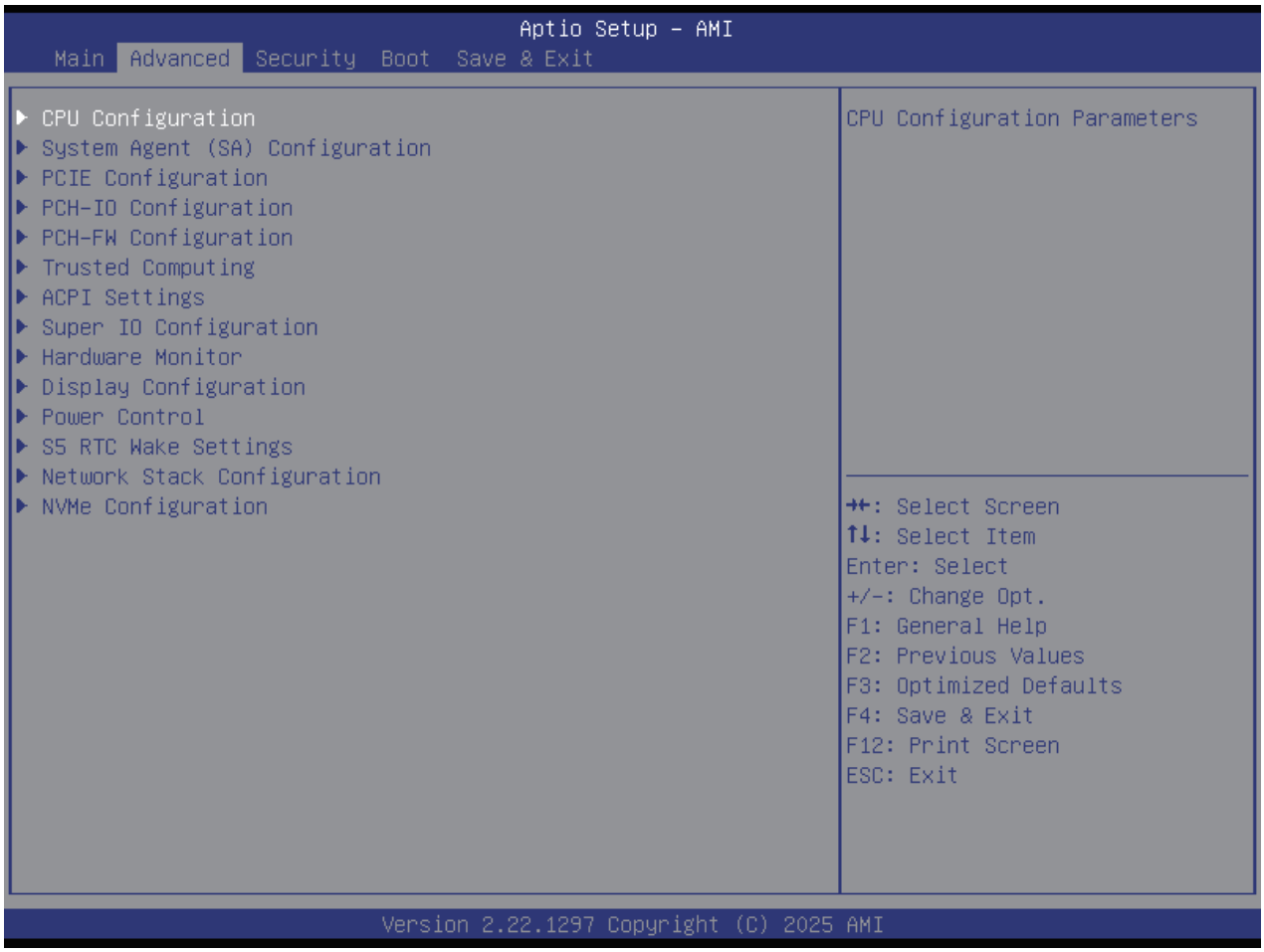
- **System Time**

  Set the time. Use Tab to switch between time elements.

- **Access Level**

  Display the current access level to BIOS Setup Utility.

3.3.2 Advanced



This menu contains the following information:

- **CPU Configuration**

  CPU Configuration Parameters.

- **System Agent (SA) Configuration**

  System Agent (SA) Parameters.

- **PCIE Configuration**

  PCIE Parameters.

- **PCH-IO Configuration**

  PCH Parameters.

- **PCH-FW Configuration**

  Configure Management Engine Technology Parameters.

- **Trusted Computing**

  Trusted Computing Settings.

- **ACPI Settings**

  System APCI Parameters.

- **Super IO Configuration**

  System Super IO Chip Parameters.

- **Hardware Monitor**

  Monitor hardware status.

- **Display Configuration**

  Display Configuration Parameters.

- **Power Control**

  Power Control Settings.

- **S5 RTC Wake Settings**

  Enable system to wake from S5 using RTC alarm.

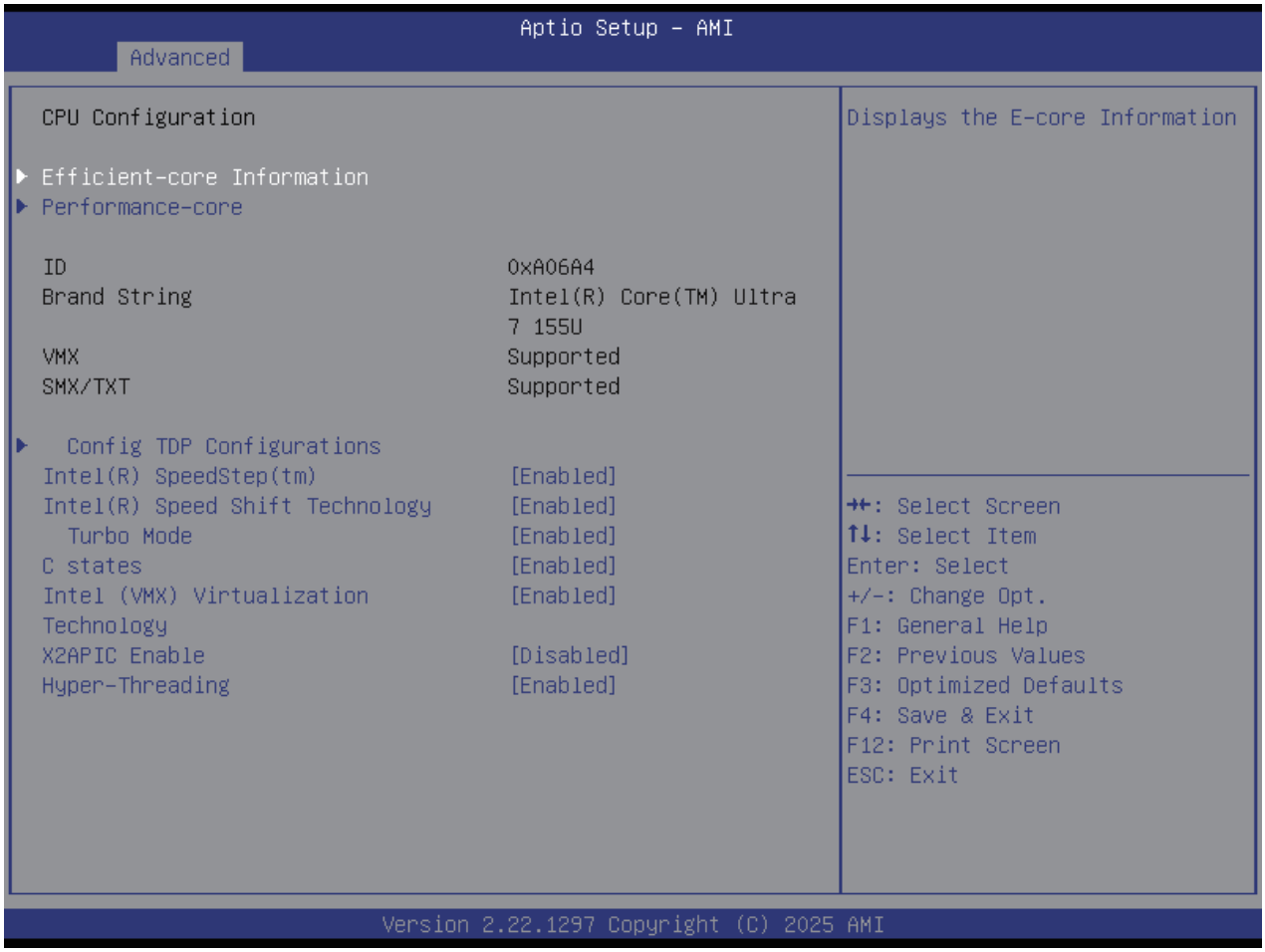
- **Network Stack Configuration**

  Network Stack Settings.

- **NVMe Configuration**

  NVMe Device Options Settings.

CPU Configuration



This menu contains the following information:

- **Efficient-core Information**

  Displays the E-core Information.

- **Performance-core**

  Displays the P-core Information.

- **ID**

  Display the Processor ID.

- **Brand String**

  Display the Brand String.

- **VMX**

  VMX (Virtual-Machine Extensions) supported or not.

- **SMX/TXT**

  SMX (Safer Mode Extensions) /TXT (Trusted Execution Technology) Supported or not.

- **Config TDP Configurations**

  cTDP (Assured Power) Configurations.

- **Intel(R) Speed Step(tm)**

  Allows more than two frequency ranges to be supported.

- **Intel(R) Speed Shift Technology**

  Enable or Disable Intel(R) Speed Shift Technology support. Enabling will expose the CPPC v2 interface to allow for hardware controlled P-states.

- **Turbo Mode**

  Enable or Disable processor Turbo Mode.

- **C states**

  Enable or Disable CPU Power Management. Allows CPU to go to C states when it's not 100% utilized.

- **Intel (VMX) Virtualization Technology**

  When enabled, a VMM can utilize the additional hardware capabilities provided by Vanderpool Technology.

- **X2APIC Enable**

  Enable or Disable X2APIC Operating Mode.

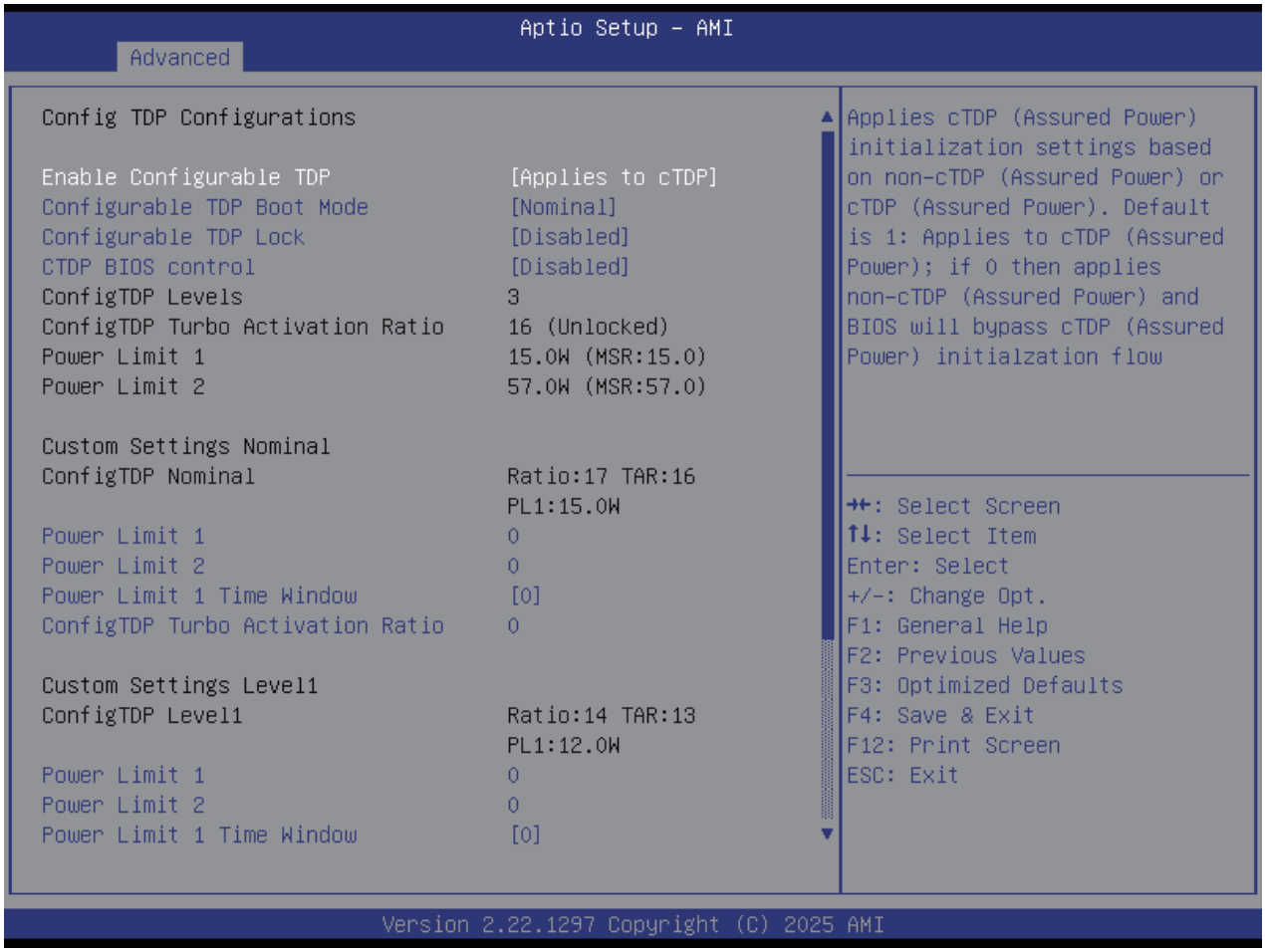
  When this option is configured as ‘Enabled’, ‘VT-d’ option must be ‘Enabled’ and ‘X2APIC Opt Out’ option must be ‘Disabled’ as well.

  This option will be grayed out when ‘VT-d’ option is configured as ‘Disabled’.

- **Hyper-Threading**

  Enable or Disable Hyper-Threading Technology.

Config TDP Configurations



This menu contains the following information:

- **Enable Configurable TDP**

  Applies cTDP (Assured Power) initialization settings based on non-cTDP (Assured Power) or cTDP (Assured Power). Default is 1: Applies to cTDP (Assured Power); if 0 then applies non-cTDP(Assured Power) and BIOS will bypass cTDP (Assured Power) initialization flow.

- **Configurable TDP Boot Mode**

  cTDP (Assured Power) Mode as Nominal/Level1/Level2/Deactivate TDP (Base Power) selection.

  Deactivate option will set MSR to Nominal and MMIO to Zero.

- **Configurable TDP Lock**

  cTDP (Assured Power) Mode Lock sets the Lock bits on TURBO_ACTIVATION_RATIO and CONFIG_TDP_CONTROL.

Note: When CTDP (Assured Power) Lock is enabled Custom ConfigTDP Count will be forced to 1 and Custom ConfigTDP Boot Index will be forced to 0.

- **CTDP BIOS control**

  Enables cTDP (Assured Power) control via runtime ACPI BIOS methods. This 'BIOS only' feature does not require EC or driver support.

  Custom Settings Nominal/Level1/Level2

- **ConfigTDP Nominal/Level1/Level2**

  The options are as follows:

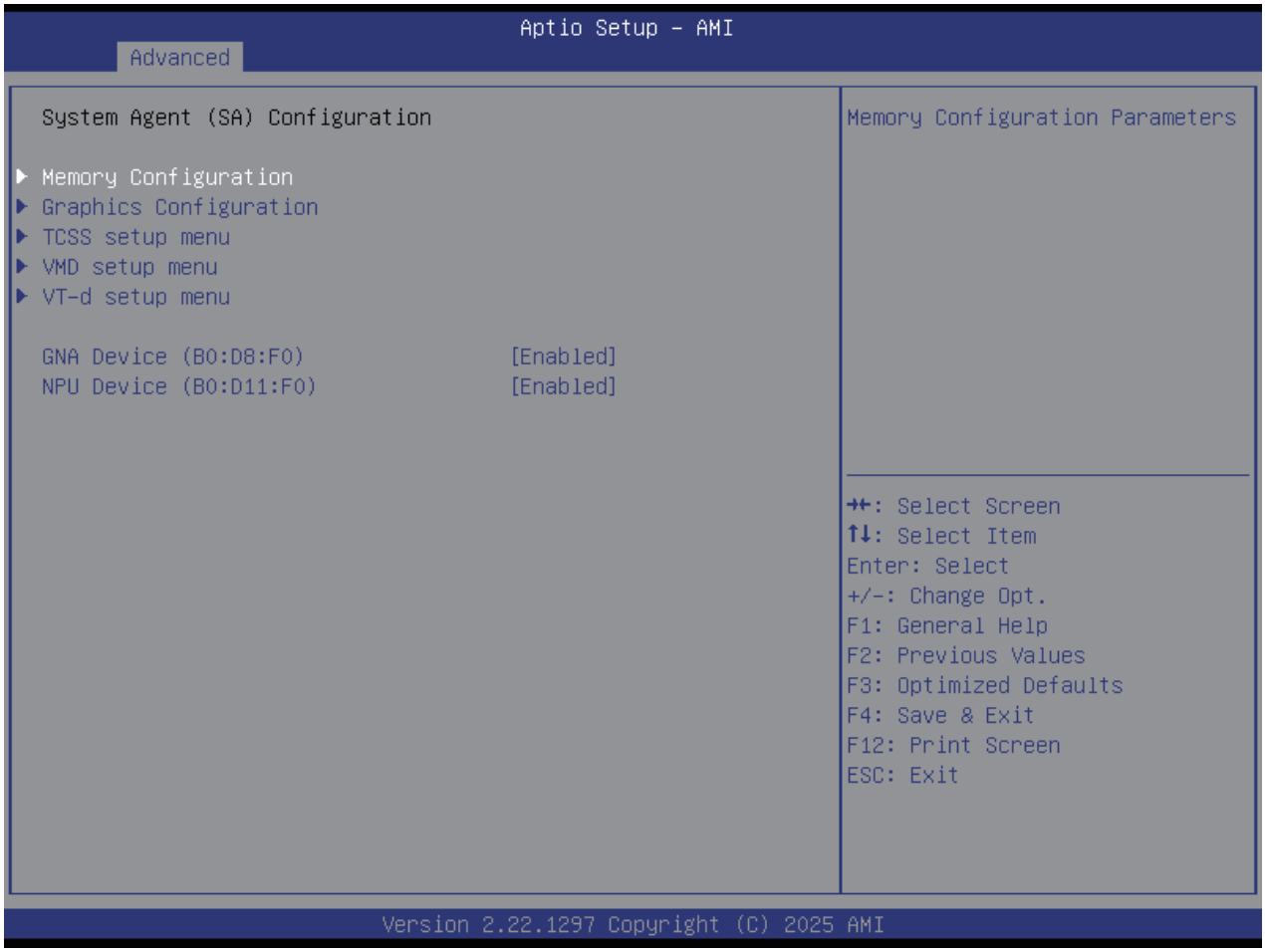
  Power Limit 1: Power Limit 1 in Milli Watts. BIOS will round to the nearest 1/8W when programming. 0 = no custom override. For 12.50W, enter 12500. Overclocking SKU: Value must be between Max and Min Power Limit and Processor Base Power (TDP) Limit.

  Power Limit 2: Power Limit 2 value in Milli Watts. BIOS will round to the nearest 1/8W when programming. 0 = no custom override. For 12.50W, enter 12500. Processor applies control policies such that the package power does not exceed this limit.

  Power Limit 1 Time Window: Power Limit 1 Time Window value in seconds. The value may vary from 0 to 128. 0 = default value (28 sec for Mobile and 8 sec for Desktop). Defines time window which Processor Base Power (TDP) value should be maintained.

  ConfigTDP Turbo Activation Ratio: Custom value for Turbo Activation Ratio. Needs to be configured with valid values from LFM to Max Turbo. 0 means don't use custom value.

System Agent (SA) Configuration



This menu contains the following information:

- **Memory Configuration**

  Memory Configuration Parameters.

- **Graphics Configuration**

  Graphics Configuration.

- **TCSS setup menu**

  TCSS Configuration settings.

- **VMD setup menu**

  VMD Configuration settings.

- **VT-d setup menu**

  VT-d Configuration settings.

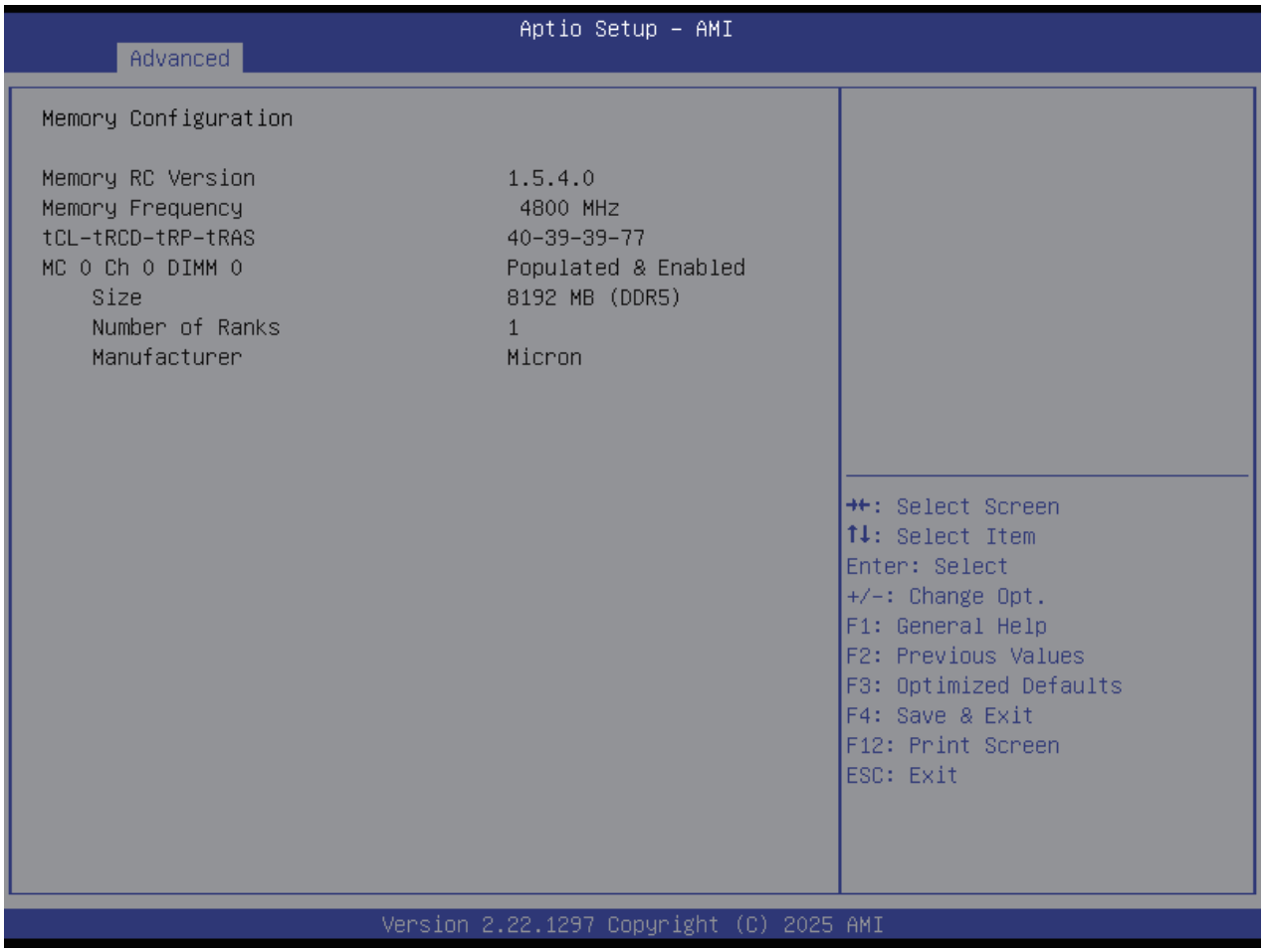
- **GNA Device (B0:D8:F0)**

  Enable or Disable SA GNA Device.

- **NPU Device (B0:D11:F0)**

  Enable or Disable NPU (Neural Processing Unit) Device.

Memory Configuration

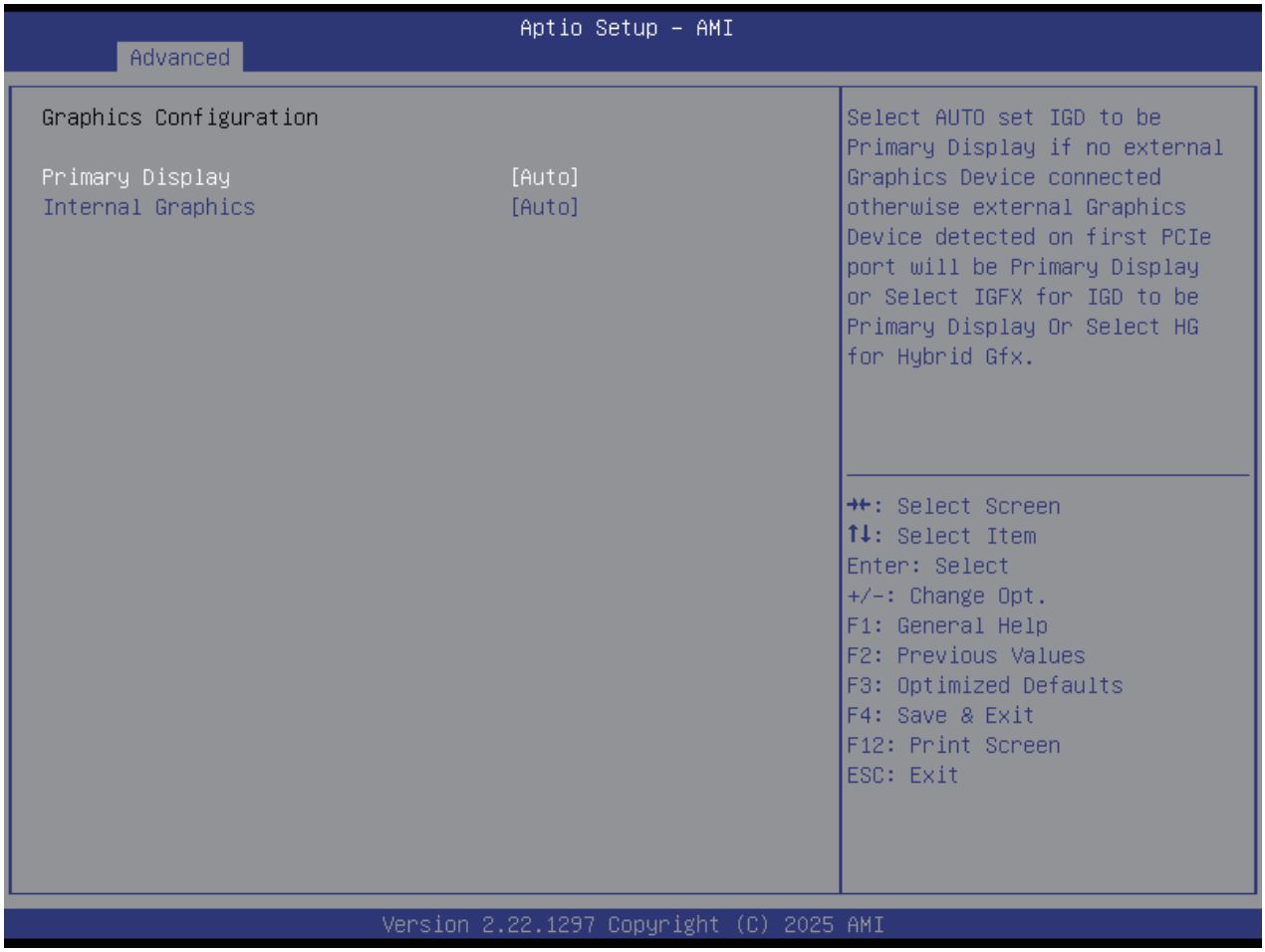


This menu contains the following information:

- **Memory Configuration**

  Display the Memory RC Version, Data Rate, Timings, etc.

Graphics Configuration



This menu contains the following information:

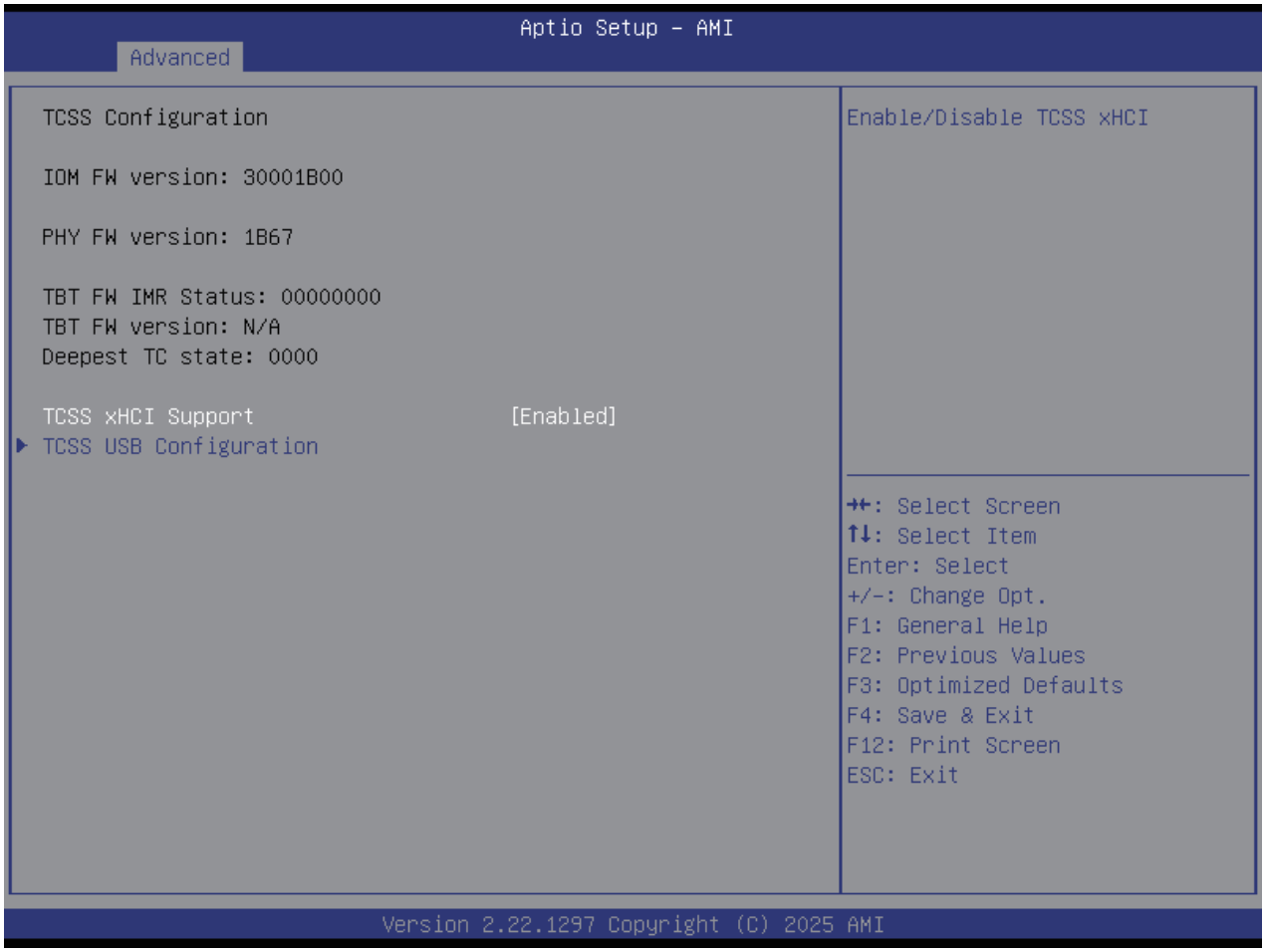
- **Primary Display**

  Select AUTO set IGD to be Primary Display if no external Graphics Device connected otherwise external Graphics Device detected on first PCIE port will be Primary Display or Select IGFX for IGD to be Primary Display Or select HG for Hybrid Gfx.

- **Internal Graphics**

  Keep IGFX enabled based on the setup options.

TCSS setup menu



This menu contains the following information:

- **TCSS xHCI Support**

  Enable or Disable TCSS xHCI.

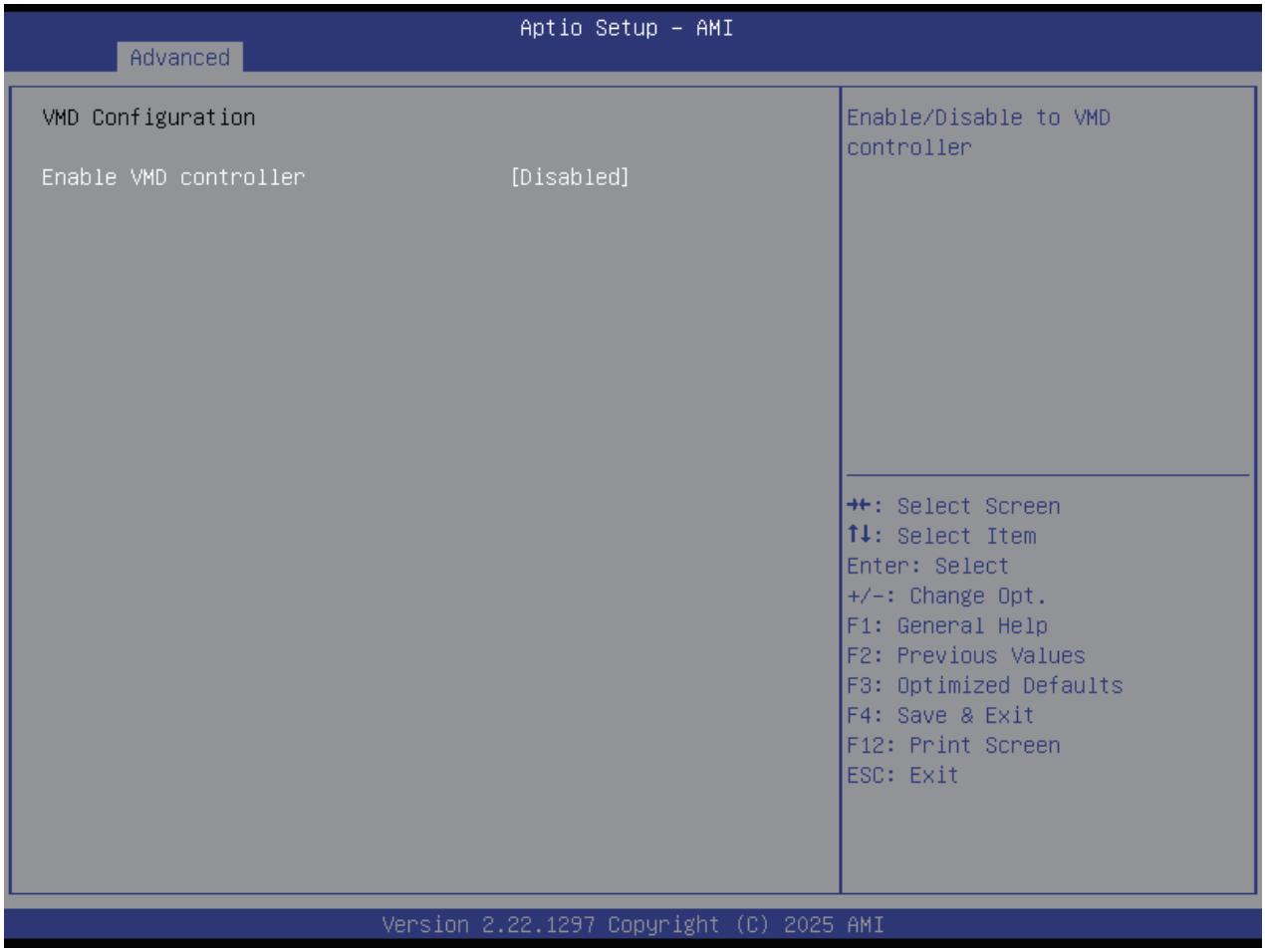
- **TCSS USB Configuration**

  SA TCSS USB Configuration settings.

  The option after entering the interface settings above is as follows:

  TCSS CPU USB Port Disable Override: Selectively Enable or Disable the corresponding USB port from reporting a Device Connection to the controller.

VMD setup menu

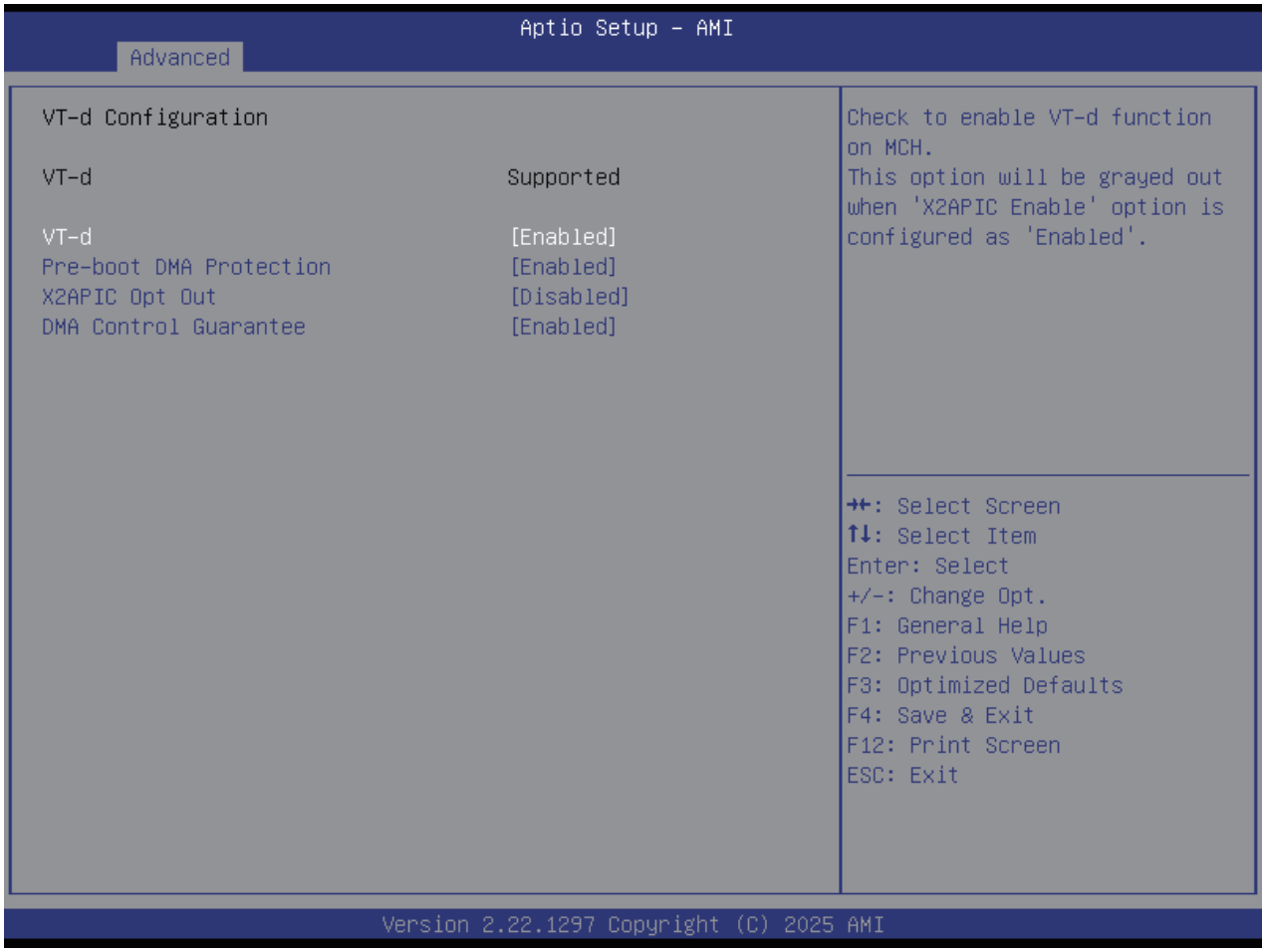


This menu contains the following information:

- **Enable VMD controller**

  Enable or Disable to VMD controller.

VT-d setup menu



This menu contains the following information:

- **VT-d**

  Check to enable VT-d function on MCH.

  This option will be grayed out when ‘X2APIC Enable’ option is configured as ‘Enabled’.

- **Pre-boot DMA Protection**

  Enable DMA Protection in Pre-boot environment (If DMAR table is installed in DXE and If VTD_INFO_PPI Is installed in PEI).

- **X2APIC OPT OUT**

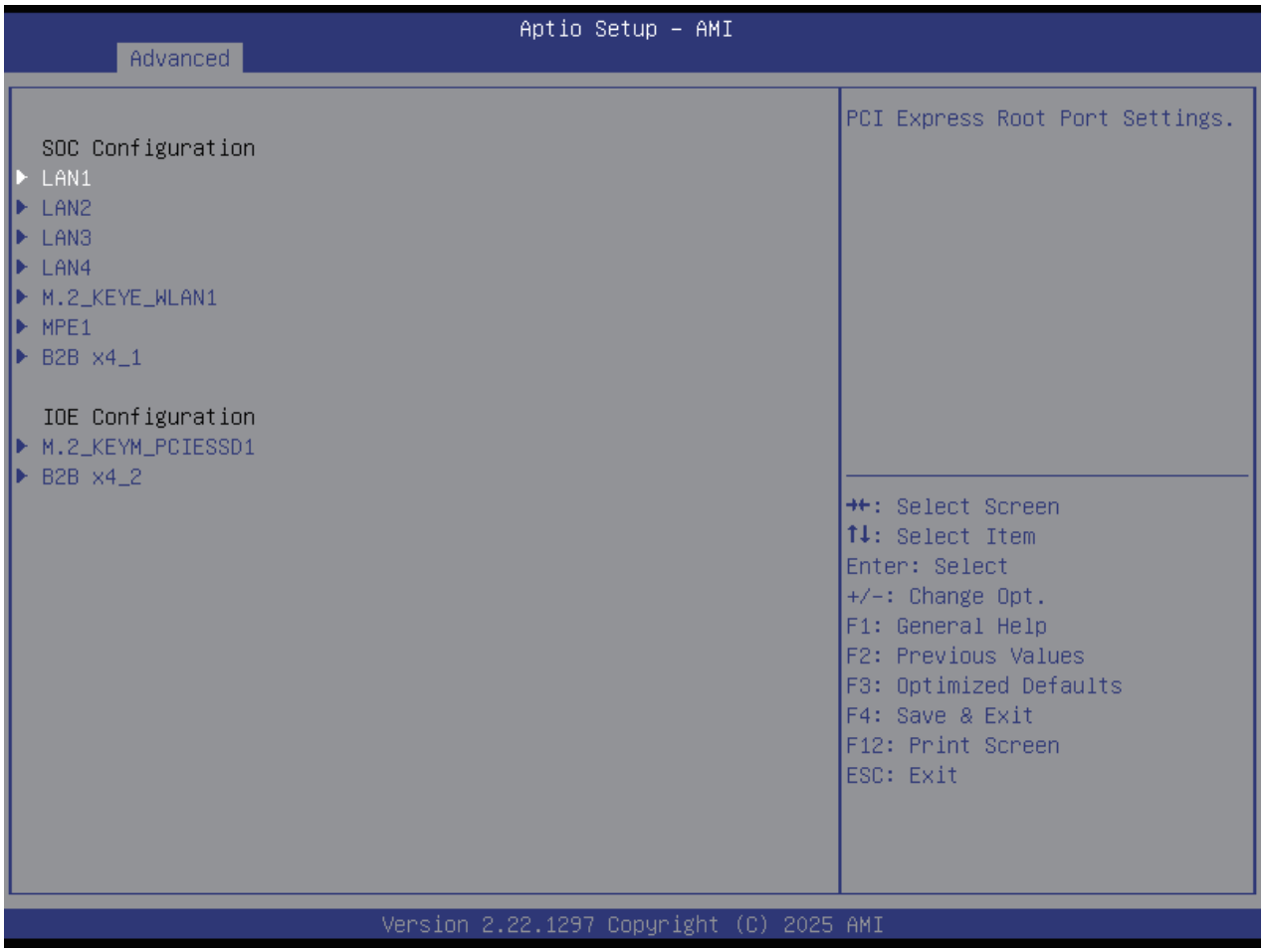
  Enable or Disable X2APIC_OPT_OUT bit.

  This option will be grayed out when ‘X2APIC Enable’ option is configured as ‘Enabled’.

- **DMA Control Guarantee**

  Enable or Disable DMA_CONTROL_GUARANTEE bit.

PCIE Configuration



This menu contains the following information:

- **LAN1/LAN2/LAN3/LAN4/M.2_KEYE_WLAN1/MPE1/B2B x4_1/M.2_KEYM_PCIESSD1/ B2B x4_2:**

  PCI Express Root Port Settings.

  The options after entering the interface settings above are as follows.

  PCI Express Root Port: Control the PCI Express Root Port.

  PCIe Speed: Configure PCIe Speed.

PCH-IO Configuration



This menu contains the following information:

- **SATA Configuration**

  SATA Device Options Settings.

- **USB Configuration**

  USB Configuration Settings.

- **HD Audio Configuration**

  HD Audio Subsystem Configuration Settings.

SATA Configuration



This menu contains the following information:

- **SATA Controller(s)**

  Enable or Disable SATA Device.

- **SATA Mode Selection**

  Determines how SATA controller(s) operate.

- **M.2_KEYM_PCIESSD1**

  Display M.2_KEYM_PCIESSD1 Information.

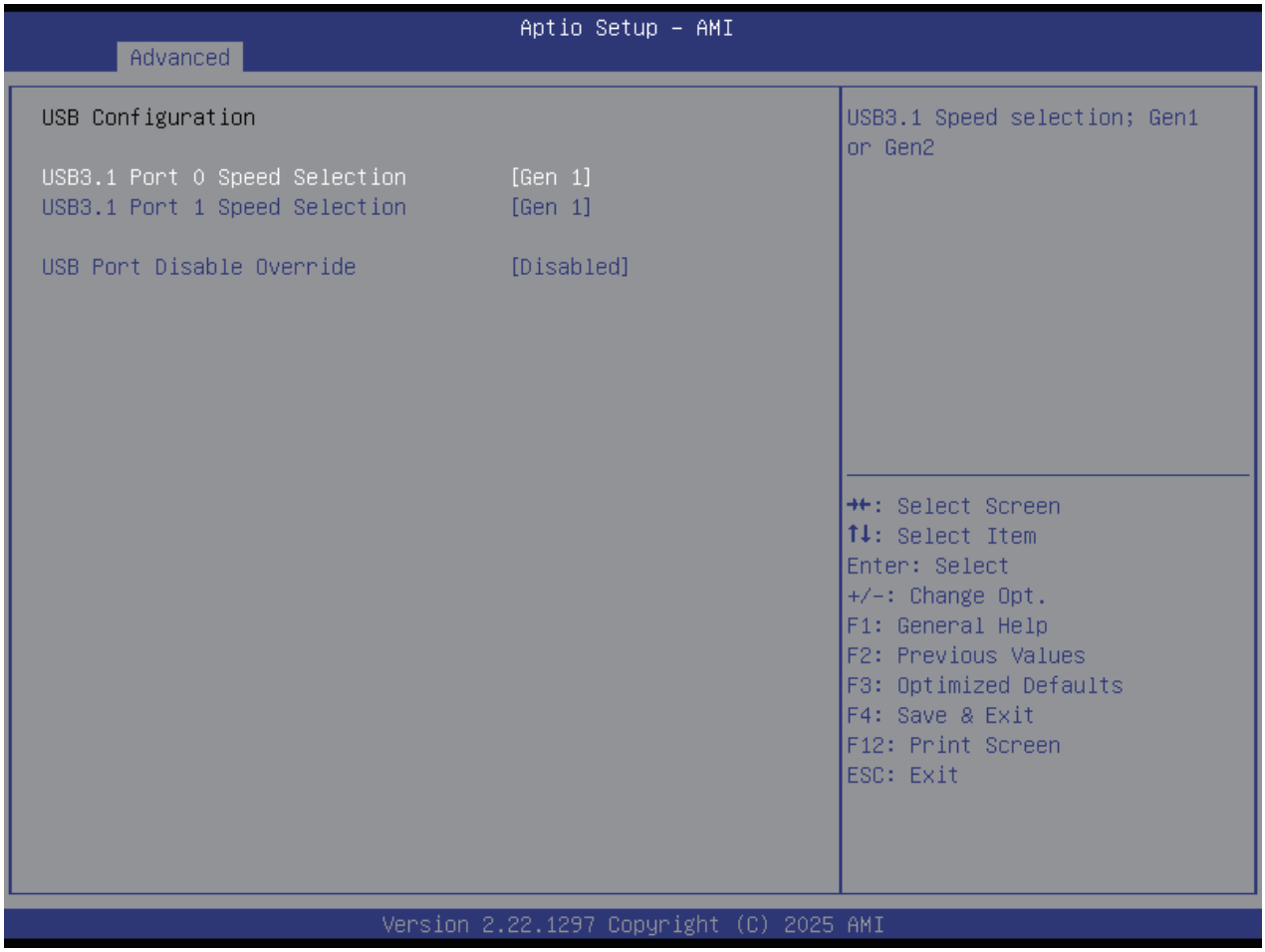
- **SATA1**

  Display SATA1 Information.

- **Port X**

  Enable or Disable SATA Port X.

USB Configuration



This menu contains the following information:

- **USB3.1 Port 0 Speed Selection**

  USB3.1 Speed selection; Gen1 or Gen2.

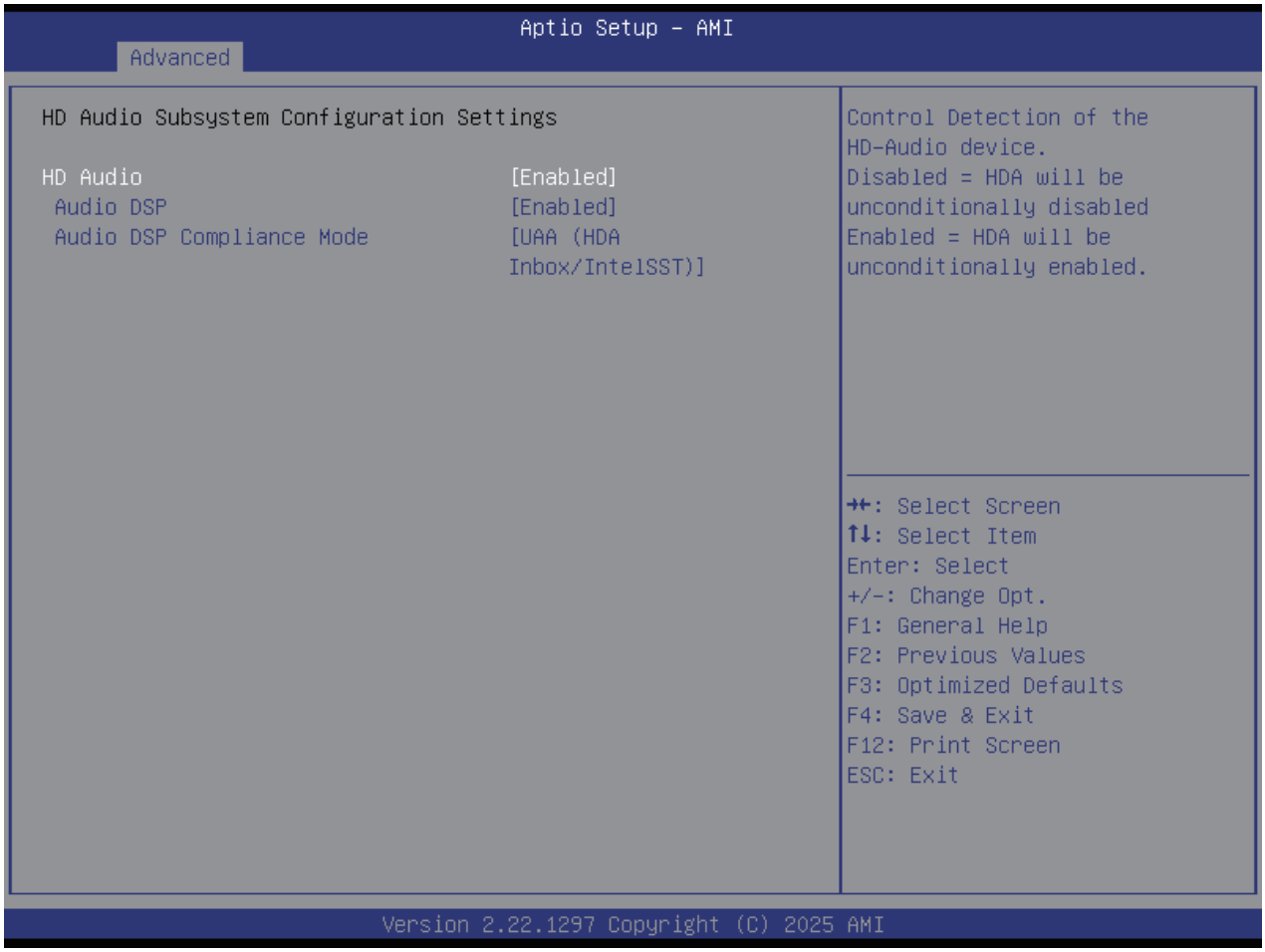
- **USB3.1 Port 1 Speed Selection**

  USB3.1 Speed selection; Gen1 or Gen2.

- **USB Port Disable Override**

  Selectively Enable or Disable the corresponding USB port from reporting a Device Connection to the controller.

HD Audio Configuration



This menu contains the following information:

- **HD Audio**

  Control Detection of the HD-Audio device.

  Disabled = HAD will be unconditionally disabled.

  Enabled = HAD will be unconditionally enabled.

- **Audio DSP**

  Enable or Disable Audio DSP.

- **Audio DSP Compliance Mode**

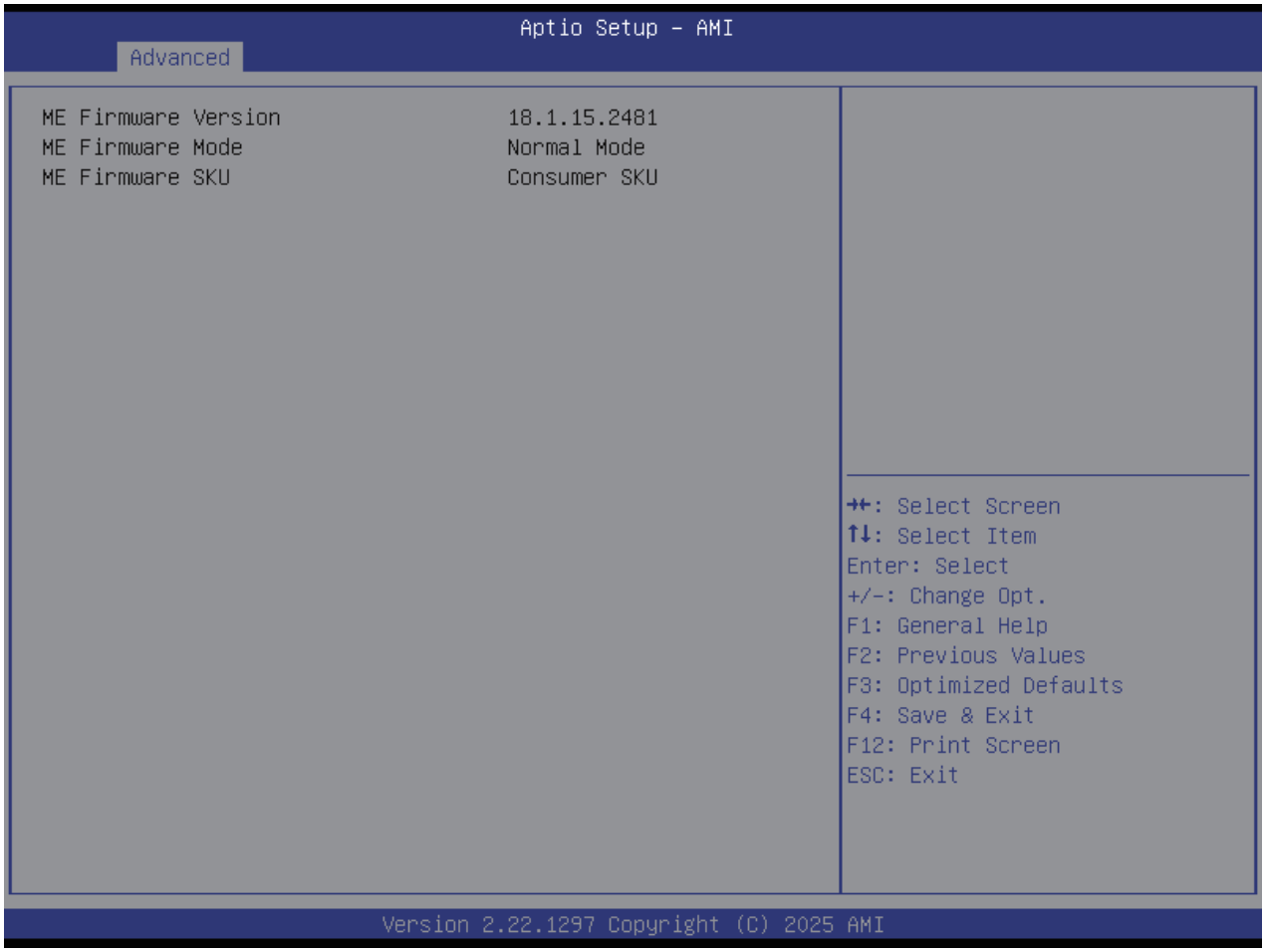
  Specifies DSP enabled system compliance:

  1. Non-UAA (Intel SST driver support only – CC_040100).

  2. UAA (HD Audio Inbox or Intel SST driver support – CC_040380).

  Note: NHLT (DMIC/BT/I2S configuration) is published for non-UAA only.

PCH-FW Configuration



This menu contains the following information:

- **ME Firmware Version**

  Display the ME Firmware information.

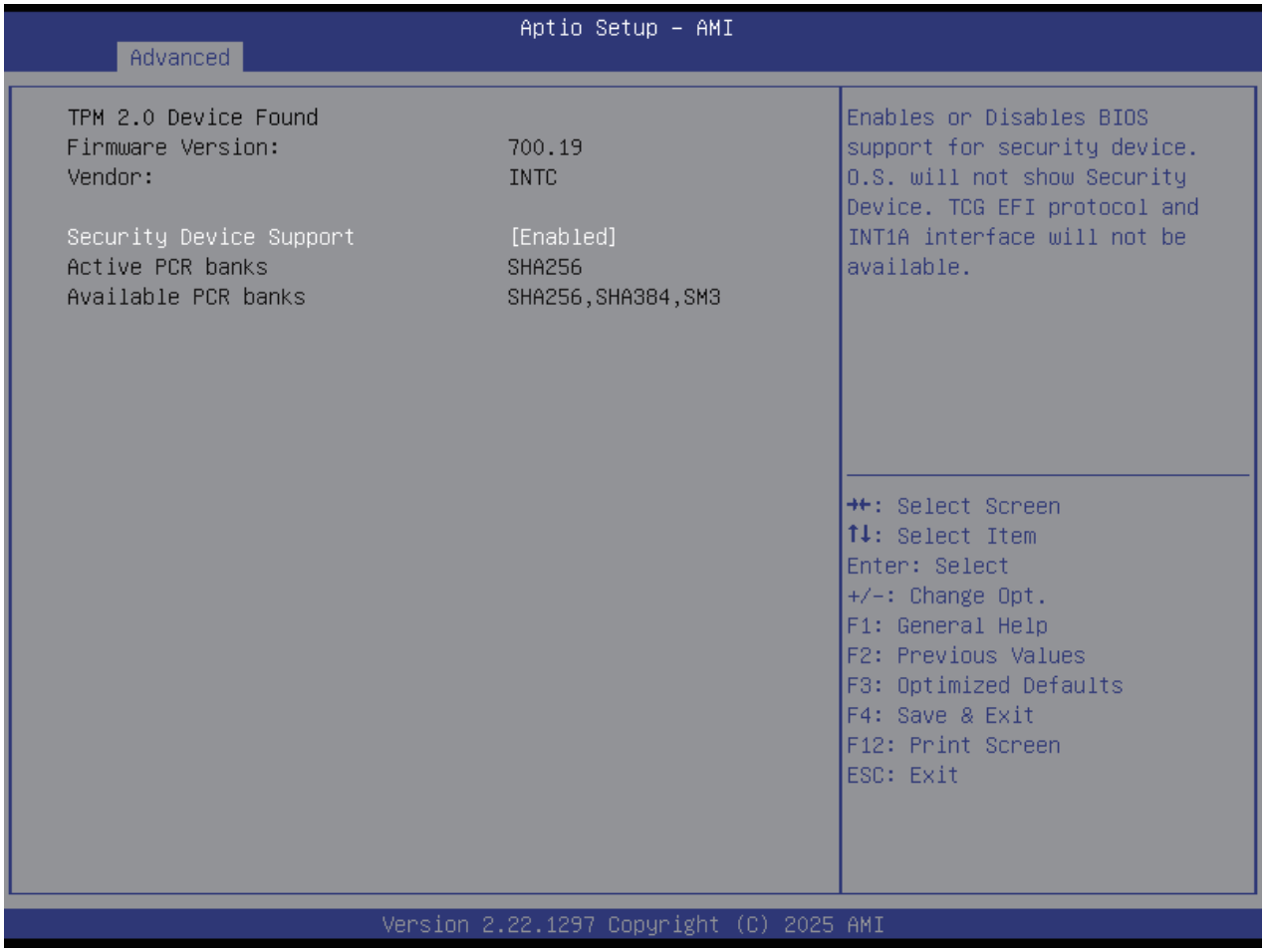
- **Me Firmware Mode**

  Display the Me Firmware Mode.

- **Me Firmware SKU**

  Display the Me Firmware SKU.

Trusted Computing



This menu contains the following information:

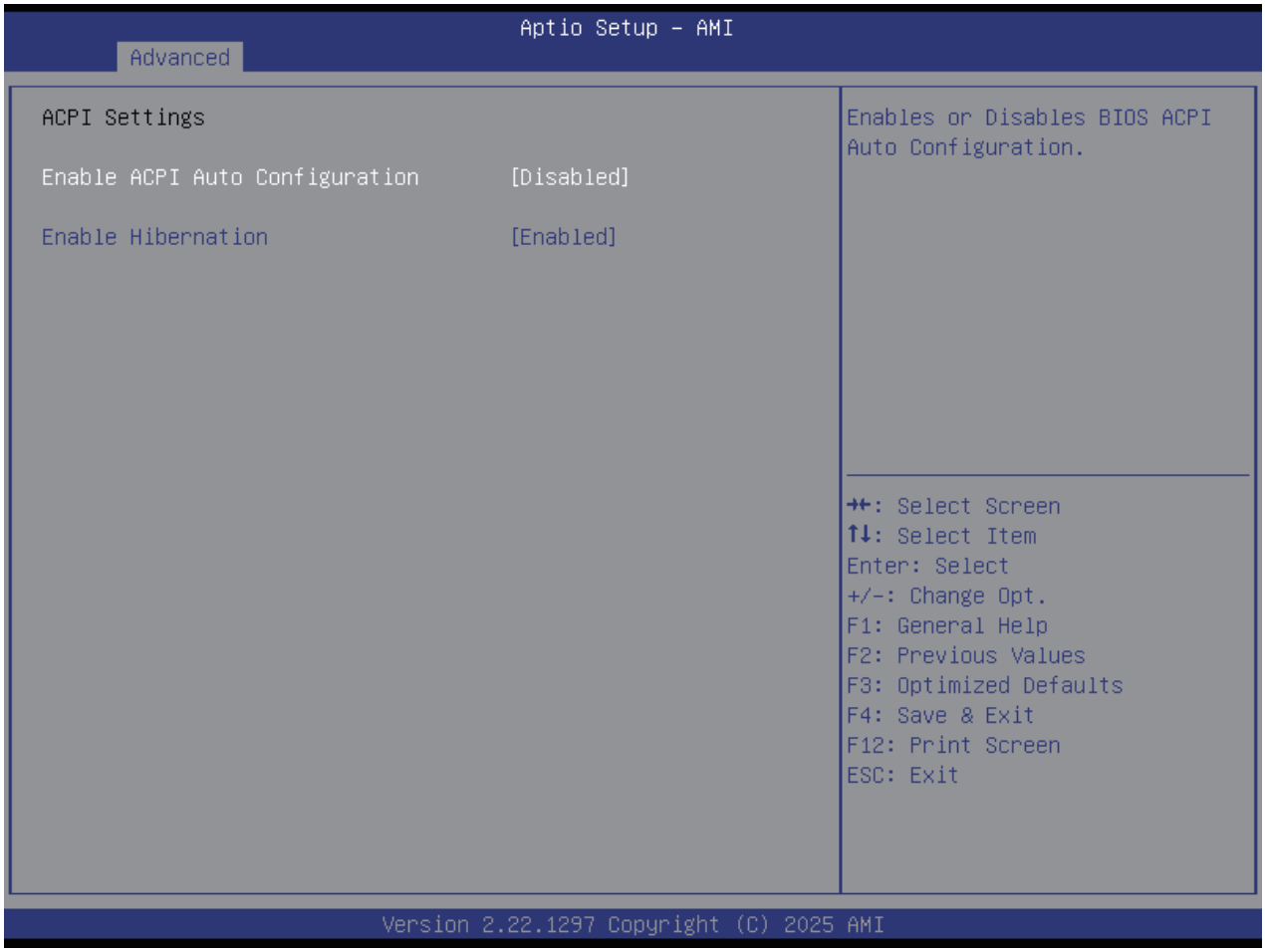
- **TPM 2.0 Device Found**

  Display the information of the TPM 2.0 Device: Firmware Version, Vendor.

- **Security Device Support**

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

ACPI Settings



This menu contains the following information:

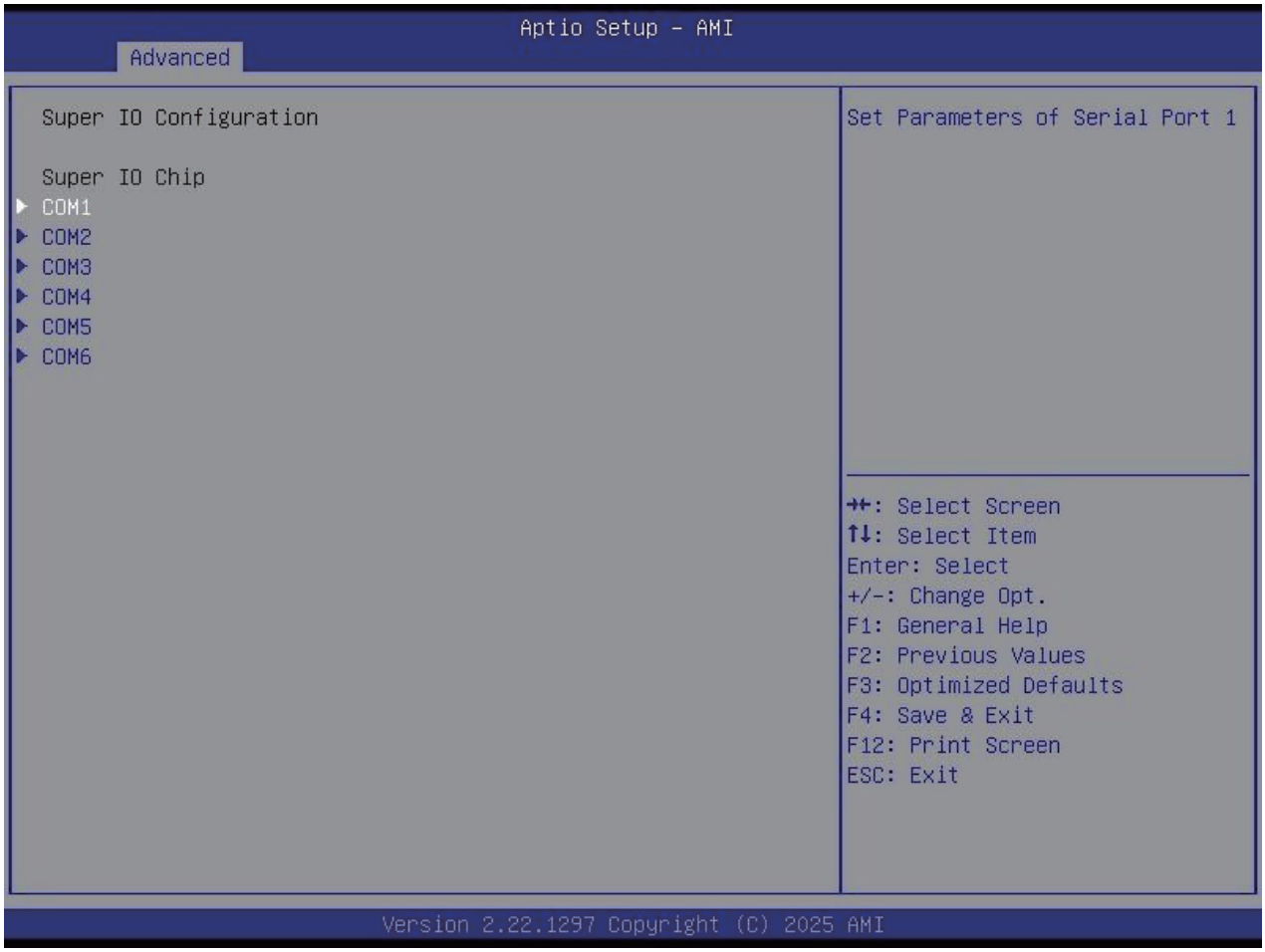
- **Enable ACPI Auto Configuration**

  Enable or Disable BIOS ACPI Auto Configuration.

- **Enable Hibernation**

  Enable or Disable System ability to Hibernate (OS/S4 Sleep State). This option may not be effective with some operating systems.

Super IO Configuration



This menu contains the following information:

- **COM X**

  Set Parameters of Serial Port x.

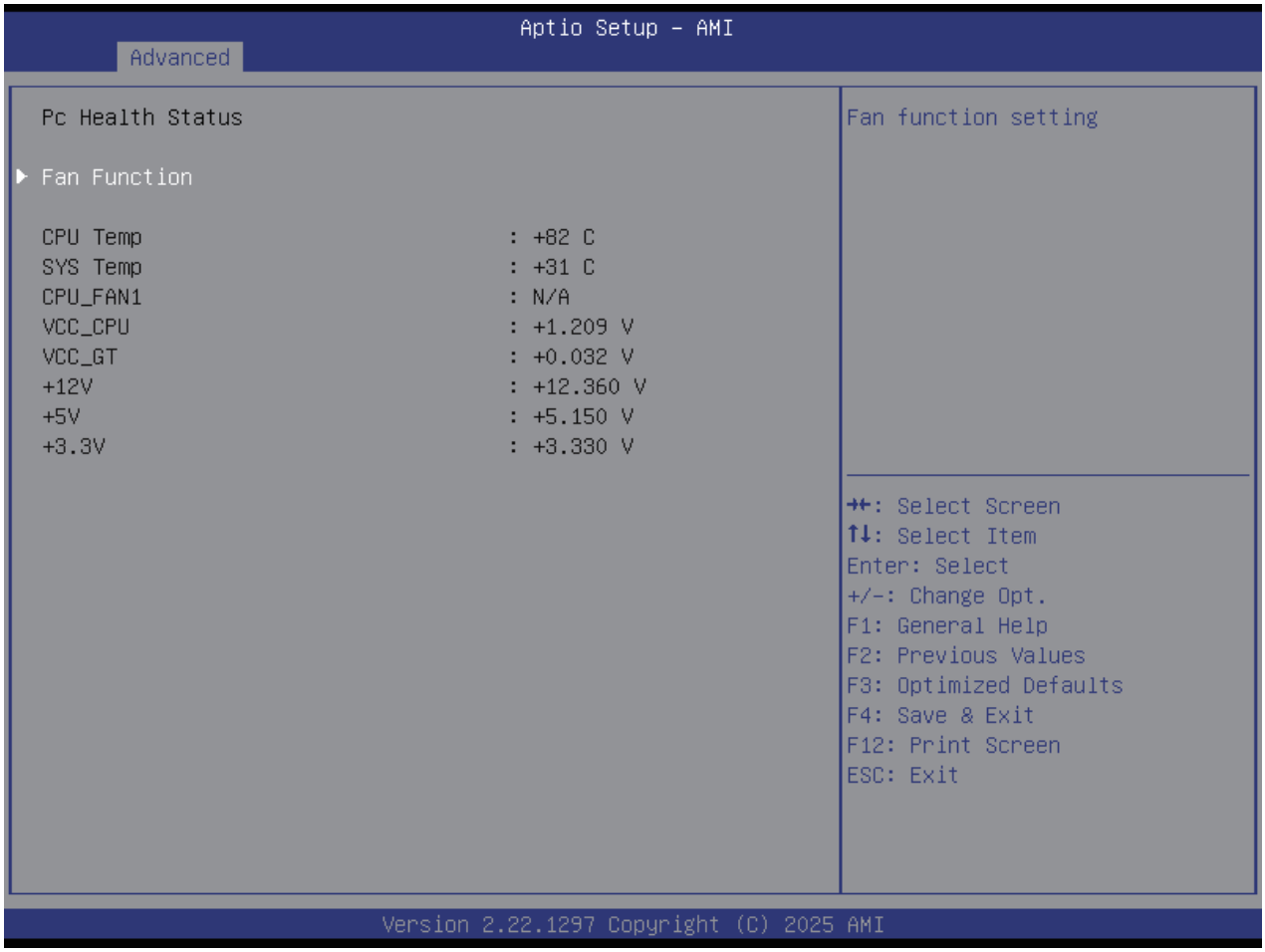
  The options after entering the interface settings above are as follows.

  Serial Port: Enable or Disable Serial Port (COM).

  Device Settings: Display the Current Device Settings.

  Change Settings: If the board supported, this menu shows. Select an optimal setting for Super IO Device.

Hardware Monitor



This menu contains the following information:

- **Fan Function (If the board supported, this menu shows.)**

  Fan Function setting.

  CPU_FAN1 Mode: FAN Mode Select.

  Full on Mode; Automatic Mode; Manual Mode. Refer to section 2.7 of this manual for setting up.

  Fan off temperature limit: Fan will off when temperature lower than this limit.

  Fan start temperature limit: Fan will work when temperature higher than this limit

  Fan start PWM: Fan will start with this PWM value

  PWM SLOPE SETTING: PWM SLOPE Selection.

- **CPU Temp**

  Display CPU Temperature.

- **SYS Temp**

  Display System Temperature.

- **CPU_FAN1**

  Display CPU_FAN1 Speed.

- **SYS_FAN1**

  Display SYS_FAN1 Speed.

- **VCC_CPU**

  Display CPU Core Voltage Value.

- **VCC_GT**

  Display Integrated Graphics Core Voltage Value.

- **+12V**

  Display +12 Voltage Value.

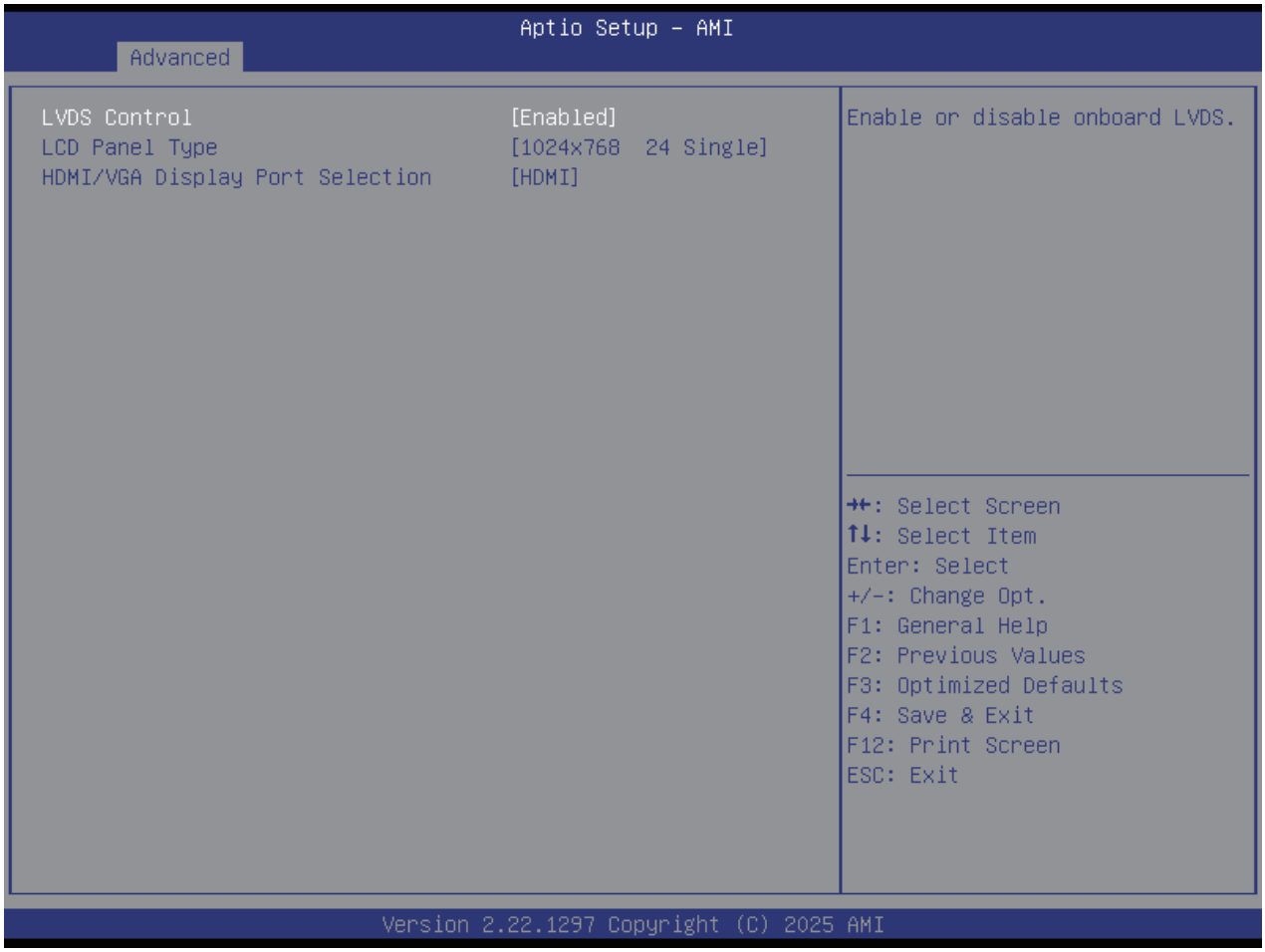
- **+5V**

  Display +5 Voltage Value.

- **+3.3V**

  Display +3.3 Voltage Value.

Display Configuration



This menu contains the following information:

- **LVDS Control**

  Enable or Disable onboard LVDS.

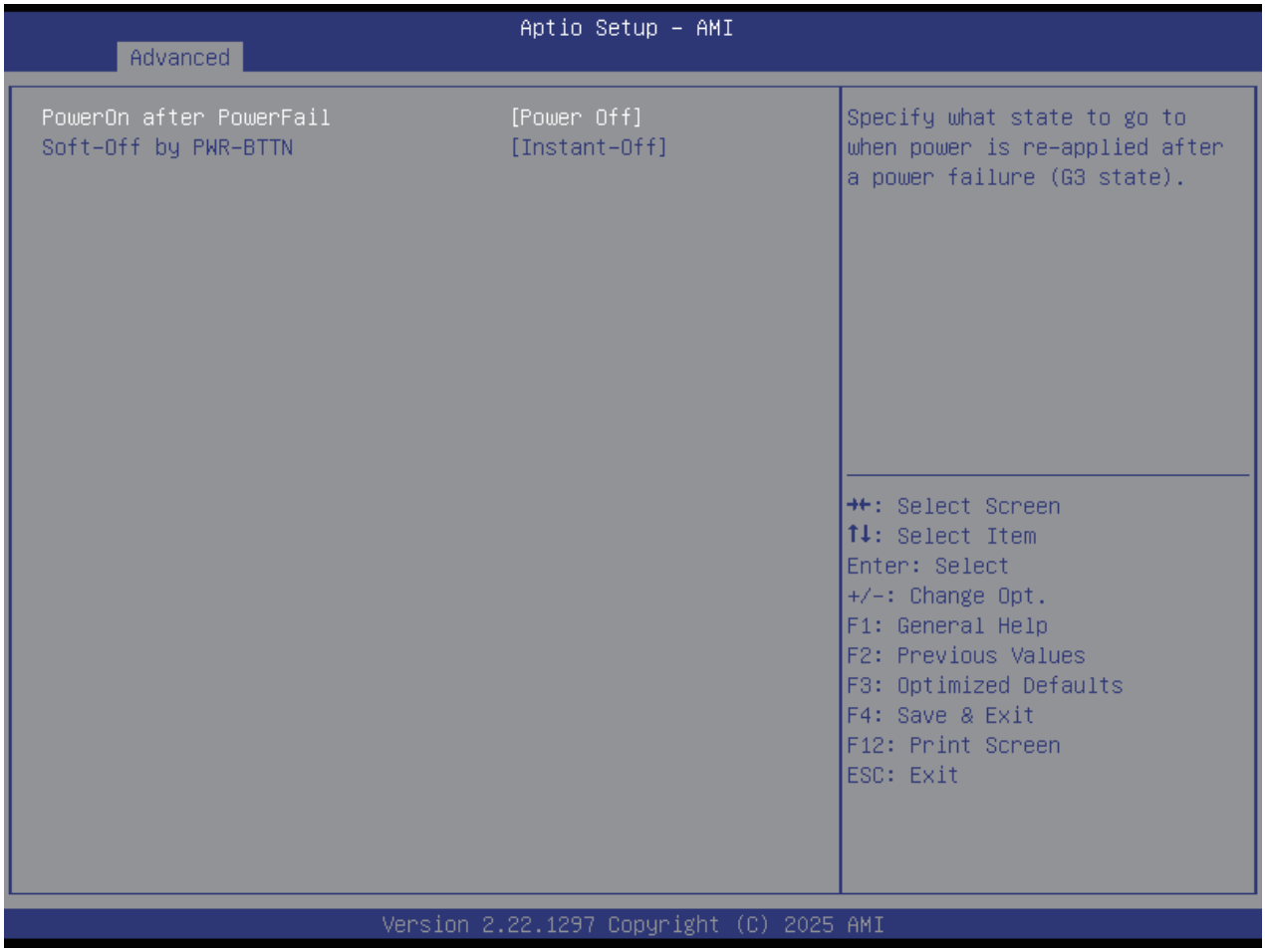
- **LCD Panel Type**

  Select LCD Panel Type.

- **HDMI/VGA Display Port Selection**

  Select HDMI/VGA Display Port Selection.

Power Control



This menu contains the following information:

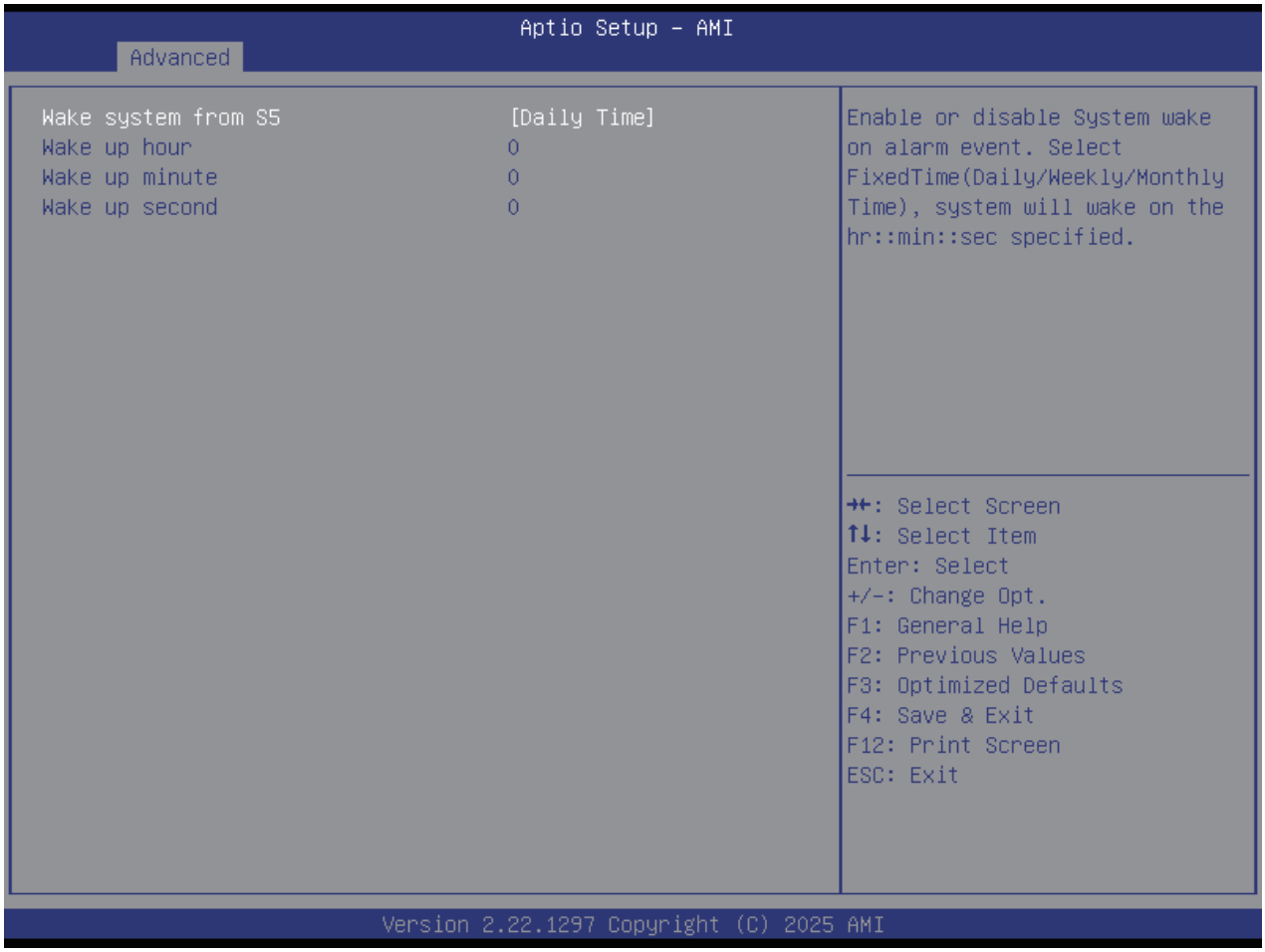
- **Power On after Power Fail**

  Specify what state to go to when power is re-applied after a power failure (G3 state).

- **Soft-off by PWR-BTTN**

  Soft-off by PWR-BTTN Settings: Delay 4 Sec. or Instant – Off.

S5 RTC Wake Settings



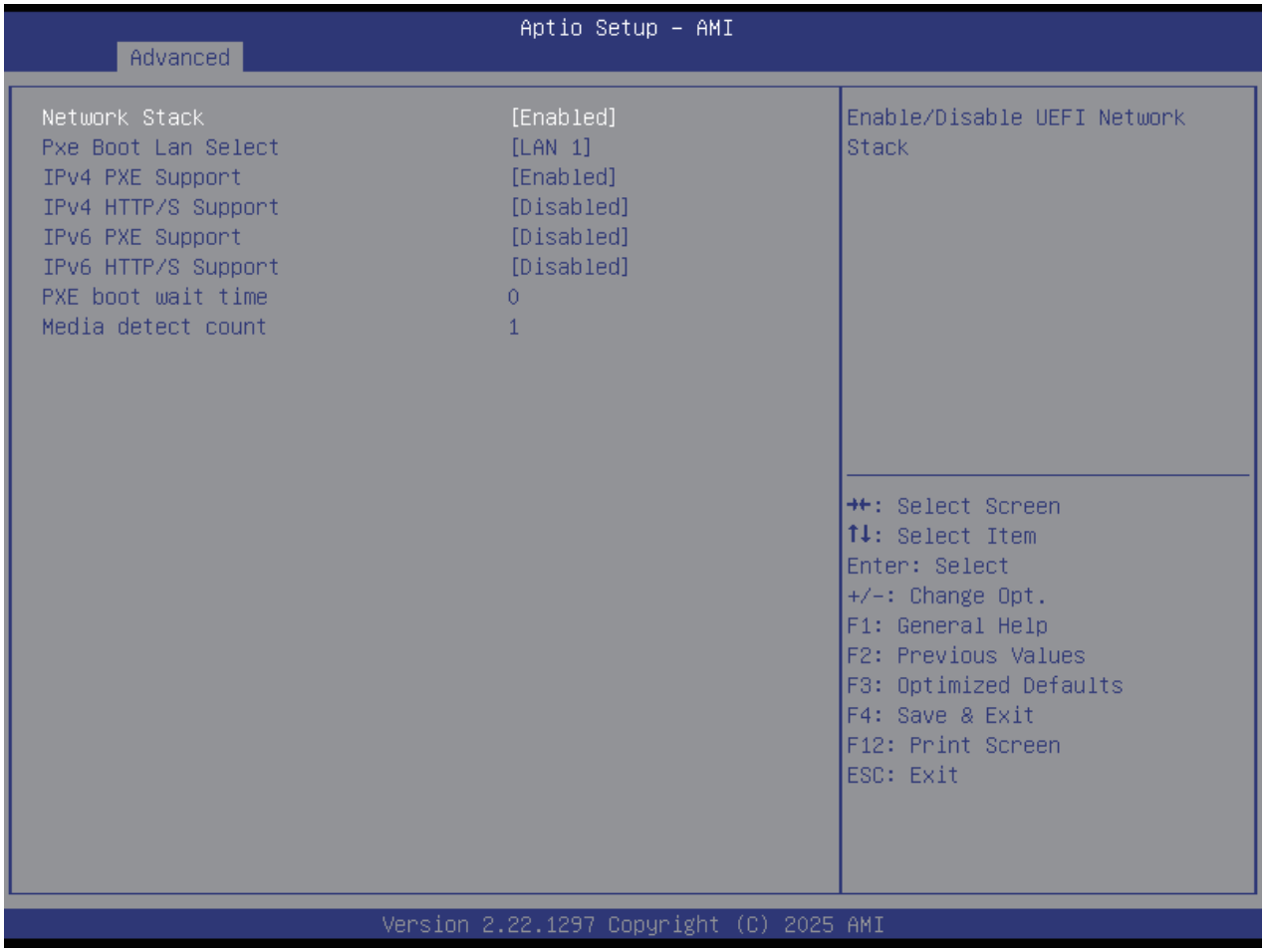
This menu contains the following information:

- **Wake system from S5**

  Enable or Disable System wake on alarm event. Select FixedTime (Daily/Weekly/Monthly Time), system will wake on the hr::min::sec specified.

  Refer to section 2.6 of this manual for setting up.

Network Stack Configuration



This menu contains the following information:

- **Network Stack**

  Enable or Disable UEFI Network Stack.

- **PXE Boot Lan Select**

  Select the LAN port to Support PXE Boot.

- **IPv4 PXE Support**

  Enable or Disable IPv4 PXE boot support. If disabled, IPv4 PXE boot support will not be available.

- **IPv4 HTTP Support**

  Enable or Disable IPv4 HTTP boot support. If disabled, IPv4 HTTP boot support will not be available.

- **IPv6 PXE Support**

  Enable or Disable IPv6 PXE boot support. If disabled, IPv6 PXE boot support will not be available.

- **IPv6 HTTP Support**

  Enable or Disable IPv6 HTTP boot support. If disabled, IPv6 HTTP boot support will not be available.

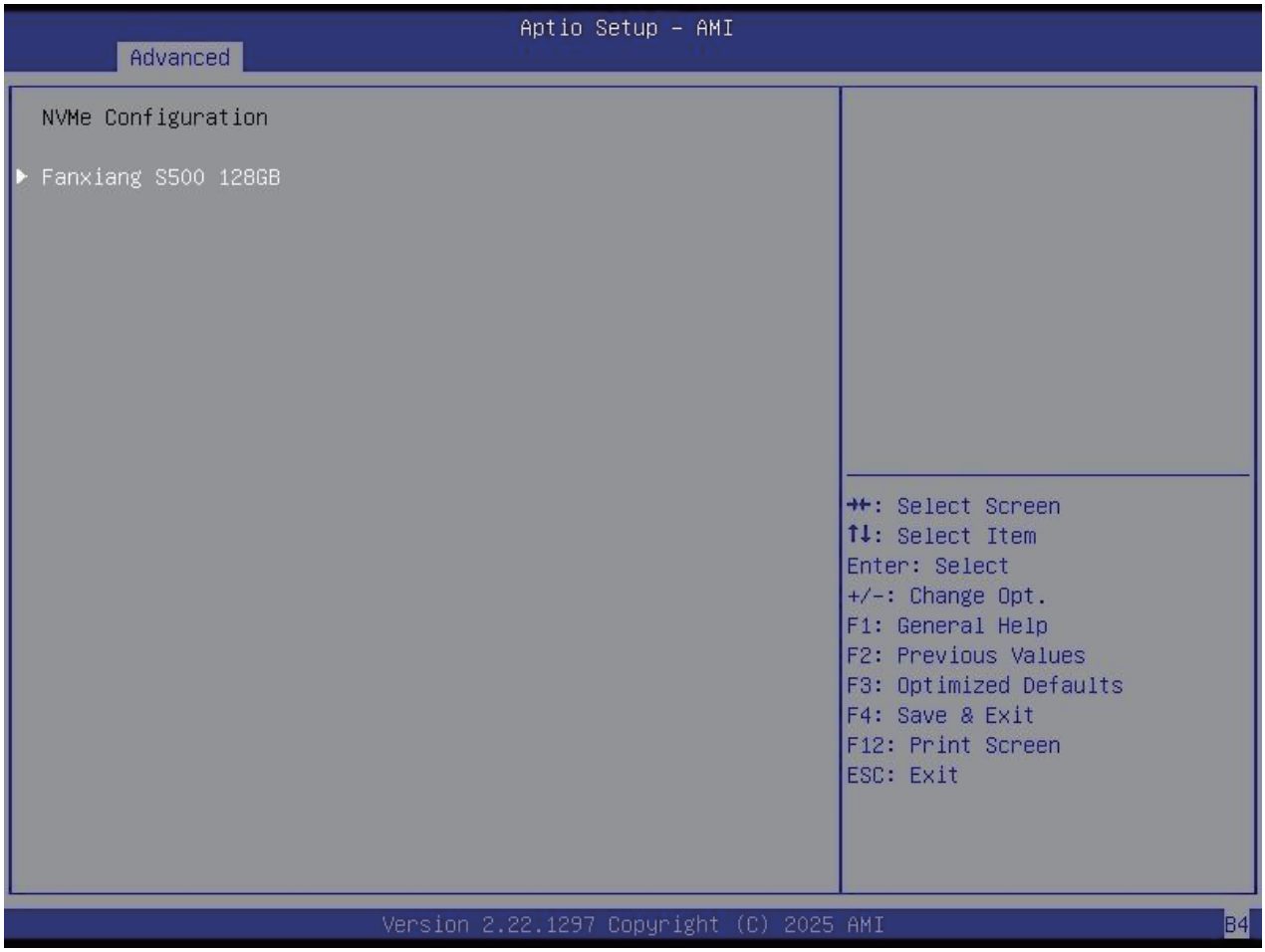
- **PXE boot wait time**

  Wait time in seconds to press ESC key to abort the PXE boot. Use either +/- or numeric keys to set the value.

- **Media detect count**

  Number of times the presence of media will be checked. Use either +/- or numeric keys to set the value.

NVMe Configuration

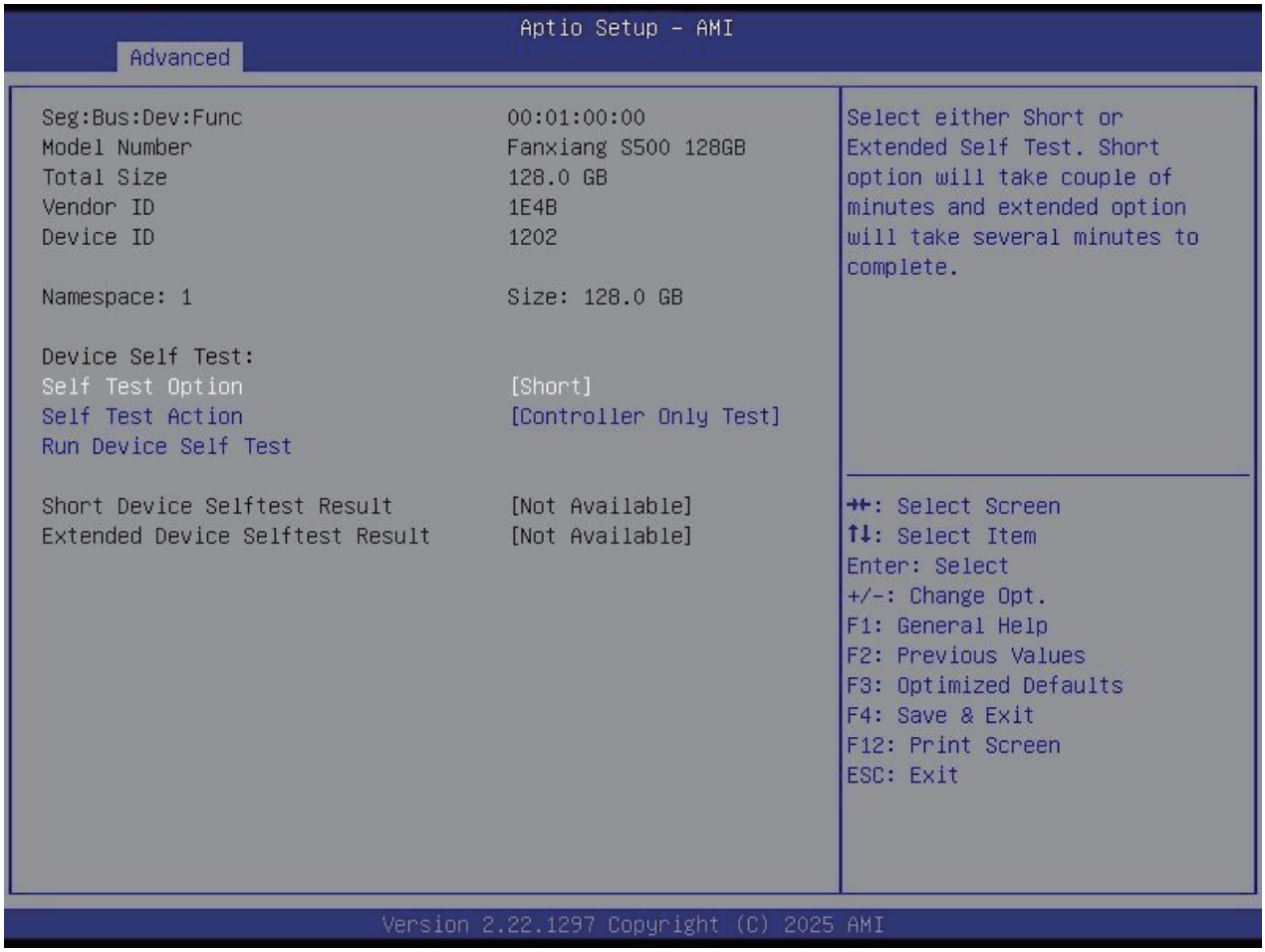


This menu contains the following information:

- **NVMe Configuration**

  NVMe Device Options Settings. (When no device is connected, the Option display ‘No NVMe Device Found’).

NVMe Device Options Settings



This menu contains the following information:

- **Display the NVME device information**
- **Device Self Test**

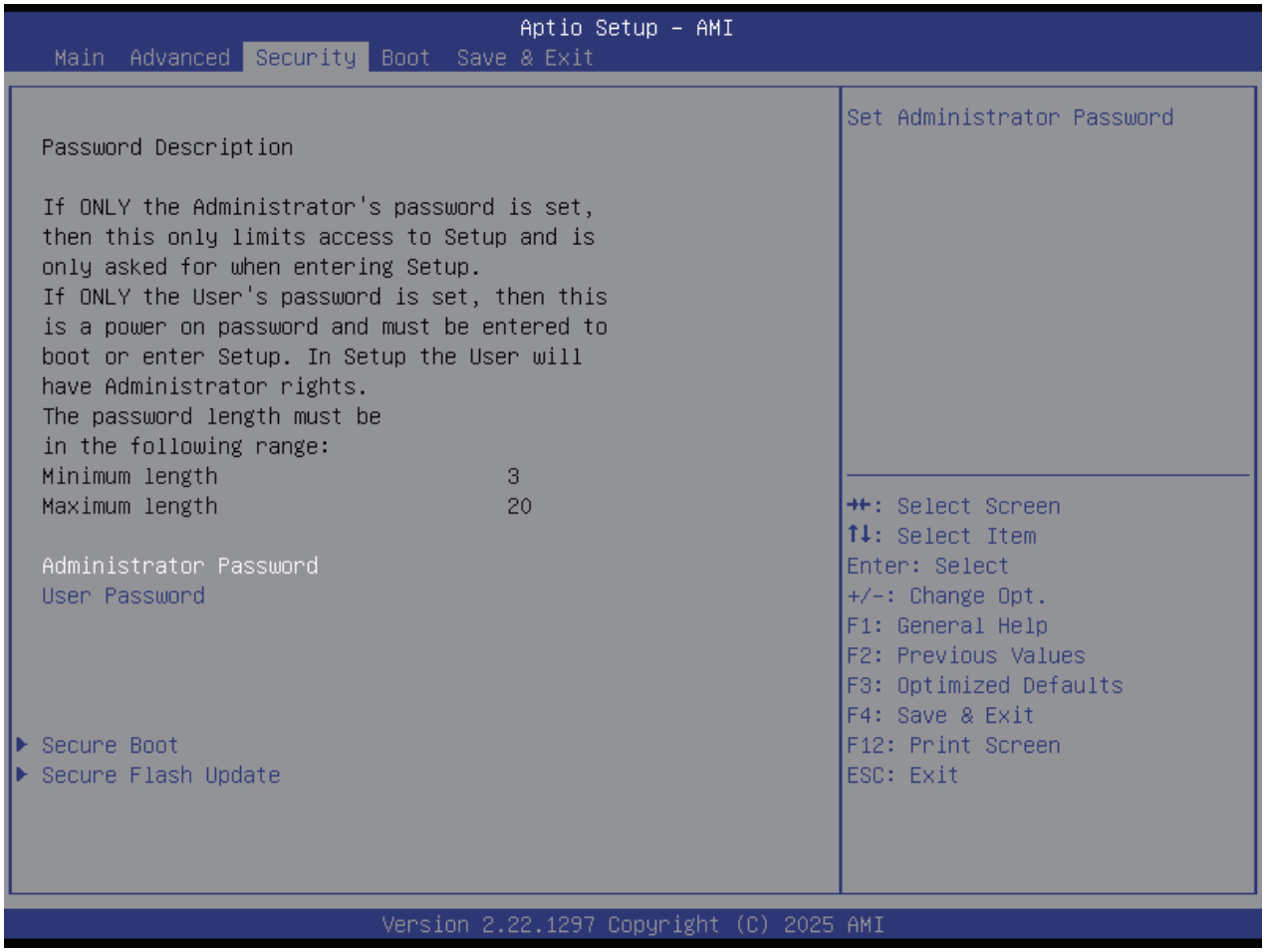
  Self Test Option: Select either Short or Extended Self Test. Short option will take couple of minutes and extended option will take several minutes to complete.

  Self Test Action: Select either to test Controller alone or Controller and Name Space. Selecting Controller and Name Space option will take lot longer to complete the test.

  Run Device Self Test: Perform device self test for the corresponding Option and Action selected by user.

  Pressing 'Esc' key will abort the test. Result shown below is the recent result logged in the device.

3.3.3 Security



This menu contains the following information:

- **Administrator Password**

  Set Setup Administrator Password.

- **User Password**

  Set User Password.

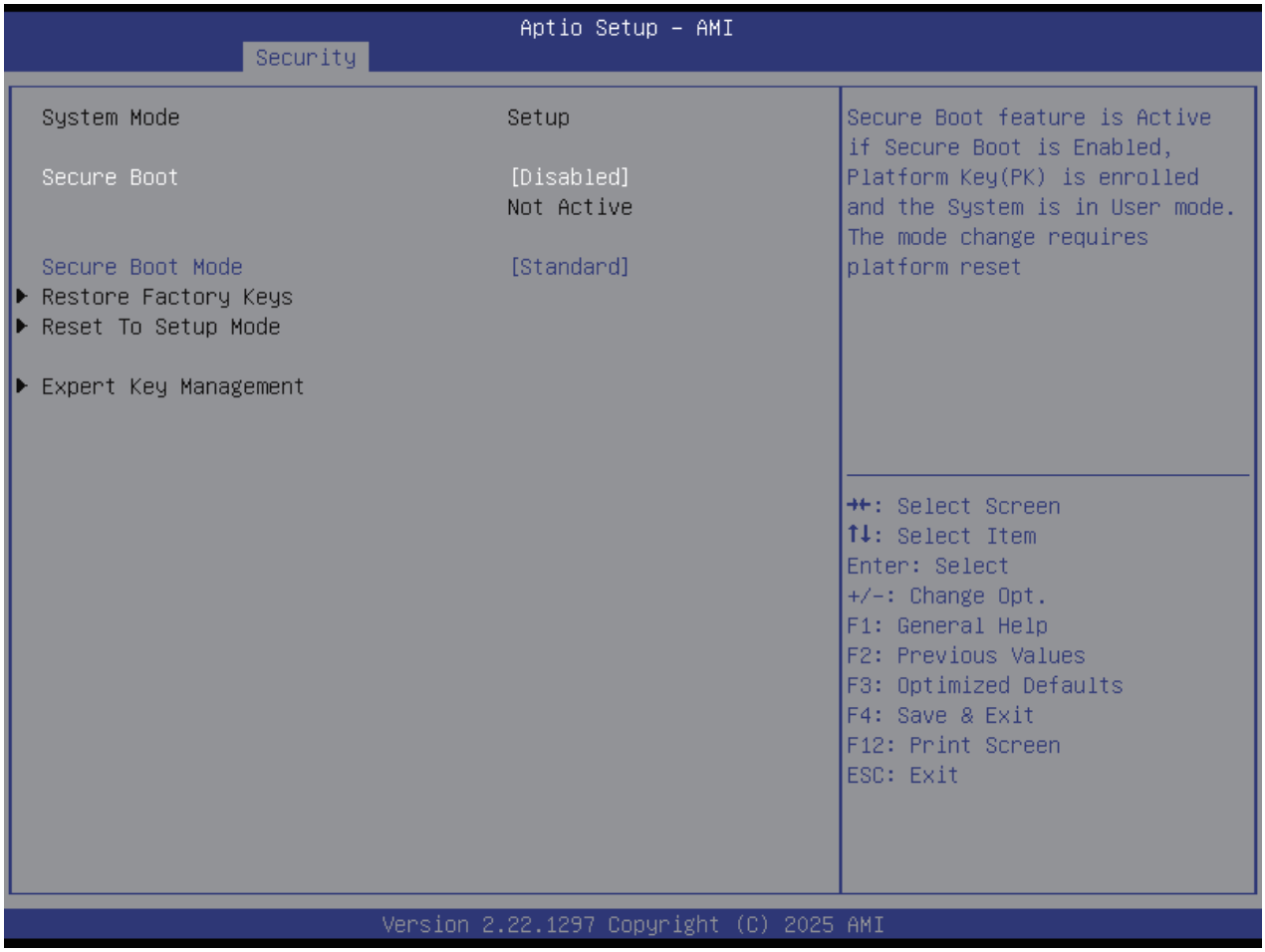
- **Secure Boot**

  Secure Boot configuration.

- **Secure Flash Update**

  Secure Flash Update Support

Secure Boot



This menu contains the following information:

- **System Mode**

  Display the System Mode.

- **Secure Boot**

  Secure Boot feature is Active if Secure Boot is Enabled, Platform Key (PK) is enrolled and the System is in User mode. The mode change requires platform reset.

- **Secure Boot Mode**

  Secure Boor mode options: Standard or Custom.

  In Custom mode, Secure Boot Policy variables can be configured by a physically present user without full authentication.

- **Restore Factory Keys**

  Force System to User Mode. Install factory default Secure Boot key databases.

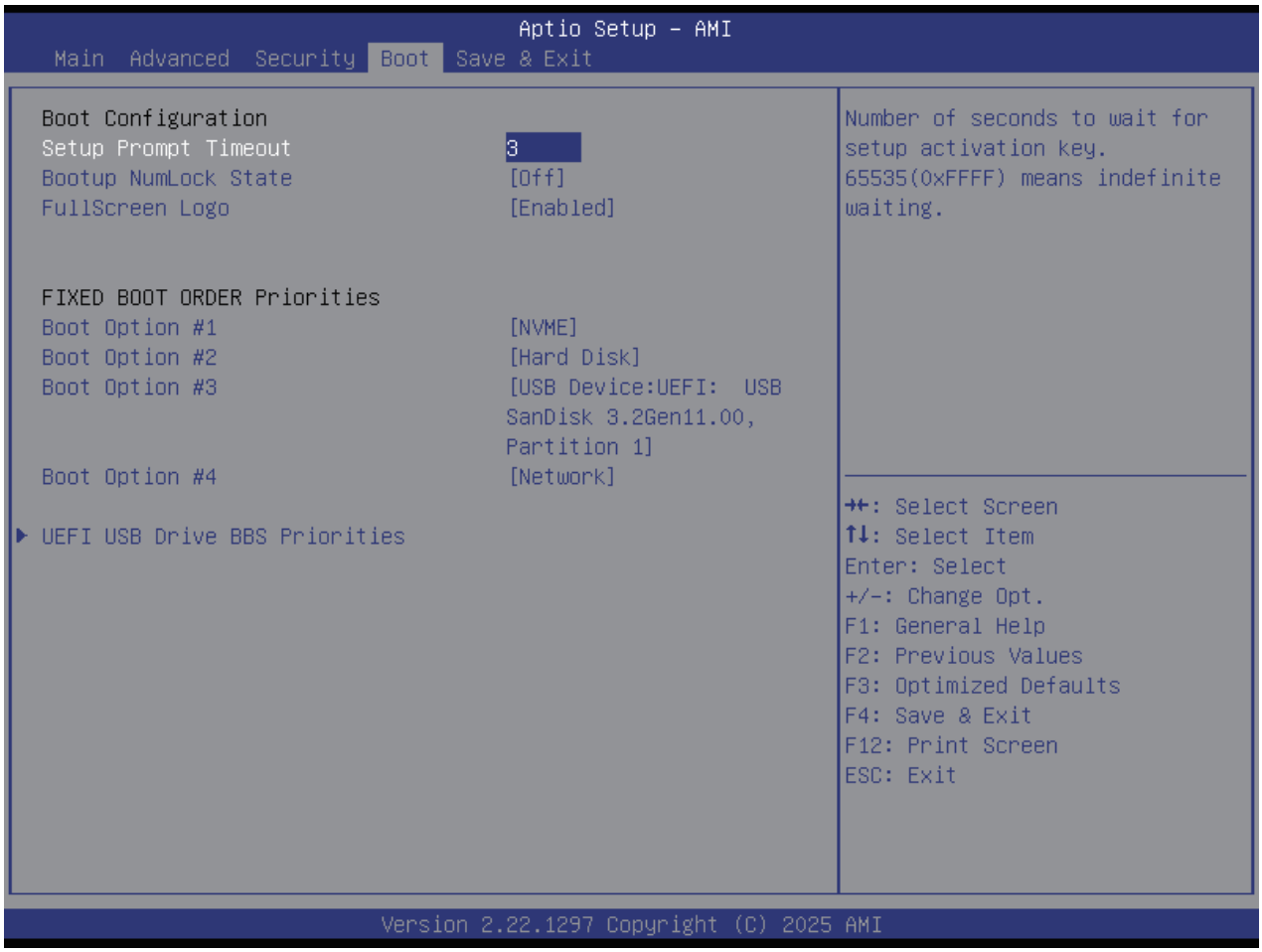
- **Reset To Setup Mode**

  Delete all Secure Boot key databases from NVRAM.

- **Expert Key Management**

  Enables expert users to modify Secure Boot Policy variables without full authentication.

3.3.4 Boot



This menu contains the following information:

- **Setup Prompt Timeout**

  Number of seconds to wait for setup activation key. 65535(0xFFFF) means indefinite waiting.

- **Bootup Num Lock State**

  Select the keyboard Num Lock state.

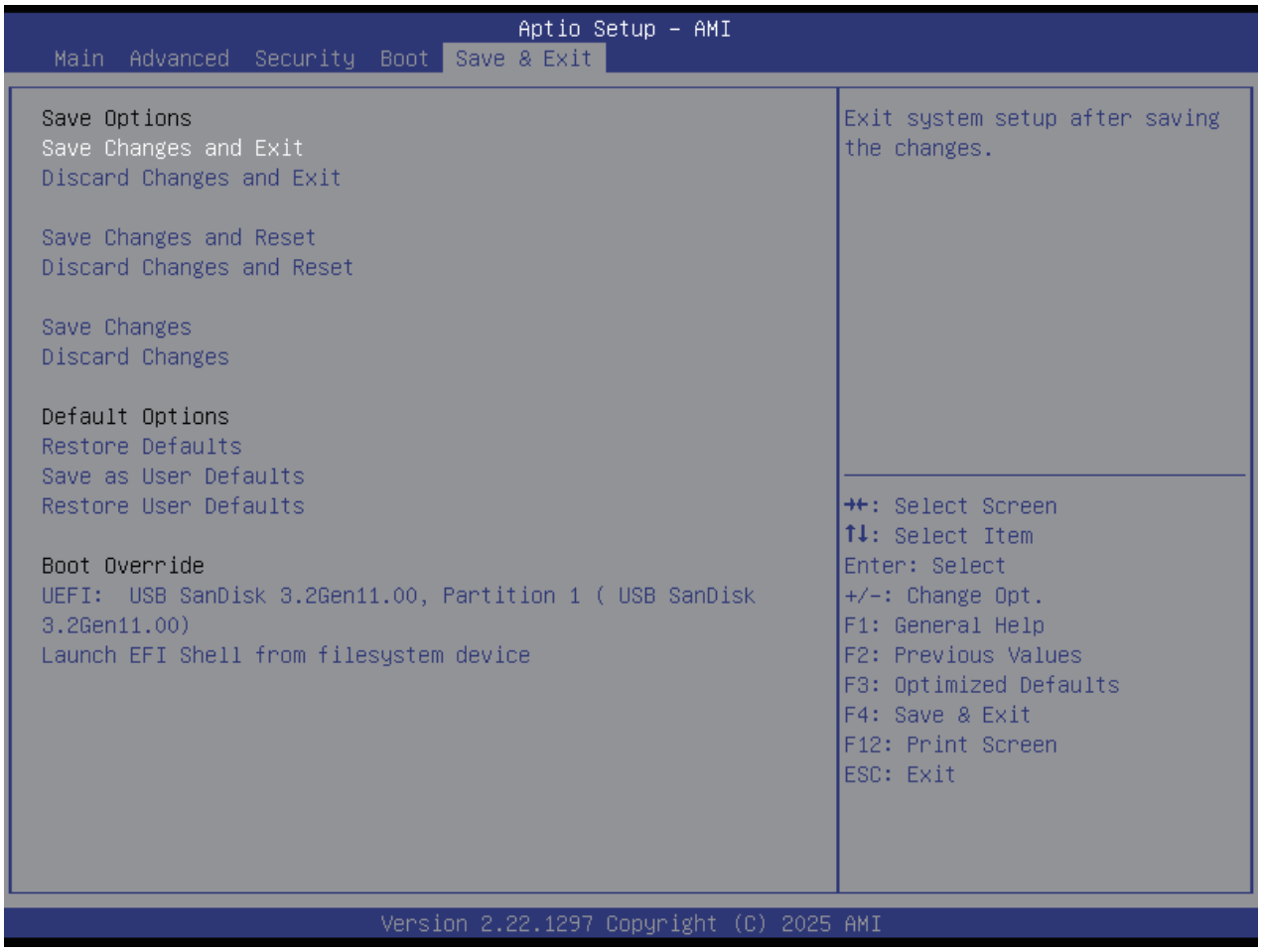
- **Full Screen Logo**

  Enable or Disable Full Screen Logo option.

- **FIXED BOOT ORDER Priorities**

  Display Boot order, and allow to set NVME/Hard Drive/USB/XXX boot order in this group.

3.3.5 Save & Exit



This menu contains the following information:

Save Options:

- **Save Changes and Exit**

  Exit the system after saving the changes.

- **Discard Changes and Exit**

  Exit system setup without saving any changes.

- **Save Changes and Reset**

  Reset the system after saving the changes.

- **Discard Changes and Reset**

  Reset system setup without saving any changes.

- **Save Changes**

  Save Changes done so far to any of far to any of the setup options.

- **Discard Changes:**

  Discard Changes done so far to any of the setup options.

Default Options

- **Restore Defaults**

  Restore or Load Default values for all the setup options.

- **Save as User Defaults**

  Save the changes done so far as User Defaults.

- **Restore User Defaults**

  Restore the User Defaults to all the setup options.

Boot Override

- **Launch EFI Shell from filesystem device**

  Attempts to launch EFI Shell application (Shell.efi) from one of the available filesystem devices.

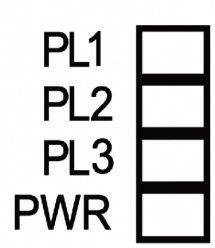
Chapter 4

PL1&PL2&PL3 Sample Code

4.1 Panel Light Control

The BPC-221 provides three software-controlled panel indicators: PL1, PL2 and PL3. Use SIO_GPIO_RW.exe to switch each light on or off, or to read its control bit. Writing 1 turns the selected light ON; writing 0 turns it OFF.

The indicators are arranged from top to bottom as PL1, PL2, PL3 and PWR. This guide covers PL1 through PL3. PWR is not assigned a control bit in this procedure.



Control addresses

Panel light	Signal	I/O address	Bit	ON / OFF
PL1	SIO_GPO86	0xA07	6	1 / 0
PL2	SIO_GPO87	0xA07	7	1 / 0
PL3	SIO_GPO84	0xA07	4	1 / 0

All three lights share one 8-bit I/O register. The bit number selects the light; it is not the number printed on the panel. Use only bits 6, 4 and 7 for the operations described here.

No connector access or wiring changes are required. The mapping and active-high behavior above reflect the supplied BPC-221 panel configuration.

4.2 Switch the Panel Lights

Start the utility

Open Command Prompt with Run as administrator. Change to the folder containing the existing SIO_GPIO_RW.exe and its matching runtime files. Replace the example path with your installation folder.

```
cd /d "C:\BPC221_GPIO\Release\X64"
```

For the x64 executable, keep SvApiLibx64.dll and SvIoCtrlx64.sys in the deployment folder. For the x86 executable, retain SvApiLib.dll and the supplied SvIoCtrl driver files. Use the executable and DLL of the same architecture.

Turn one light on or off

Light	Turn ON	Turn OFF
PL1	SIO_GPIO_RW.exe -w 0xA07 6 1	SIO_GPIO_RW.exe -w 0xA07 6 0
PL2	SIO_GPIO_RW.exe -w 0xA07 7 1	SIO_GPIO_RW.exe -w 0xA07 7 0
PL3	SIO_GPIO_RW.exe -w 0xA07 4 1	SIO_GPIO_RW.exe -w 0xA07 4 0

Run each command on its own line. With valid arguments, the tool changes the selected bit and preserves the other bits in the value written. Do not run competing control programs at the same time.

Turn all three lights on

```
SIO_GPIO_RW.exe -w 0xA07 6 1
SIO_GPIO_RW.exe -w 0xA07 7 1
SIO_GPIO_RW.exe -w 0xA07 4 1
```

Turn all three lights off

```
SIO_GPIO_RW.exe -w 0xA07 6 0
SIO_GPIO_RW.exe -w 0xA07 7 0
SIO_GPIO_RW.exe -w 0xA07 4 0
```

These commands run sequentially; they do not switch all three lights at exactly the same instant. PWR is unaffected by these mapped operations.

4.3 Read and Confirm Light State

Light	Read control bit
PL1	SIO_GPIO_RW.exe -r 0xA07 6
PL2	SIO_GPIO_RW.exe -r 0xA07 7
PL3	SIO_GPIO_RW.exe -r 0xA07 4

A returned bit of 1 indicates the ON control state; 0 indicates the OFF control state. Reading the register reports the electrical control value. It does not detect a failed lamp or independently confirm visible light output.

Example for PL1

```
SIO_GPIO_RW.exe -w 0xA07 6 1
SIO_GPIO_RW.exe -r 0xA07 6
```

Example messages for the ON state are shown below. Spacing may vary.

```
Initialize Driver OK...
Writing Data at address,bit    a07,6 is 1
Initialize Driver OK...
Reading Data at address,bit    a07,6 is 1
```

Check that PL1 is visibly illuminated. To finish the test with PL1 off, run:

```
SIO_GPIO_RW.exe -w 0xA07 6 0
```

Process exit codes

The original read/write implementation returns the observed bit as its process exit code. A return of 1 is therefore the ON state, not a conventional failure code. A return of 0 can mean OFF, displayed help or an unrecognized option. Initialization and some parameter failures return -1. Check the printed messages as well as the exit code.

Command arguments

Use lowercase -w to write and -r to read. The address is hexadecimal, the bit number is decimal and the intended values are 0 or 1. Always provide every argument. The tool does not fully validate missing or out-of-range values.

```
SIO_GPIO_RW.exe -h
```

Help requires driver initialization. The source edition includes Aicotek panel-light help text; an older executable can display generic examples until rebuilt. Use the commands in this guide for the BPC-221 lights.

4.4 Functions Used for Light Control

Initialization and release

```
BOOL InitApiLib(HMODULE *hModule);
BOOL DeinitApiLib(HMODULE *hModule);
```

InitApiLib loads the architecture-matched DLL, resolves its required exports and initializes hardware access. hModule receives the loaded module handle. It returns FALSE on a load or export failure; otherwise it returns the initialization result. DeinitApiLib releases the initialized library, frees the DLL and clears the handle. The supplied main function does not call the release function before exit.

Read and write a byte

```
BYTE IoRead8(WORD port);
BOOL IoWrite8(WORD port, BYTE value);
```

IoRead8 reads an 8-bit value from the I/O address. For panel lights, port is 0xA07. The selected light state is (value >> bit) & 1. This wrapper ignores the underlying read status; a failed read is not reliably reported.

IoWrite8 writes an entire byte and returns the underlying BOOL status. Before calling it, main reads the old byte and replaces only the requested bit:

```
new_value = (old_value & ~(1 << bit)) | (value << bit);
```

For PL1 use bit 6; for PL2 use bit 7; for PL3 use bit 4. value is 1 for ON or 0 for OFF. main then reads the byte again and reports the selected bit. It does not check the write API status. The read–modify–write sequence is not atomic; serialize access when integrating multiple callers.

Command handling

```
void printfHelp(char *argv0);
int main(int argc, char **argv);
```

printfHelp prints the command syntax and panel-light examples. argv0 supplies the executable name. main initializes the API, processes -r, -w or -h, converts the arguments, performs the requested operation and prints the result. The complete application listing follows in Section 4.6.

Checked reads in an application

```
BYTE value = 0;
BOOL ok = ReadIoPortByte(0xA07, &value);
if (ok) {
    bool pl1_on = ((value >> 6) & 1) != 0;
    bool pl2_on = ((value >> 7) & 1) != 0;
    bool pl3_on = ((value >> 4) & 1) != 0;
}
```

Call this fragment only after successful InitApiLib. ReadIoPortByte is the loaded byte-read API pointer; checking its BOOL result avoids relying on the unchecked IoRead8 wrapper. Release the initialized library when finished.

4.5 Build and Troubleshooting

Build the Aicotek source edition

The companion Aicotek_BPC221_GPIO_Source.zip contains the complete application source, both API headers and Visual Studio project files. It also contains README_EN.txt. It is a source package; it does not include a rebuilt executable or the DLL and driver implementation.

1. Extract the package and open SIO_GPIO_RW\SIO_GPIO_RW.sln.
2. Use the project's Visual C++ v142 toolset and Windows SDK 10.0, or deliberately retarget to a compatible installed toolchain. Select Release and Win32 or x64.
3. Build the solution and deploy the resulting executable beside the existing matching DLL and driver files. Run as administrator.

The source comments and help text have been updated for Aicotek and the three panel lights. The underlying hardware read/write logic is unchanged. Include SvApiInit.h in one translation unit because it defines global function pointers and function bodies.

Troubleshooting

Symptom	Action
Can't load dynamic library	Check that the matching DLL is present and has the same architecture as the executable.
Can't find export function	Use the matching supplied runtime library. Initialization checks more exports than those used for panel lights.
Initialize Driver Error	Run as administrator and check the supplied driver files and target Windows compatibility.
Wrong light changes	Verify PL1 = bit 6, PL2 = bit 7 and PL3 = bit 4, all at address 0xA07.
Control reads 1 but lamp is dark	Check the visible lamp condition and the unit configuration. Register readback is not optical feedback.
Script reports failure on ON	An exit code of 1 is the ON bit value in this utility. Check printed messages before treating it as an error.

Integration notes

Validate all arguments and check low-level API status in production applications. The write path does not check that the value argument exists. The utility does not implement blinking, timing guarantees or saved power-on states. No hardware command was executed, and this source edition was not compiled during document preparation.

4.6 Application Source

Complete SIO_GPIO_RW.cpp from the companion Aicotek source edition. The two API headers and project files are included in the ZIP.

```
//---------------------------------------------------------------------------
// Aicotek BPC-221 panel indicator control
// Customer source edition
// License : The modified BSD license
//---------------------------------------------------------------------------
#include <stdio.h>
#include <windows.h>
#include <winioctl.h>
#include <string.h>
#include <cstdio>
#include <iostream>
#include "SvApiInit.h"
HMODULE module;
// Address 0xA07: PL1 = GPO86 / bit 6, PL2 = GPO87 / bit 7,
// PL3 = GPO84 / bit 4. Write 1 for ON and 0 for OFF.
// PWR is not controlled by these commands.
// Windows command line application
void printfHelp(char *argv0)
{
    printf("Aicotek BPC-221 panel light control\n");
    printf("PL1: 0xA07 bit 6; PL2: bit 7; PL3: bit 4\n");
    printf("Write 1 = ON, 0 = OFF\n\n");
    printf("<%s> <-w/-r> <address> <bit> <0/1>\n", argv0);
    printf("For Example:\n");
    printf("%s -w 0xA07 6 1  \n", argv0);
    printf("%s -r 0xA07 6  \n", argv0);
    printf("\n%s -h:for help!\n", argv0);
}
int main(int argc, char** argv)
{
    WORD address;
    BYTE BYTEVal;
    BYTE Temp;
    UINT8 bit;  // Byte register bit index: 0-7
    UINT8 val;  // Panel light state: 0 = OFF, 1 = ON
    if (!InitApiLib(&module))
    {
        printf("Initialize Driver Error...\n\n");
        return  -1;
    }
    printf("Initialize Driver OK...\n\n");
    if (argv[1] == NULL || !strcmp(argv[1], "-h"))
    {
        printfHelp(argv[0]);
        return 0;
    }
```

```
    if (!strcmp(argv[1], "-r"))
    {
        if (argv[2] != NULL && argv[3] != NULL)
        {
            address = (WORD)strtoul(argv[2], NULL, 16);
            bit = (UINT8)strtoul(argv[3], NULL, 10);
        }
        else
        {
            printf("Wrong parameter!\n");
            return  -1;
        }
        BYTEVal = IoRead8(address);
        Temp = ((BYTEVal >> bit) & 0x1);
        printf("Reading Data at address,bit\t");
        printf("%lx,%d is %x\n", address, bit, Temp);
        return Temp;
    }
    else if (!strcmp(argv[1], "-w"))
    {
        if (argv[2] != NULL && argv[3] != NULL)
        {
            address = (WORD)strtoul(argv[2], NULL, 16);
            bit = (UINT8)strtoul(argv[3], NULL, 10);
            val = (UINT8)strtoul(argv[4], NULL, 16);
        }
        else
        {
            printf("Wrong parameter!\n");
            return  -1;
        }
        BYTEVal = IoRead8(address);
        Temp = ((BYTEVal&(~(1 << bit))) | (val << bit));
        IoWrite8(address, Temp);
        //printf("Finishing Data Write Process......\n");
        printf("Writing Data at address,bit\t\t");
        BYTEVal = IoRead8(address);
        printf("%lx,%d is %x\n", address, bit, ((BYTEVal >> bit) & 0x1));
        return ((BYTEVal >> bit) & 0x1);
    }
    else
    {
        printf("Wrong parameter!\n");
    }
    return 0;
}
```

Chapter 5

Watchdog Control Sample Code

5.1 Watchdog Control

This guide explains how to use the supplied WatchDogITE Windows utility to start, refresh and stop the ITE watchdog. It includes a function reference, the complete application source listing and a companion Visual C++ source project.

A watchdog counts down independently of the feeding program. If software stops refreshing it, timeout is intended to reset the computer. Save your work before running a timeout test.

Prepare the utility

Extract the original Win_WatchDogITE.zip package. Open Command Prompt with Run as administrator and change to the appropriate Release folder. Keep the executable, DLL and driver together.

Folder	Required files
Release\X64	WatchDogITE.exe, SvApiLibx64.dll, SvIoCtrlx64.sys
Release\X86	WatchDogITE.exe, SvApiLib.dll, SvIoCtrl.sys

Choose matching executable and DLL architectures. A tested Windows version matrix is not supplied. The commands below assume that the current directory contains WatchDogITE.exe.

Commands at a glance

Command	Behavior
WatchDogITE.exe start 10	Starts a 10-second countdown and exits. It does not keep feeding.
WatchDogITE.exe enable	Starts a 6-second countdown and feeds every 3 seconds in a continuous loop.
WatchDogITE.exe stop	Calls the watchdog stop API and exits.
WatchDogITE.exe -h	Displays help after initializing the driver.

Prepared from the supplied readme and C++ project. The application help reports version 20240924. Hardware behavior has not been tested for this document.

5.2 Operating the Watchdog

Run a deliberate timeout test

```
WatchDogITE.exe start 10
```

The utility calls WatchdogStart(10) and prints "watchdog is starting...". The original readme describes a restart after 10 seconds. The process then exits; the countdown remains armed. To cancel the test, run the following command before timeout:

```
WatchDogITE.exe stop
```

Run continuous monitoring

```
WatchDogITE.exe enable
```

Keep this process running. It starts the watchdog with a 6-second timeout, sleeps for 3000 milliseconds, then reloads the counter with 6. The loop repeats indefinitely. There is no configurable timeout argument for enable mode.

If the process or operating system stops feeding the timer, timeout is intended to cause a restart. The exact reset behavior and timing depend on the installed hardware and firmware. This example feeds unconditionally; it does not check the health of your application.

Stop continuous monitoring

Prepare a second administrator Command Prompt with the stop command ready. Terminate the enable process, then immediately run WatchDogITE.exe stop from that second prompt, within the remaining timeout. The sample provides no graceful loop shutdown and does not automatically disable the watchdog when terminated.

Do not rely on stop while enable is still running: the next loop iteration writes the timeout register again. For a production application, coordinate loop cancellation and WatchdogStop() in the same shutdown path.

Timeout input

start parses a decimal argument with strtoul and casts it to an 8-bit BYTE. The API parameter is named seconds. The sample does not validate numeric syntax or the hardware-supported range. Use the documented 10-second example for initial verification; values above 255 are truncated and zero has no documented meaning in the supplied materials.

5.3 Function Reference

Application entry points

main(int argc, char** argv) initializes the API library, selects start, stop or enable, and executes that branch. Initialization failure or a missing start argument returns -1. Help and other completed paths return 0; an unknown command also falls through with 0. A zero exit code is therefore not proof that the watchdog operation succeeded.

printfHelp(char* argv0) prints the utility version, command syntax and examples. argv0 supplies the executable name. main initializes the driver before handling help.

Library initialization and cleanup

BOOL InitApiLib(HMODULE* hModule) loads SvApiLibx64.dll for an x64 build or SvApiLib.dll otherwise, resolves the required exported functions and calls InitializeLib(). It returns FALSE if loading or symbol lookup fails, otherwise the initialization result. Its required export list includes other hardware services in addition to watchdog functions.

BOOL DeinitApiLib(HMODULE* hModule) calls DeinitializeLib(), frees the DLL and clears the handle. A null handle returns TRUE. This helper is defined in SvApiInit.h but is not called by the supplied main. Releasing the library is separate from disabling the watchdog.

Watchdog API functions

VOID WatchdogStart(BYTE seconds) is a function pointer bound to the DLL export SvWatchdogStart. It starts the watchdog with the requested seconds value. start passes the parsed byte; enable passes 6. The DLL implementation and supported hardware timeout range are not included in the source.

VOID WatchdogStop(VOID) is bound to SvWatchdogStop. It requests watchdog disable. Both watchdog functions return VOID, so the application cannot obtain a success code from these calls. Its printed messages report the selected path rather than verified hardware state.

Byte input and output helpers

BOOL IoWrite8(WORD port, BYTE value) calls WriteIoPortByte and returns its status. The feeding loop uses this helper but ignores the returned status. BYTE IoRead8(WORD port) reads a byte using ReadIoPortByte; its wrapper ignores the read status. It is available in the shared header but is not used by this application.

5.4 Feeding Sequence and Source Project

The enable loop writes the following sequence after each Sleep(3000). This table describes the supplied implementation; it is not a general register specification for other ITE devices.

Orde	Port	Values written	Purpose in the sample
1	0x2E	0x87, 0x01, 0x55, 0x55	Enter the configuration sequence.
2	0x2E / 0x2F	0x22 / 0x80	Write configuration register 0x22.
3	0x2E / 0x2F	0x07 / 0x07	Select logical device 7.
4	0x2E / 0x2F	0x73 / 6	Reload the watchdog timeout.
5	0x2E / 0x2F	0x02 / 0x02	Exit the configuration sequence.

Build the companion source

Extract Aicotek_WatchDogITE_Source.zip and open WatchDogITE\WatchDogITE.sln. The project uses Visual C++ platform toolset v142 and Windows SDK 10.0. Select Release with Win32 or x64. The project specifies output folders ..\..\Release\X86 and ..\..\Release\X64 relative to the project directory.

Deploy the resulting executable with the original matching DLL and driver listed in Section 5.1. The companion archive contains source and project files only; it does not contain a rebuilt binary or the implementation of the DLL and kernel driver.

File	Purpose
WatchDog.cpp	Command handling, help and continuous feeding loop.
SvApiInit.h	Dynamic API loading, cleanup and hardware I/O wrappers.
SvApiInitDef.h	API function pointer type declarations.
WatchDogITE.sln and WatchDog.vcxproj	Solution and build configuration.

Integration and troubleshooting

For DLL load errors, check architecture and file placement. For missing export errors, use the matching supplied DLL. For initialization errors, check administrator rights and driver availability. The sample has limited argument validation and error reporting, no graceful loop cancellation and no automatic API cleanup. Address these when integrating it into an application.

Only comments were changed for Aicotek branding and English wording. Executable source content and project settings were preserved. This source edition has not been compiled or tested on hardware.

5.5 Application Source

Complete WatchDog.cpp from the companion Aicotek source package. Both supporting headers are included in full in the ZIP.

```
//--------------------------------------------------------------------------
//     Author : Aicotek Software Research Department
//       Mail : bcc@Aicotek.com
//        Web : https://www.Aicotek.com/
//    License : The modified BSD license
//
//    Copyright 2003-2022 Aicotek.com. All rights reserved.
//--------------------------------------------------------------------------
#include <stdio.h>
#include <windows.h>
#include <winioctl.h>
#include <string.h>
#include <cstdio>
#include <iostream>
#include "SvApiInit.h"
HMODULE module;
//Ver 0.0.1
//Windows command line application
void printfHelp(char *argv0)
{
    printf("Ver: 20240924\n\n");
    printf("<%s> <start/stop/enable> <time> \n", argv0);
    printf("For Example:\n");
    printf("%s start 10  \n", argv0);
    printf("%s enable \n", argv0);
    printf("%s stop  \n", argv0);
    printf("\n%s -h:for help!\n", argv0);
}
int main(int argc, char** argv)
{
    BYTE time;
    if (!InitApiLib(&module))
    {
        printf("Initialize Driver Error...\n\n");
        return  -1;
    }
    printf("Initialize Driver OK...\n\n");
    if (argv[1] == NULL || !strcmp(argv[1], "-h"))//Print help when no command or -h is supplied
    {
        printfHelp(argv[0]);
        return 0;
    }
    if (!strcmp(argv[1], "start"))
    {
        if (argv[2] != NULL)
        {
            time = (UINT8)strtoul(argv[2], NULL, 10);
        }
        else
        {
            printf("Wrong parameter!\n");
            return  -1;
        }
```

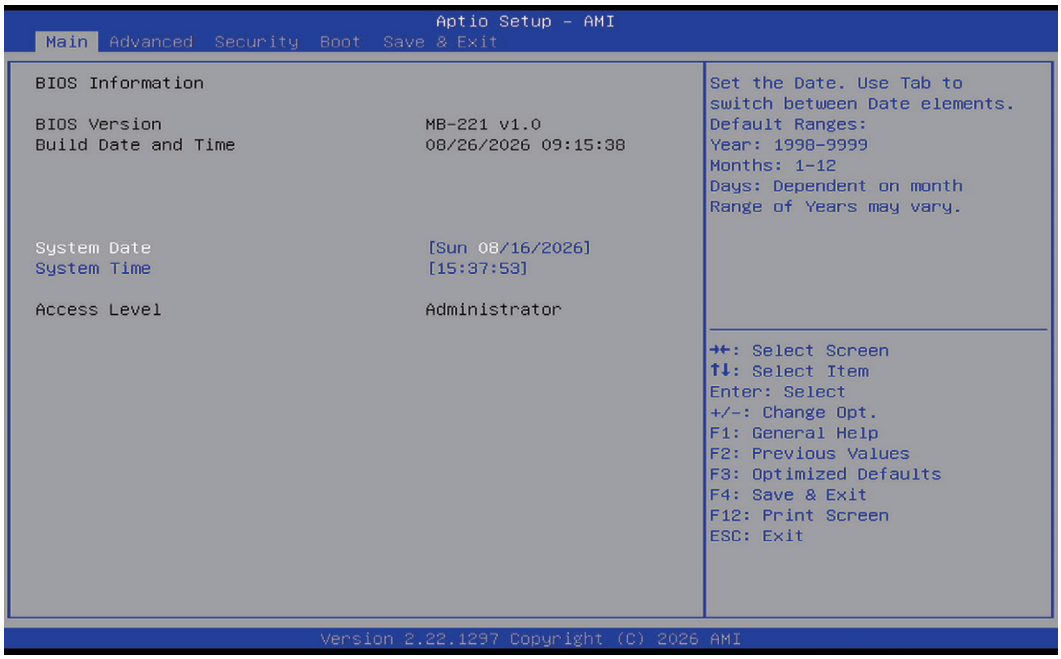
```
        WatchdogStart(time);
        printf("watchdog is starting...\n");
    }
    else if (!strcmp(argv[1], "stop"))
    {
        WatchdogStop();
        printf("watchdog is stopping...\n");
    }
    else if (!strcmp(argv[1], "enable"))
    {
        WatchdogStart(6);
        printf("watchdog is enable...\n");
        while (1)
        {
            Sleep(3000);
            IoWrite8(0x2E, 0x87);
            IoWrite8(0x2E, 0x01);
            IoWrite8(0x2E, 0x55);
            IoWrite8(0x2E, 0x55);
            IoWrite8(0x2E, 0x22);
            IoWrite8(0x2F, 0x80);
            IoWrite8(0x2E, 0x07);
            IoWrite8(0x2F, 0x07);
            IoWrite8(0x2E, 0x73);
            IoWrite8(0x2F, 6);
            IoWrite8(0x2E, 0x02);
            IoWrite8(0x2F, 0x02);
        }
    }
    else
    {
        //printf("Wrong parameter!\n");
    }
    return 0;
}
```

Appendix A

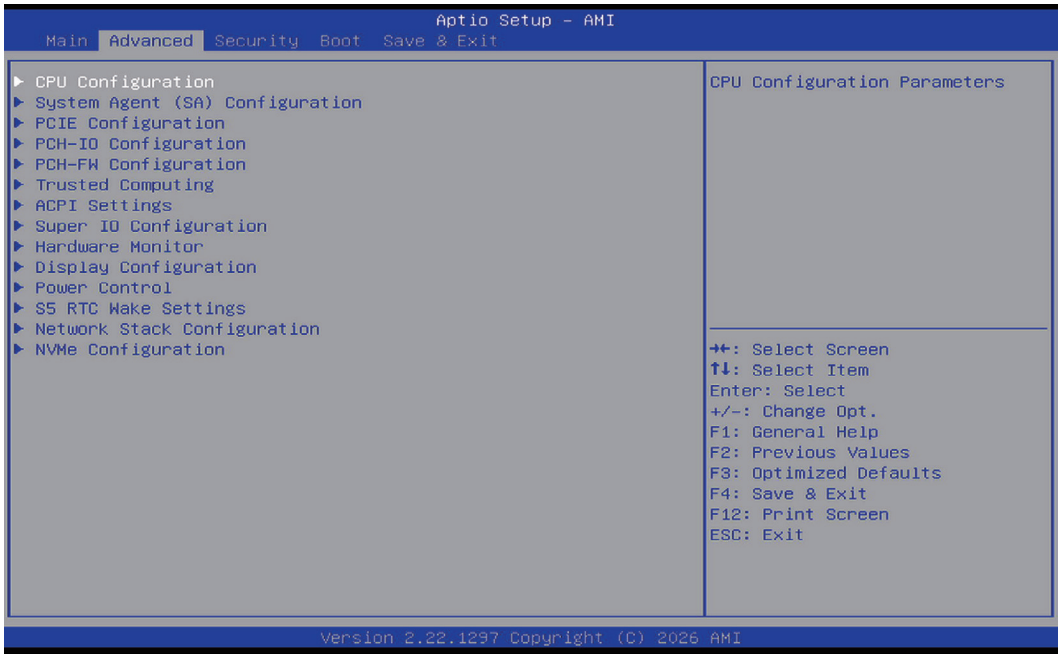
BIOS Hidden Menu Access Tutorial

A.1 Enter BIOS Setup

Step 1: Turn on the computer and repeatedly press the Delete key during startup to enter BIOS Setup.



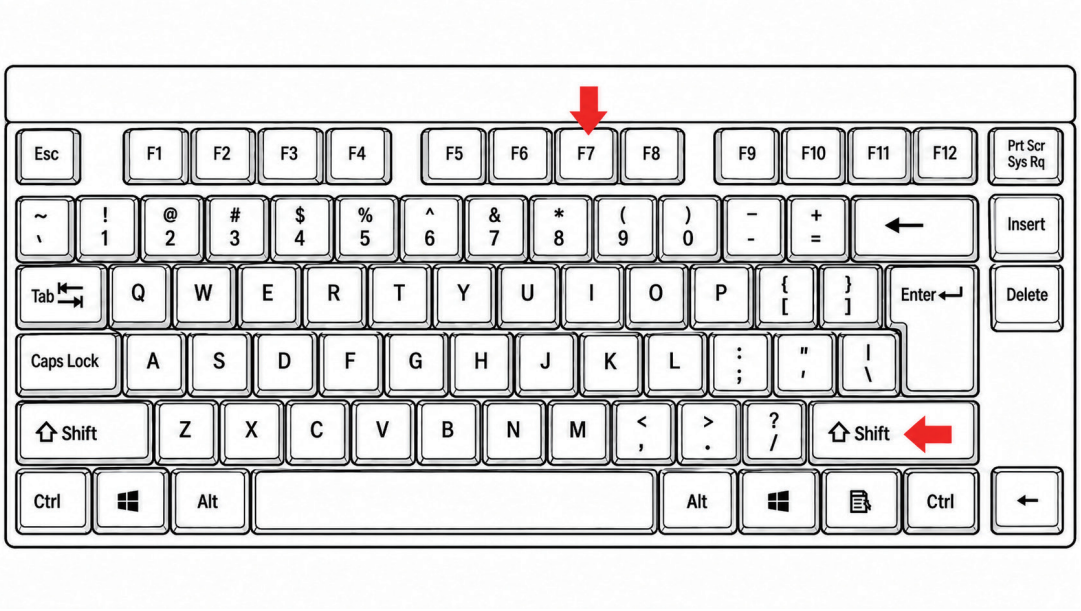
At this point, the Advanced tab does not show the hidden menus.



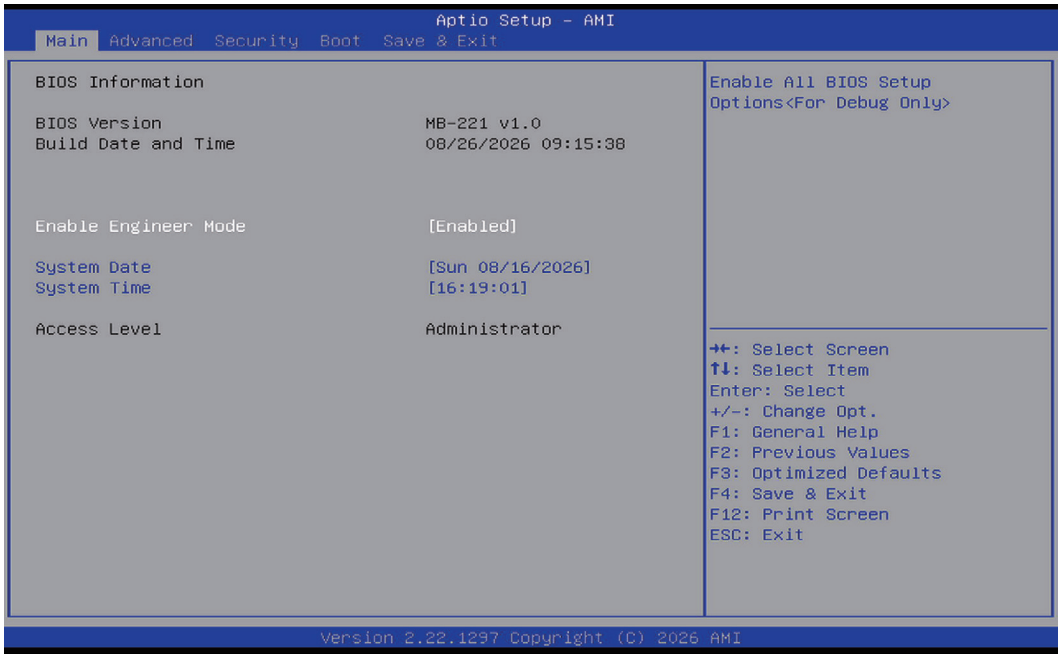
A.2 Reveal the Debug Setup Option

Step 2: Press Right Shift and F7 at the same time.

Important: Use the Shift key on the right side of the keyboard.

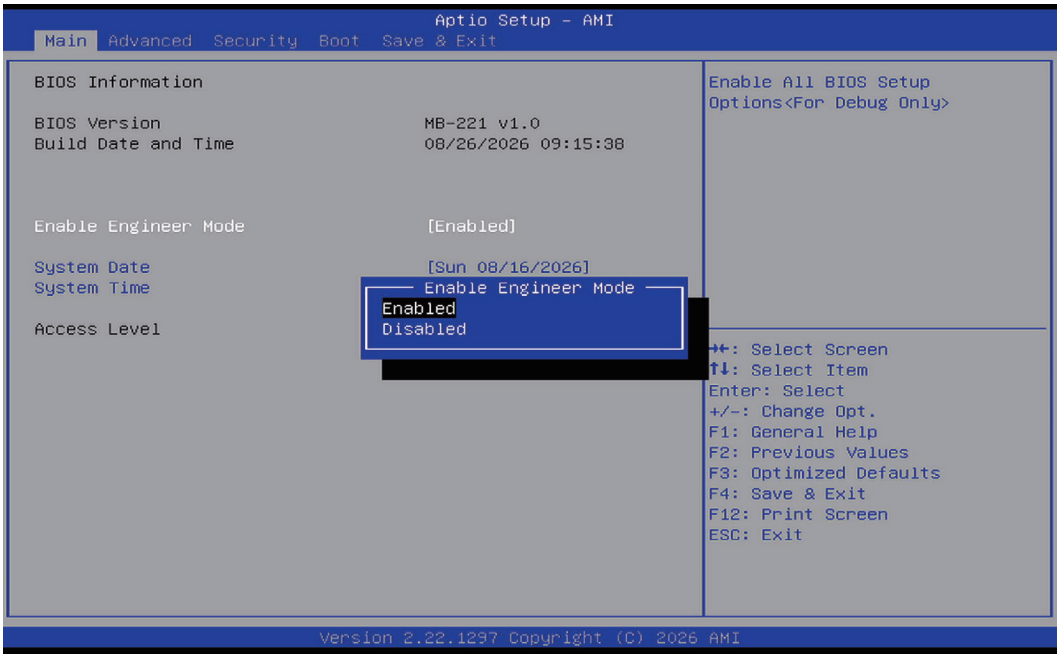


The "Enabled Engineer Mode" option appears on the Main tab.



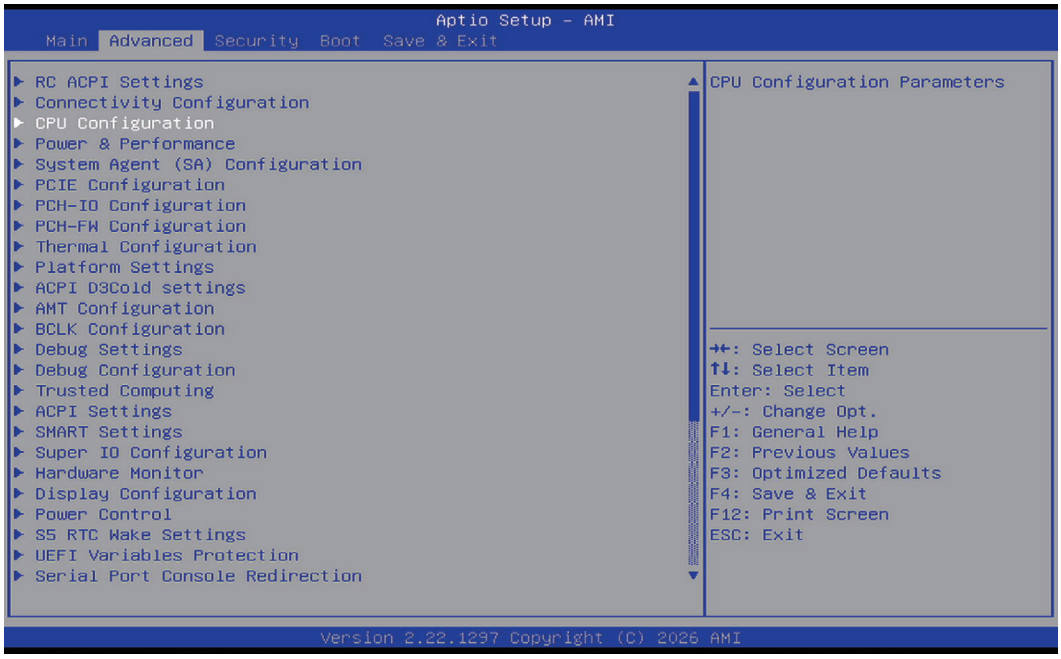
A.3 Enable Access to Hidden Menus

Step 3: Select "Enabled Debug Setup Menu" and press Enter.
Choose "Disabled" to make the hidden menus available.



A.4 Open the Advanced Tab

Step 4: Select the Advanced tab. The hidden menus are now displayed, as shown below.



Appendix B

Alarm and Protection Functions

B.1 Buzzer Alarm Indications

Buzzer Indication	Description
Continuous Beeping	Memory not detected
One Short Beep	Memory POST passed; system continues booting

B.2 Power Protection Specifications

Protection Function	Threshold / Description
Over-Voltage Protection	26V
Under-Voltage Protection	10V
Over-Current Protection	6A
Reverse-Polarity Protection	Supported

Baltesakker 17, 5625TC
Eindhoven, The Netherlands
www.aicotek.com
info@aicotek.com

AICOTEK

Empowering the world with AI solutions