

Arrakis Pico MK4

Version:
v1.0.0

Date:
12.12.2024



Contents

1	Copyright	2
2	Regulatory Compliances	3
2.1	CE and UKCA Notice	3
2.2	FCC PART 15 VERIFICATION STATEMENT	4
2.3	ICES-003 ISSUE 7 VERIFICATION STATEMENT	5
3	Safety Instructions	6
4	Product Introduction	7
4.1	About Arrakis Pico MK4	7
4.2	Product Specification	8
5	System Information	10
5.1	System Drawing	11
5.2	Mainboard Block Diagram	12
6	I/O Ports	13
6.1	Connector Guide	13
6.2	Power Connector Pin-Out	14
6.3	COM and DI/DO Connector	15
7	Interfaces and Connections	17
7.1	Arrakis Pico Mk4 Series	18
7.2	Arrakis Pico Mk4 Series (with optional Radio Module)	19
8	Radio Modules (only relevant with optional LTE/WiFi Modules)	20
9	BIOS	21
9.1	Introduction:	21
9.2	Enter BIOS	21
9.3	BIOS menu and function keys	21
9.4	BIOS Help	23
9.5	Main Menu	24
9.6	Advanced	24
9.7	UART Port 1 Configuration	29
9.8	Restore on Power Loss	30
9.9	Security	31
9.10	Power	32
9.11	Boot	32
9.12	Exit	33
10	Driver Installation	35
11	Appendix A: Power consumption	36
12	Appendix B: F75111N DIO & Watchdog Device	37
12.1	Watchdog Timer under DOS	37
12.2	Watchdog Timer and DIO under Windows:	38
12.3	IO Device: F75111 VB6 under Windows	42
12.4	Watchdog Timer and DIO under Linux	44

1 Copyright

Copyright and Trademarks, 2024 Publishing. All Rights Reserved

This manual, software and firmware described in it are copyrighted by their respective owners and protected under the laws of the Universal Copyright Convention. You may not reproduce, transmit, transcribe, store in a retrieval system, or translate into any language, in any form or by any means, electronic, mechanical, magnetic, optical, chemical, biological, molecular, manual, or otherwise, any part of this publication without the express written permission of the publisher.

All products and trade names described within are mentioned for identification purpose only. No affiliation with or endorsement of the manufacturer is made or implied. Product names and brands appearing in this manual are registered trademarks of their respective companies. The information published herein has been checked for accuracy as of publishing time. No representation or warranties regarding the fitness of this document for any use are made or implied by the publisher.

We reserve the right to revise this document or make changes in the specifications of the product described therein at any time without notice and without obligation to notify any person of such revision or change.

2 Regulatory Compliances

2.1 CE and UKCA Notice

The Arrakis Pico Mk4 Series may contain the following LTE Module:

Trademark: Quectel

Model Number: EM05-G

Complies with: RADIO EQUIPMENT DIRECTIVE 2014/53/EU and Radio Equipment Regulations 2017 (SI 2017 No. 1206)

Article 3.1a Safety: Low Voltage Directive 2014/35/EU

- EN 62368-1:2018
- EN IEC 62368-1 :2020+A11:2020
- BS EN IEC 62368-1 :2020+A11:2020
- EN IEC 62368-1 :2020+A11:2020 (module)

Article 3.1a Health:

- EN IEC 62311:2020 (module)

Article 3.1b EMC: EMC Directive 2014/30/EU

- EN 55032:2015+AC:2016
- EN 55035:2017
- EN 61000-3-2:2014
- EN 61000-3-3:2013
- EN 61000-4-2:2009
- EN 61000-4-3:2006+A1
- EN 61000-4-4:2012
- EN 61000-4-5:2014+A1:2017
- EN 61000-4-6:2014+AC:2015
- EN 61000-4-8:2010
- EN 61000-4-11:2004+A1:2017
- BS EN 50121-4:2006+A1:2019
- BS EN 61000-6-4:2014
- BS EN 61000-4-2:2009
- BS EN IEC 61000-4-3:2020
- BS EN 61000-4-4:2012
- BS EN 61000-4-5:2014+A1:2017
- BS EN 61000-4-6:2014
- BS EN 61000-4-8:2010

- EN 301 489-1 V2.2.3 (module)
- Draft EN 301 489-19 V2.2.0 (module)
- EN 301 489-52 V1.2.1 (module)
- EN 55032:2015+A1:2020 (module)
- EN 55035:2017+A11:2020 (module)

Article 3.2 Radio:

- EN 301 908-1 V15.1.1 (module)
- EN 301 908-2 V13.1.1 (module)
- EN 301 908-13 V13.2.1 (module)
- EN 303 413 V1.2.1 (module)

RoHS 2 Directive 2011/65/EU & 2015/863/EU

- Exemption(s) used:
 - 6a, 6b, 6c

The corresponding markings are located on the device:



2.2 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

2.3 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 aux deux 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 keep them for future reference.

1. Disconnect Before Cleaning

Unplug this equipment from the power outlet before cleaning. Do not use liquid or sprayed detergent. Use a moist cloth or sheet.

2. Avoid Humidity

Keep this equipment away from humidity.

3. Proper Power Cord Handling

Ensure the power cord is arranged to prevent people from stepping on it. Do not place anything over the power cord.

4. Follow Equipment Cautions

Note all cautions and warnings marked on the equipment.

5. Long Periods of Inactivity

If the equipment is unused for a long period, disconnect it from the main power to avoid damage from transient overvoltage.

6. Voltage Requirements

Prolonged usage with less than 12V may damage the PSU or destroy the mainboard.

7. Prevent Liquid Entry

Never pour any liquid into the openings of the equipment, as this could cause fire or electrical shock.

8. Service Personnel Assistance

If any of the following situations occur, have the equipment checked by qualified service personnel:

- The power cord or plug is damaged.
- Liquid has entered the equipment.
- The equipment has been exposed to moisture or condensation.
- The equipment is not functioning properly or fails to operate according to the user manual.
- The equipment has been dropped or physically damaged.

9. Storage Temperature

Do not leave this equipment in an unconditioned environment. Prolonged exposure to temperatures below -20°C or above 60°C may damage the equipment.

10. Unplug for Servicing

Unplug the power cord when performing any service or when adding optional kits.

11. Lithium Battery Caution

- Danger of explosion if the battery is replaced incorrectly. Replace only with the same or equivalent type recommended by the manufacturer. Dispose of used batteries according to the manufacturer's instructions.
- Do not remove the cover; there are no user-serviceable components inside. Take the unit to an authorized service center for service and repairs.

4 Product Introduction

4.1 About Arrakis Pico MK4

Thank you for choosing Welotec Arrakis Pico MK4. The Arrakis Pico Mk4 is a ultra compact fan less embedded system for industrial computing application. It features industrial grade components and is suitable to operate in harsh and critical environments.

The Arrakis Pico MK4 is powered by a 1.5/3.0 GHz Quadcore Intel Atom® x6413E Processor, and its standard Version has 8 GB or 16 GB LP-DDR4 Memory on board.

It is capable to support 1x HDMI with the resolution up to 4K. The system features M.2 NVMe storage, one additional M.2 socket and an external Nano SIM Socket. Up to three USB 3.0, one USB 2.0 and three 2.5 Gigabit-LAN Ports, one Digital In and Out each, support for 4G/5G WWLAN and one COM port.

With its quick mounting design, you have the option to setup between Wall mount and DIN mount.

Combining the above-mentioned features with its aluminium housing and dust-proof design makes Arrakis Pico Mk4 a perfect industrial embedded system to operate in environments with harsh and critical conditions.

4.2 Product Specification

Feature	Specification	Arrakis Pico Mk4 - Standard	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® x6200FE Processor, 1.0 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-IT)
	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 WNFAQ-262ACNI(BT) with SATA-SSD	Sparklan WNFAQ-262ACNI(BT) with SATA-SSD	Sparklan WNFAQ-262ACNI(BT) 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

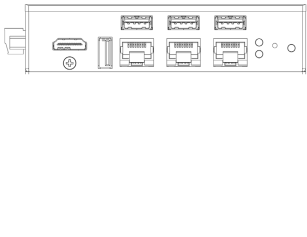
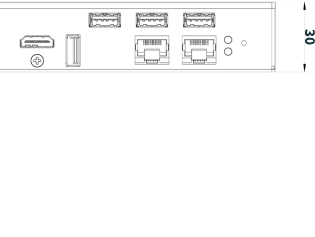
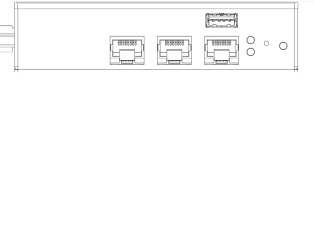
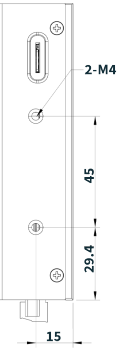
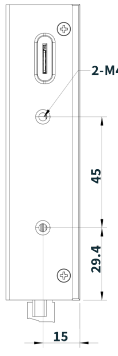
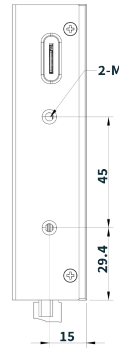
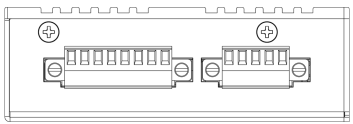
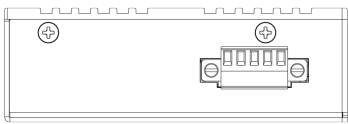
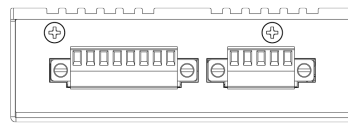
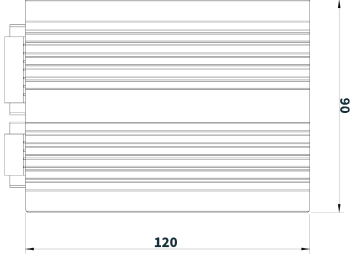
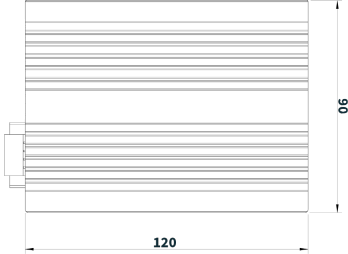
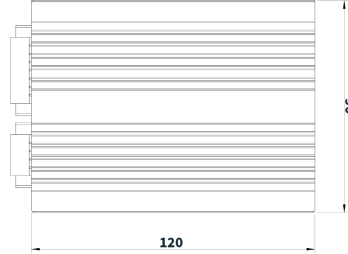
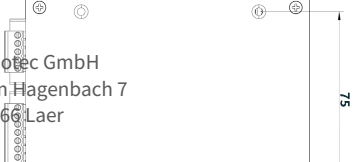

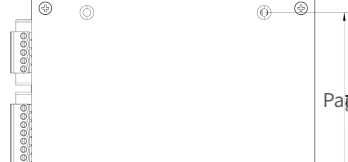
Attention: When opening the chassis make sure to slide the chassis top to the rear. Lifting the top up may shear the SIM Slot from the PCB.



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.

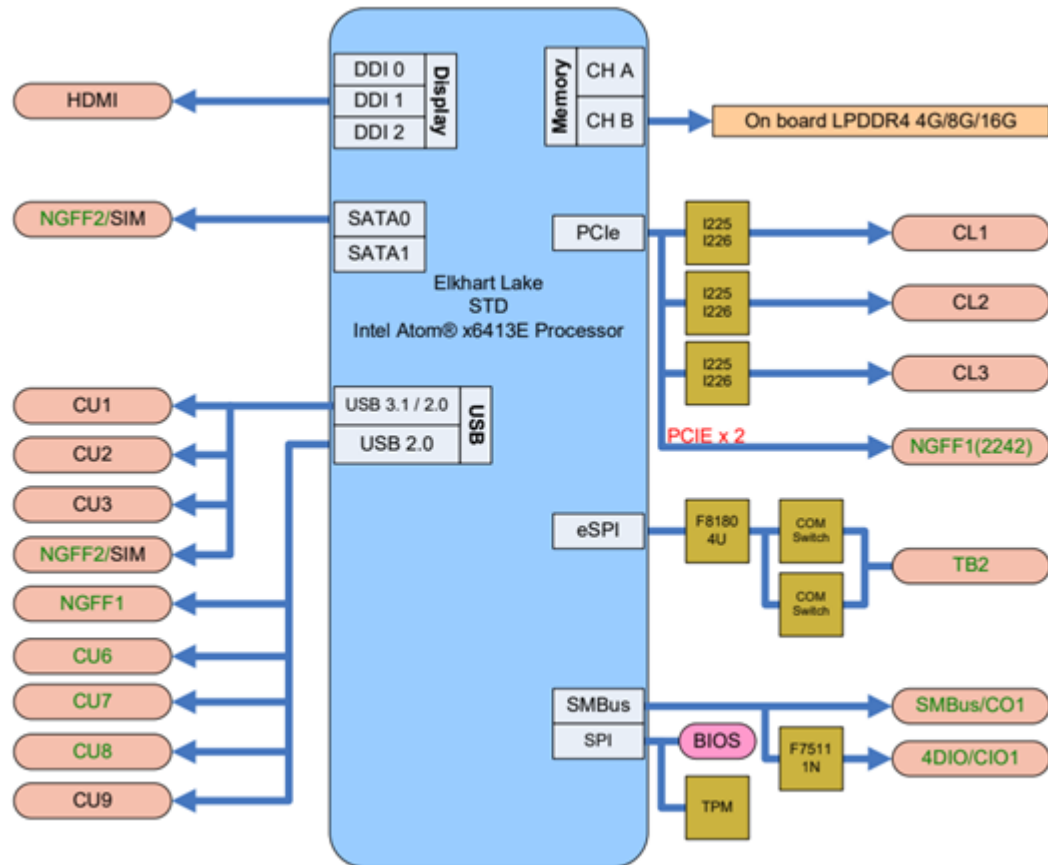
To allow for sufficient heat removal we recommend: 30mm distance on either side of the Arrakis Pico Mk4 when mounted on a DIN-Rail 100mm headroom above the Arrakis Pico Mk4 when mounted horizontally. The heatsink should be on top.

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 I/O Ports

6.1 Connector Guide

This section provides a detailed overview of all connectors, switches, and indicators needed for peripheral setup. Before installation, ensure that the system is powered off and disconnected from any power sources.

6.1.1 Switches and Jumpers

- **JSB1: CMOS Data Clear**
 - 1-2: Normal set
 - 2-3: CMOS data clear

6.1.2 Quick List of Connectors

- **CBT1:** CMOS Battery in a 1x2 pin (1.25mm) wafer.
- **CU1, CU2, CU3:** USB3.1 type A connectors.
- **CU9:** USB2.0 type A connector.
- **CU6, CU7, CU8:** USB 2.0 ports, each with a 1x4 pin (1.25mm) wafer.
- **CL1, CL2, CL3:** RJ45 LAN connectors for network connections.
- **TB1:** Power button / DC-IN with a TB 5-pin (3.5mm) connector.
- **TB2:** TB 8-pin (3.5mm) connector.
- **CFP1:** Front Panel connector with a 2x5 pin (2.0mm) wafer.
- **CO1:** SMBus connection via a 1x4 pin (1.25mm) wafer.
- **SIM1:** Nano SIM card socket for mobile connectivity.
- **NGFF1:** M.2 B key socket for 2242/2280 form factors with H=8.5, featuring 75 pins.
- **NGFF2:** M.2 B key socket for 2242/3042 form factors with H=8.5, featuring 75 pins.
- **HDMI1:** HDMI type A video output connector for displays.

6.1.3 Safety and Handling

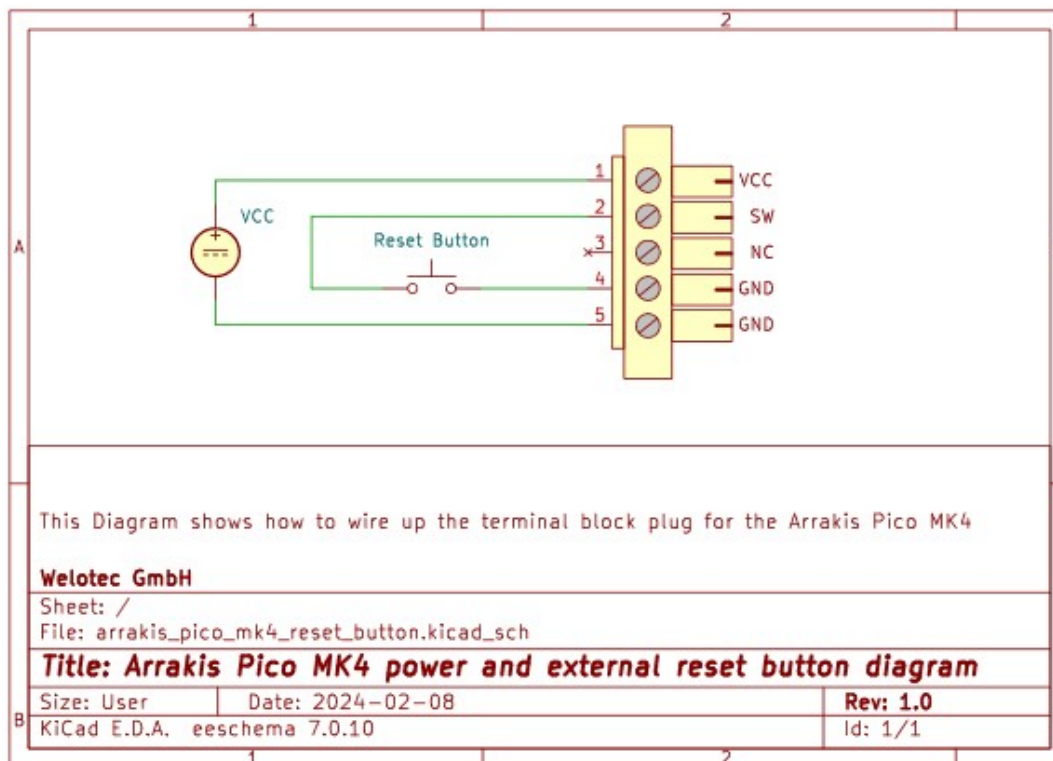
- **Safety:** Always ensure the system is completely powered off and disconnected from any power source before connecting or disconnecting peripherals.
- **Handling:** Carefully handle connectors and avoid forceful insertion to prevent damage to pins and ports.

6.2 Power Connector Pin-Out

Attention: Do not connect Voltage to the SW Pin on the Power Connector!

Power Connector Pin-Out	
Vcc	12-24V
SW	External Power Switch Connector
NC	Not Connected
GND	Ground
GND	Ground

6.2.1 Power Connector wiring schematic



6.3 COM and DI/DO Connector

COM1 RS232 or RS485 form BIOS select , default set RS232

TB2-TB 8pin connector (3.5mm)(RS232 Mode)

PIN NO.	DESCRIPTION	PIN NO.	DESCRIPTION
1	NTXD	5	
2	NC	6	
3	NRXD	7	
4	GND	8	

TB2-TB 8pin connector (3.5mm)(R485 Mode)

PIN NO.	DESCRIPTION	PIN NO.	DESCRIPTION
1	NC	5	
2	RS485_DATA-	6	
3	RS485_DATA+	7	
4	GND	8	

2-4 Digital Input / Output / Watch Dog Time

2-4-1 TB2-TB 8pin connector (3.5mm)

Note

1. F75111N-1 I2C bus address 0x9C.
2. DI input range is +12V or +24V.
3. DI pin default pull up 10KΩ to +3.3V.
4. DO output voltage rail from DC-in(+12V or +24V) and current limit max 2A.

For F75111N I2C watch dog timer device:

Input low Voltage (VIL):+0.8 Max,

Input High Voltage (VIH): +2V Min

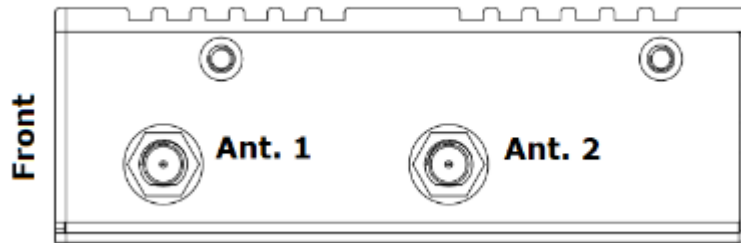
Output low Current (IOL):10mA (Min) VOL=0.4V

Output High Current (IOH):-10mA (Min) VOL=2.4V

Watch Dog Time value 0~255 sec

The system will be issued reset. When WDT is enable the hardware start down counter to zero. The reset timer has 10~20% tolerance upon the Temperature.

6.3.1 Antennae Connector Position LTE/5G:



6.3.2 Antennae Connector Position Wi-Fi:

Ant. 1: WiFi 1 Ant. 2: WiFi 2 The WiFi Connectors are equivalent.

Connector Type: SMA-RP

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).

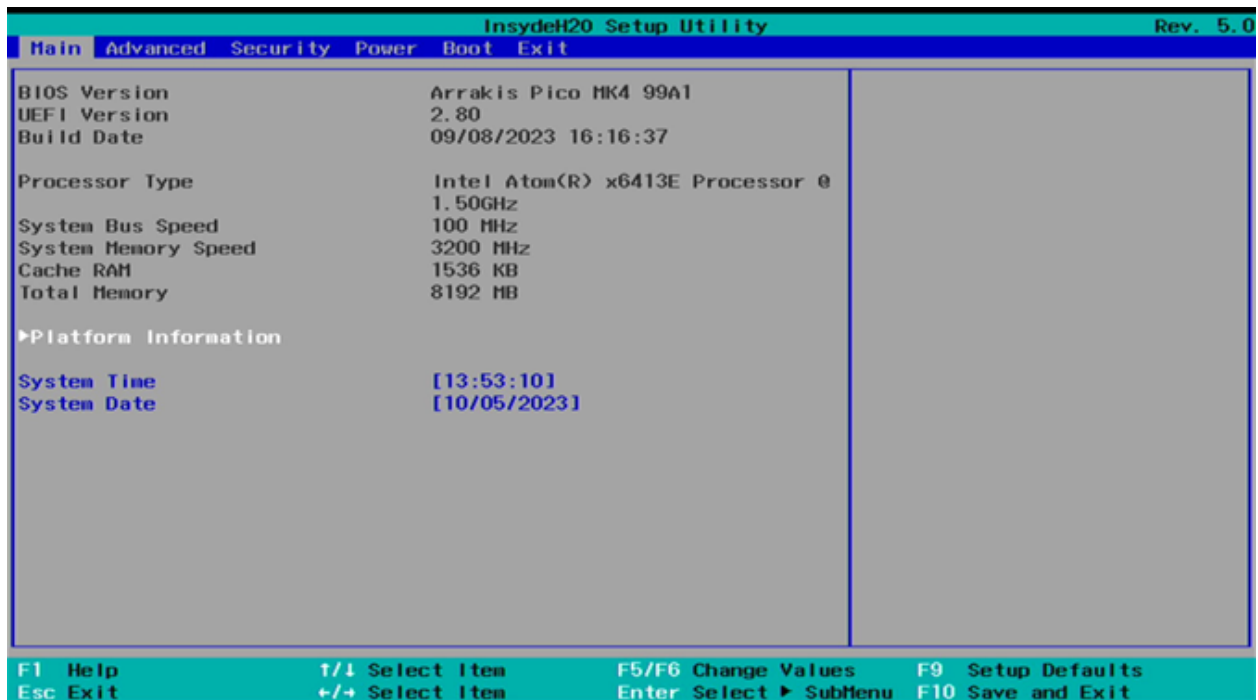
Since the BIOS is the sole channel for hardware and software communication, it is crucial for system stability and optimal performance. In the BIOS Setup main menu, you can see several options. These options will be explained in detail down below. First, let's look at the function keys you may use here:

- 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:

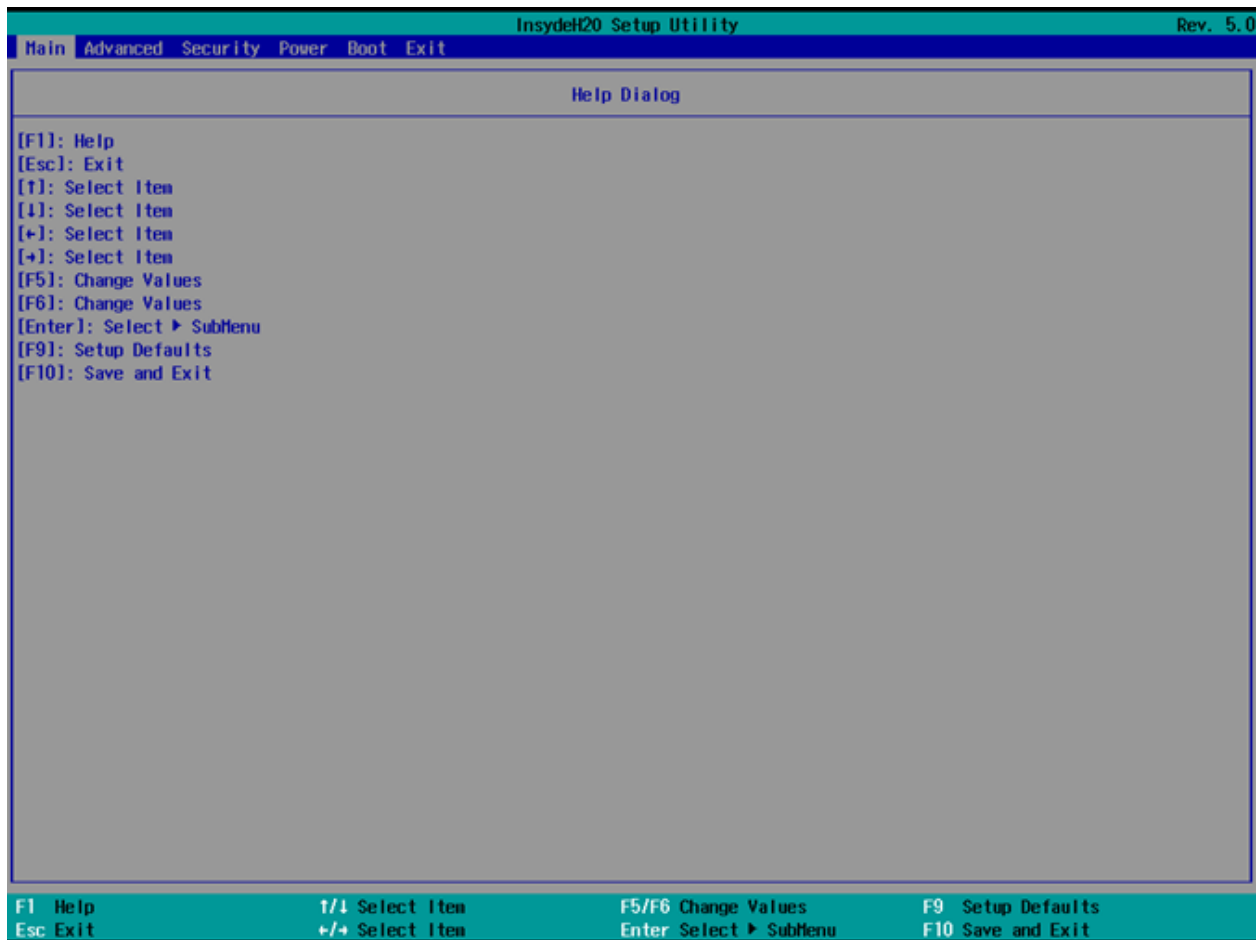
- Press ←→ (left, right) to select the screen.
- Press ↑↓ (up, down) to choose the option you want to confirm or modify.
- Press Enter to select.
- Press + or - to modify the BIOS parameters for the active option.
- F1: General help.
- F2: Previous value.
- F3: Optimized defaults.
- F4: Save & Reset.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

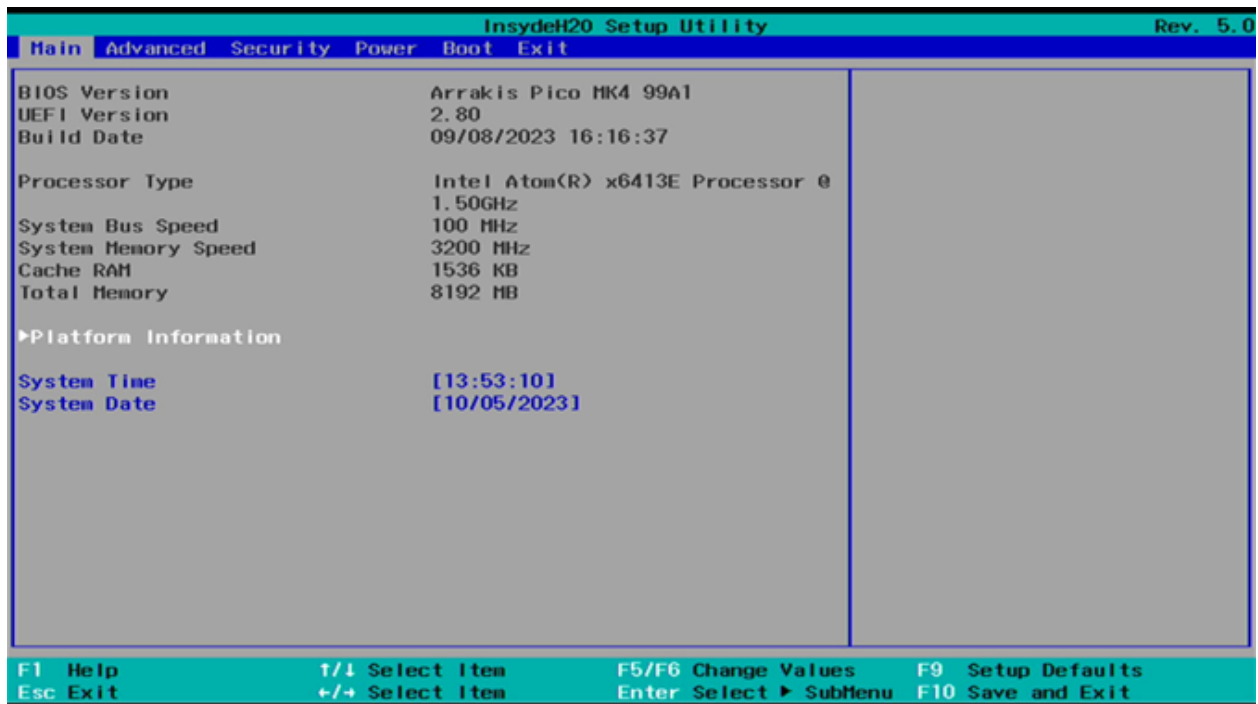
9.4 BIOS Help



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.

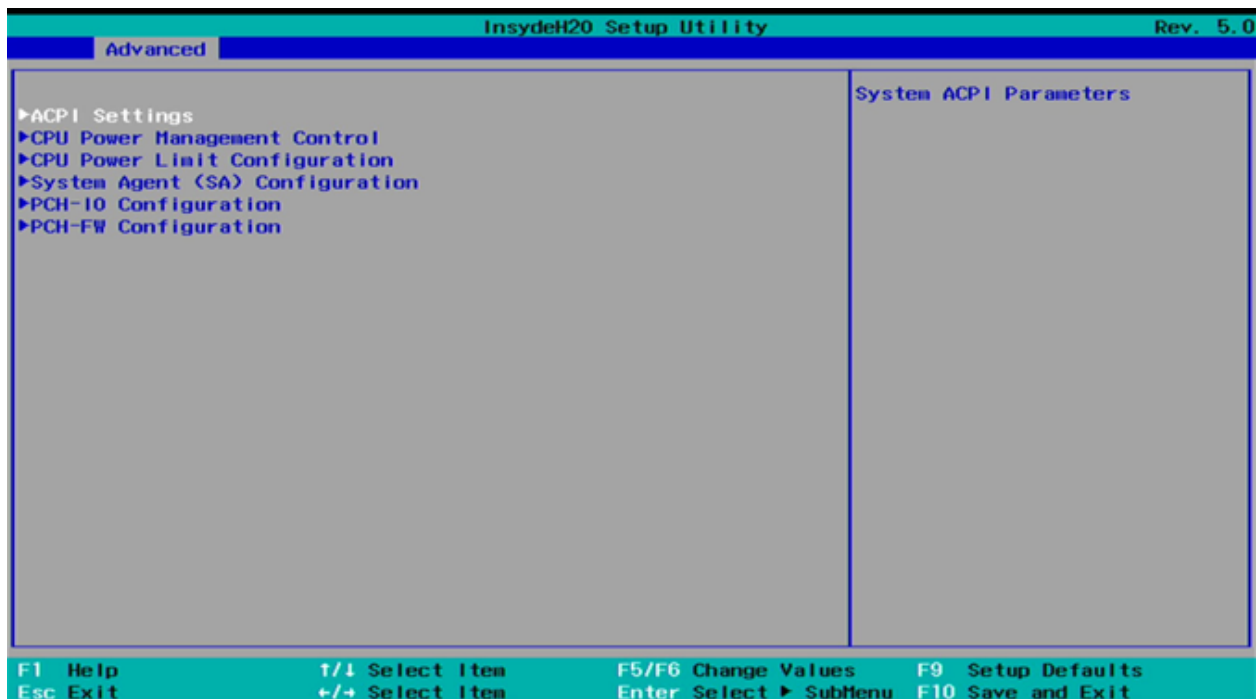
9.5 Main Menu



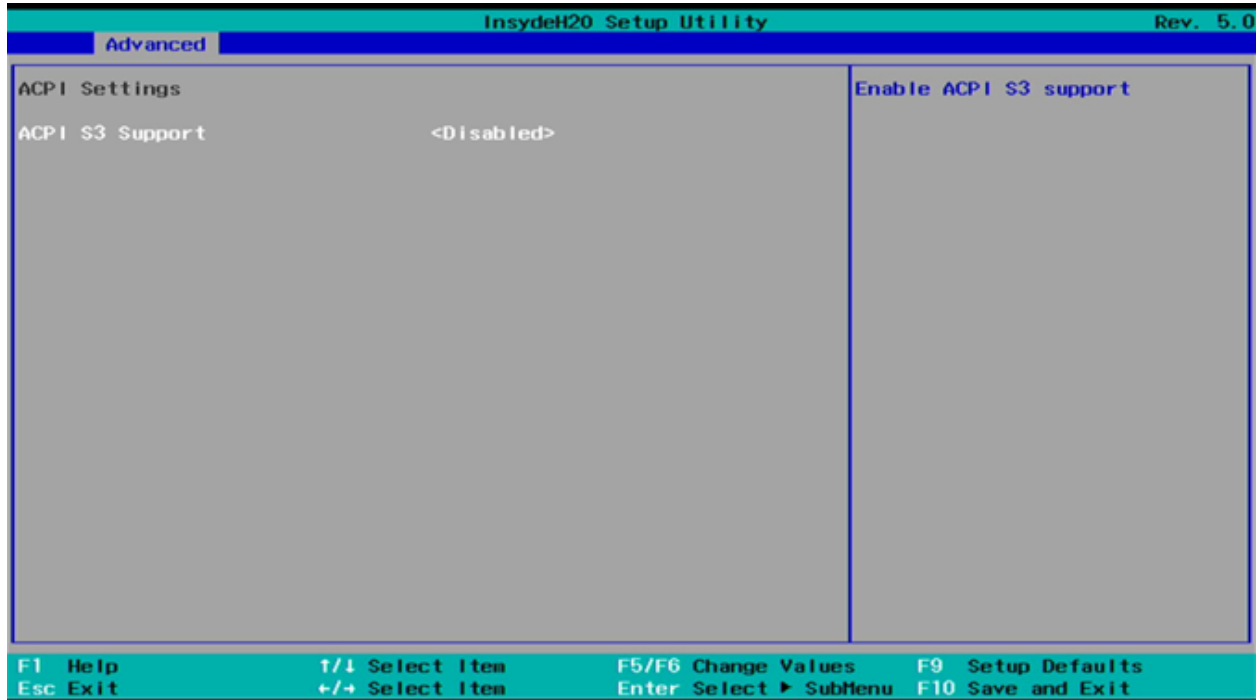
The Main menu screen includes some basic system information. Highlight the item and then use the + or - keys and numerical keyboard keys to select the value you want in each item.

- **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

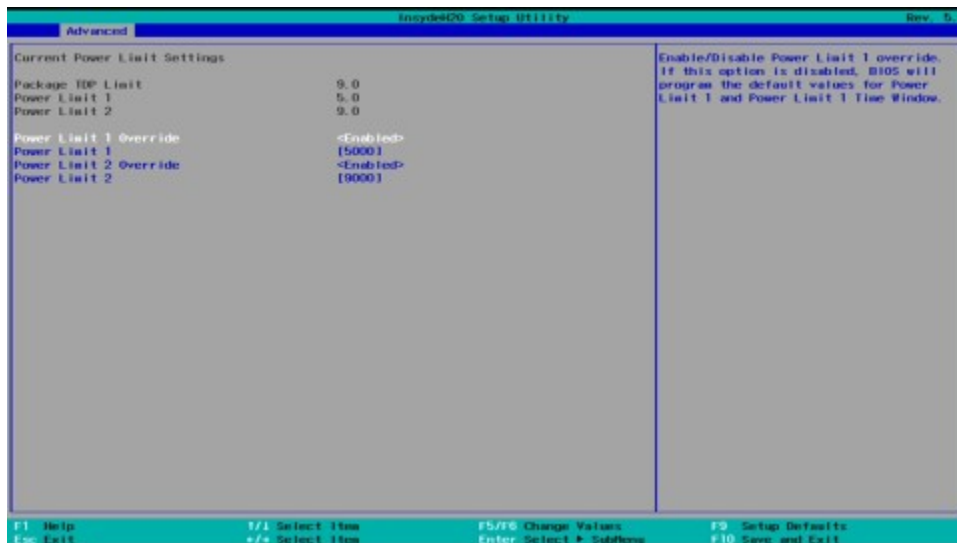


9.6.1 ACPS Settings



Toggle to enable/disable ACPI S3 support.

9.6.2 CPU Power Limit Configuration

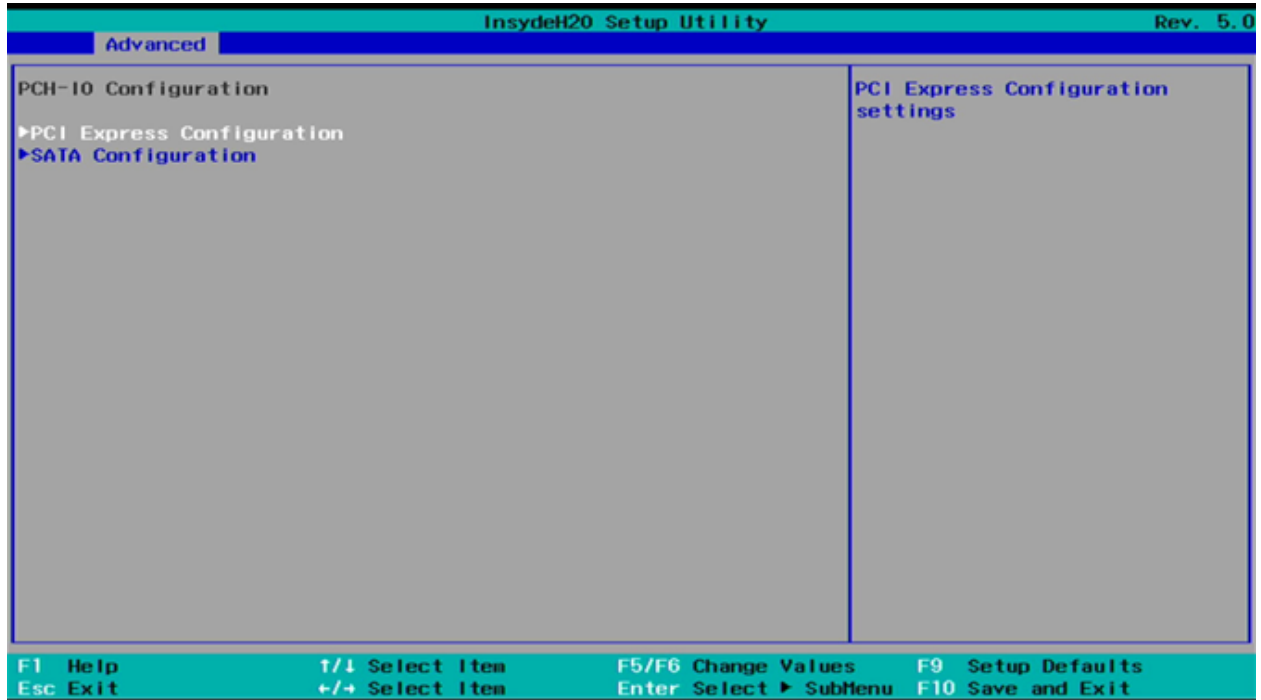


By default, both power limit overrides are enabled. Recommended values are:

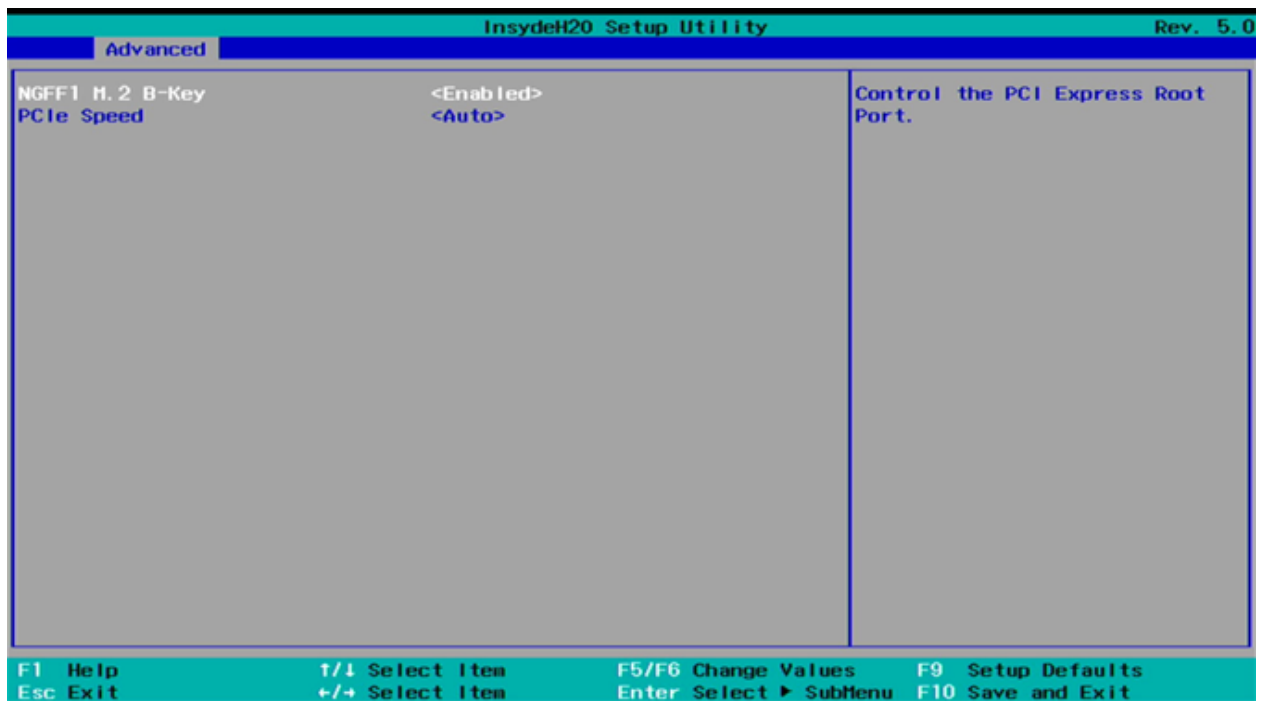
- **Power Limit 1:** 5000
- **Power Limit 2:** 9000

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



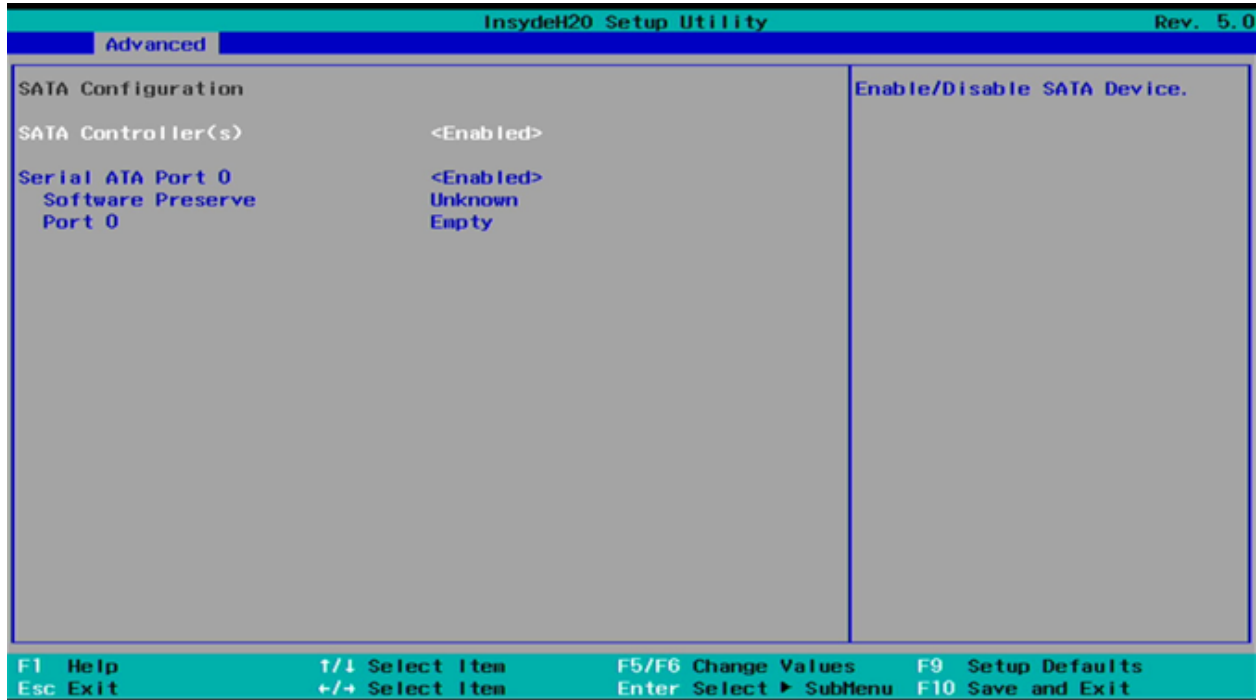
9.6.4 PCI-Express Configuration



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



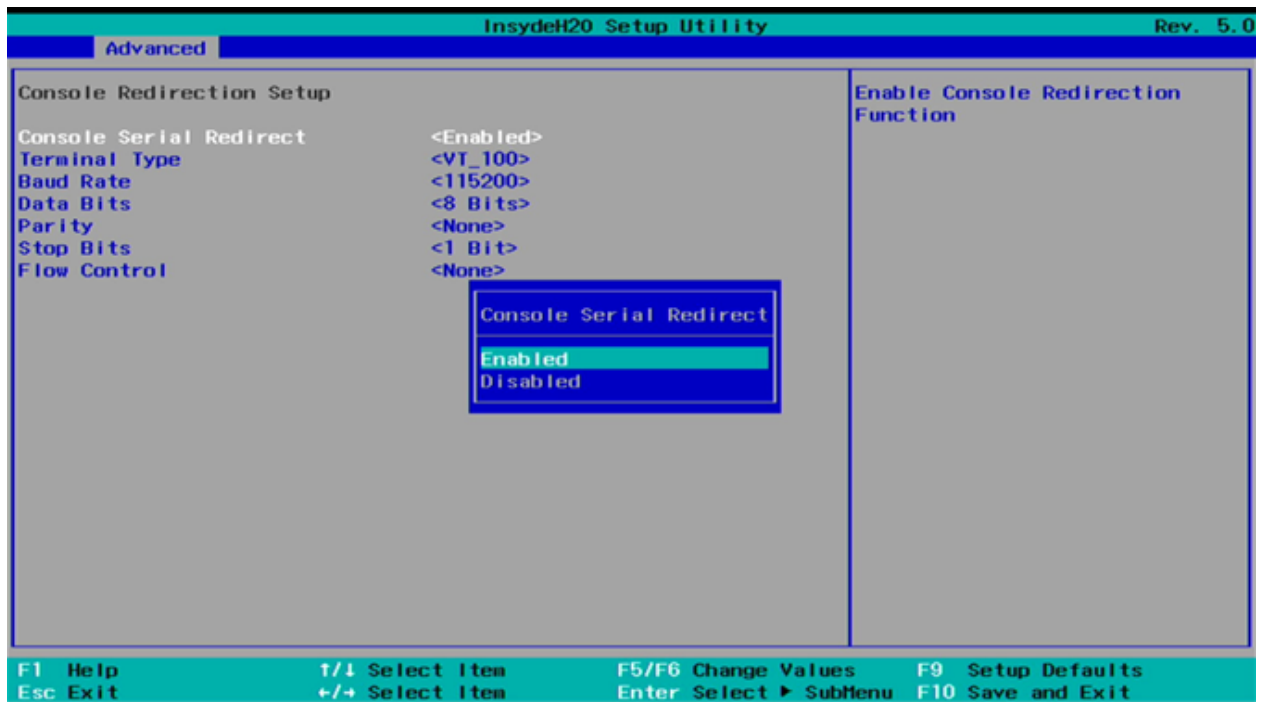
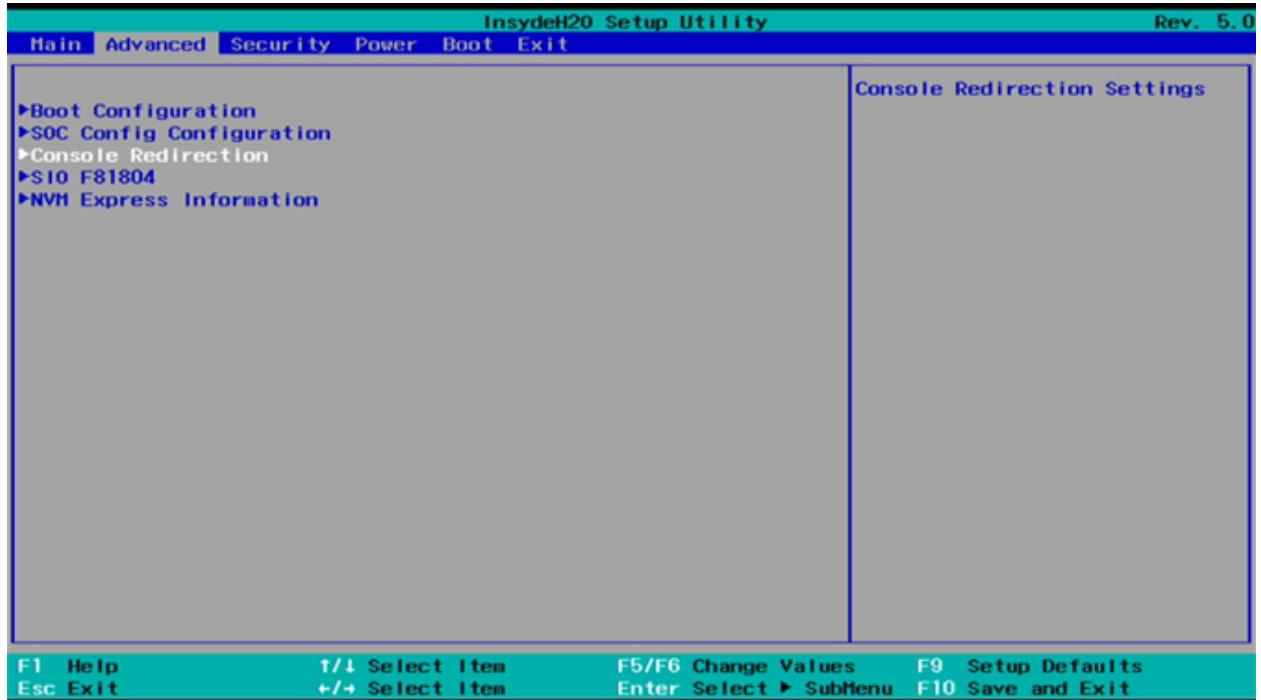
There is usually no need for users to make any changes here. The default settings are:

- **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

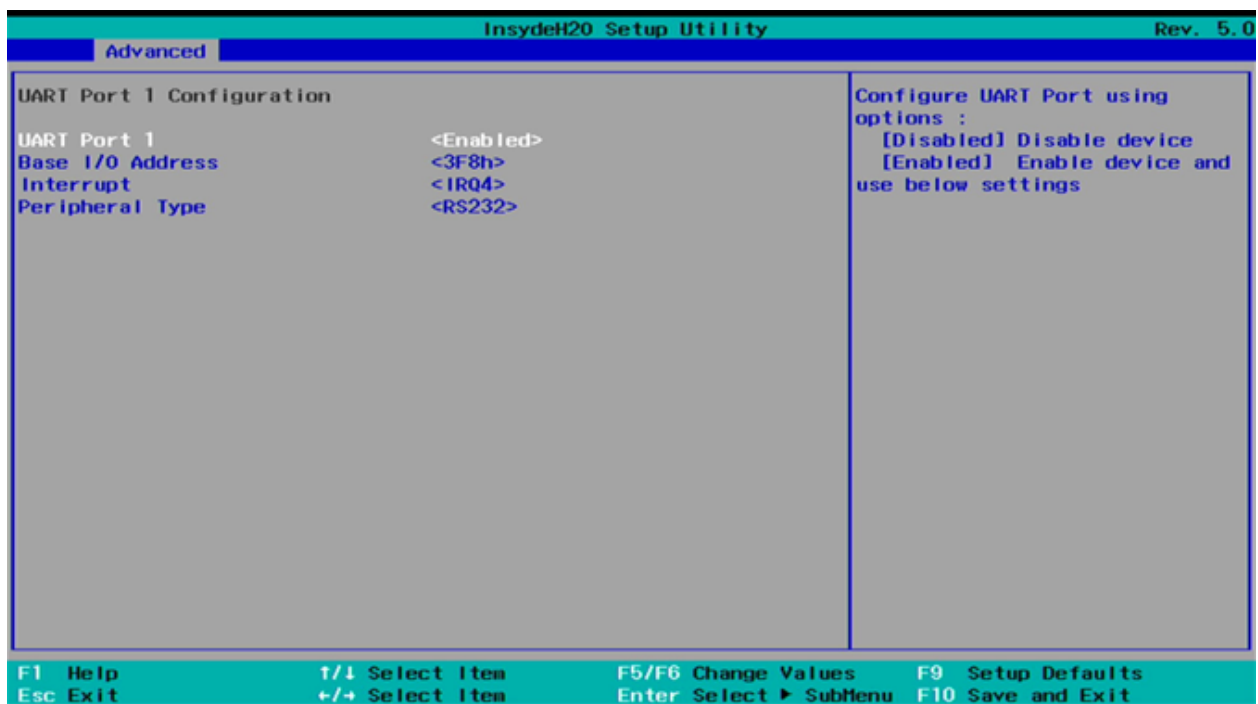
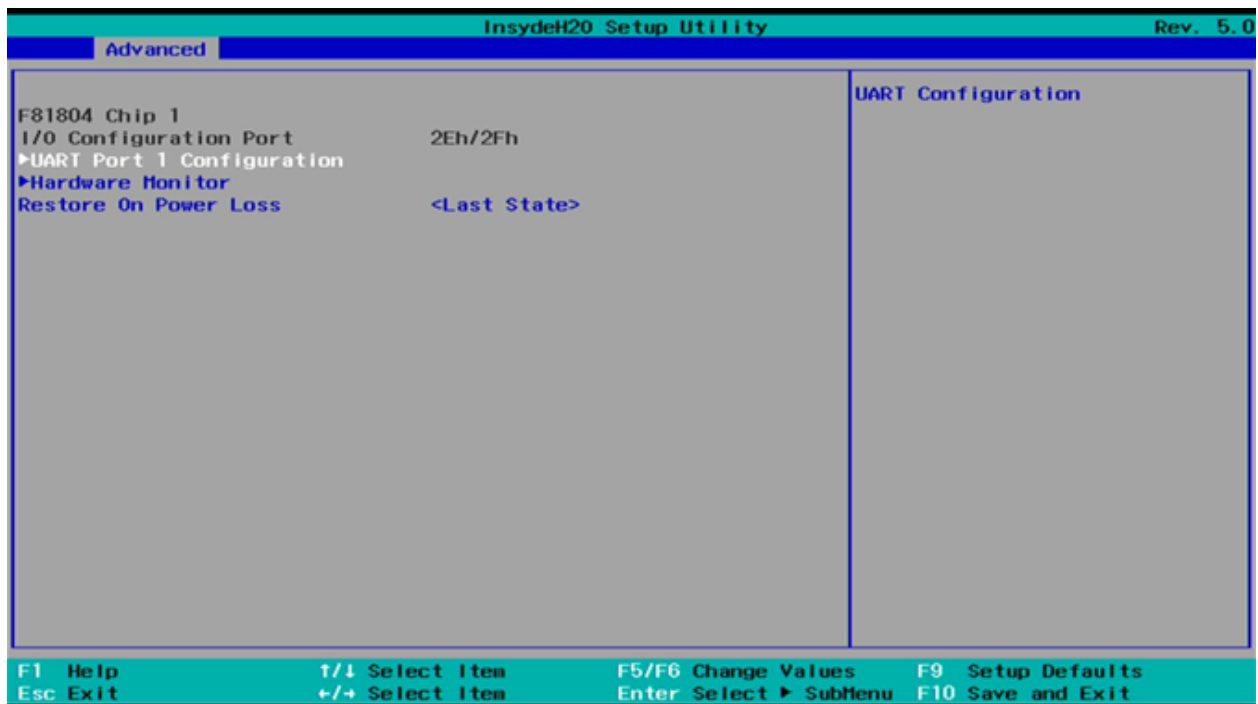


Toggle Console Serial Redirect to enable/disable the function.

- **Default:** Enabled

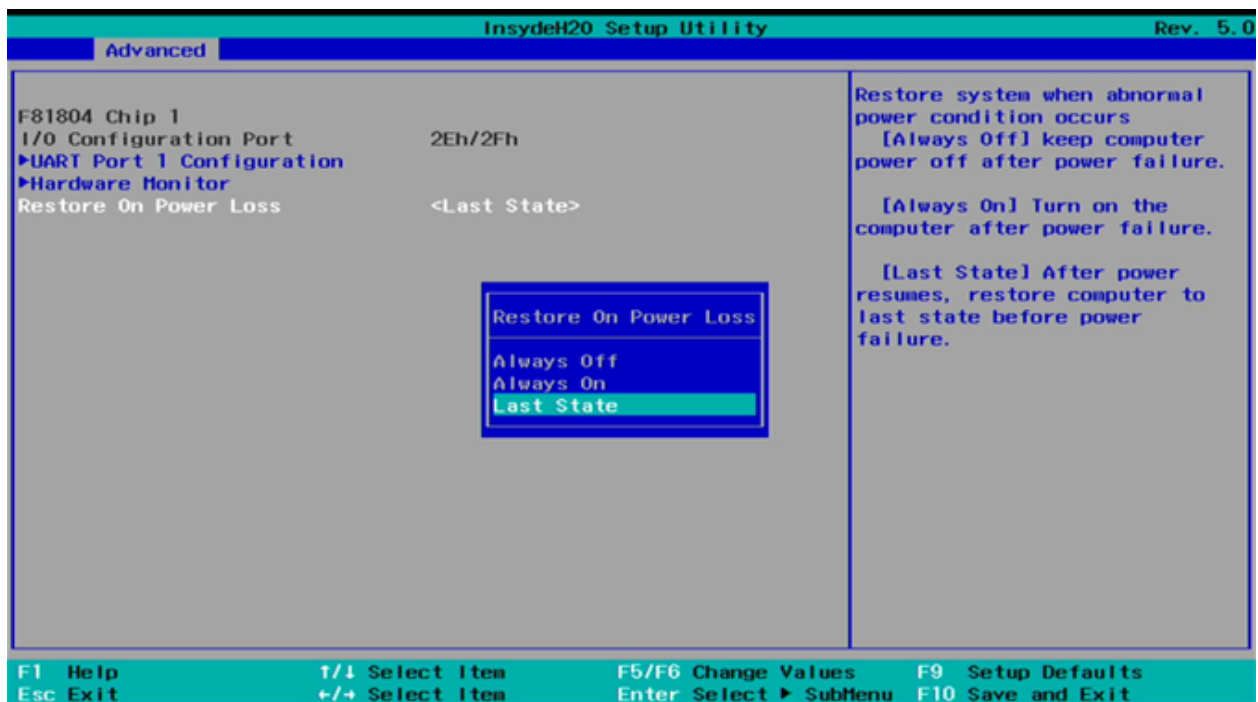
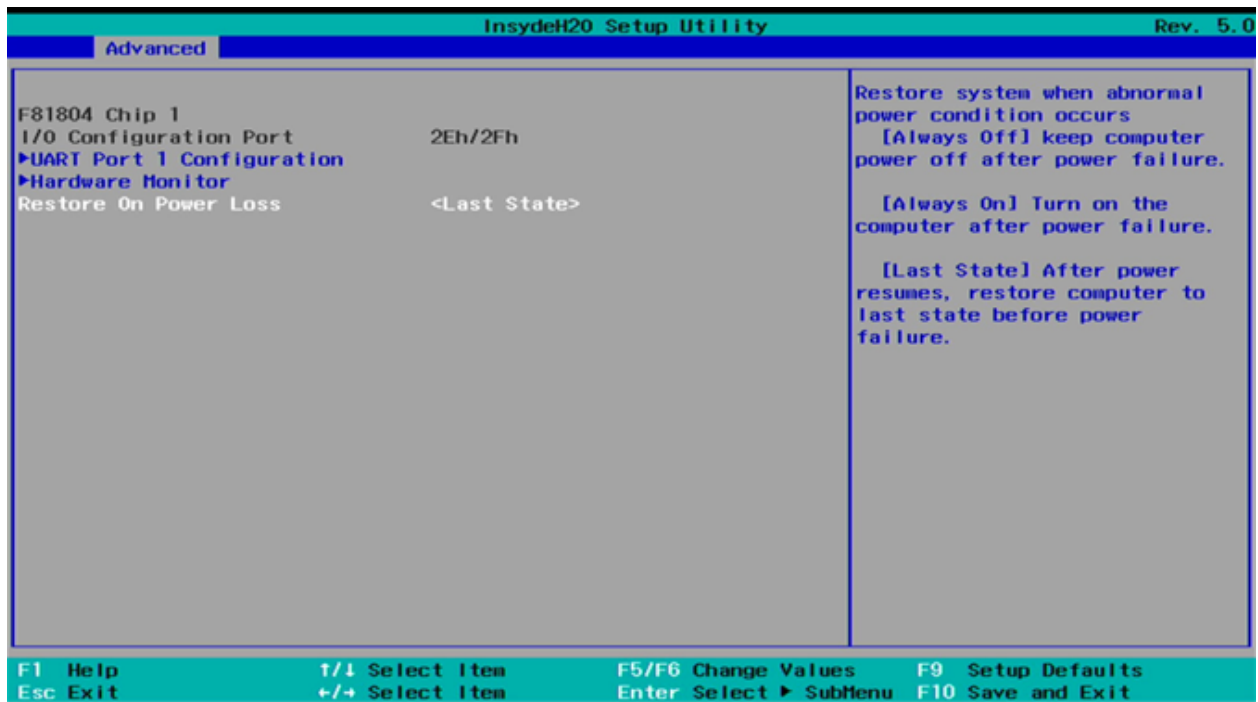
You can configure the terminal settings in this dialogue.

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.

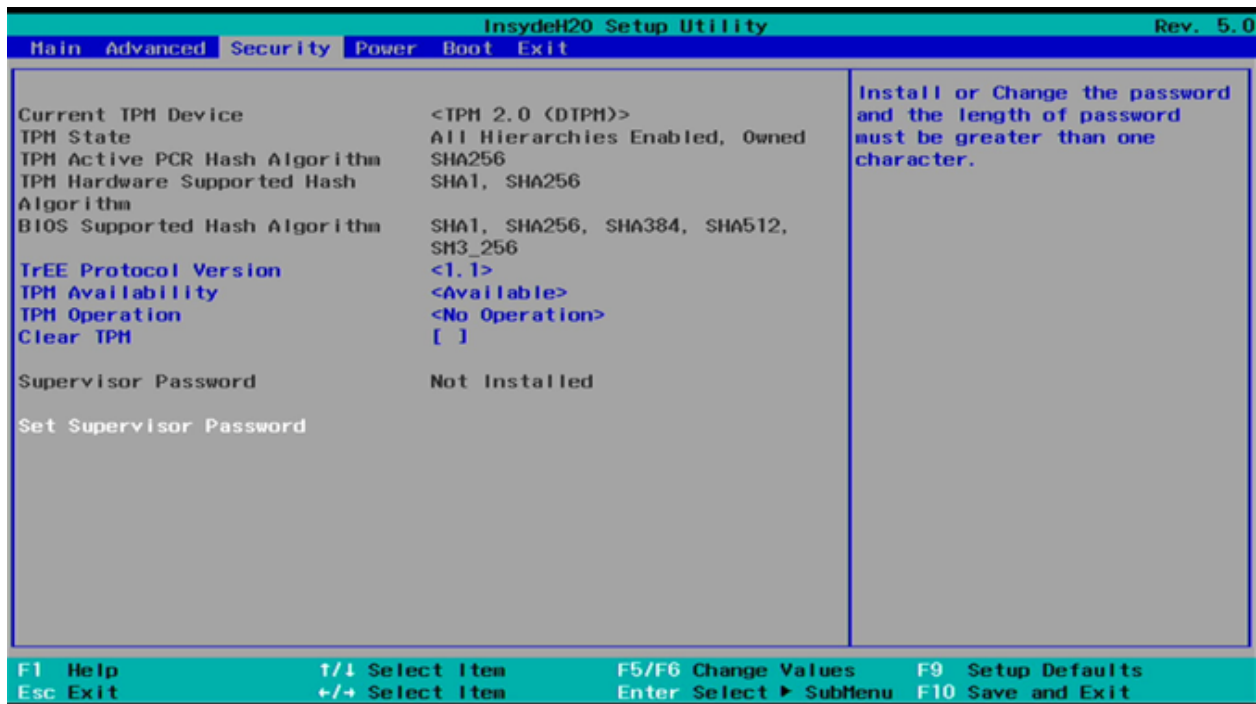
9.8 Restore on Power Loss



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:

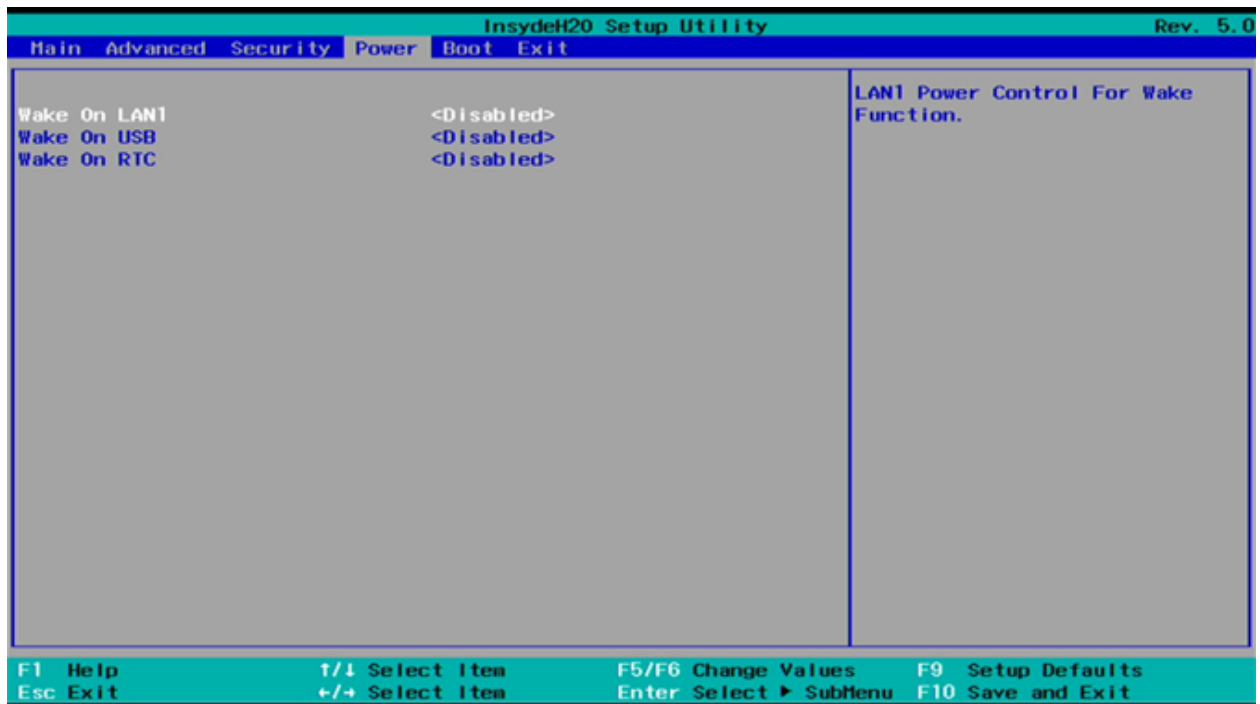
1. Select **Supervisor Password**. A “Create New Password” dialog will pop up.
2. Enter your desired password (must be between 3 and 10 characters).
3. Press **Enter** to submit.

9.9.2 Security Advisory

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

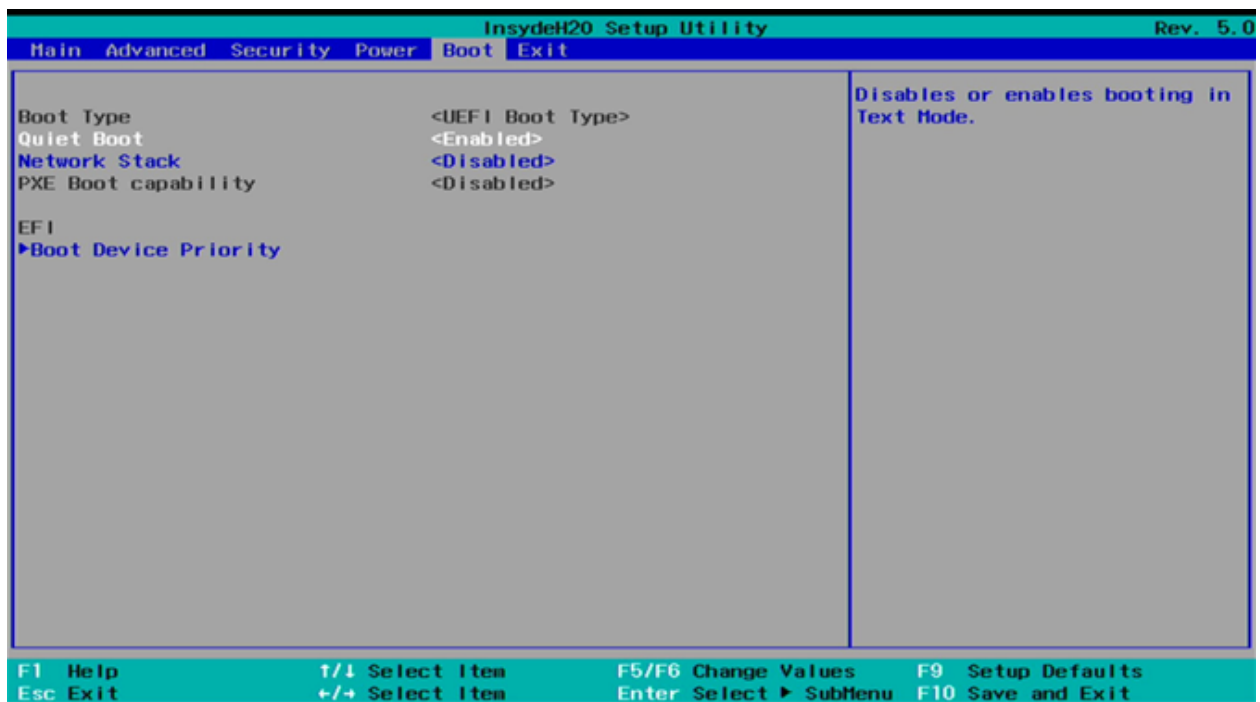
1. Create an **Admin Password** in the BIOS -> **Security** section.
2. Deactivate all unnecessary boot media in the BIOS -> **Boot** section.

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

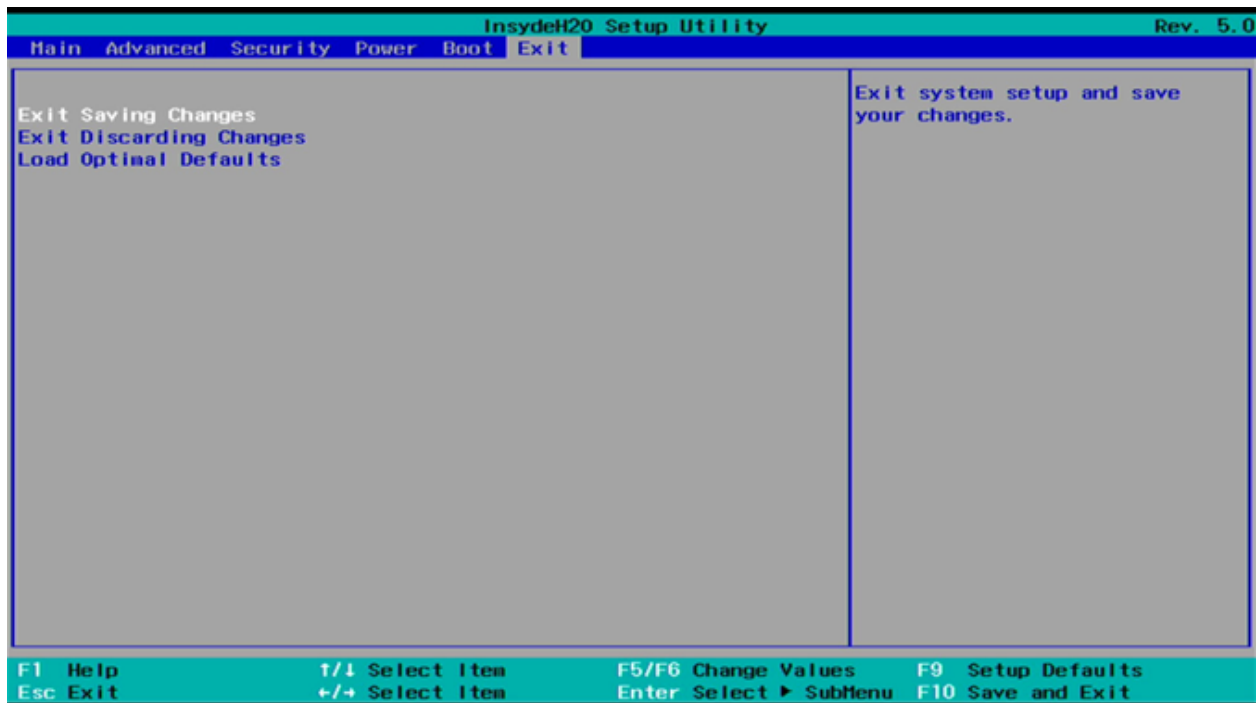
This item determines the protocol used during PXE boot:

- 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.

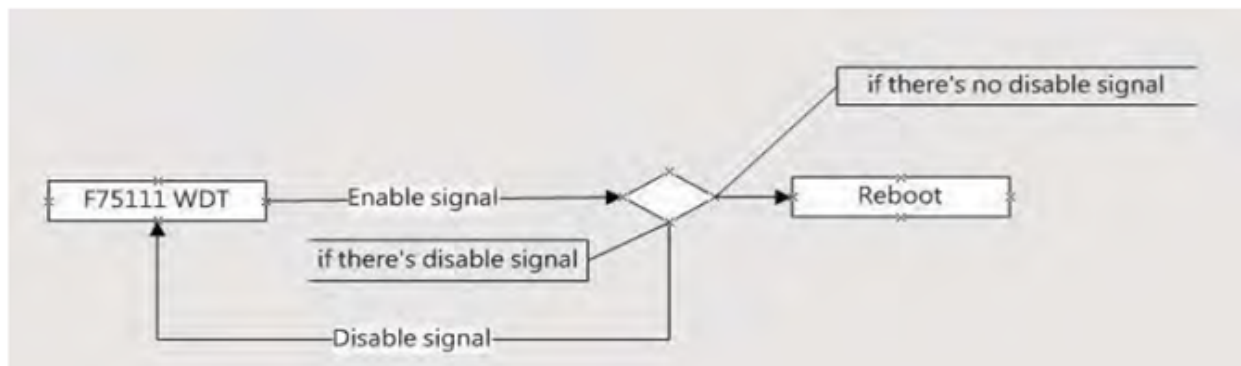
12.1 Watchdog Timer under DOS

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:

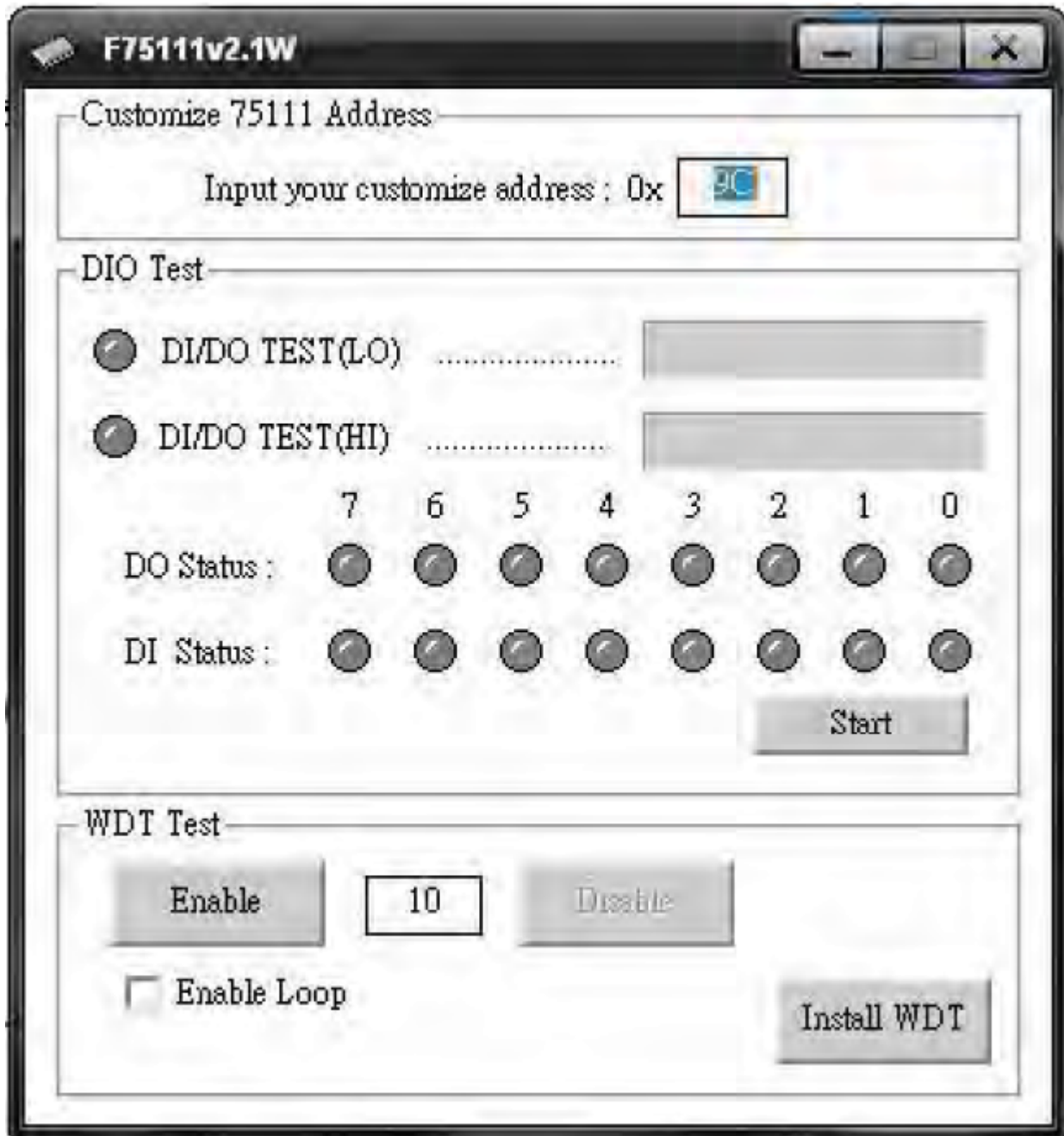
```
void pause(int time) {
    asm mov ah,0h; // Ah = 00 Read System Time Counter
    asm int 1ah; // Read time from Time Counter and store it in DX register
    asm add dx, time;
    asm mov bx, dx;
label:
    asm int 1ah;
    asm cmp bx, dx;
    asm jne label;
}
```

12.2 Watchdog Timer and DIO under Windows:

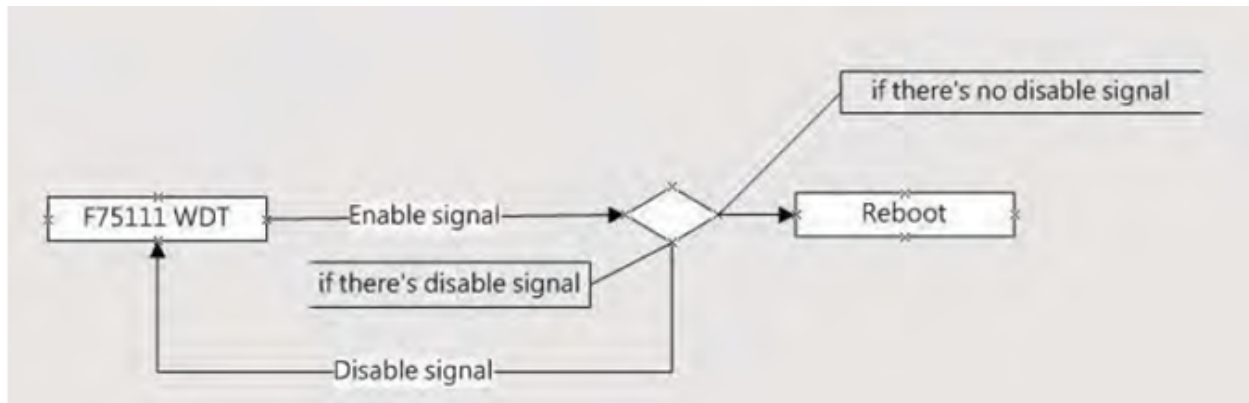
The necessary software resources for programming the watchdog timer can be accessed from the Driver Download section:

- **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:

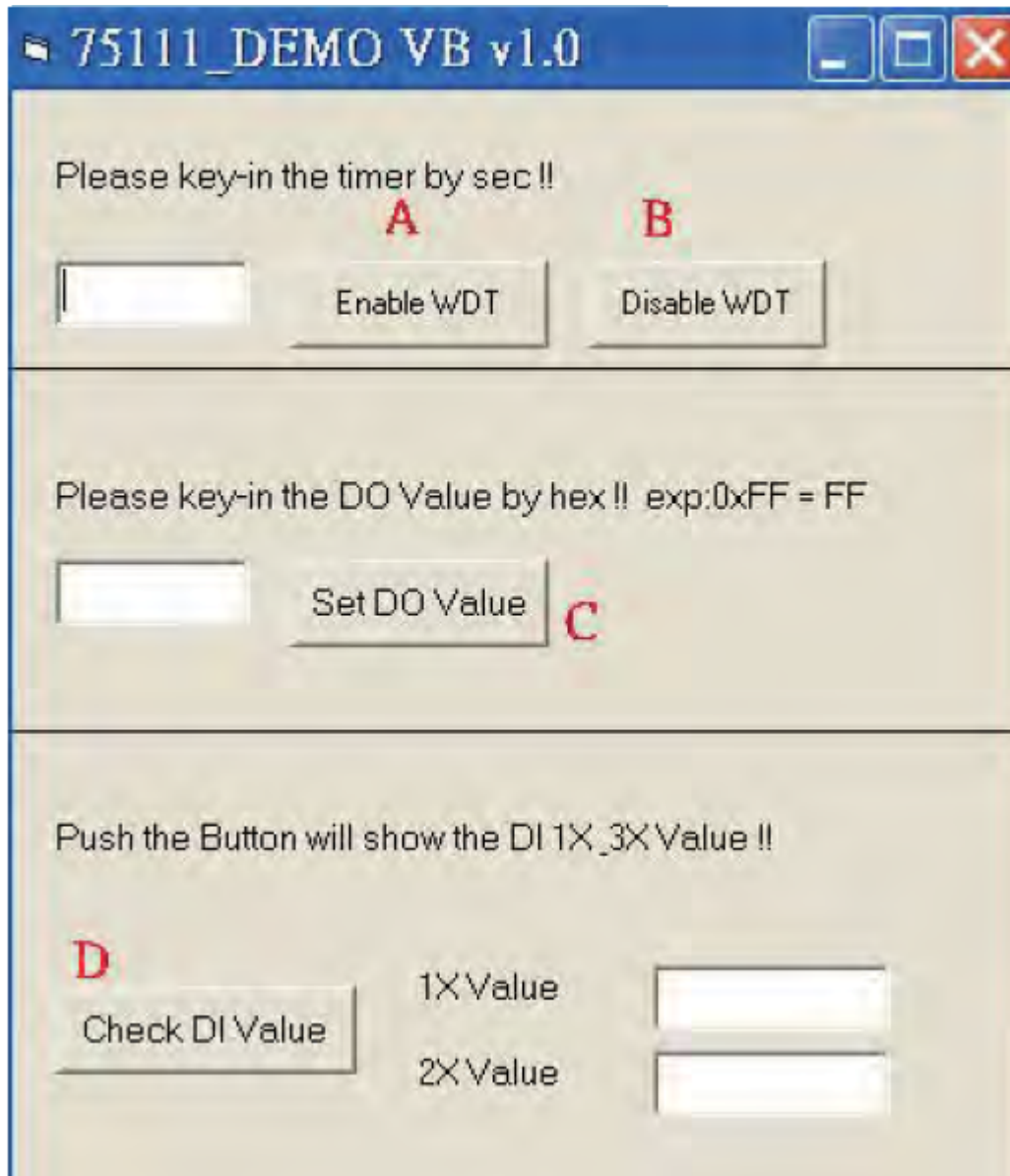
```
void F75111_SetWDTDisable() {
    WriteByte(F75111_INTERNAL_ADDR, WDT_CONFIGURATION, 0x00); // Disable Watchdog
}
```

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, doalvalue)
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”:

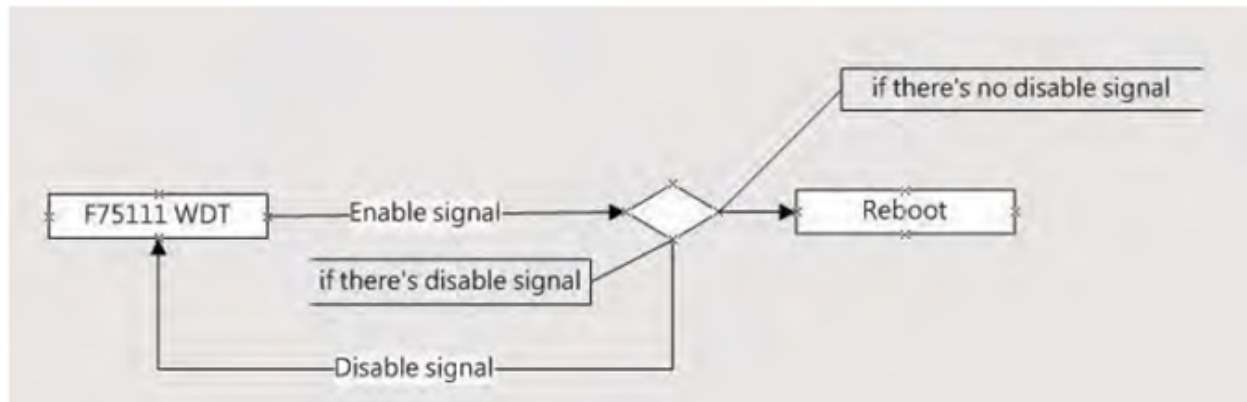
- Navigate to the F75111 directory: `cd F75111`.
- Compile the source: `make`.
- Execute the binary file: `src/f75111`.

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 & 0xFF);
}
  
```

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
}
```