Arrakis Pico Mk4 Series

Version:
v1.0.1

Date:
15.08.2025

Contents

1 Copyright

2 Regulatory Compliances

2.1 Complies with the following EU directives

Radio Equipment Directive (2014/53/EU) only applies to devices containing radio module EM05-G.

No	Short Name
2014/35/EU	Low Voltage Directive (LVD)
2014/53/EU	Radio Equipment Directive (RED)
2014/30/EU	Electromagnetic Compatibility (EMC)
2011/65/EU	Restriction of the use of certain hazardous substances in electrical and electronic equipment Directive (RoHS2)
2015/863/EU	Amendment to Annex II in Directive 2011/65/EU regards the list of restricted substances (RoHS3)

2.2 References of standards applied

Standard	Reference	Issue
EN 18031-1	Common security requirements for radio equipment - Part 1: Internet connected radio equipment	2024
EN 55032	Electromagnetic compatibility of multimedia equipment - Emission Requirements	2015+A11:2020+A1:2020
EN 55035	Electromagnetic compatibility of multimedia equipment - Immunity requirements	2017+A11:2020
EN (IEC) 61000-3-2	Electromagnetic compatibility (EMC) - Part 3-2: Limits - Limits for harmonic current emissions	2014 2019+A1:2021
EN 61000-3-3	Electromagnetic compatibility (EMC) - Part 3-3: Limits - Limitation of voltage changes, voltage fluctuations and flicker in public low-voltage supply systems	2013 2013+A2:2021+AC:2022
EN 61000-4-2	Electromagnetic compatibility (EMC). Testing and measurement techniques. Electrostatic discharge immunity test	2009
EN IEC 61000-4-3	Electromagnetic compatibility (EMC) - Part 4-3: Testing and measurement techniques - Radiated, radio-frequency, electromagnetic field immunity test	2020
EN 61000-4-4	Electromagnetic compatibility (EMC) - Part 4-4 : Testing and measurement techniques - Electrical fast transient/burst immunity test	2012
EN 61000-4-5	Electromagnetic compatibility (EMC) - Part 4-5: Testing and measurement techniques - Surge immunity test	2014+A1:2017
EN 61000-4-6	Electromagnetic compatibility (EMC) - Part 4-6: Testing and measurement techniques - Immunity to conducted disturbances, induced by radio-frequency fields	2014+AC:2015
EN 61000-4-8	Electromagnetic compatibility (EMC) - Part 4-8: Testing and measurement techniques - Power frequency magnetic field immunity test	2010
EN IEC 61000-4-11	Electromagnetic compatibility (EMC) - Part 4-11: Testing and measurement techniques - Voltage dips, short interruptions and voltage variations immunity tests	2020+AC:2022
EN 301 489-1 (module)	ElectroMagnetic Compatibility (EMC) standard for radio equipment and services; Part 1: Common technical requirements; Harmonised Standard for ElectroMagnetic Compatibility	V2.2.3
EN 301 489-52 (module)	ElectroMagnetic Compatibility (EMC) standard for radio equipment and services; Part 52: Specific conditions for Cellular Communication User Equipment (UE) radio and ancillary equipment; Harmonised Standard for ElectroMagnetic Compatibility	V1.2.1
Draft EN 301 489-19 (module)	ElectroMagnetic Compatibility (EMC) standard for radio equipment and services - Part 19: Specific conditions for Receive Only Mobile Earth Stations (ROMES) operating in the 1,5 GHz band providing data communications and GNSS receivers operating in the RNSS band (ROGNSS) providing positioning, navigation and timing data	V2.2.0
ETSI EN 301 908-1	IMT cellular networks; Harmonised Standard for access to radio spectrum; Part 1: Introduction and common requirements Release 15	V15.1.1

2.3 FCC PART 15 VERIFICATION STATEMENT

WARNING

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

Notice: The changes or modifications not expressly approved by the party responsible for compliance could void the user's authority to operate the equipment.

This device complies with part 15 of the FCC Rules. Operation is subject to the following two conditions: (1) This device may not cause harmful interference, and (2) this device must accept any interference received, including interference that may cause undesired operation.

May Contain transmitter module:

- RYK-WNFQ262ACNIBT
- N7NEM75T
- XMR2021EM05G

2.4 ICES-003 ISSUE 7 VERIFICATION STATEMENT

CAN ICES3(A)/NMB3(A)

This device complies with CAN ICES-003 Issue 7 Class A. Operation is subject to the following two conditions: (1) This device may not cause harmful interference, and (2) this device must accept any interference received, including interference that may cause undesired operation. Cet appareil est conforme à la norme CAN ICES-003 Issue 7 Class A. Le fonctionnement est soumis auxdeux conditions suivantes : (1) cet appareil ne doit pas causer d'interférences nuisibles et (2) cet appareil doit accepter toute interférence reçue, y compris les interférences pouvant opération indésirable.

May Contain transmitter module:

- 2417C-EM75T
- 6158A-FQ262ACNIBT

3 Safety Instructions

Please read these instructions carefully and retain them for future reference.

1. Disconnect this equipment from the power outlet before cleaning. Do not use liquid or sprayed detergent for cleaning. Use a moist cloth or sheet.
2. Keep this equipment away from humidity.
3. Ensure the power cord is positioned to prevent tripping hazards and do not place anything on top of it.
4. Pay attention to all cautions and warnings on the equipment.
5. If the equipment is not used for an extended period, disconnect it from the main power to avoid damage from transient over-voltage.
6. **Prolonged usage with less than 12V may damage the PSU or destroy the mainboard.**
7. Never pour any liquid into openings as this could cause fire or electrical shock.
8. Have the equipment checked by service personnel if:
    - The power cord or plug is damaged.
    - Liquid has penetrated the equipment.
    - The equipment has been exposed to moisture in a condensation environment.
    - The equipment does not function properly, or you cannot get it to work by following the user manual.
    - The equipment has been dropped and damaged.
9. Do not leave this equipment in an unconditioned environment, with storage temperatures below -20 degrees or above 60 degrees Celsius for extended periods, as this may damage the equipment.
10. Unplug the power cord when performing any service or adding optional kits.
11. Lithium Battery Caution:
    - Risk of explosion if the battery is replaced incorrectly. Replace only with the original or an equivalent type recommended by the manufacturer. Dispose of used batteries according to the manufacturer's instructions.
    - Do not remove the cover, and ensure no user-serviceable components are inside. Take the unit to a service center for service and repair.

4 Product Specifications

Feature	Specification	Arrakis Pico Mk4 - Standard 8GB/16GB	Arrakis Pico Mk4 - Light	Arrakis Pico Mk4 - Headless
Processor	CPU	Intel Atom® x6413E Processor, 1.5/3.0 GHz	Intel Atom® x6211E Processor, 1.3/3.0 GHz	Intel Atom® x6211E Processor, 1.3/3.0 GHz
Memory	RAM	8GB / 16GB LP-DDR4	4GB LP-DDR4	4GB LP-DDR4
Storage	SSD	1x NVMe SSD on M.2 socket 2 (128 GB up to 1 TB)	1x NVMe SSD on M.2 socket 2 (128 GB up to 1 TB)	1x NVMe SSD on M.2 socket 2 (128 GB up to 1 TB)
Security	TPM	TPM 2.0 (Infineon SLB 9670VQ2.0)	TPM 2.0 (Infineon SLB 9670VQ2.0)	TPM 2.0 (Infineon SLB 9670VQ2.0)
	Secure Boot	Yes	Yes	Yes
I/O Ports	HDMI	1x HDMI	1x HDMI	-
	Gigabit Ethernet	3x 2.5 GbE (i226-IT)	2x 2.5 GbE (i226-V)	3x 2.5 GbE (i226-V)
	USB 3.0	3x USB 3.0	3x USB 3.0	1x USB 3.0
	USB 2.0	1x USB 2.0	1x USB 2.0	-
	Serial Ports	1x RS232/RS-485 (Bios selectable)	-	1x RS232/RS-485 (Bios selectable)
	DIO	1x DI, 12-24V 1x DO, 12-24V max. 2A (voltage defined by DC input)	-	1x DI, 12-24V 1x DO, 12-24V max. 2A (voltage defined by DC input)
	M.2	Socket 1: M.2 3042 B-Key, USB 3.0/2.0/SATA for SSD or LTE Socket 2: M.2 2242 B-Key, USB 2.0/PCIe x2 for NVMe SSD/WiFi/AI	Socket 1: M.2 3042 B-Key, USB 3.0/2.0/SATA for SSD or LTE Socket 2: M.2 2242 B-Key, USB 2.0/PCIe x2 for NVMe SSD/WiFi	Socket 1: M.2 3042 B-Key, USB 3.0/2.0/SATA for SSD or LTE Socket 2: M.2 2242 B-Key, USB 2.0/PCIe x2 for NVMe SSD/WiFi
	SIM Slot	1 push-push Type Nano-SIM Slot	1 push-push Type Nano-SIM Slot	1 push-push Type Nano-SIM Slot
Connectivity	LTE (optional)	Quectel EM05-G Quectel EM05-E	Quectel EM05-G Quectel EM05-E	Quectel EM05-G Quectel EM05-E
	WLAN (optional)	Sparklan WNFQ-262ACNI(BT) with SATA-SSD	Sparklan WNFQ-262ACNI(BT) with SATA-SSD	Sparklan WNFQ-262ACNI(BT) with SATA-SSD
Extension	AI Booster	Hailo-8™ edge AI processor with SATA-SSD		-

Model	RAM	LTE	WiFi	AI processor
Arrakis Pico Mk4 Standard 8 GB	8 GB LPDDR4			
Arrakis Pico Mk4 Standard 16 GB	16 GB LPDDR4			
Arrakis Pico Mk4 Light	4 GB LPDDR4			
Arrakis Pico Mk4 Headless	4 GB LPDDR4			
Arrakis Pico Mk4 Standard 8 GB w/ LTE	8 GB LPDDR4	Cat. 4		
Arrakis Pico Mk4 Standard 16 GB w/ LTE	16 GB LPDDR4	Cat. 4		
Arrakis Pico Mk4 Light w/ LTE	4 GB LPDDR4	Cat. 4		
Arrakis Pico Mk4 Headless w/ LTE	4 GB LPDDR4	Cat. 4		
Arrakis Pico Mk4 Standard 8 GB w/ Wifi	8 GB LPDDR4		WiFi Client/ Soft AP	
Arrakis Pico Mk4 Standard 16 GB w/ Wifi	16 GB LPDDR4		WiFi Client/ Soft AP	
Arrakis Pico Mk4 Light w/ Wifi	4 GB LPDDR4		WiFi Client/ Soft AP	
Arrakis Pico Mk4 Headless w/ Wifi	4 GB LPDDR4		WiFi Client/ Soft AP	
Arrakis Pico Mk4 Standard 8 GB w/ AI	8 GB LPDDR4			Hailo-8™ edge AI processor
Arrakis Pico Mk4 Standard 16 GB w/ AI	16 GB LPDDR4			Hailo-8™ edge AI processor

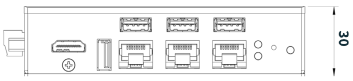
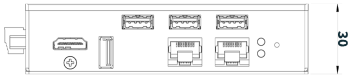
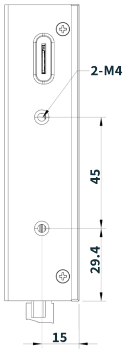
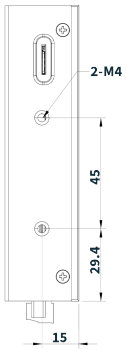
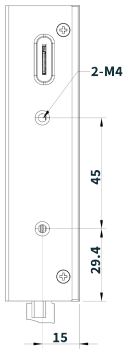
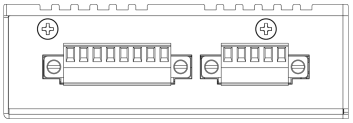
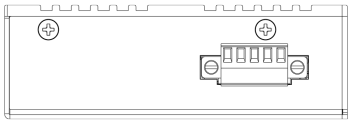
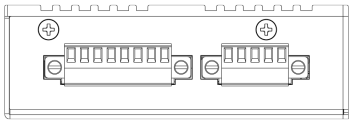
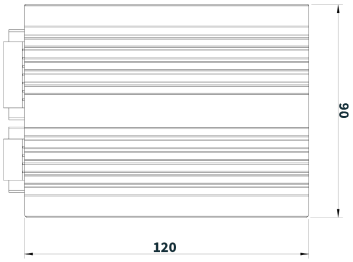
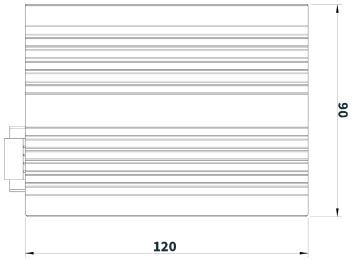
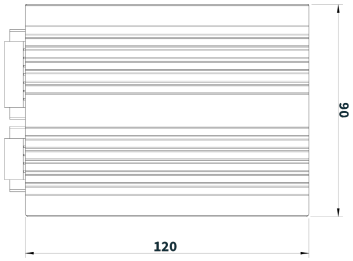
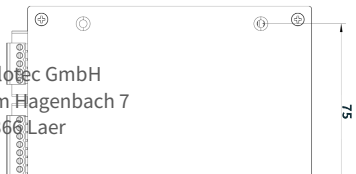
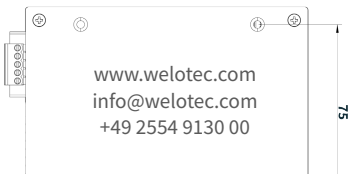
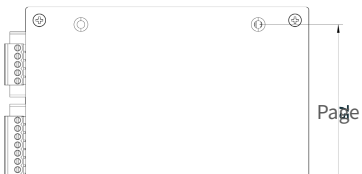
5 System Information

Attention: When opening the chassis make sure to slide the chassis top to the rear. Lifting the top up may shear the SIM Slot from the PCB.

Being a powerful, yet small fanless system, the Arrakis Pico Mk4 may reach very high surface temperatures in excess of 60°C/140°F with risk of injury. Users should ensure sufficient protection against touching.

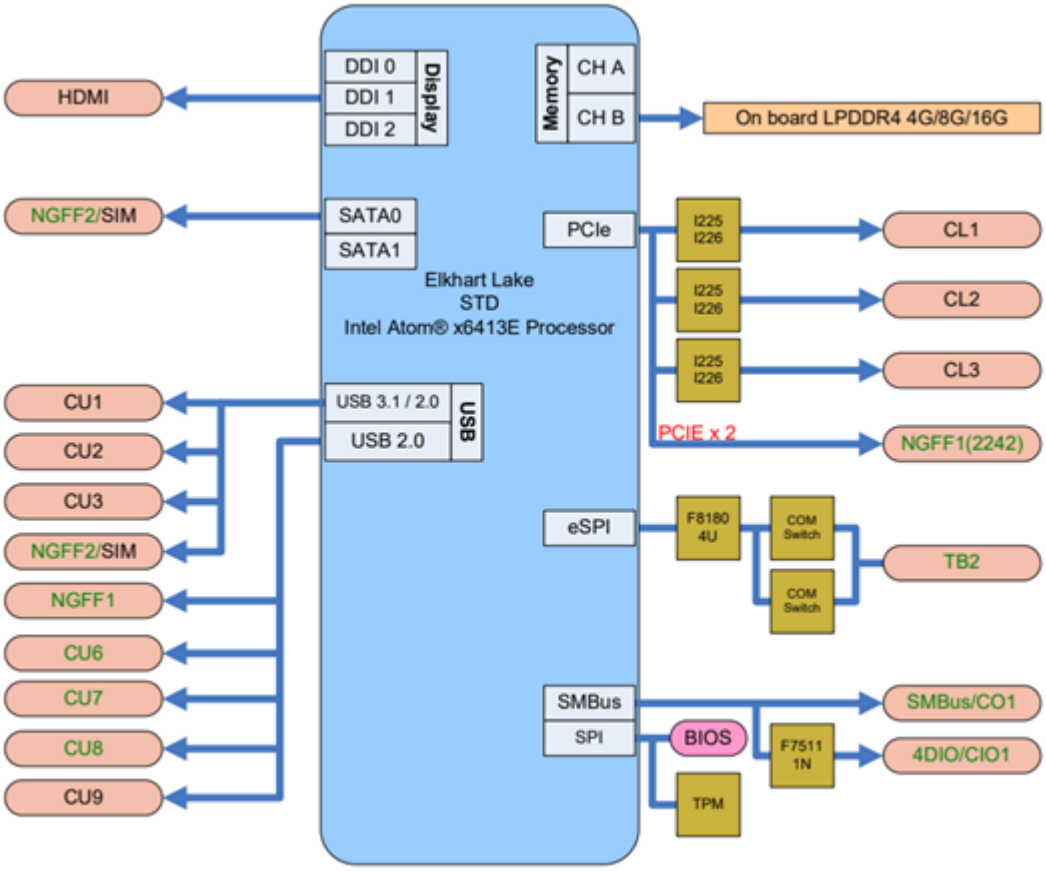
To allow for sufficient heat removal we recommend: 30mm distance on either side of the Arrakis Pico Mk4 when mounted on a DIN-Rail 100mm headroom above the Arrakis Pico Mk4 when mounted horizontally. The heatsink should be on top.

5.1 System Drawing

Arrakis Pico Mk4	Arrakis Pico Mk4 Light	Arrakis Pico Mk4 Headless
		
		
		
		
		

5.2 Mainboard Block Diagram

This block diagram describes the relationship among all interfaces and modules on the mainboard.



6 Power Supply

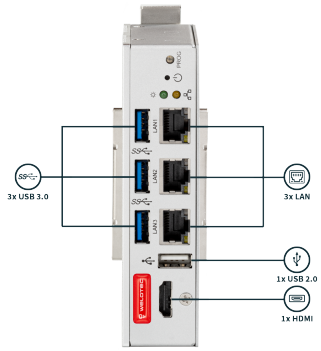
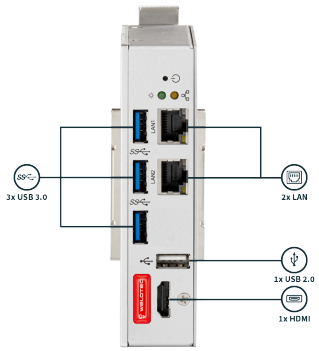
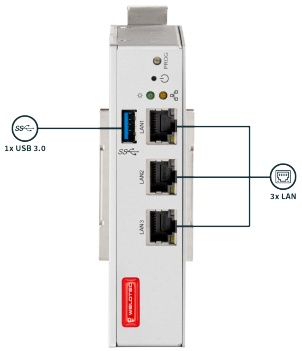
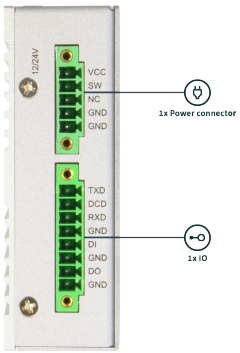
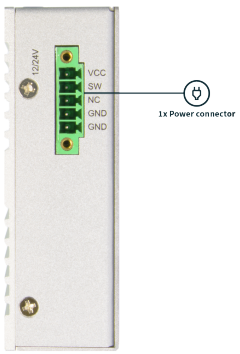
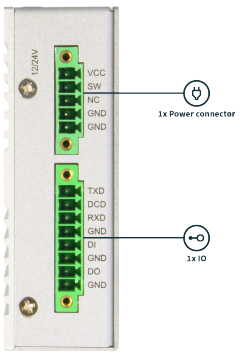
⚠ Please ensure no external voltage is applied to SW! This could cause damage.

Use the terminal block to connect the Arrakis Pico Mk4 to a 12-24V DC power source.

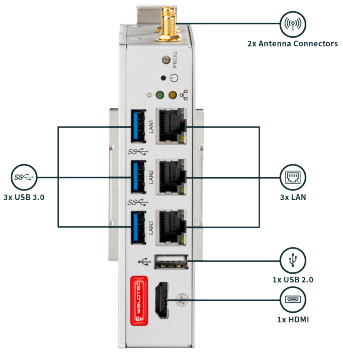
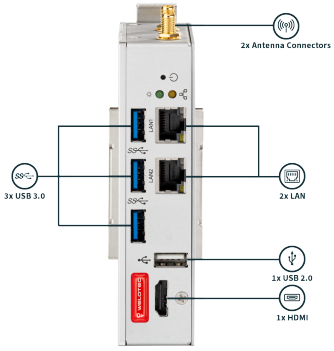
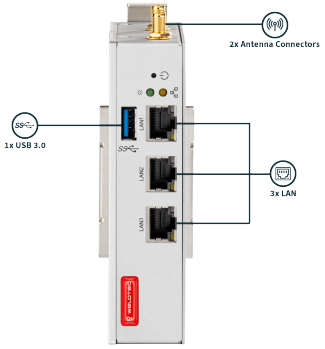
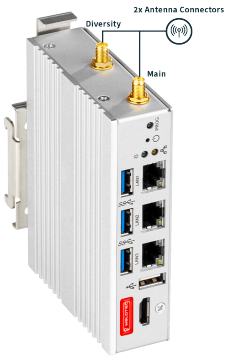
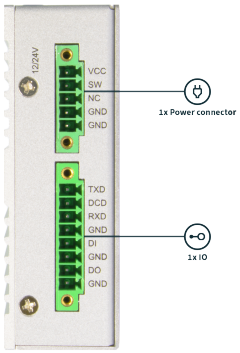
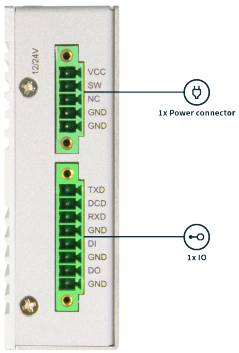
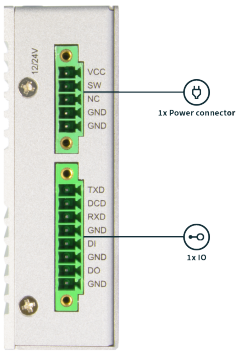
Pin	Description
Pin 0 – VCC	V+ 12-24V
Pin 1 – SW	External power switch
Pin 2 – NC	Not connected
Pin 3 – GND	Ground
Pin 4 – GND	Ground

7 Interfaces and Connections

7.1 Arrakis Pico Mk4 Series

Arrakis Pico Mk4 Standard	Arrakis Pico Mk4 Light	Arrakis Pico Mk4 Headless
		
		
		
		

7.2 Arrakis Pico Mk4 Series (with optional Radio Module)

Arrakis Pico Mk4 Standard	Arrakis Pico Mk4 Light	Arrakis Pico Mk4 Headless
		
		
		
		

8 Radio Modules (only relevant with optional LTE/WiFi Modules)

The Arrakis Pico Mk4 may contain one of the following RF Modules:

LTE:

Quectel EM05-G	Supported Bands
LTE	FDD B1/ B2/ B3/ B4/ B5/ B7/B8/ B12/B13/B14/ B18/ B19/B20/ B25/ B26/ B28/B66/B71TDD B38/ B39/ B40/ B41
WCDMA	B1/ B2/ B4/ B5/ B6/ B8/ B19

Sierra Wireless EM7590	Supported Bands
LTE	FDD B1/ B2/ B3/ B4 /B5 /B7 /B8 /B12 /B13 /B14 /B18 /B19 /B20 /B25 /B26 /B28 /B29 /B32 /B66 / B71 TDD B38 /B39 /B40 /B41 /B42 /B43 /B48
WCDMA	B1 /B2 /B4 /B8 /B19 /B5 /B6 /B9

WiFi:

SparkLAN WNFQ-262ACNI(BT) industrial Wifi with Qualcomm Atheros QCA6174A chipset. 802.11a/b/g/n/ac/ac wave 2

9 BIOS

9.1 Introduction:

The BIOS is a program stored in the Flash Memory on the motherboard, acting as a bridge between the hardware and the operating system. When you start the computer, the BIOS gains control and performs an auto-diagnostic test called POST (Power on Self Test) to check all necessary hardware. It detects all hardware devices and configures their parameters for synchronization. Once these tasks are completed, the BIOS hands control over to the operating system (OS).

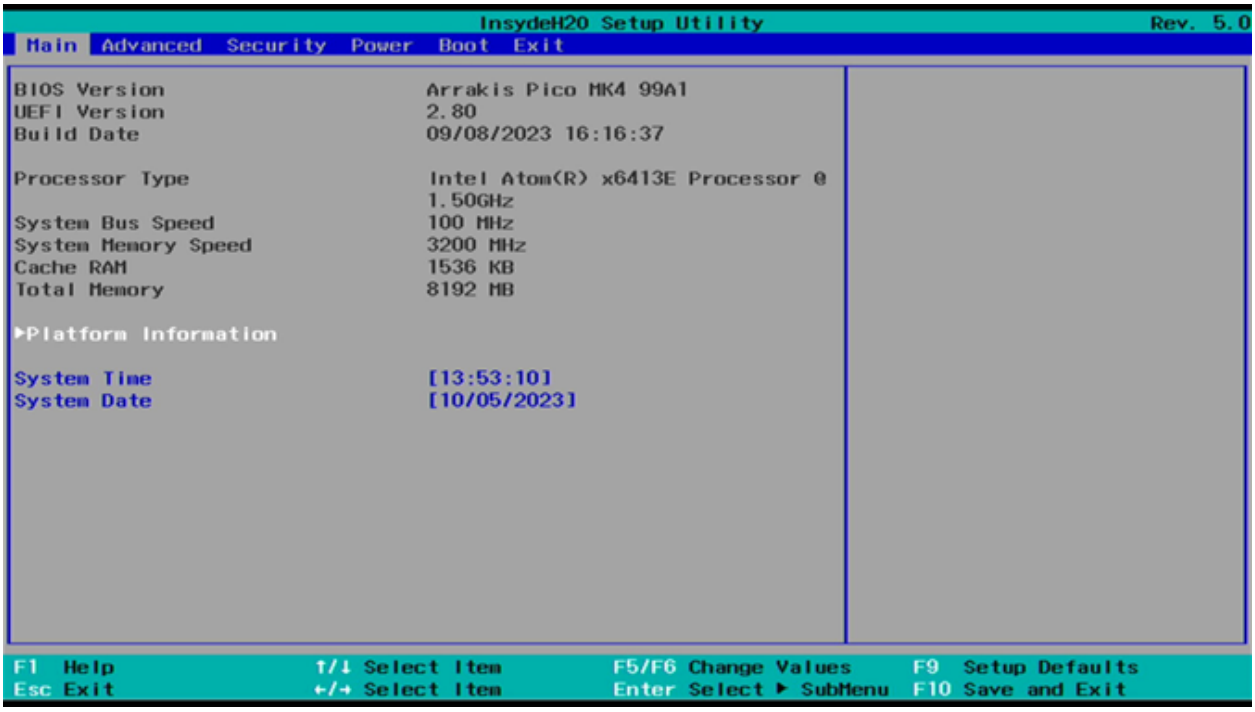
Since the BIOS is the sole channel for hardware and software communication, it is crucial for system stability and optimal performance. In the BIOS Setup main menu, you can see several options. These options will be explained in detail down below. First, let's look at the function keys you may use here:

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

9.2 Enter BIOS

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

9.3 BIOS menu and function keys



In the above BIOS Setup main menu, you can see several options. These options will be explained step by step. First, let's look at a brief description of the function keys you may use here:

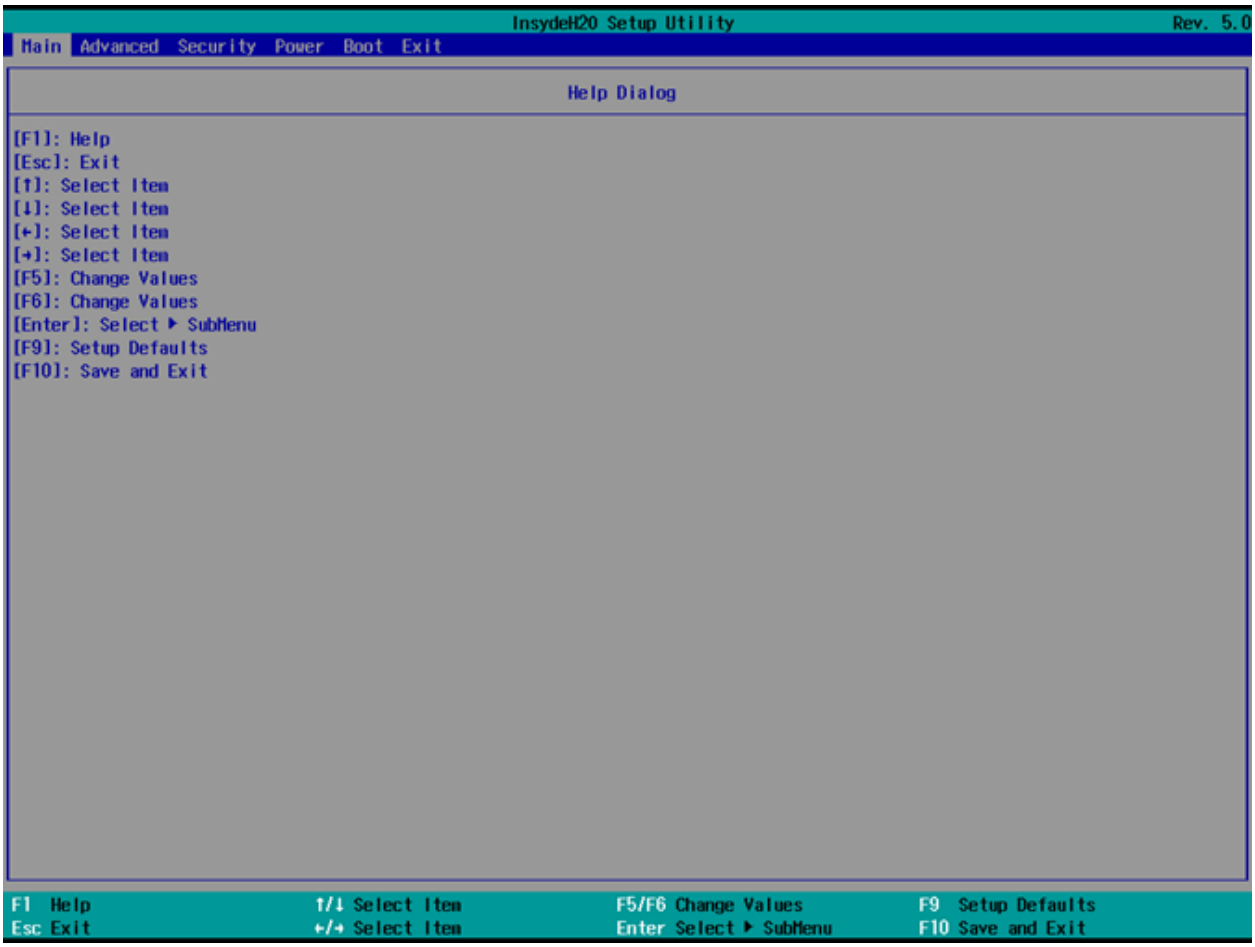
- Press ↔ (left, right) to select the screen.
- Press ↑↓ (up, down) to choose the option you want to confirm or modify.
- Press `Enter` to select.
- Press + or − to modify the BIOS parameters for the active option.
- `F1`: General help.
- `F2`: Previous value.
- `F3`: Optimized defaults.
- `F4`: Save & Reset.
- Press `Esc` to quit the BIOS Setup.

There are six menu bars on top of the BIOS screen:

- **Main**: To change system basic configuration
- **Advanced**: To change system advanced configuration
- **Security**: BIOS Password settings
- **Power**: ACPI and wake device settings
- **Boot**: To change system boot configuration
- **Exit**: Save settings, loading, and exit options

The selected menu bar is highlighted.

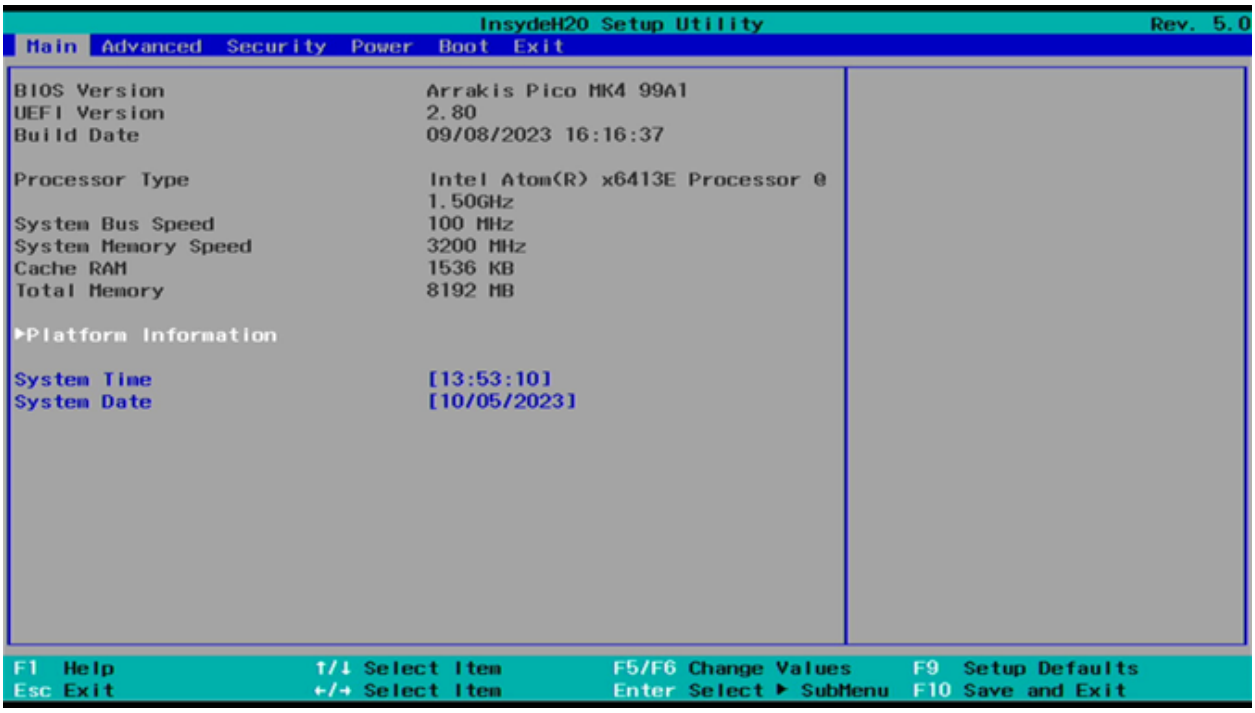
9.4 BIOS Help



Status Page Setup Menu/Option Page Setup Menu

Press F1 to open a help window that describes the appropriate keys to use and the possible selections for the highlighted item. To exit the Help Window, press Esc.

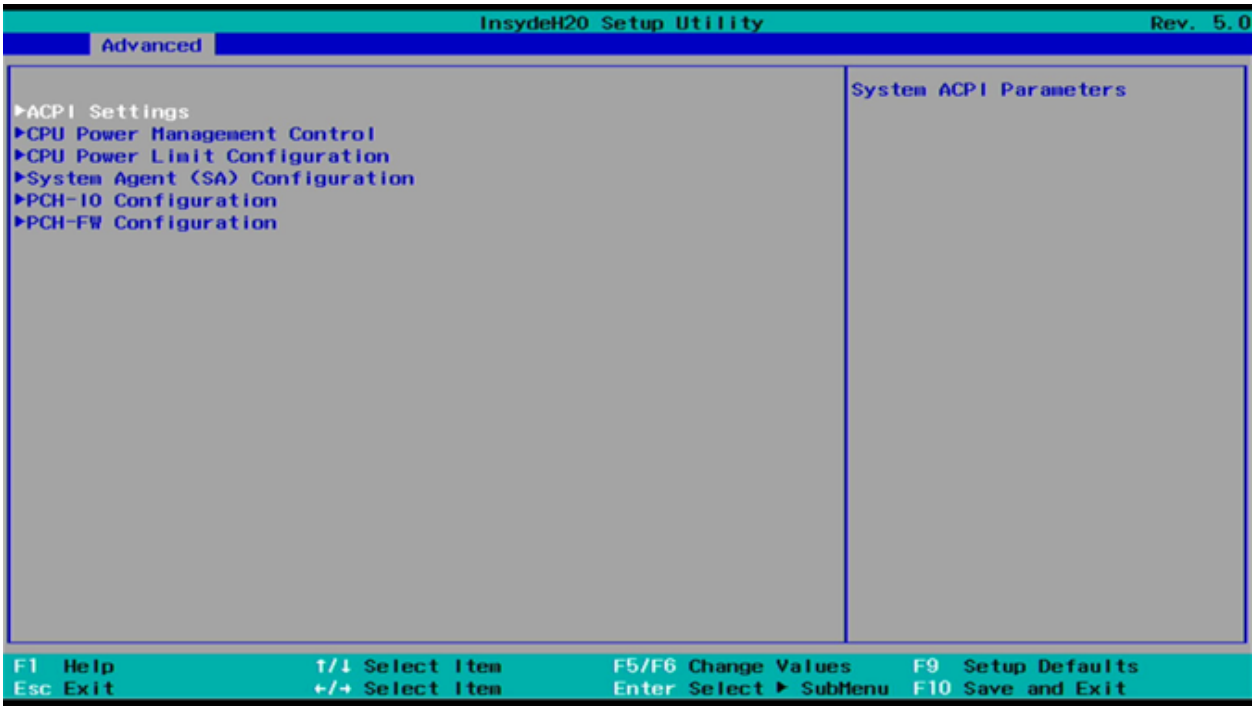
9.5 Main Menu



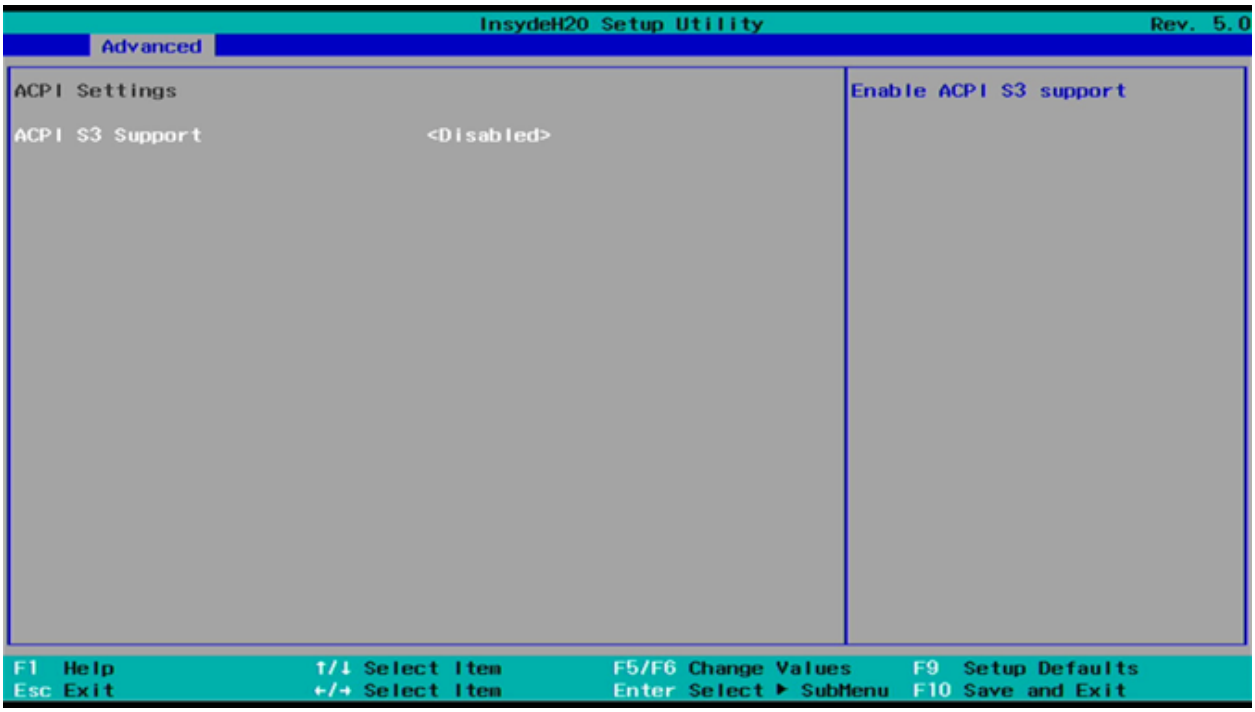
The Main menu screen includes some basic system information. Highlight the item and then use the + or − keys and numerical keyboard keys to select the value you want in each item.

- **System Date**: Set the Date. Use `Tab` to switch between date elements.
- **System Time**: Set the Time. Use `Tab` to switch between time elements.

9.6 Advanced

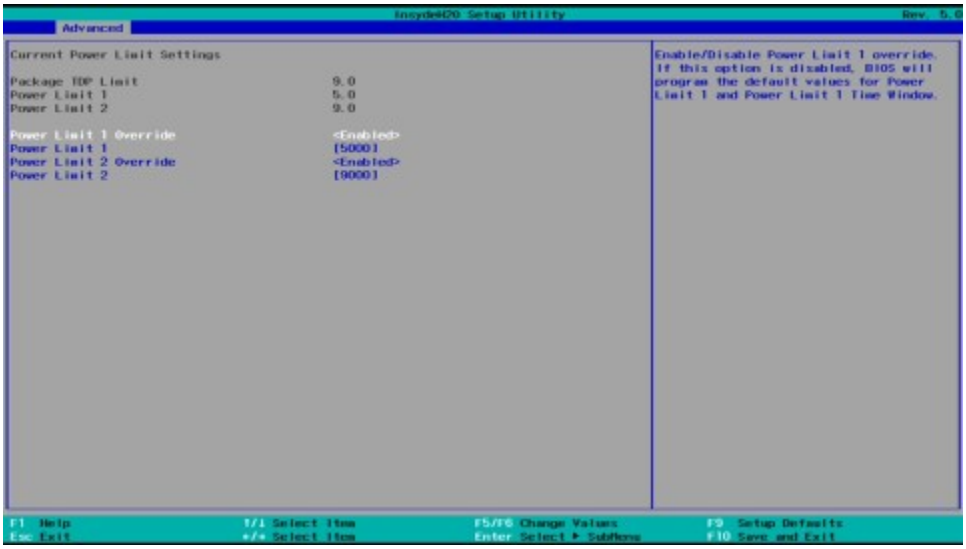


9.6.1 ACPS Settings



Toggle to enable/disable ACPI S3 support.

9.6.2 CPU Power Limit Configuration

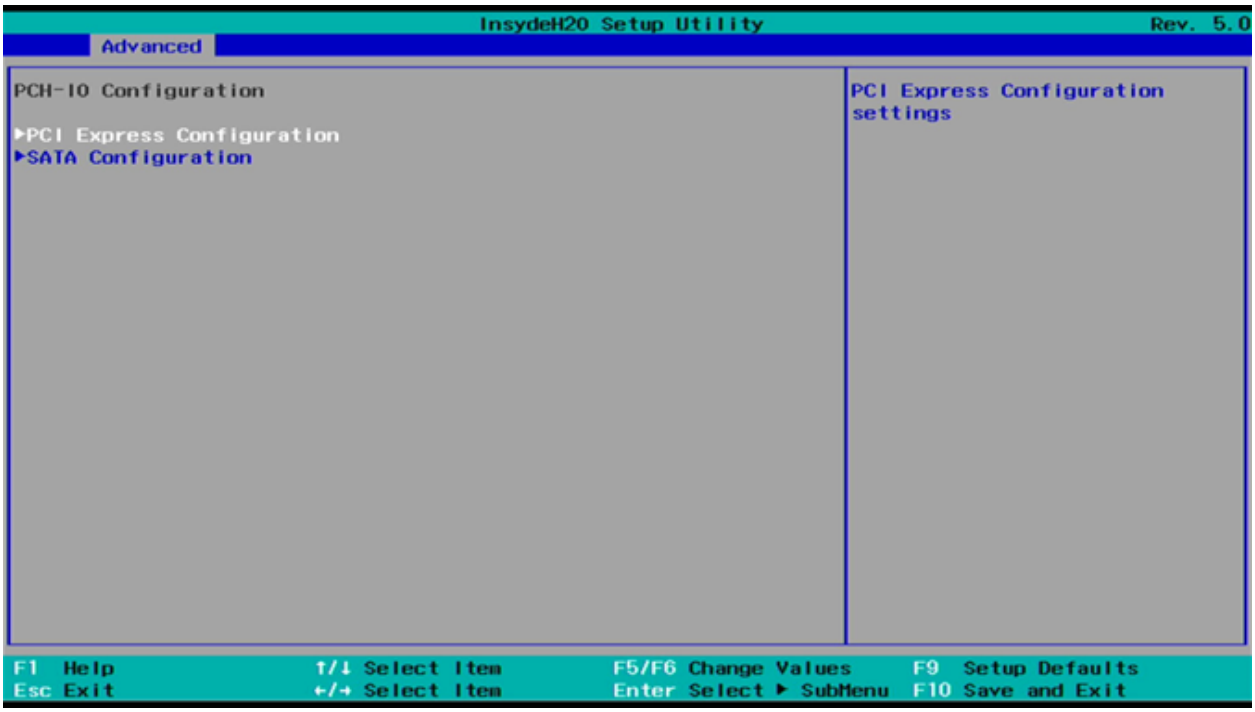


By default, both power limit overrides are enabled. Recommended values are:

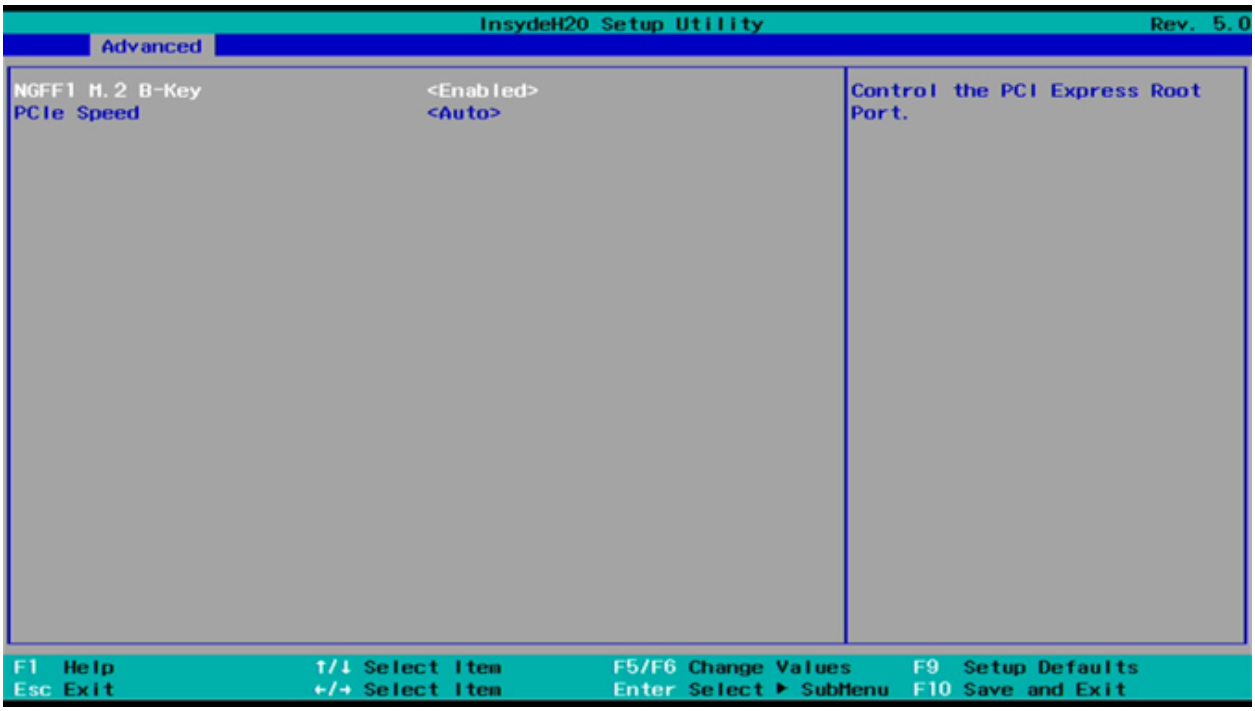
- **Power Limit 1**: 5000
- **Power Limit 2**: 9000

Increasing these values will result in higher power consumption, which will raise the case temperature and reduce the operating temperature range. Lowering these values will reduce device performance and may lead to undesired behavior.

9.6.3 PCH-IO Configuration



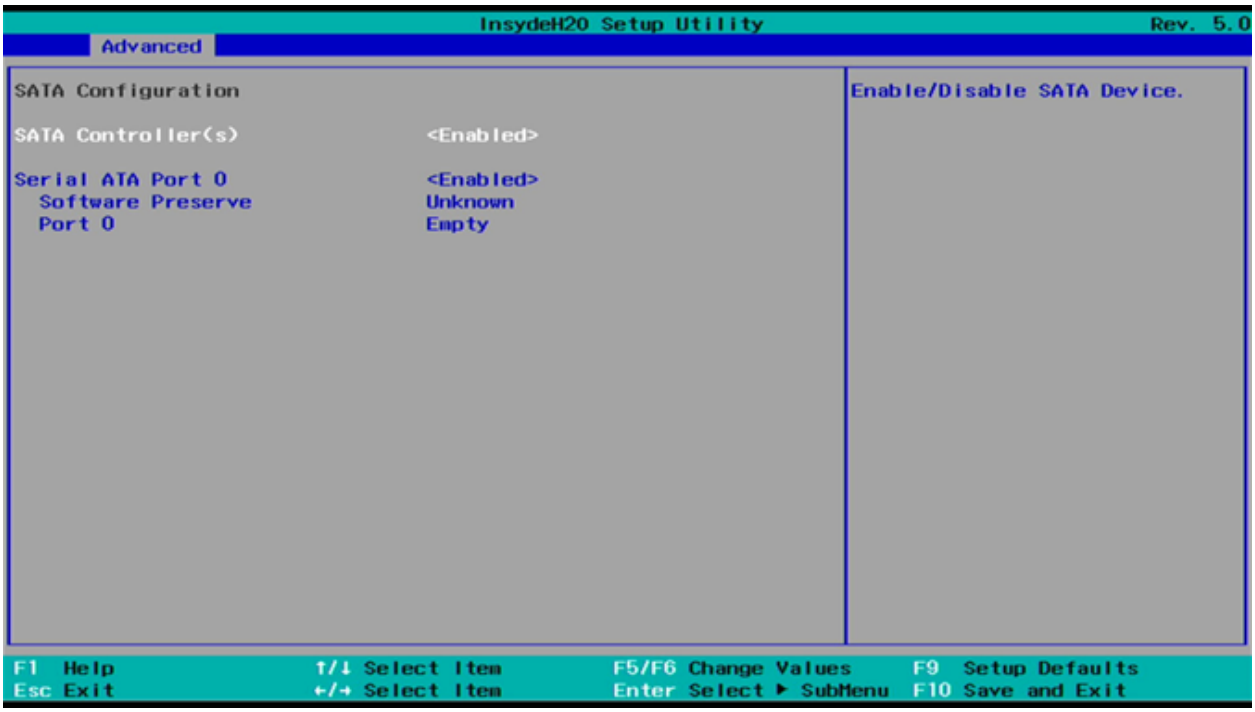
9.6.4 PCI-Express Configuration



There is usually no need for users to make any changes here. The default settings are:

- **MGFF1 M.2 B-Key**: Enabled
- **PCIe Speed**: Auto

9.6.5 SATA Configuration

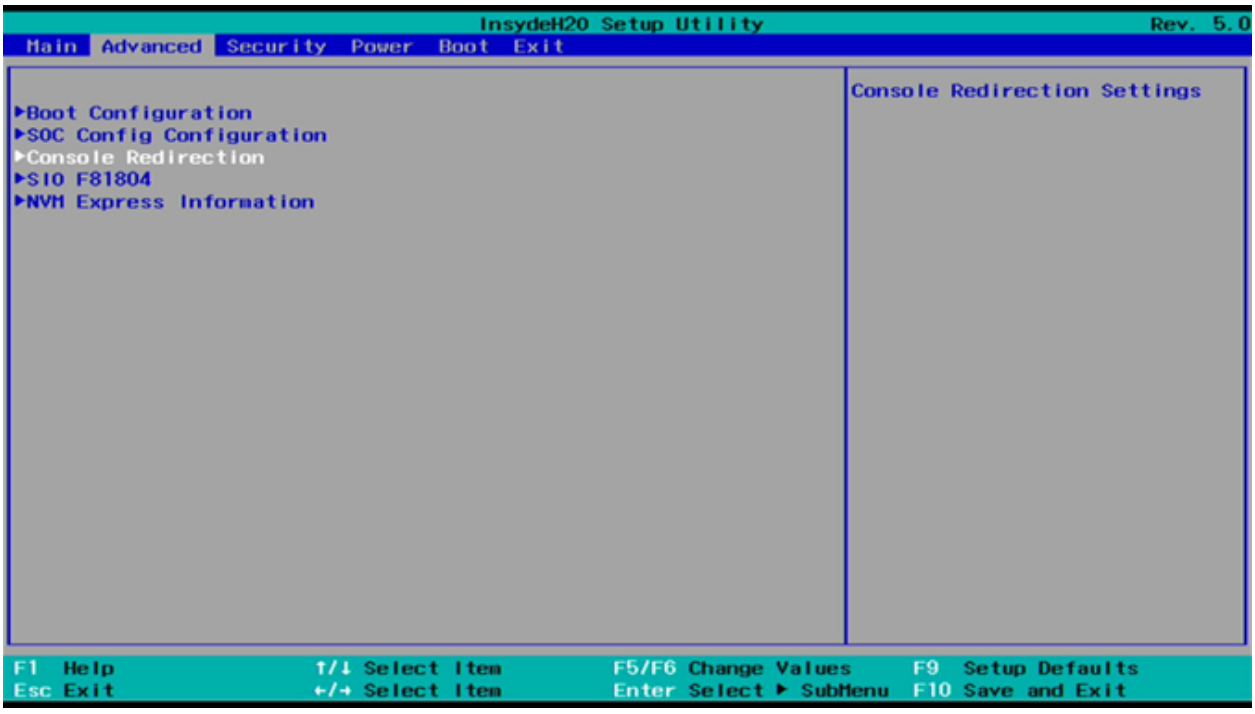


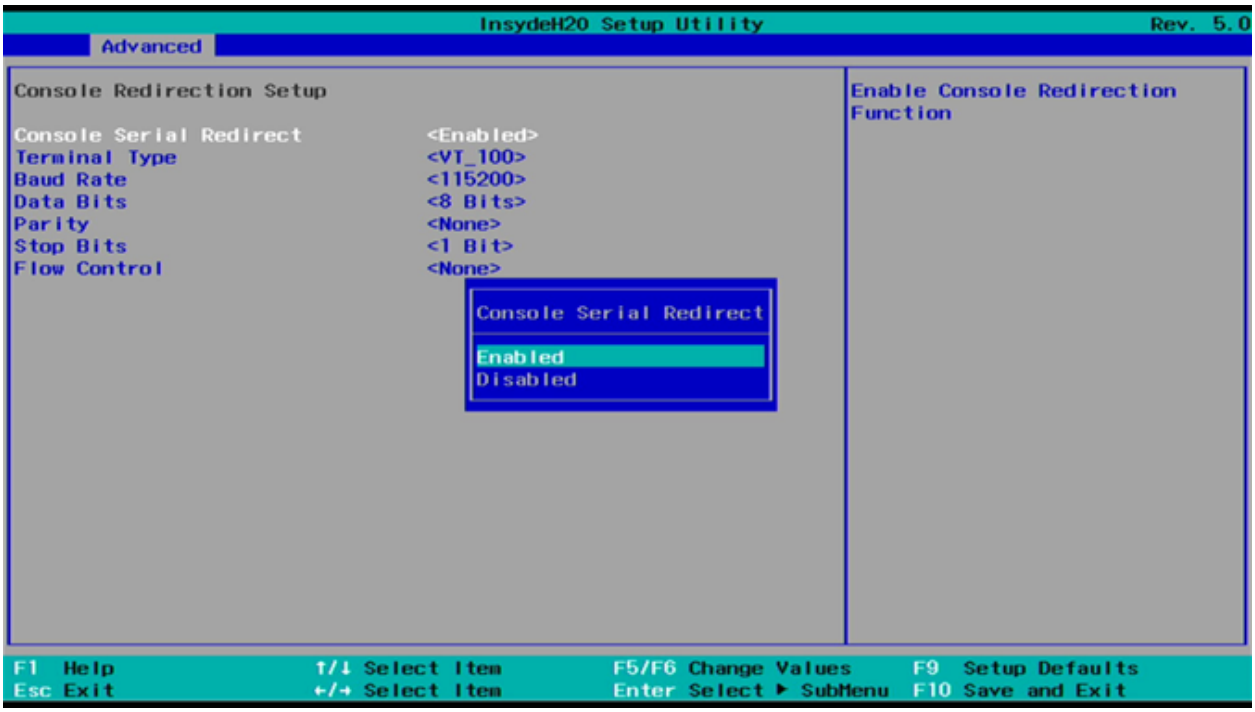
There is usually no need for users to make any changes here. The default settings are:

- **SATA Controller(s)**: Enabled
- **Serial ATA Port 0**: Enabled

Port 0 will show the type and capacity of any installed SATA drives.
Please note that the **Arrakis Pico Mk4** uses NVMe storage by default, so this field will often appear empty.

9.6.6 Console Redirect



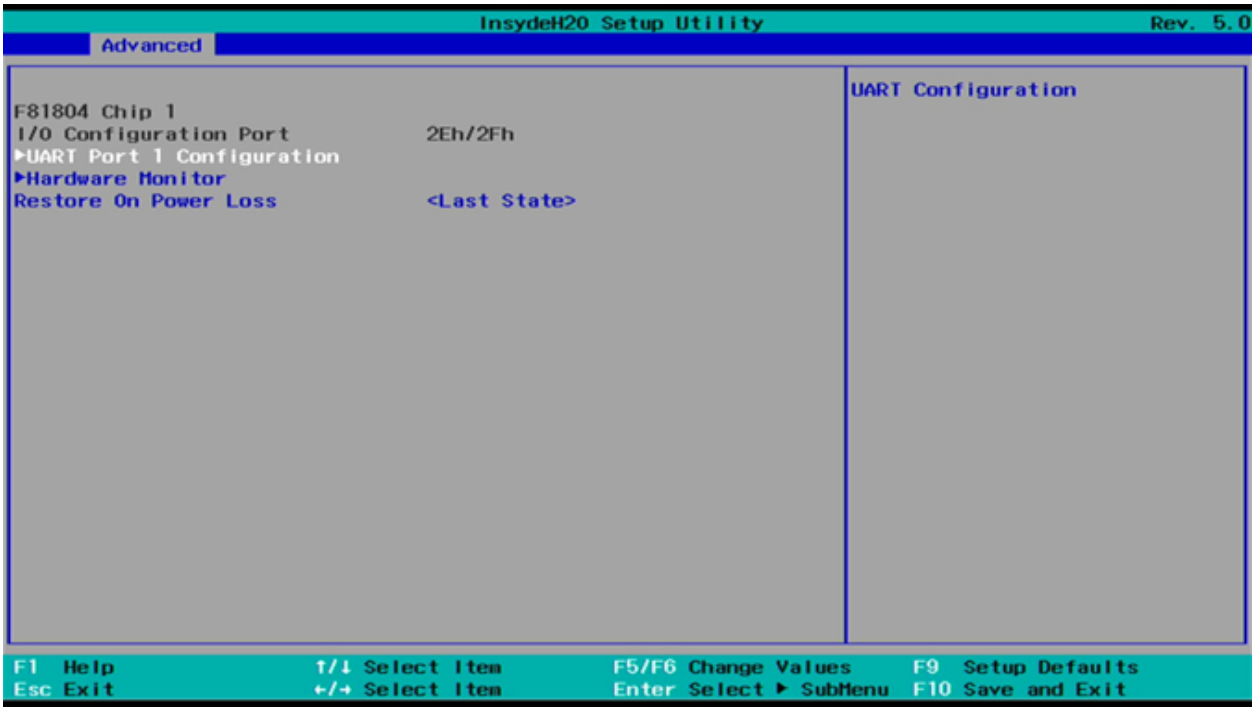


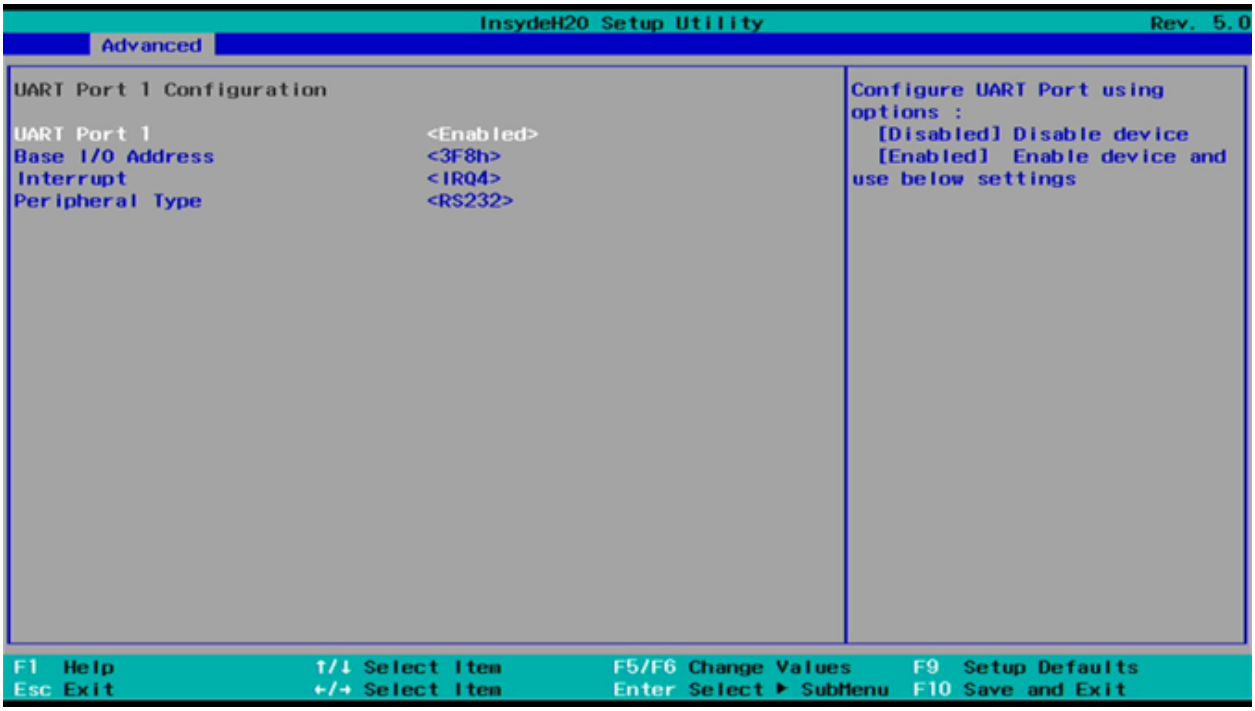
Toggle **Console Serial Redirect** to enable/disable the function.

- **Default**: Enabled

You can configure the terminal settings in this dialogue.

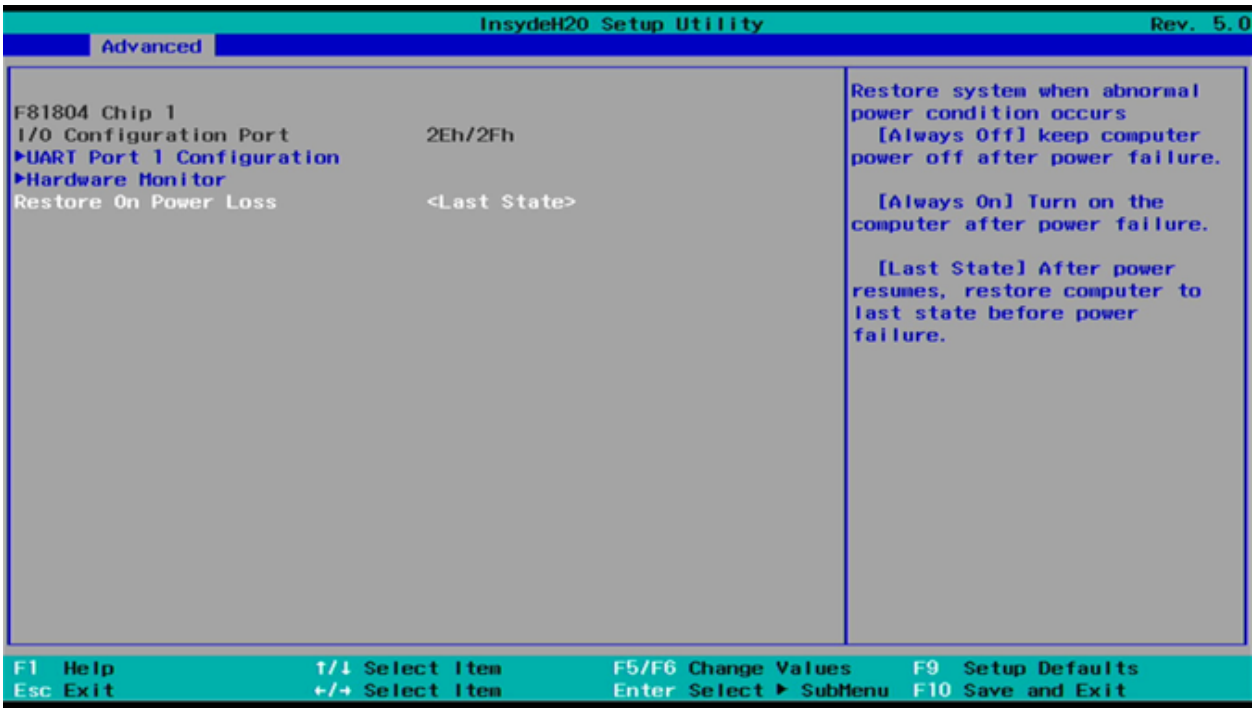
9.7 UART Port 1 Configuration

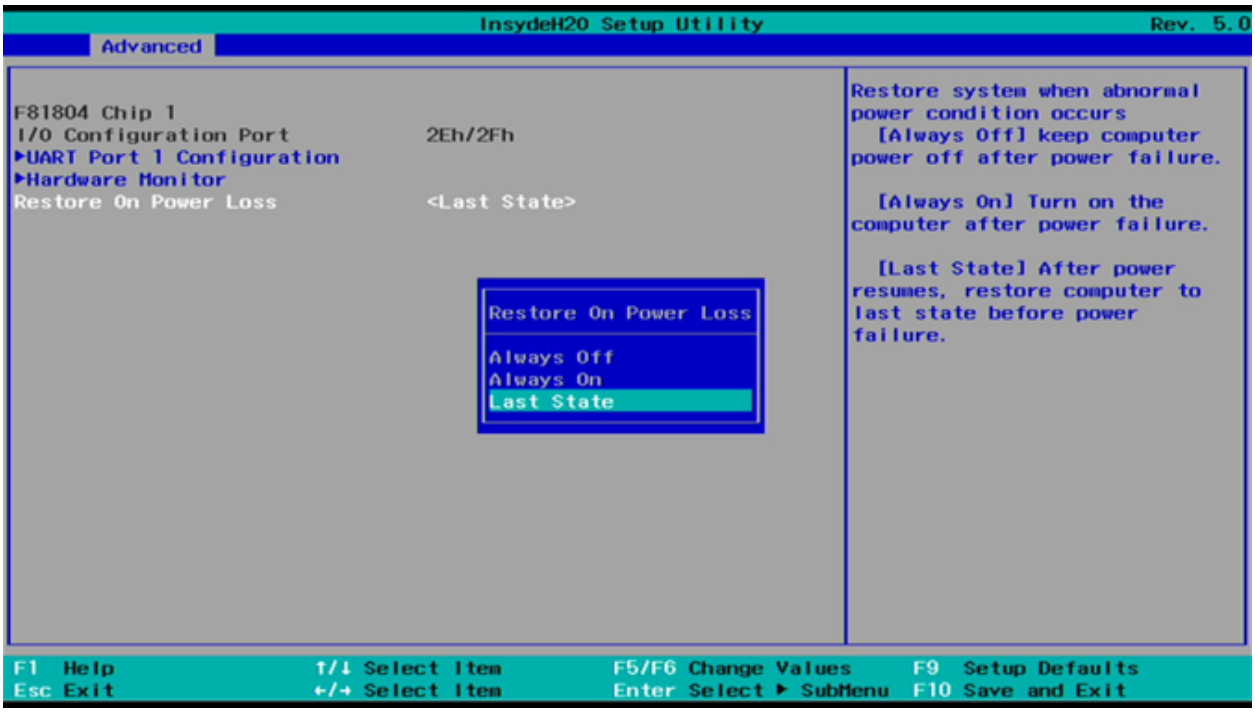




Toggle to enable/disable the serial interface.
Be advised that in **RS232 mode**, only **RX/TX lines** are supported.

9.8 Restore on Power Loss

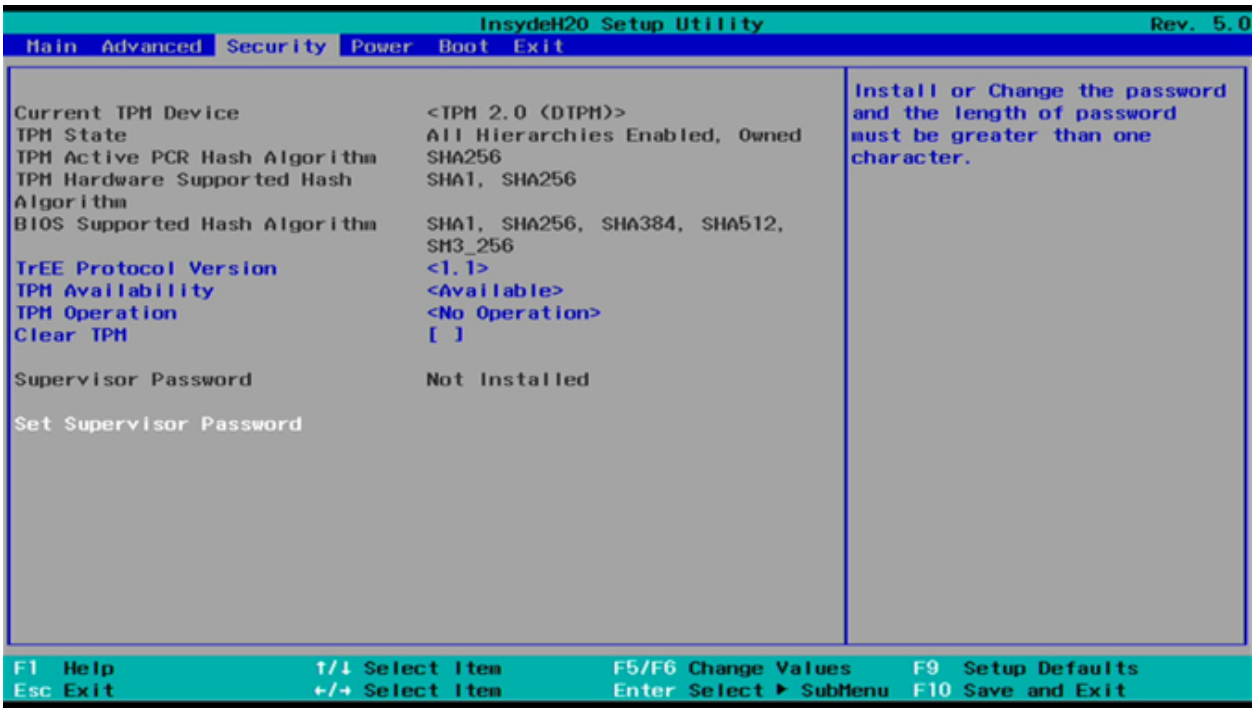




Toggle to configure the desired behavior after power loss. The available options are:

- **Last State (default)**: After power resumes, the Arrakis Pico Mk4 is restored to the state it was in before the power loss.
- **Always On**: The Arrakis Pico Mk4 powers on automatically when power is applied.
- **Always Off**: The Arrakis Pico Mk4 remains off after power is restored.

9.9 Security



- **Configure the TPM**
- **Clear TPM**

9.9.1 Supervisor Password

To set up a **Supervisor password**:

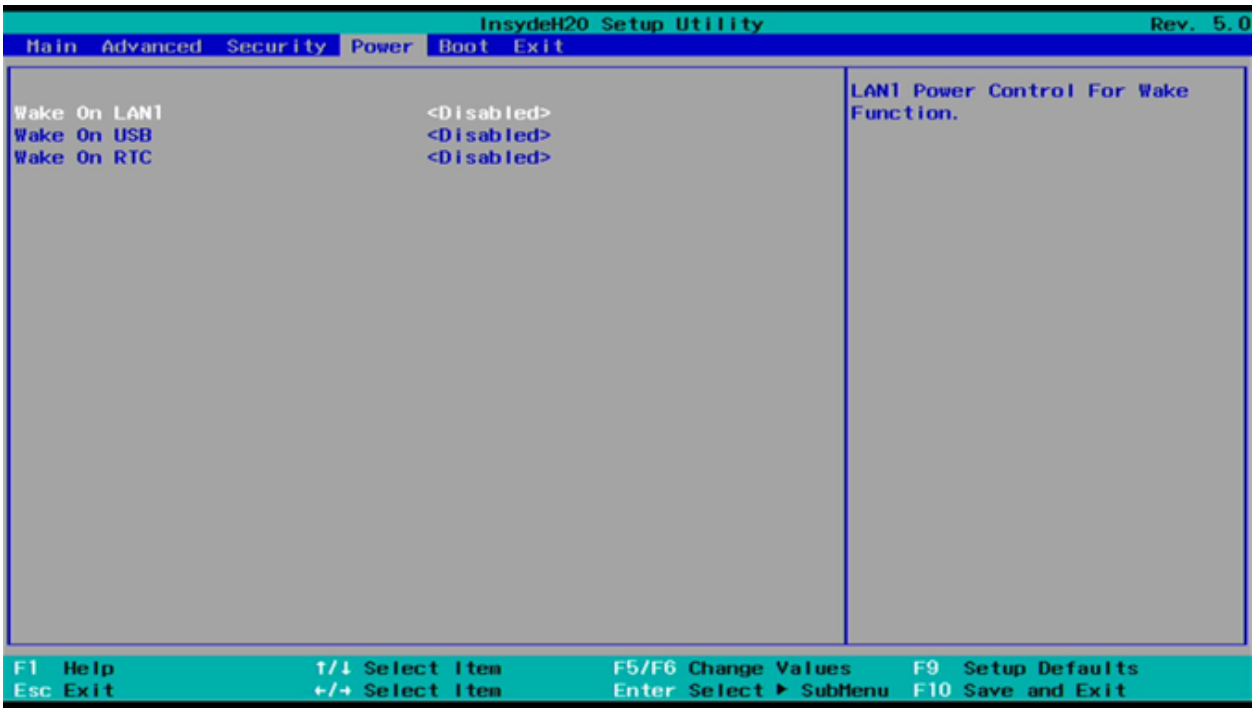
1. Select **Supervisor Password**. A “Create New Password” dialog will pop up.
2. Enter your desired password (must be between 3 and 10 characters).
3. Press `Enter` to submit.

9.9.2 Security Advisory

To enhance device security, we recommend the following steps in the BIOS:

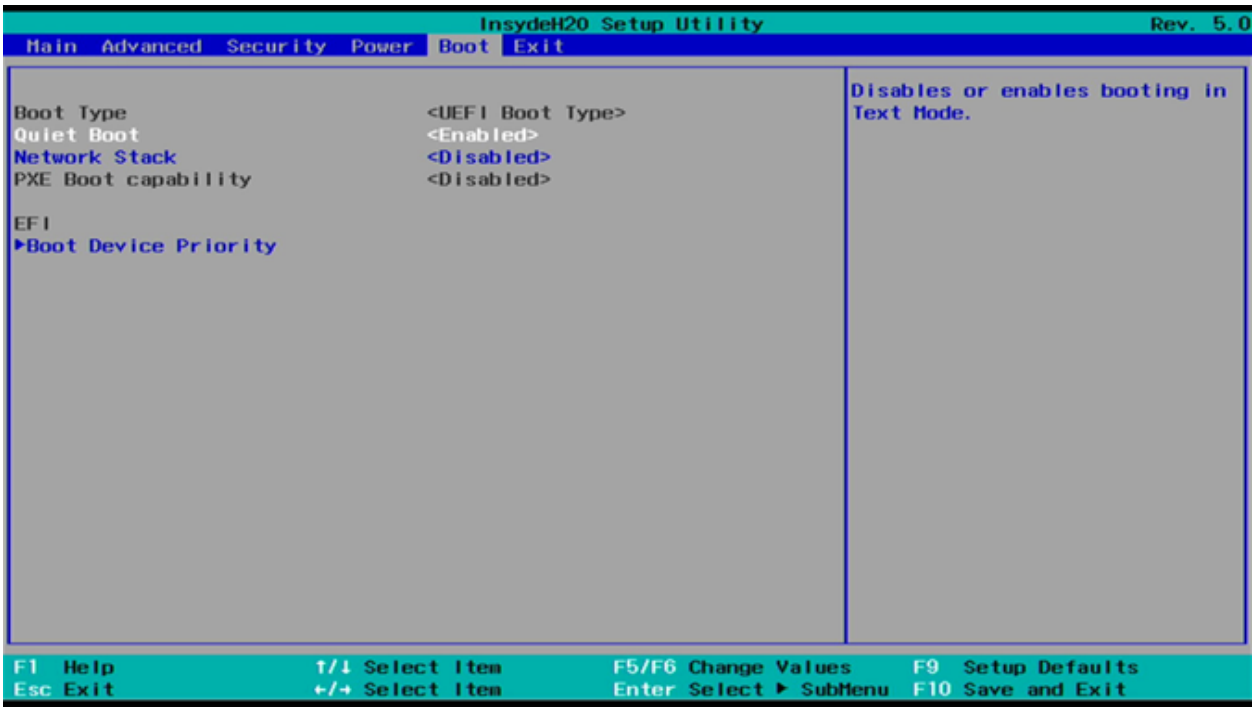
1. Create an **Admin Password** in the **BIOS -> Security** section.
2. Deactivate all unnecessary boot media in the **BIOS -> Boot** section.

9.10 Power



Toggle to enable desired **Wake-up Events**.

9.11 Boot



9.11.1 Boot Type

The **Arrakis Pico Mk4** is a UEFI boot-only system.

9.11.2 Quiet Boot

- **Options**:
  - **Enabled** (default)
  - **Disabled**

9.11.3 Network Stack

Enable this option if you need PXE functionality.

- **Default**: Disabled

9.11.4 PXE Boot Capability

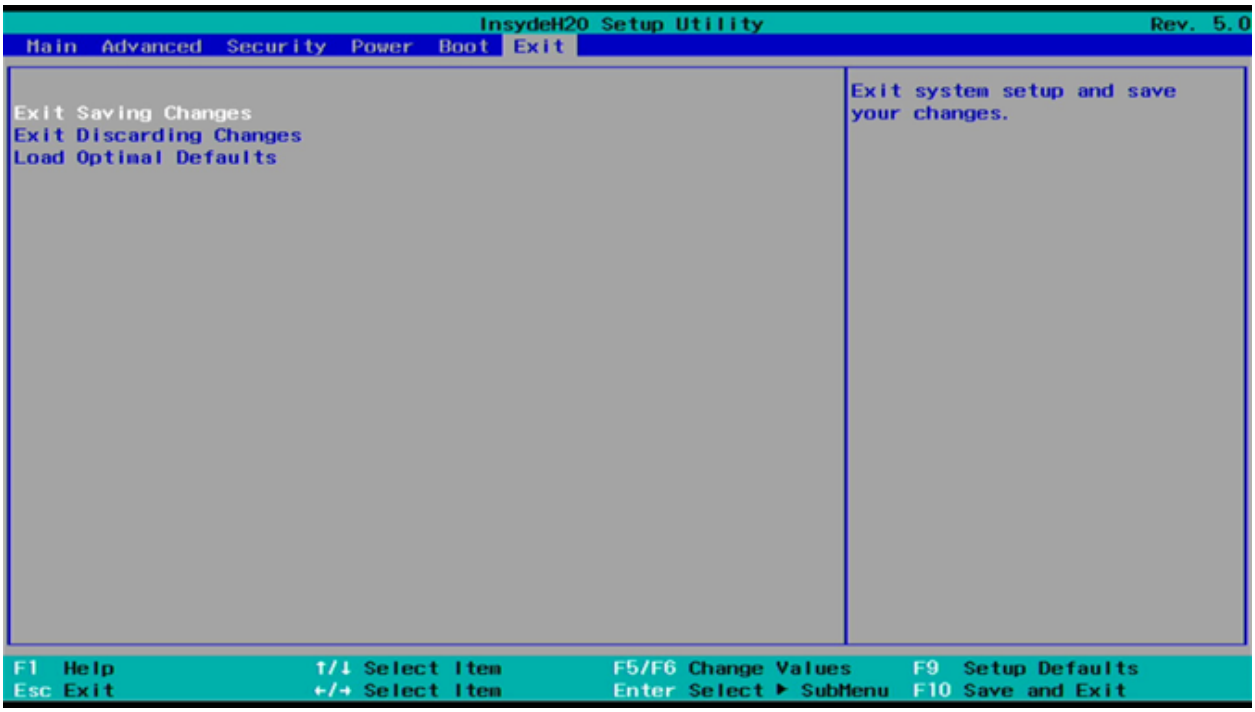
This item determines the protocol used during PXE boot:

- **Disabled** (default)
- **UEFI: IPv4**
- **UEFI: IPv6**

9.11.5 EFI

Determine which **EFI storage device** the Arrakis will boot from. This item will only appear if EFI is present on the storage media.

9.12 Exit



9.12.1 Exit Saving Changes

This option allows the user to reset the system after saving any changes made.

9.12.2 Exit Discarding Changes

This option allows the user to restart the system without saving any changes.

9.12.3 Load Optimal Defaults

Use this option to restore the optimal default settings for all setup options.

10 Driver Installation

The Arrakis Pico Mk4 is usually shipped with an Operating System preinstalled (recommended)

In case you have chose to purchase an Arrakis Mk4 without preinstalled operating system or need to reinstall, you can download all available System drivers from this address:

Welotec Download Service

To Install the Drivers, please execute the driver installation programs according to the on-screen instructions.

11 Appendix A: Power Consumption

Item	Specification
CPU	Intel Atom® x6413E Processor
RAM	LP-DDR4 8GB 3200MHz
Operating System	Windows 10 IoT 2021 LTSC
Test Program	3DMark06
mSATA	128GB

Results are for reference only!

Voltage	Power Off	Start up max.	Start up stable	Burn in Max	Shut Down
12V	0.07A	1.48A	0.63A	1.70A	1.31A
24V	0.04A	0.73A	0.35A	0.91A	0.65A

The Power Consumption depends on options and Software.

12 Appendix B: F75111N DIO & Watchdog Device

The Arrakis Pico MK4 includes optional DIO Ports. This Appendix provides an introduction to programming these ports.

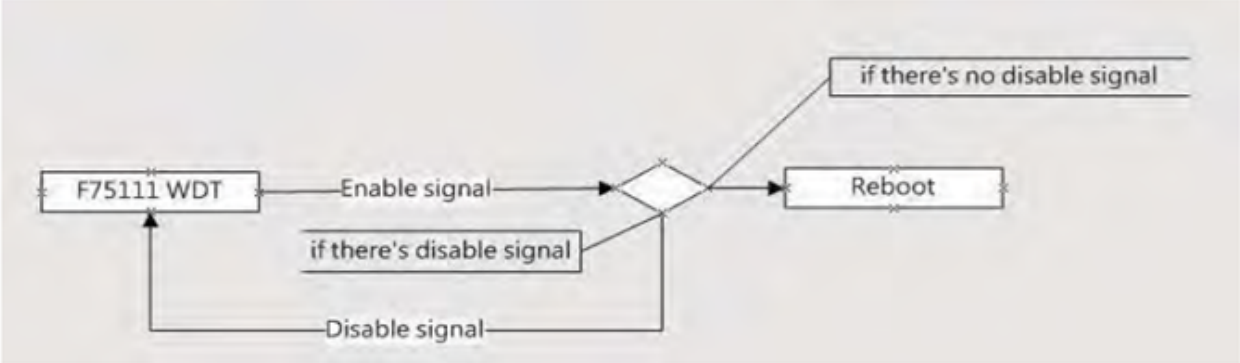
12.1 Watchdog Timer under DOS

The necessary software resources for programming the watchdog timer can be accessed from the Driver Download section:

- **Source file**: F75111_Dos_Src.rar
- **Binary file**: F75111_Dos_Bin.rar
- **USERNAME & PASSWORD**: sf

12.1.1 How to Use the Demo Application:

1. Boot into the MS-DOS Operating System.
2. Execute the `75WDT.EXE` binary file.
3. Input `1` to enable the WDT timer or `0` to disable it.
4. Input the number of seconds for the chip countdown and reset the computer.



12.1.2 Introduction:

How to use the Watchdog Timer Demo in different ways:

```
WriteI2CByte(I2CADDR, CONFIG, 0x03); // Set Watchdog Timer function
WriteI2CByte(I2CADDR, WDT_TIMER, timer); // Set Watchdog Timer range from 0-255
WriteI2CByte(I2CADDR, WDT_TIMER_CTL, 0x73); // Enable Watchdog Timer in second and pulse mode
```

Or:

```
WriteI2CByte(I2CADDR, WDT_TIMER_CTL, 0x00);
```

Or:

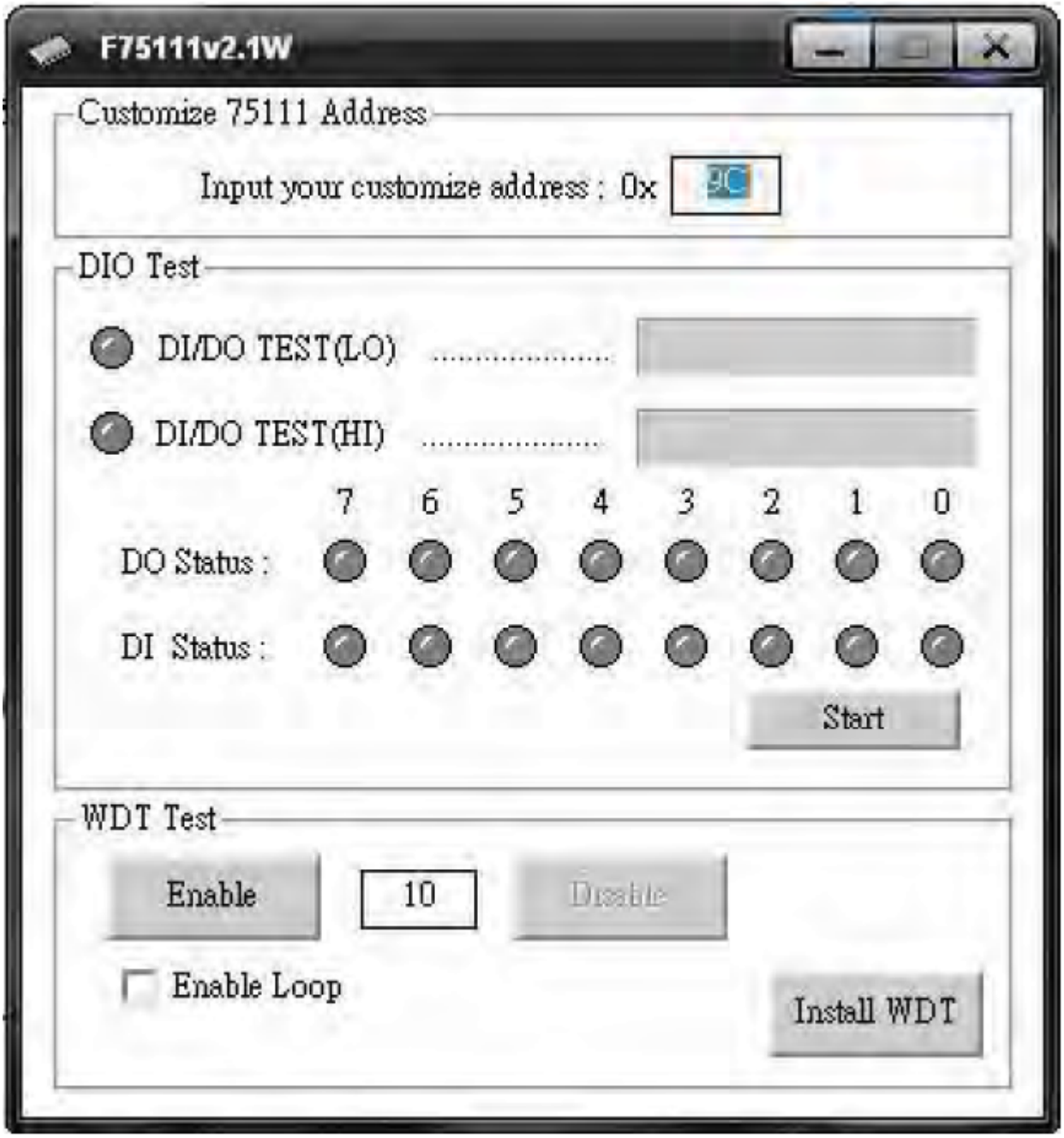
```
void pause(int time) {
    asm mov ah,0h; // Ah = 00 Read System Time Counter
    asm int 1ah;   // Read time from Time Counter and store it in DX register
    asm add dx, time;
    asm mov bx, dx;
label:
    asm int 1ah;
    asm cmp bx, dx;
    asm jne label;
}
```

12.2 Watchdog Timer and DIO under Windows:

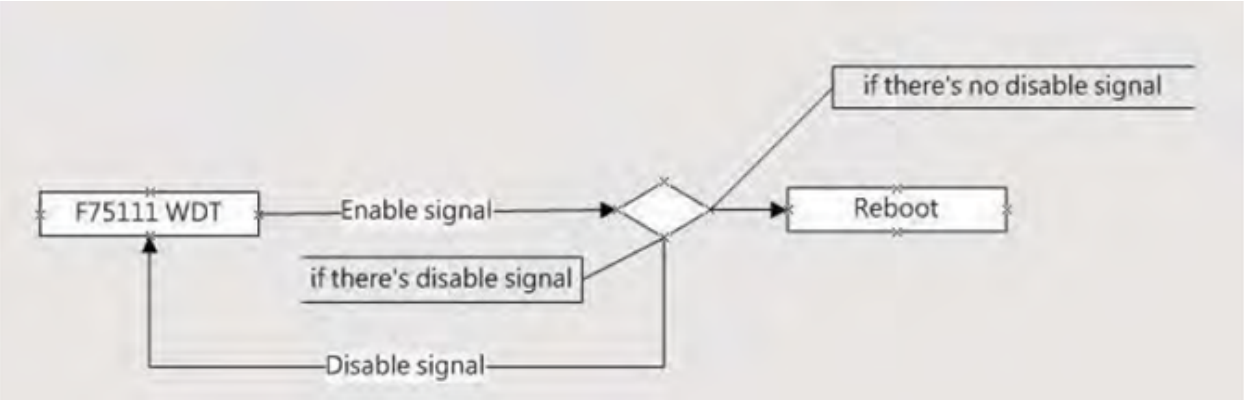
The necessary software resources for programming the watchdog timer can be accessed from the Driver Download section:

- **Source file**: F75111_DIOSrc.rar
- **Binary file**: F75111_DemoBin.rar
- **USERNAME & PASSWORD**: sf

12.2.1 How to Use the Demo Application:



1. Press the `Start` button to test the DIO function.
2. Press the `Enable` button to test the WDT function.
3. Press the `Disable` button to disable the WDT.
4. Check the `Enable Loop` box and press `Enable` to do a WDT loop test.
5. Press `Install WDT` to set the system to autorun this application when booting. Press it again to remove the application from booting. The icon will show when active.



The F75111 will send `F75111_SetWDTEnable(BYTE byteTimer)` including a `timer` parameter. If there's no disable signal (`F75111_SetWDTDisable()`) to stop it before the timer countdown reaches 0, the system will reboot. If a disable signal is received, it will reset the Enable WDT signal to prevent a reboot loop.

12.2.2 Introduction:

Initial Internal F75111 port address (0x9c) Define GPIO1X, GPIO2X, GPIO3X as input or output and enable the WDT function pin.

12.2.3 Set F75111 DI/DO (Sample Code Below to Get Input Value/Set Output Value):

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

12.2.4 Enable/Disable WDT:

- **Enable**: `F75111_SetWDTEnable(BYTE byteTimer)`
- **Disable**: `F75111_SetWDTDisable()`

12.2.5 Pulse Mode:

Example to set GP33, 32, 31, 30 output to 1mS low pulse signal:

```
{
    this->Write_Byte(F75111_INTERNAL_ADDR, GPIO3X_PULSE_CONTROL, 0x00); // Set low pulse output
    this->Write_Byte(F75111_INTERNAL_ADDR, GPIO3X_PULSE_WIDTH_CONTROL, 0x01); // Set pulse width␣
↪to 1mS
    this->Write_Byte(F75111_INTERNAL_ADDR, GPIO3X_CONTROL_MODE, 0x0F); // Set GP33, 32, 31, 30 to␣
↪output function
    this->Write_Byte(F75111_INTERNAL_ADDR, GPIO3X_Output_Data, 0x0F); // Set GP33, 32, 31, 30␣
↪output data
}
```

12.2.6 Initialize Internal F75111:

```
void F75111::InitInternalF75111() {
    this->Write_Byte(F75111_INTERNAL_ADDR, GPIO1X_CONTROL_MODE, 0x00); // Set GPIO1X to input␣
↪function
    this->Write_Byte(F75111_INTERNAL_ADDR, GPIO3X_CONTROL_MODE, 0x00); // Set GPIO3X to input␣
↪function
    this->Write_Byte(F75111_INTERNAL_ADDR, GPIO2X_CONTROL_MODE, 0xFF); // Set GPIO2X to output␣
↪function
    this->Write_Byte(F75111_INTERNAL_ADDR, F75111_CONFIGURATION, 0x03); // Enable WDT OUT function
}
```

12.2.7 Set Output Value:

```
void F75111::InterDigitalOutput(BYTE byteValue) {
    BYTE byteData = 0;
    byteData = (byteData & 0x01) ? byteValue + 0x01 : byteValue;
    byteData = (byteData & 0x02) ? byteValue + 0x02 : byteValue;
    byteData = (byteData & 0x04) ? byteValue + 0x04 : byteValue;
    byteData = (byteData & 0x80) ? byteValue + 0x08 : byteValue;
    byteData = (byteData & 0x40) ? byteValue + 0x10 : byteValue;
    byteData = (byteData & 0x20) ? byteValue + 0x20 : byteValue;
    byteData = (byteData & 0x10) ? byteValue + 0x40 : byteValue;
    byteData = (byteData & 0x08) ? byteValue + 0x80 : byteValue; // Get value bit by bit
    this->Write_Byte(F75111_INTERNAL_ADDR, GPIO2X_OUTPUT_DATA, byteData); // Write byteData value␣
↪via GPIO2X output pin
}
```

12.2.8 Get Input Value:

```
BYTE F75111::InterDigitalInput() {
    BYTE byteGPIO1X = 0;
    BYTE byteGPIO3X = 0;
    BYTE byteData = 0;
    this->Read_Byte(F75111_INTERNAL_ADDR, GPIO1X_INPUT_DATA, &byteGPIO1X); // Get value from GPIO1X
    this->Read_Byte(F75111_INTERNAL_ADDR, GPIO3X_INPUT_DATA, &byteGPIO3X); // Get value from GPIO3X
    byteGPIO1X = byteGPIO1X & 0xF0; // Mask unuseful value
    byteGPIO3X = byteGPIO3X & 0x0F; // Mask unuseful value
    byteData = (byteGPIO1X & 0x10) ? byteData + 0x01 : byteData;
    byteData = (byteGPIO1X & 0x80) ? byteData + 0x02 : byteData;
    byteData = (byteGPIO1X & 0x40) ? byteData + 0x04 : byteData;
    byteData = (byteGPIO3X & 0x01) ? byteData + 0x08 : byteData;
    byteData = (byteGPIO3X & 0x02) ? byteData + 0x10 : byteData;
    byteData = (byteGPIO3X & 0x04) ? byteData + 0x20 : byteData;
    byteData = (byteGPIO3X & 0x08) ? byteData + 0x40 : byteData;
    byteData = (byteGPIO1X & 0x20) ? byteData + 0x80 : byteData; // Get correct DI value from␣
↪GPIO1X & GPIO3X
    return byteData;
}
```

12.2.9 Enable Watchdog:

```
void F75111_SetWDTEnable(BYTE byteTimer) {
    WriteByte(F75111_INTERNAL_ADDR, WDT_TIMER_RANGE, byteTimer); // Set Watchdog range and timer
    WriteByte(F75111_INTERNAL_ADDR, WDT_CONFIGURATION, WDT_TIMEOUT_FLAG | WDT_ENABLE | WDT_PULSE |␣
↪WDT_PSWIDTH_100MS);
    // Enable Watchdog, Setting Watchdog configure
}
```

12.2.10 Disable Watchdog:

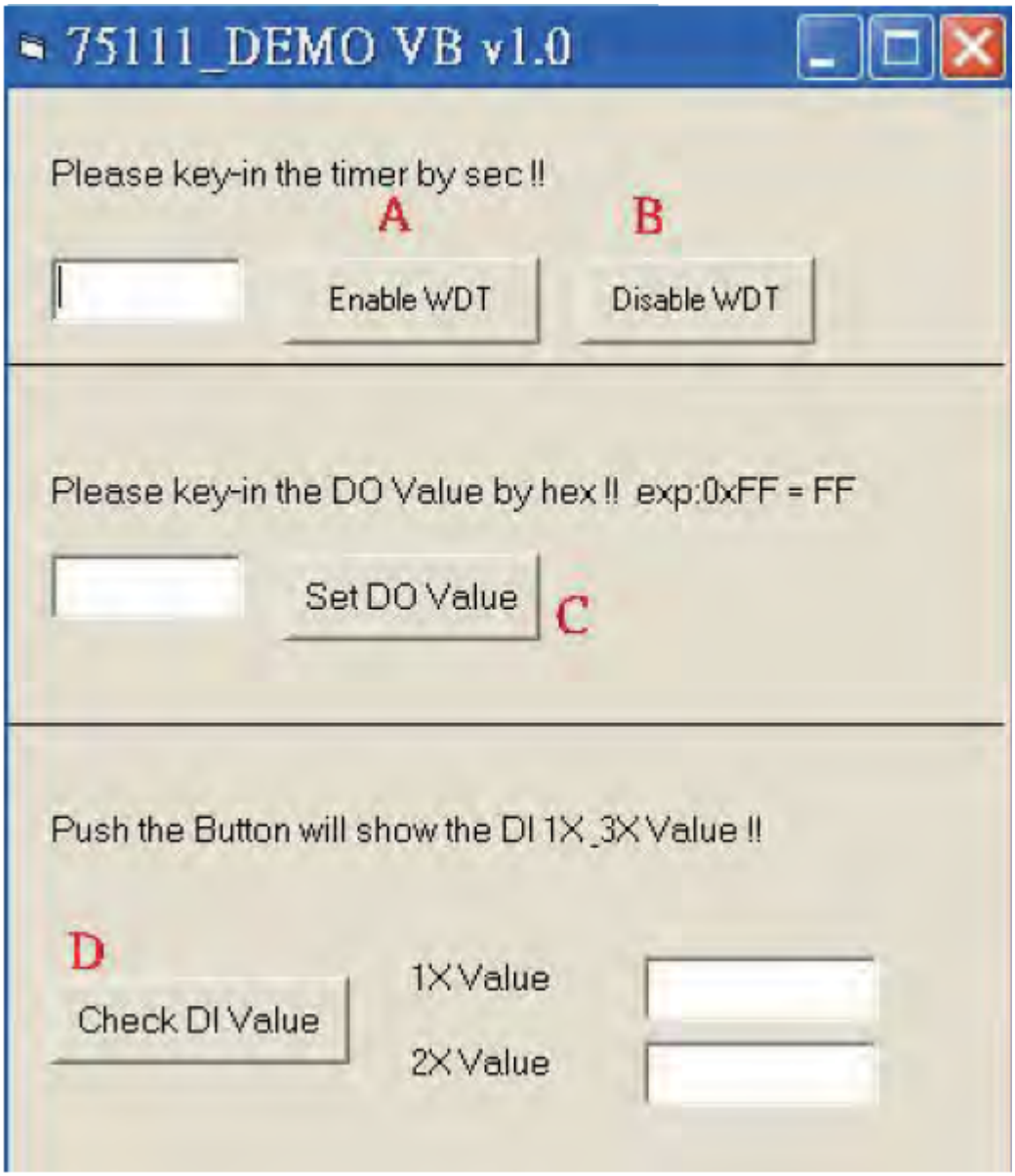
```
void F75111_SetWDTDisable() {
    WriteByte(F75111_INTERNAL_ADDR, WDT_CONFIGURATION, 0x00); // Disable Watchdog
}
```

12.3 IO Device: F75111 VB6 under Windows

The necessary software resources for programming the watchdog timer can be accessed from the Driver Download section:

- **Source file**: 75111_VB_v10.rar
- **Binary file**: 75111_VB_Src.rar111_DemoBin.rar
- **USERNAME & PASSWORD**: sf

12.3.1 How to Use the Demo Application



- **A** Function - Enable WDT timer: Enter the value in seconds, then the system will reboot after the specified time.
- **B** Function - Disable WDT timer: Press the button to clear the WDT timer value.
- **C** Function - Set DO Value: Enter the DO value in hex, then press the button.
- **D** Function - Check DI Value: The two text boxes on the right display DI 1X & 2X values when you press the button.

12.3.2 SDK Function Introduction

Function EnableWDT:

```
Function EnableWDT(timer As Integer)
    Call WriteI2CByte(&H3, &H3)
    Call WriteI2CByte(&H37, timer)
    Call WriteI2CByte(&H36, &H73)
End Function
```

Function DisableWDT:

```
Function DisableWDT()
    Call WriteI2CByte(&H36, &H0)
End Function
```

Function SetDOValue:

```
Function SetDOValue(dovalue As Integer)
    Call WriteI2CByte(&H23, &H0)
    Call WriteI2CByte(&H20, &HFF)
    Call WriteI2CByte(&H2B, &HFF)
    Call WriteI2CByte(&H21, dovalue)
End Function
```

Function CheckDIValue:

```
Function CheckDIValue()
    Dim GPIO1X As Integer
    Dim GPIO3X As Integer
    Dim DI1Xhex As String
    Dim DI3Xhex As String

    Call ReadI2CByte(&H12, GPIO1X)
    Call ReadI2CByte(&H42, GPIO3X)

    DI1Xhex = Hex(GPIO1X)
    DI3Xhex = Hex(GPIO3X)

    Text3.Text = "0x" + DI1Xhex
    Text4.Text = "0x" + DI3Xhex
End Function
```

12.4 Watchdog Timer and DIO under Linux

The necessary software resources for programming the watchdog timer can be accessed from the Driver Download section:

- **Source file**: F75111v2.0L.tar.gz
- **Binary file**: F75111v2.0LBin.tar.gz
- **USERNAME & PASSWORD**: sf

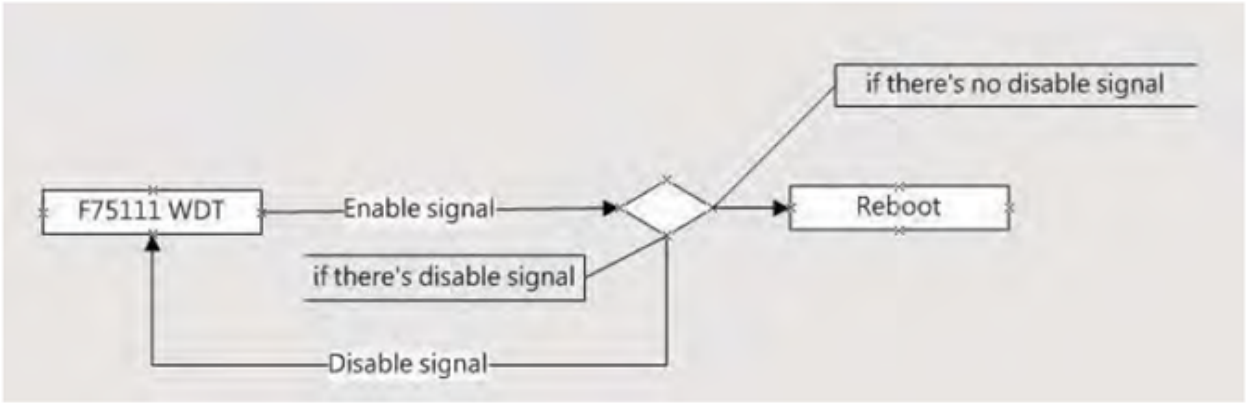
12.4.1 How to Compile the Source Code

1. **Compile with Code::Blocks:**
   - Download and install Code::Blocks with the command `apt-get install codeblocks`.
   - Open the existing project (F75111.cbp) in Code::Blocks and click the compile button.
   - Add the option `pkg-config --libs gtk+-2.0 gthread-2.0` in "Project -> Build Option -> Linker Setting -> Other linker option".
2. **Compile with "make":**
   - Navigate to the F75111 directory: `cd F75111`.
   - Compile the source: `make`.
   - Execute the binary file: `src/f75111`.

12.4.2 How to Use the Demo Application

1. Press the "Start" button to test the DIO function.
2. Press the "Enable" button to test the WDT function.
3. Press the "Disable" button to disable the WDT.
4. Check the "Enable Loop" box and press "Enable" to do a WDT loop test.

5. Press “Install” to set the system to autorun this application at boot, press “Uninstall” to remove it from boot.
6. If WDT is enabled, the system icon will blink.



The F75111 will send `F75111_SetWDTEnable(BYTE byteTimer)` with a parameter `timer`. If no disable signal (`F75111_SetWDTDisable()`) is received before the timer counts down to 0, the system will reboot. If a disable signal is received, it will resend the enable WDT signal to prevent a reboot loop.

12.4.3 Introduction

IO Function in the file SMBus.c:

```
void SMBusIoWrite(BYTE byteOffset, BYTE byteData) {
    outb(byteData, m_SMBusMapIoAddr + byteOffset);
}

BYTE SMBusIoRead(BYTE byteOffset) {
    DWORD dwAddrVal;
    dwAddrVal = inb(m_SMBusMapIoAddr + byteOffset);
    return (BYTE)(dwAddrVal & 0x0FF);
}
```

Init Internal F75111:

```
void F75111::InitInternalF75111() {
    this->Write_Byte(F75111_INTERNAL_ADDR, GPIO1X_CONTROL_MODE, 0x00); // Set GPIO1X to Input␣
↪function
    this->Write_Byte(F75111_INTERNAL_ADDR, GPIO3X_CONTROL_MODE, 0x00); // Set GPIO3X to Input␣
↪function
    this->Write_Byte(F75111_INTERNAL_ADDR, GPIO2X_CONTROL_MODE, 0xFF); // Set GPIO2X to Output␣
↪function
    this->Write_Byte(F75111_INTERNAL_ADDR, F75111_CONFIGURATION, 0x03); // Enable WDT OUT function
}
```

Set Output Value:

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

(continues on next page)

(continued from previous page)

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

Get Input Value:

```
BYTE F75111::InterDigitalInput() {
    BYTE byteGPIO1X = 0;
    BYTE byteGPIO3X = 0;
    BYTE byteData = 0;
    this->Read_Byte(F75111_INTERNAL_ADDR, GPIO1X_INPUT_DATA, &byteGPIO1X); // Get value from GPIO1X
    this->Read_Byte(F75111_INTERNAL_ADDR, GPIO3X_INPUT_DATA, &byteGPIO3X); // Get value from GPIO3X
    byteGPIO1X = byteGPIO1X & 0xF0; // Mask unnecessary value
    byteGPIO3X = byteGPIO3X & 0x0F; // Mask unnecessary value
    byteData = (byteGPIO1X & 0x10) ? byteData + 0x01 : byteData;
    byteData = (byteGPIO1X & 0x80) ? byteData + 0x02 : byteData;
    byteData = (byteGPIO1X & 0x40) ? byteData + 0x04 : byteData;
    byteData = (byteGPIO3X & 0x01) ? byteData + 0x08 : byteData;
    byteData = (byteGPIO3X & 0x02) ? byteData + 0x10 : byteData;
    byteData = (byteGPIO3X & 0x04) ? byteData + 0x20 : byteData;
    byteData = (byteGPIO3X & 0x08) ? byteData + 0x40 : byteData;
    byteData = (byteGPIO1X & 0x20) ? byteData + 0x80 : byteData; // Get correct DI value from␣
↪GPIO1X & GPIO3X
    return byteData;
}
```

Enable WatchDog:

```
void F75111_SetWDTEnable(BYTE byteTimer) {
    WriteByte(F75111_INTERNAL_ADDR, WDT_TIMER_RANGE, byteTimer); // Set WatchDog range and timer
    WriteByte(F75111_INTERNAL_ADDR, WDT_CONFIGURATION, WDT_TIMEOUT_FLAG | WDT_ENABLE | WDT_PULSE |␣
↪WDT_PSWIDTH_100MS);
    // Enable WatchDog, Setting WatchDog configuration
}
```

Disable WatchDog:

```
void F75111_SetWDTDisable() {
    WriteByte(F75111_INTERNAL_ADDR, WDT_CONFIGURATION, 0x00); // Disable WatchDog
}
```