Arrakis Pico Mk4 Series

Version:
v1.1.0

Date:
17.11.2025

Contents

1 Copyright 3

2 Regulatory Compliances 4
- 2.1 Complies with the following EU directives . . . 4
- 2.2 References of standards applied . . . 5
- 2.3 FCC PART 15 VERIFICATION STATEMENT . . . 6
- 2.4 ICES-003 ISSUE 7 VERIFICATION STATEMENT . . . 6

3 Intended Use and IT Security Instructions 7
- 3.1 Intended Use . . . 7
- 3.2 Non-Intended Use . . . 9
- 3.3 Exposed Interfaces and Services . . . 9
- 3.4 Cyber Security . . . 10
- 3.5 Vulnerability Handling . . . 12

4 Safety Instructions 13

5 Product Specifications 14

6 System Information 18
- 6.1 System Drawing . . . 19
- 6.2 Mainboard Block Diagram . . . 20

7 Power Supply 21

8 Interfaces and Connections 22
- 8.1 Arrakis Pico Mk4 Series . . . 23
- 8.2 Arrakis Pico Mk4 Series (with optional Radio Module) . . . 24

9 Radio Modules (only relevant with optional LTE/WiFi Modules) 25

10 BIOS 26
- 10.1 Introduction: . . . 26
- 10.2 Enter BIOS . . . 26
- 10.3 BIOS menu and function keys . . . 26
- 10.4 BIOS Help . . . 28
- 10.5 Main Menu . . . 29
- 10.6 Advanced . . . 29
- 10.7 UART Port 1 Configuration . . . 34
- 10.8 Restore on Power Loss . . . 35
- 10.9 Security . . . 36
- 10.10 Power . . . 37
- 10.11 Boot . . . 37
- 10.12 Exit . . . 38

11 Driver Installation 40

12 Appendix A: Power Consumption 41

13 Appendix B: F75111N DIO & Watchdog Device 42
- 13.1 Watchdog Timer under DOS . . . 42
- 13.2 Watchdog Timer and DIO under Windows: . . . 43

13.3 IO Device: F75111 VB6 under Windows 47
13.4 Watchdog Timer and DIO under Linux 49

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 Intended Use and IT Security Instructions

This section provides crucial safety and security information and recommendations to help you configure your Welotec Industrial Computer (IPC) for optimal security in your deployment.

3.1 Intended Use

This section specifies the intended use and essential operating conditions for your Welotec Industrial Computer (hereinafter referred to as “IPC”).

The IPC is designed for use as a dedicated control, monitoring, and data acquisition unit within the enclosed control cabinet of a machine. Its primary function is to execute specific machine-control software, process operational data, provide human-machine interface (HMI) functionalities, and/or facilitate communication within the industrial automation environment. The IPC is exclusively intended for continuous operation within a controlled industrial setting.

The intended use of the IPC is strictly defined by the following conditions and requirements:

3.1.1 Physical Security and Installation Environment

- Enclosure: The IPC must be permanently installed within a secure, locked control cabinet (e.g., meeting IP54 or higher protection class) that provides adequate protection against dust, moisture, mechanical impact and unauthorized access.
- Controlled Access: Access to the control cabinet and its wiring must be restricted to authorized personnel only. Physical security measures (e.g., key locks, access control systems) are mandatory.
- Environmental Conditions:
  - Temperature: The IPC must operate within the specified ambient temperature and humidity range as outlined in the technical specifications. Adequate ventilation or active cooling within the cabinet must ensure these limits are not exceeded. This includes accounting for the unit’s own thermal dissipation and that of all other components in the cabinet.
  - Vibration and Shock: The IPC must be mounted securely within the cabinet to minimize exposure to excessive vibrations and mechanical shock, adhering to the manufacturer’s specifications.
  - Cleanliness: The inside of the cabinet must be kept free of dust, debris, and contaminants that could impair cooling or lead to electrical shorts.

3.1.2 EMC compliant electrical Installation and Power Supply

This product is designed to meet EMC standards when installed according to the following instructions. Failure to adhere to these instructions may result in the equipment failing to meet compliance standards and can cause interference with other devices. The installer is responsible for ensuring the EMC conformity of the final system.

- Power Supply: The IPC must be connected to a dedicated stable and filtered power supply within the specified voltage range. To ensure operational reliability and meet EMC requirements, the power source must provide adequate filtering against surges, transients, electrical fast transients (EFTs), and conducted RF noise common in industrial environments. An Uninterruptible Power Supply (UPS) is highly recommended to protect further against power fluctuations and outages.

- Wiring: All wiring connecting to the IPC must comply with applicable industrial wiring standards, be properly insulated, strain-relieved, and protected against mechanical damage.
- Grounding: The unit must be properly grounded according to the installation manual, typically via a low-impedance connection to the control cabinet's central grounding point.

3.1.3 Functional Safety

This unit is not certified as a standalone component for functional safety applications (e.g., SIL, PL).

Intended Use: The unit is intended for standard control and monitoring. It must not be used as the sole or primary controller for safety-critical functions (e.g., emergency stops, safety interlocks, light curtains, burner controls).

System Integration: Safety-related control logic must be executed by dedicated, certified safety controllers (e.g., Safety PLC, safety relays). This unit may be used to supervise or monitor a safety system (e.g., for HMI visualization or data logging) via a non-safety-rated communication channel, but it must not be part of the safety-critical control loop. The failure of this unit must not lead to a loss of the primary safety function.

3.1.4 Qualified and Trained Personnel

- Installation, Configuration, and Maintenance: All installation, configuration, maintenance, troubleshooting, and repair activities on the IPC and its connections within the control cabinet must be performed exclusively by qualified, trained, and authorized technical personnel. This personnel must possess proven expertise in electrical systems, IT hardware, and cybersecurity best practices.
- Security Awareness: All personnel interacting with the IPC or the network it is connected to must receive regular training on IT security awareness including password policies and reporting suspicious activities.

3.1.5 Software and Configuration

- Operating System: Only the pre-installed or manufacturer-approved operating system (OS) version may be used. The OS must be regularly updated with security patches provided by the manufacturer or OS vendor, after thorough testing in a non-production environment.
- Secure Configuration: The IPC's operating system, firmware, and installed applications must be configured according to secure hardening guidelines, including disabling unused services, ports, and protocols, and enforcing strong password policies.
- Secure Boot: Where supported Secure Boot must be enabled to prevent the loading of unsigned or malicious bootloaders.

Please refer to the section "Cyber Security" for further details.

3.1.6 Network Segmentation and "Defense in Depth" IT Security Principles

- Network Segmentation: The unit and its control network must be isolated from all other networks (e.g., corporate, guest, public internet) using industrial firewalls and network segmentation. Direct connection to the internet is considered misuse unless done via a secure, managed gateway.
- Defense in Depth: A multi-layered security approach ("Defense in Depth") must be implemented for the entire machine. This includes:
  - Network Security: Industrial Firewalls (e.g., Next-Generation Firewalls) at network boundaries, strict firewall rules (whitelist approach – only allow explicitly required traffic), VLANs for segmentation.
  - System Security: Operating system hardening (minimum services, disabled unnecessary ports), regular security updates, robust antivirus/anti-malware solutions specifically designed for industrial environments, and strong password policies.

  - Application Security: Secure configuration of all industrial applications, disabling default credentials, and ensuring application-level security features are enabled.
  - Data Integrity: Measures to ensure data integrity and availability (e.g., backups, redundant systems where appropriate).
  - Physical Security: see above

- Access Control: Remote access to the IPC (if required) must be strictly controlled, using secure connections, multi-factor authentication, and granular user permissions. Unnecessary remote access functionalities must be disabled.
- Logging and Monitoring: The IPC and connected network devices should implement logging of security-relevant events. Centralized monitoring and alerting systems are recommended for timely detection of anomalies.

3.2 Non-Intended Use

Any use of the IPC that deviates from the conditions described including but not limited to:

- Operation outside the specified environmental limits.
- Operation without a secure, enclosed control cabinet.
- Operation in hazardous locations (e.g., explosive atmospheres) for which the unit is not explicitly certified.
- Installation or maintenance by unqualified personnel.
- Connection to an unfiltered, unstable, or non-grounded power source.
- Direct connection to unsecured corporate networks or the internet without adequate protective measures.
- Installation of unauthorized software or operating systems.
- Bypassing or disabling of security features (e.g., firewall, antivirus, Secure Boot).
- Failure to implement a cyber security management plan (patching, hardening, access control).

is considered non-intended use and may result in:

- Damage to the IPC or the machine.
- Compromised data security and integrity.
- Serious personal injury or death.
- Failure to comply with regulatory requirements.

3.3 Exposed Interfaces and Services

The following interfaces are exposed:

Interface	Comment
LAN 1 … 3	only 2 interfaces in Light-Version
COM 1	BIOS configurable, not available in Light-Version
USB 1 … 4	only 1 interface in Headless-Version
HDMI	not available in Headless-Version
DI / GND	not available in Light-Version
DO / GND	not available in Light-Version
SW / GND	Power Switch

Available services highly depend on Operating System type and version.

3.4 Cyber Security

The flexibility to run common operating systems like Windows and Linux places the full responsibility of cyber security implementation on the system integrator and end-user. The unit is a component that must be integrated into a comprehensive, defense-in-depth security architecture.

The intended use requires the integrator/user to implement, at a minimum, the following:

3.4.1 Use Secure Boot

Secure Boot is a crucial security feature that helps protect your system from malware and unauthorized operating systems during the boot process. It's a component of the Unified Extensible Firmware Interface (UEFI) that ensures only trustworthy software, signed with a digital certificate, loads when your system starts. Without Secure Boot, malicious programs or unsigned operating systems could load unnoticed before the actual operating system, compromising your system's integrity and security.

We highly recommend enabling Secure Boot - please refer to "BIOS" section for further details

3.4.2 Enable Storage Encryption

Storage encryption is a critical security measure that protects your sensitive data by rendering it unreadable to unauthorized parties, even if they gain physical access to your storage device. In today's interconnected world, where devices can be lost, stolen, or compromised, ensuring the confidentiality of your information is paramount.

Windows (using BitLocker with TPM)

Windows' built-in BitLocker encryption leverages the TPM to securely store the encryption key, making the process largely automatic and secure.

- Check TPM Status: Ensure that the TPM chip is enabled in the UEFI/BIOS settings
- Open BitLocker Drive Encryption: Search for "BitLocker" in the Windows search bar and select "Manage BitLocker."
- Turn on BitLocker: Select the drive you wish to encrypt (typically your C: drive) and click "Turn on BitLocker."
- Follow the Wizard: Windows will guide you through the process. Since a TPM is present, it will typically automatically use the TPM to store the encryption key. You will be prompted to save a recovery key (e.g., to a Microsoft account, a USB drive, or print it) – this is crucial in case you ever need to access your data if the TPM is reset or unavailable.
- Start Encryption: The encryption process will begin in the background. You can continue using your computer during this time.

Linux (using LUKS with TPM consideration):

Linux uses LUKS (Linux Unified Key Setup) for full disk encryption. Integrating it with a TPM for automatic unlocking at boot can be more involved than BitLocker but offers similar benefits. This typically involves tools like clevis or systemd-cryptenroll.

- Install Necessary Tools: You'll need cryptsetup for LUKS and potentially tpm2-tools and clevis (or similar TPM integration tools) if you want to bind your LUKS key to the TPM for automatic decryption.
- Encrypt the Drive (during OS Installation or manually):
  - During Installation: Most Linux distributions (e.g., Ubuntu, Fedora) offer an option to "Encrypt the disk" during the installation process. This is the simplest way to set up LUKS.
  - Manually (Post-Installation): If encrypting an existing drive or a secondary drive, you would use cryptsetup luksFormat /dev/sdXy to format the partition for LUKS, followed by cryptsetup luksOpen /dev/sdXy my_encrypted_drive and then creating a filesystem on the opened device.

- Bind LUKS Key to TPM (Optional, for automatic unlock):
  - This is the step that utilizes the TPM. Tools like clevis can be used to “bind” a LUKS passphrase (or a key slot) to the TPM. This allows the system to automatically unlock the encrypted volume at boot if the TPM verifies the system’s integrity.
  - The exact commands vary, but it generally involves generating a new LUKS key slot and then using a TPM-binding tool to store the key in the TPM and configure the system to use it for unlocking.
- Update Boot Configuration: Ensure your bootloader (e.g., GRUB) is configured correctly to handle the encrypted root partition and, if used, to leverage the TPM for unlocking.

For both operating systems, it’s essential to:

- Backup your recovery keys/passphrases: Without them, your data can be permanently lost if there’s a hardware failure or you forget your primary password.
- Understand the implications: While encryption provides strong security, proper handling of keys and adherence to security best practices are still crucial.

3.4.3 Use Strong Passwords

Strong passwords are the first line of defense against unauthorized access. If you want to use password based access it is recommended to:

- Change the factory default password on first login
- Use passwords with a minimum length of 12 characters or more
- Use a combination of uppercase and lowercase letters, numbers, and special characters (e.g., !@#$%^&*)
- Do not use easily guessable patterns, such as sequences (e.g., “123456”, “abcdef”), repeated characters (e.g., “aaaaaa”), or dictionary words

3.4.4 System Hardening:

The operating system (Windows or Linux) must be hardened. This includes:

- Disabling all unused services, applications, and network ports.
- Enforcing strong, unique passwords for all accounts.
- Implementing a least-privilege access model for users and applications.
- Configuring OS-level firewalls (e.g., ufw, Windows Defender Firewall).

3.4.5 Patch Management

A robust process must be in place for testing and deploying security patches for the operating system and all installed third-party applications. This process must be compatible with the operational constraints of the industrial environment.

3.4.6 Endpoint Protection

Where appropriate for the application, industrial-compatible endpoint protection (e.g., anti-malware, application whitelisting, host-based intrusion detection) must be installed, maintained, and kept up-to-date.

3.4.7 Physical Security

Use of the locked control cabinet (see Section 3) to prevent unauthorized physical access and tampering (e.g., via USB ports) is a critical part of the security model.

3.5 Vulnerability Handling

Welotec has implemented a Coordinated Vulnerability Disclosure Policy - please visit the following site for further details: https://welotec.com/pages/coordinated-vulnerability-disclosure-policy

4 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.

5 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

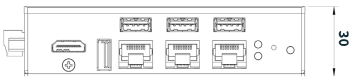
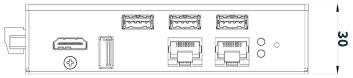
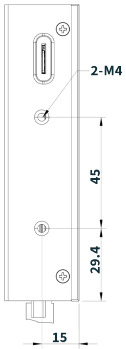
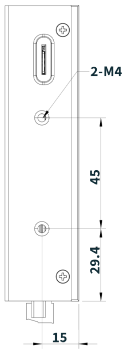
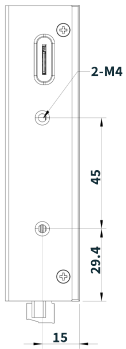
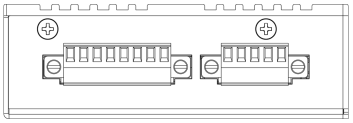
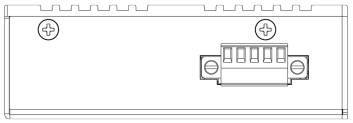
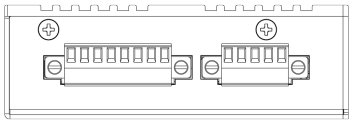
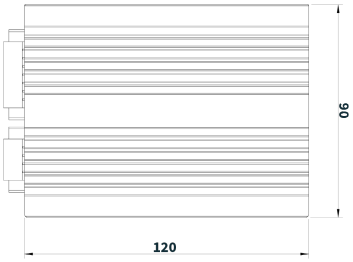
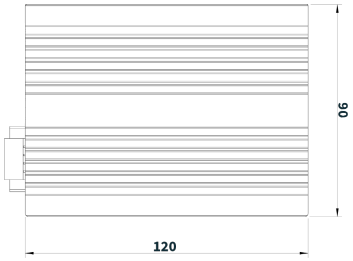
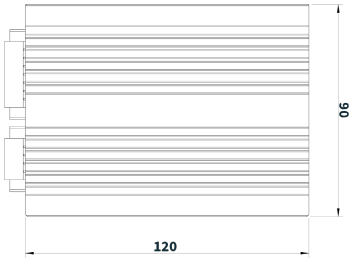
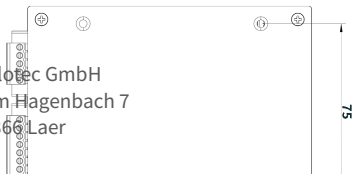
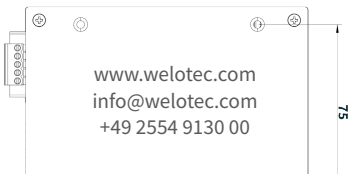
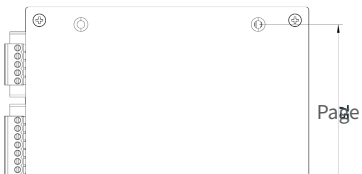
6 System Information

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.

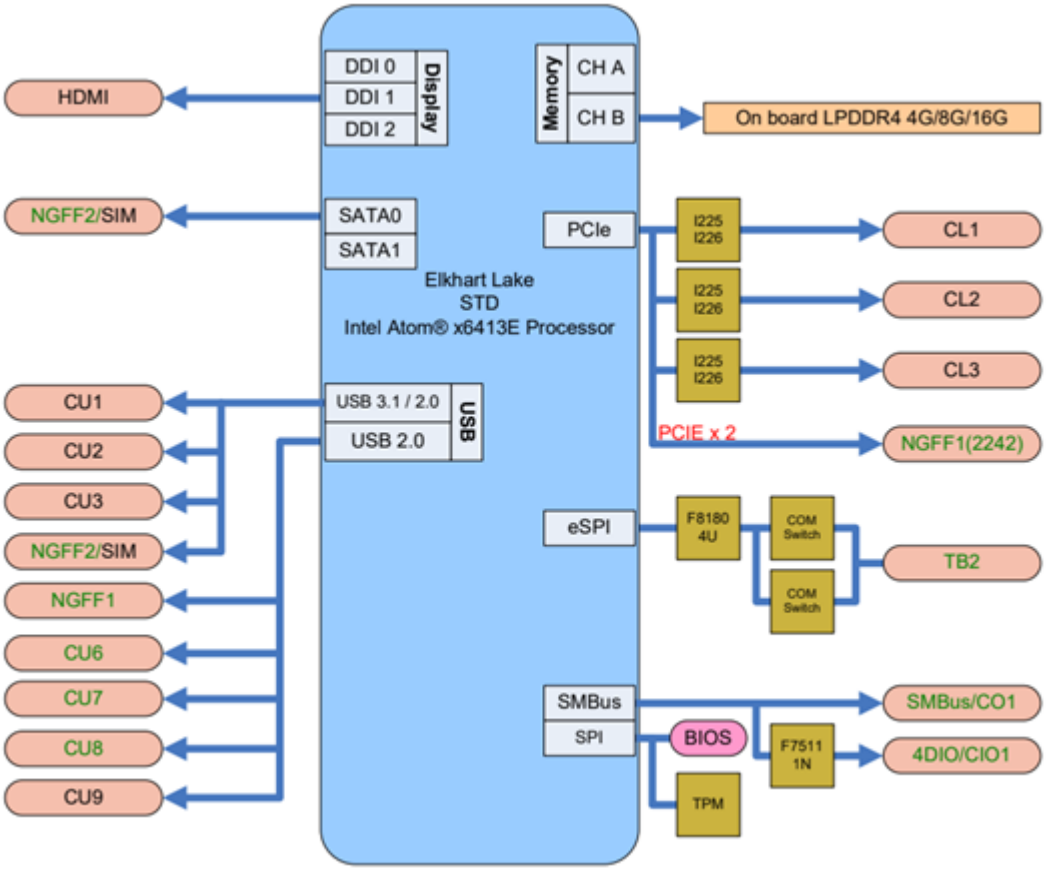
To allow for sufficient heat removal we recommend: 30mm distance on either side of the Arrakis Pico Mk4 when mounted on a DIN-Rail 100mm headroom above the Arrakis Pico Mk4 when mounted horizontally. The heatsink should be on top.

6.1 System Drawing

Arrakis Pico Mk4	Arrakis Pico Mk4 Light	Arrakis Pico Mk4 Headless
		
		
		
		
		

6.2 Mainboard Block Diagram

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



7 Power Supply

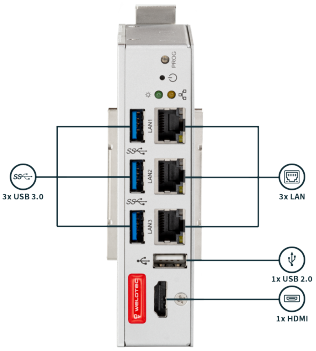
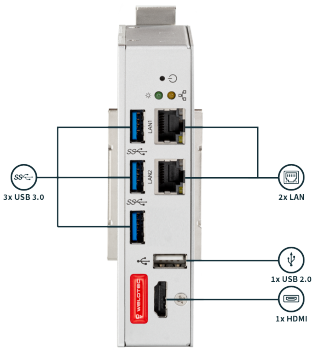
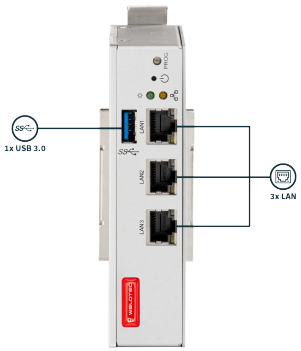
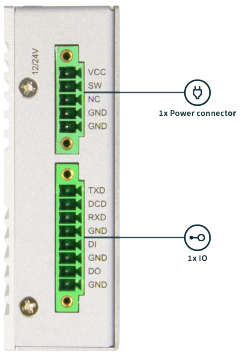
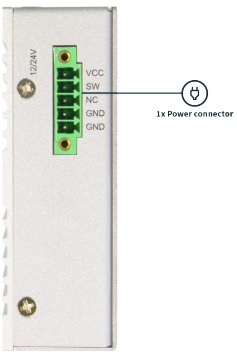
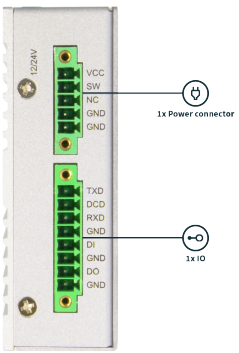
⊠ Please ensure no external voltage is applied to SW! This could cause damage.

Use the terminal block to connect the Arrakis Pico Mk4 to a 12-24V DC power source - please consider “EMC compliant electrical Installation” part in chapter “Intended Use and IT Security Instruction”

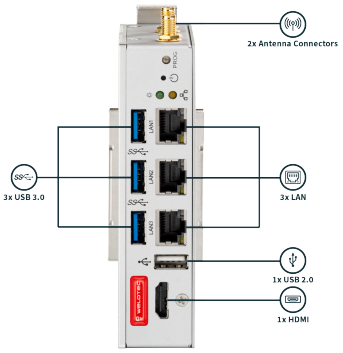
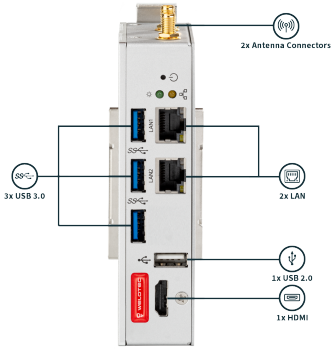
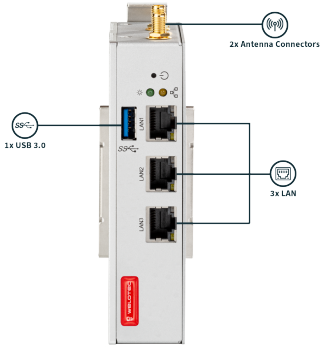
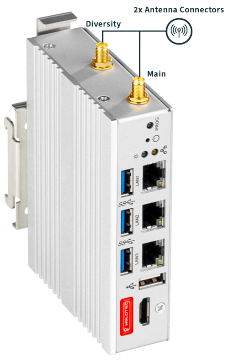
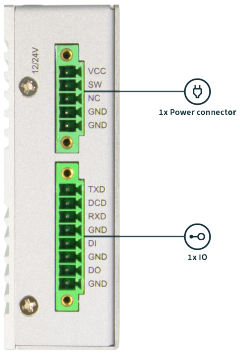
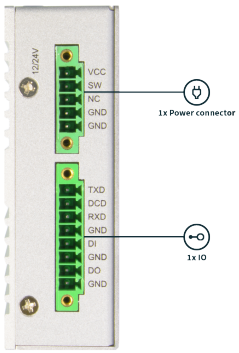
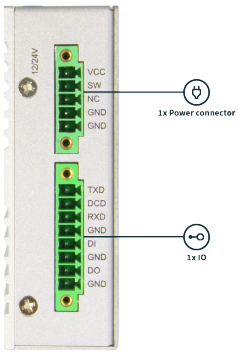
Pin	Description
Pin 0 – VCC	V+ 12-24V
Pin 1 – SW	External power switch
Pin 2 – NC	Not connected
Pin 3 – GND	Ground
Pin 4 – GND	Ground

8 Interfaces and Connections

8.1 Arrakis Pico Mk4 Series

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

8.2 Arrakis Pico Mk4 Series (with optional Radio Module)

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

9 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

10 BIOS

10.1 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).

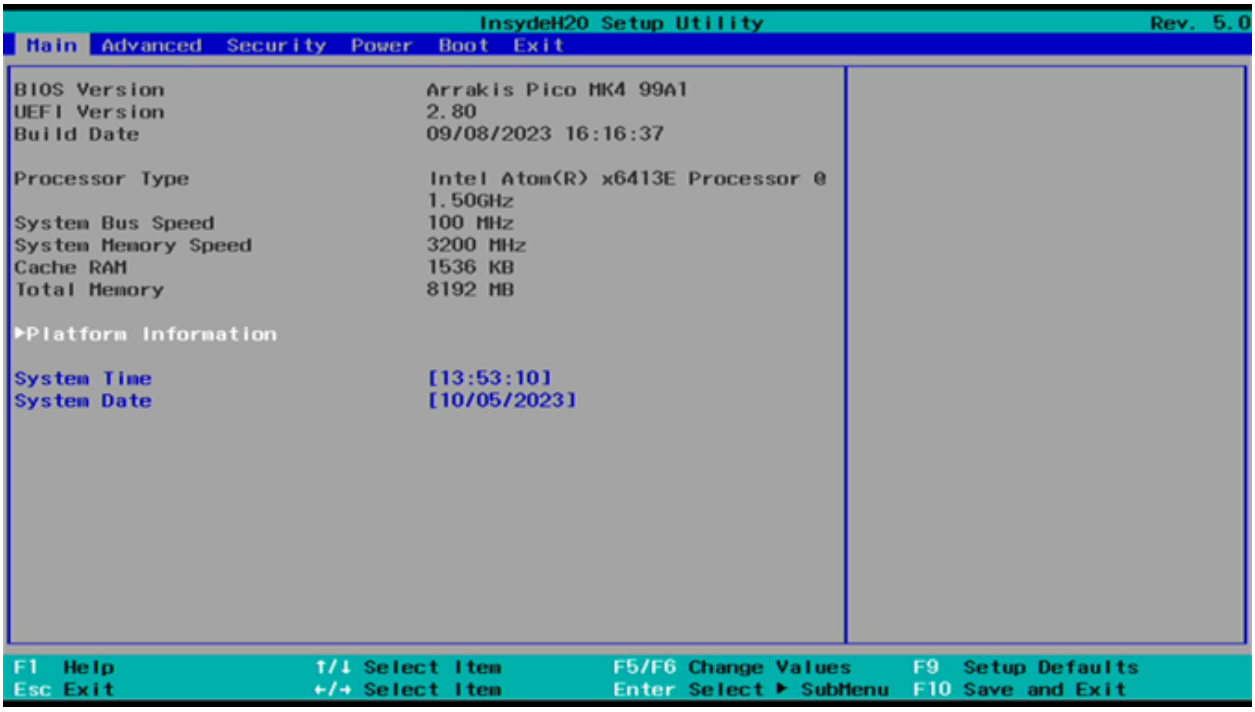
Since the BIOS is the sole channel for hardware and software communication, it is crucial for system stability and optimal performance. In the BIOS Setup main menu, you can see several options. These options will be explained in detail down below. First, let's look at the function keys you may use here:

- 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.

10.2 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.

10.3 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:

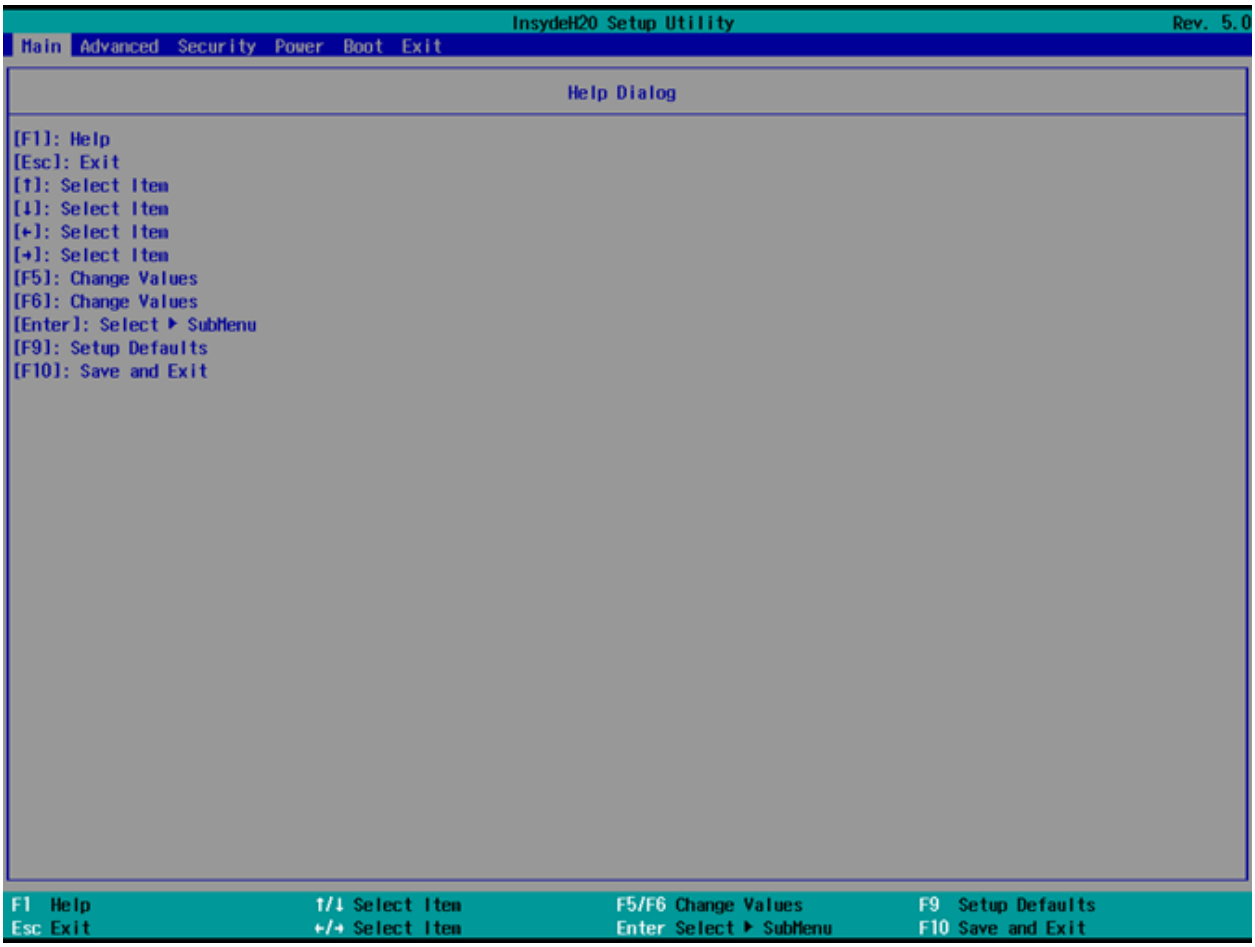
- Press ↔ (left, right) to select the screen.
- Press ↑↓ (up, down) to choose the option you want to confirm or modify.
- Press `Enter` to select.
- Press + or – to modify the BIOS parameters for the active option.
- `F1`: General help.
- `F2`: Previous value.
- `F3`: Optimized defaults.
- `F4`: Save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

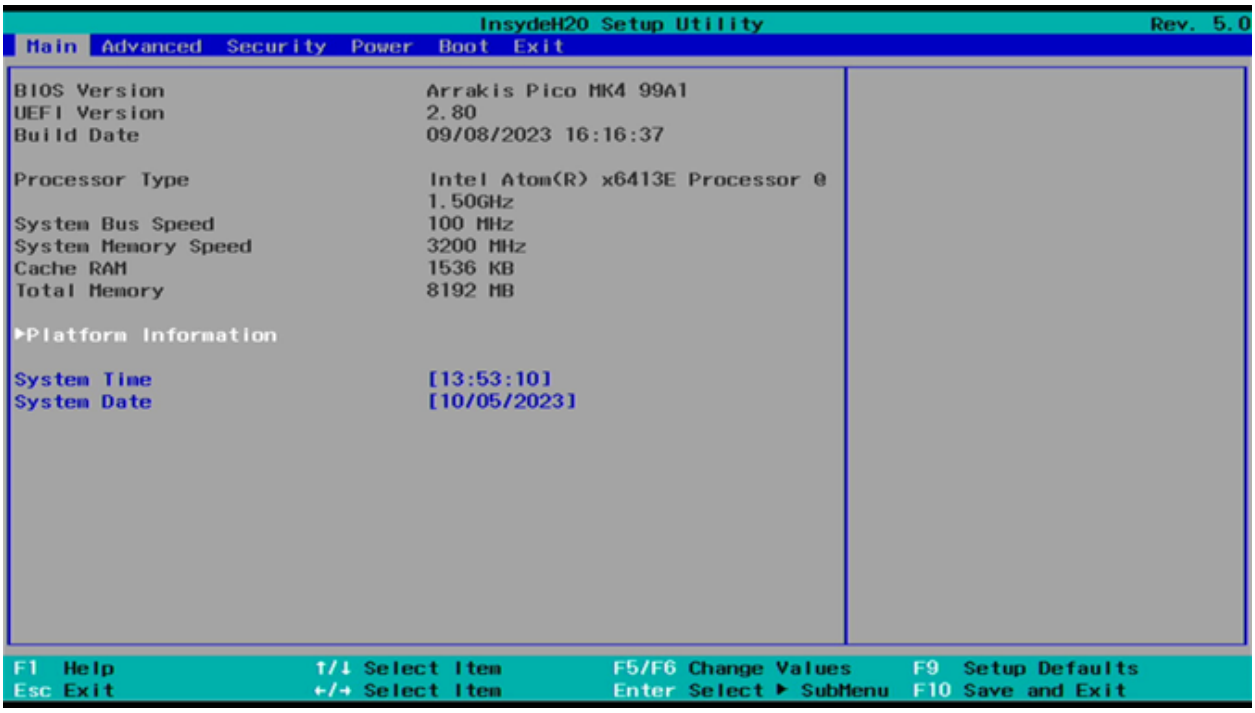
10.4 BIOS Help



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.

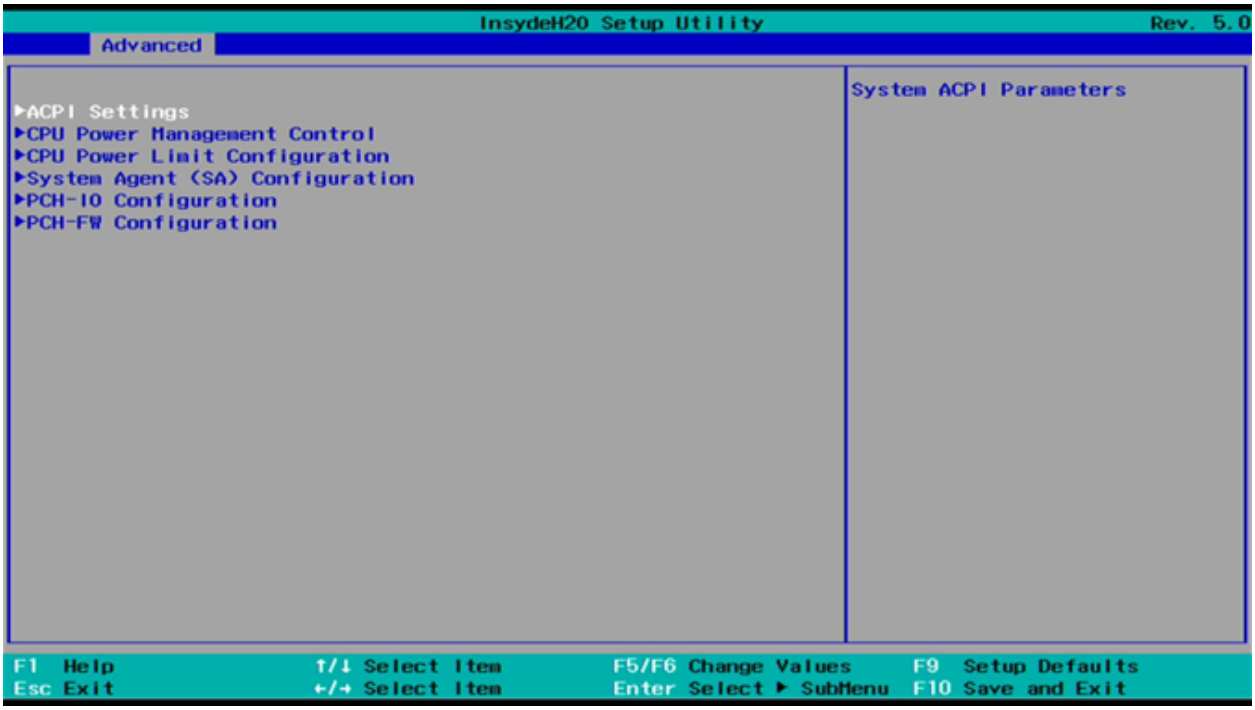
10.5 Main Menu



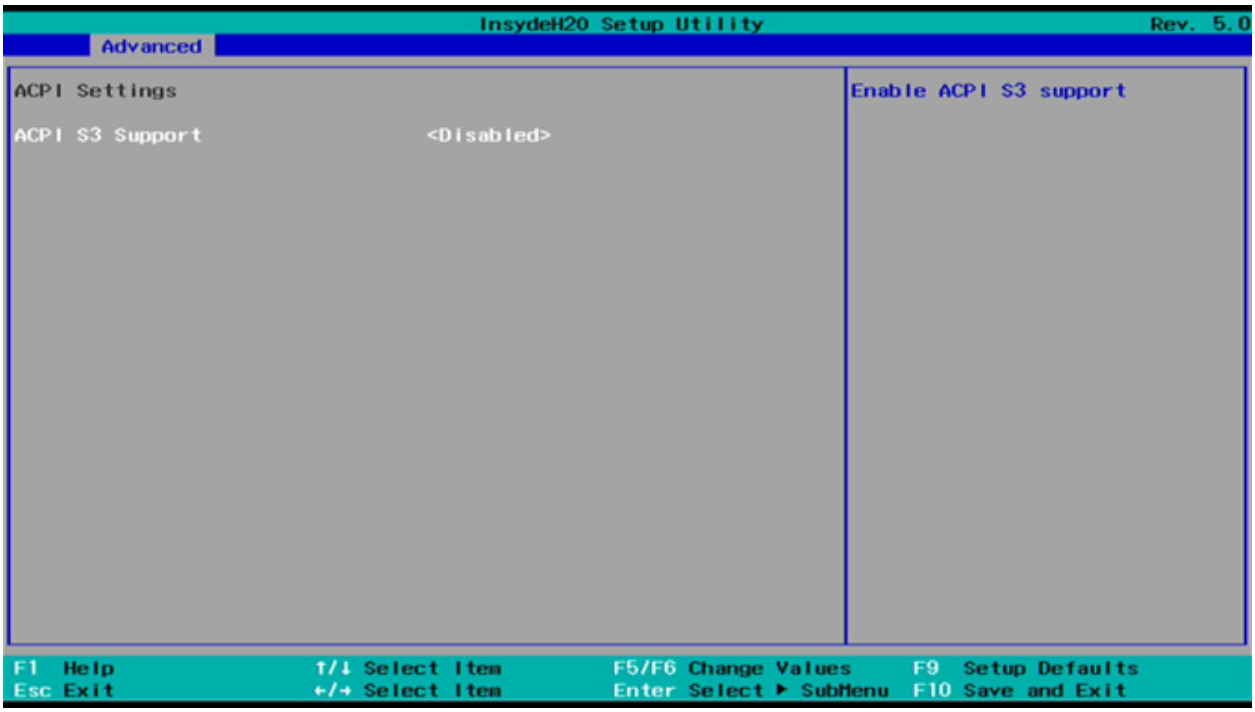
The Main menu screen includes some basic system information. Highlight the item and then use the + or − keys and numerical keyboard keys to select the value you want in each item.

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

10.6 Advanced

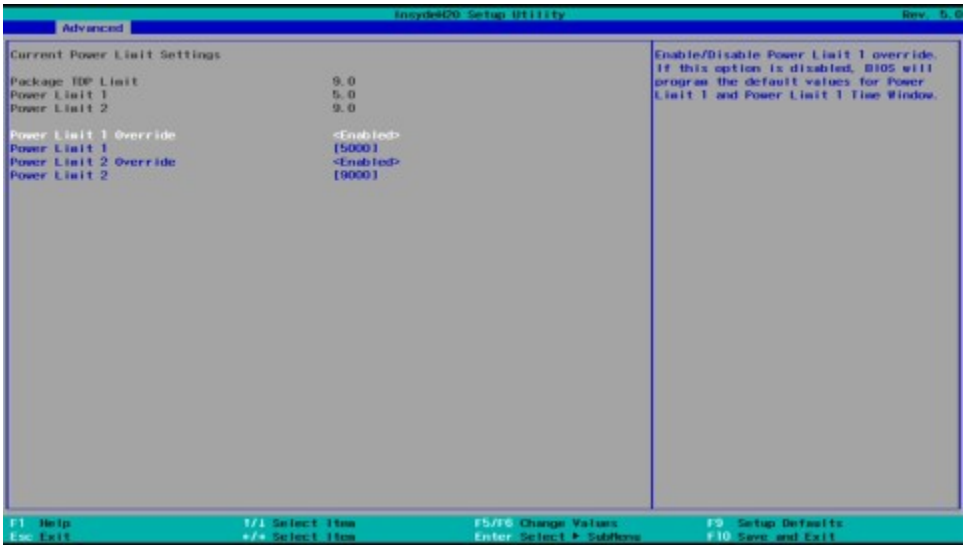


10.6.1 ACPS Settings



Toggle to enable/disable ACPI S3 support.

10.6.2 CPU Power Limit Configuration

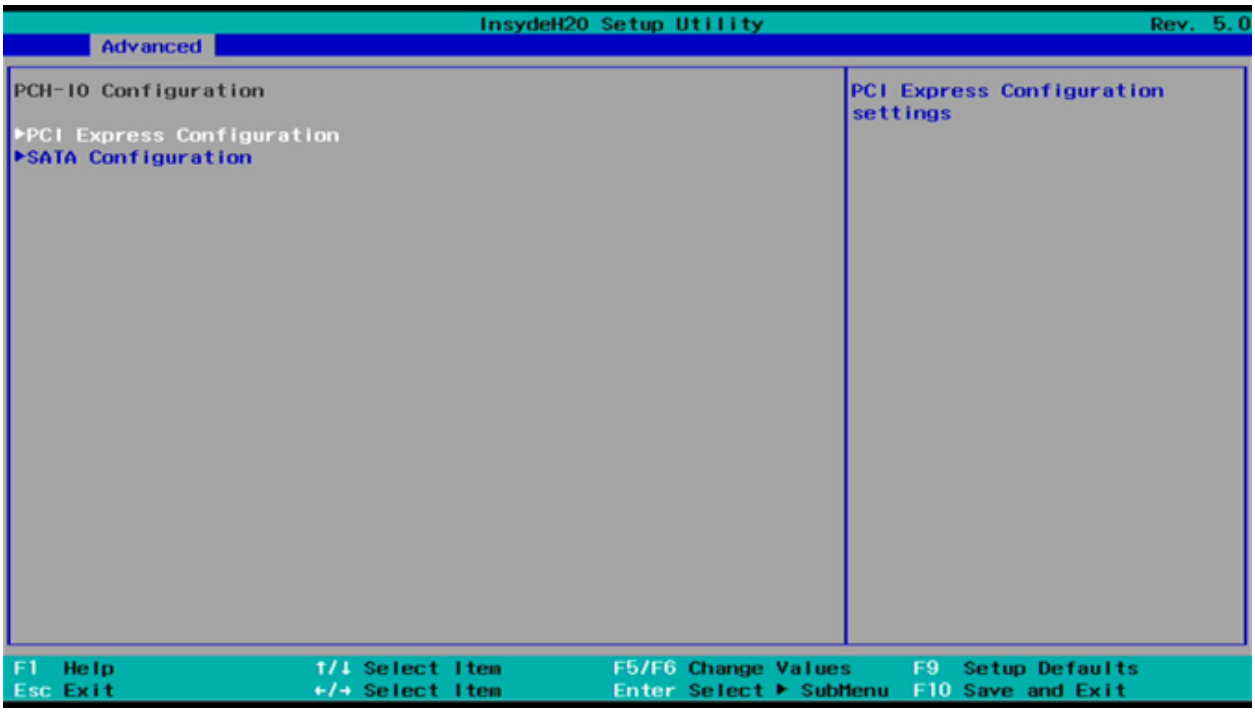


By default, both power limit overrides are enabled. Recommended values are:

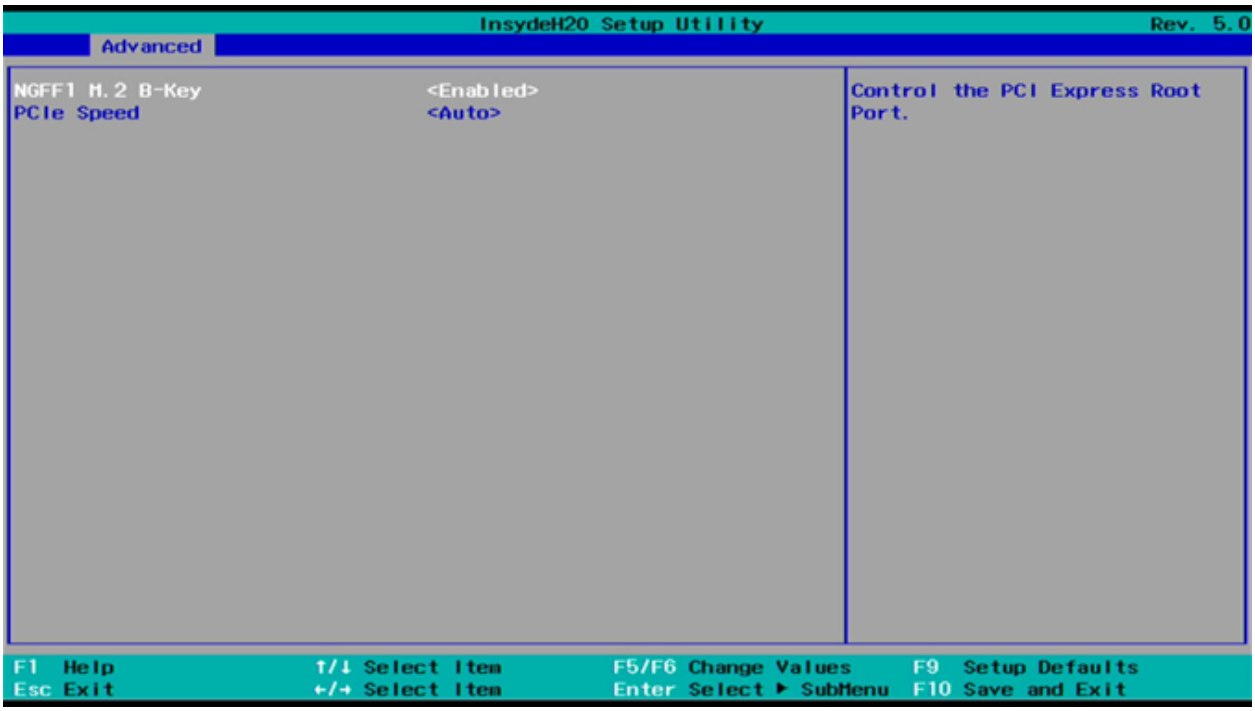
- **Power Limit 1**: 5000
- **Power Limit 2**: 9000

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.

10.6.3 PCH-IO Configuration



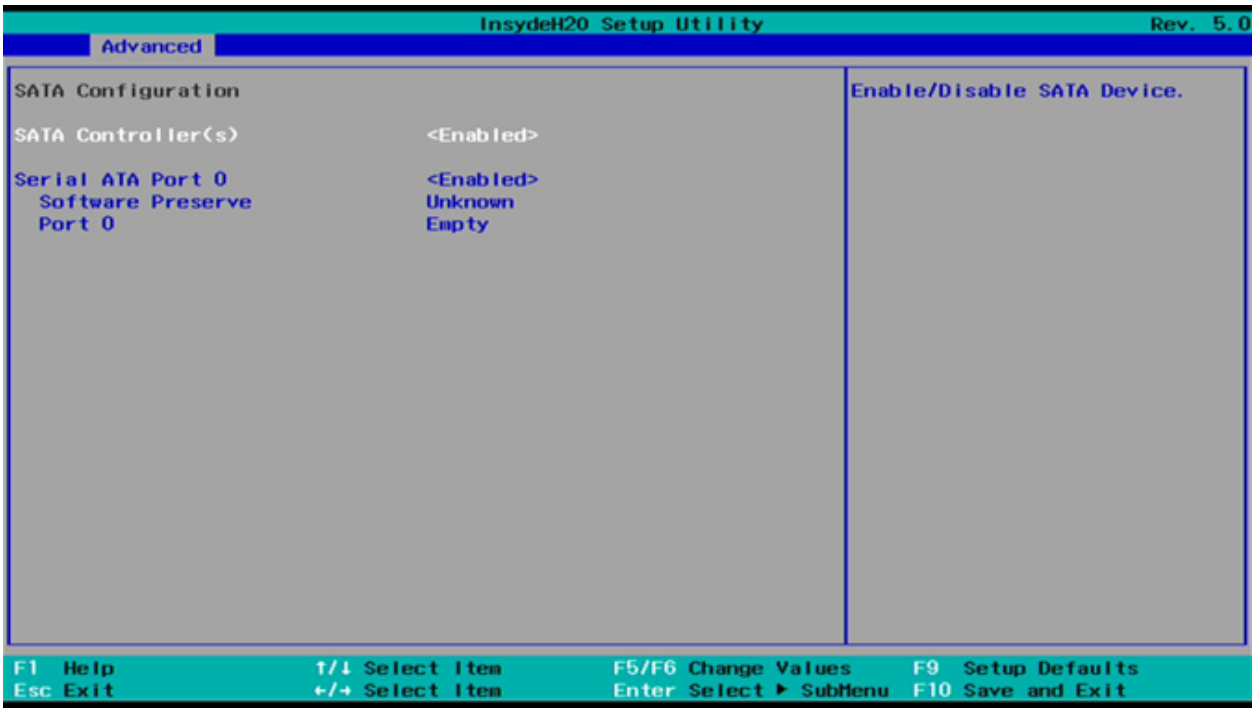
10.6.4 PCI-Express Configuration



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

10.6.5 SATA Configuration

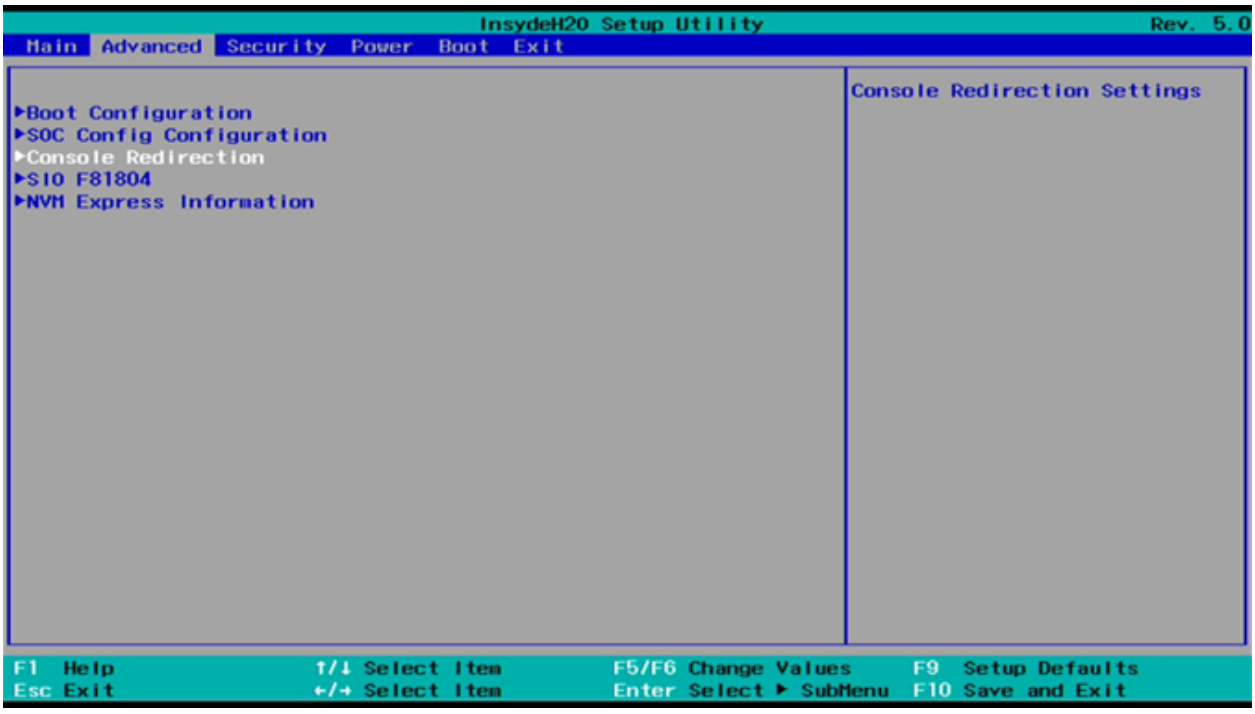


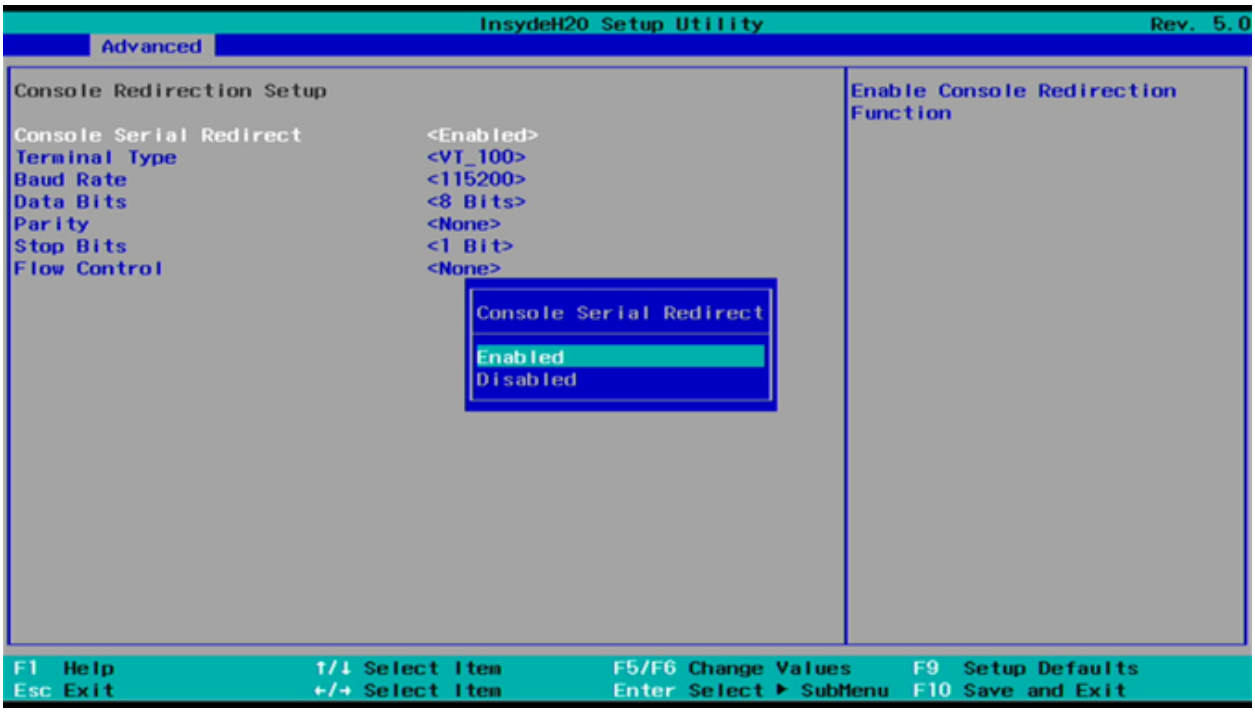
There is usually no need for users to make any changes here. The default settings are:

- **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.

10.6.6 Console Redirect



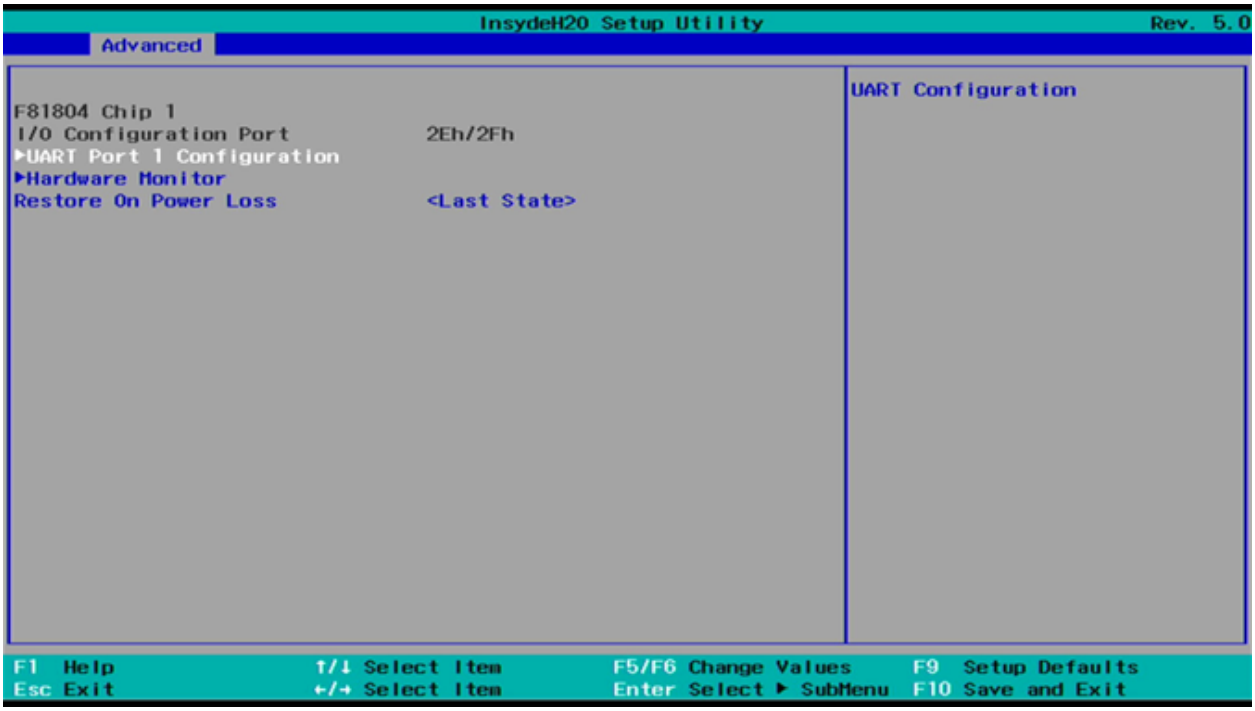


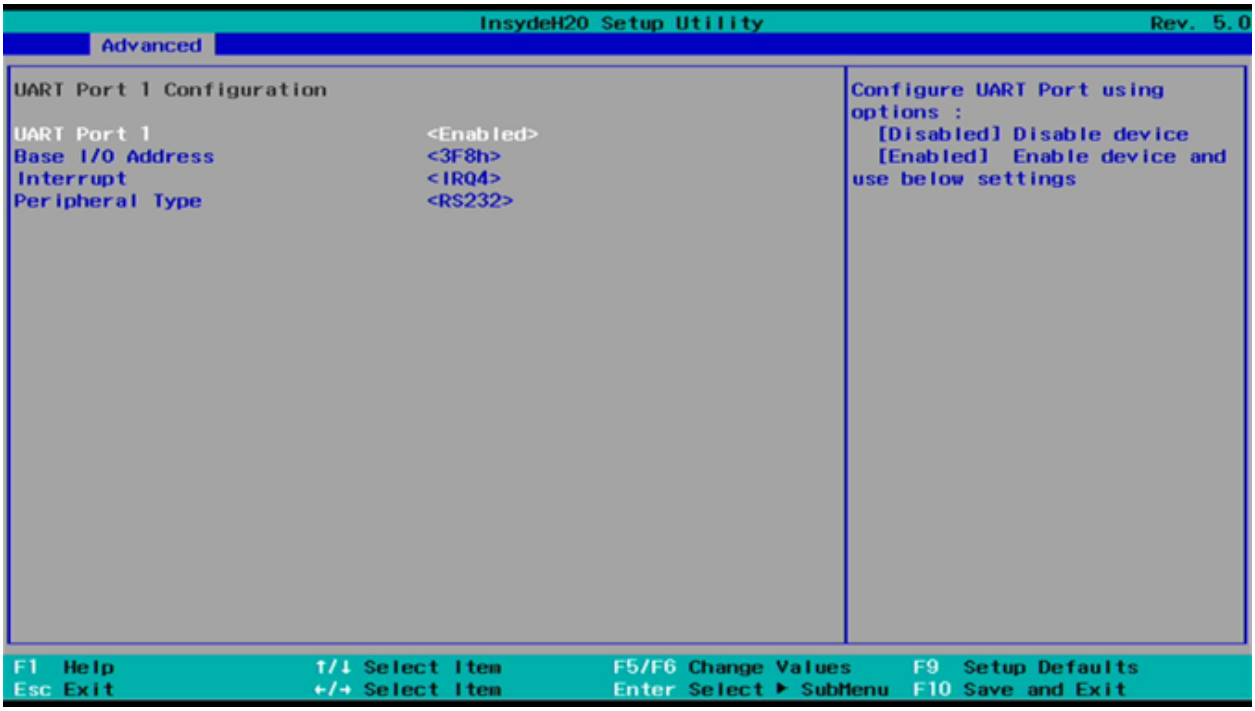
Toggle **Console Serial Redirect** to enable/disable the function.

- **Default**: Enabled

You can configure the terminal settings in this dialogue.

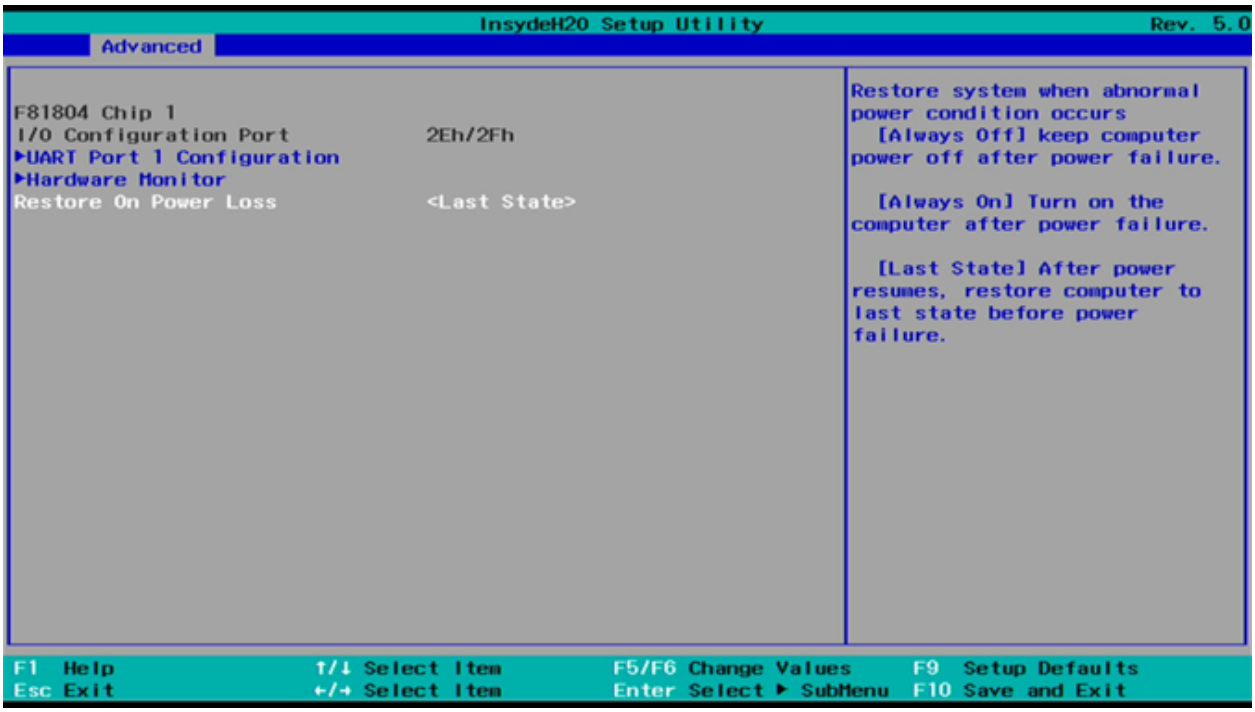
10.7 UART Port 1 Configuration

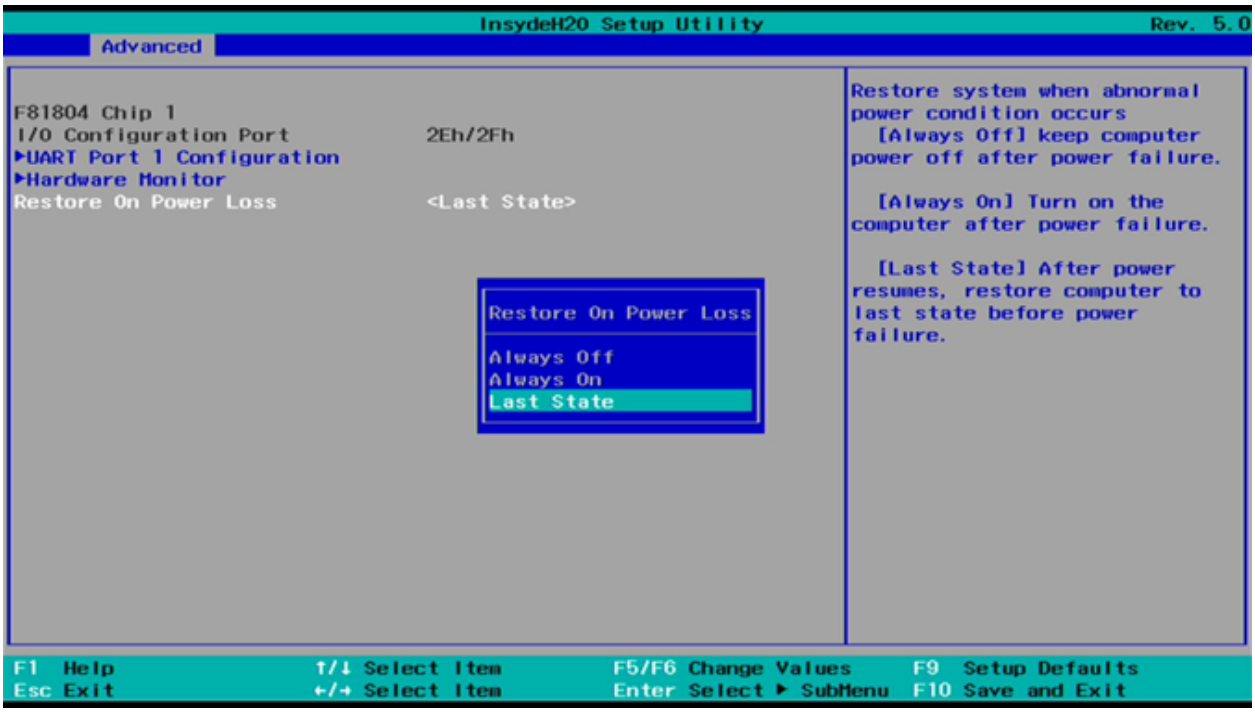




Toggle to enable/disable the serial interface.
Be advised that in **RS232 mode**, only **RX/TX lines** are supported.

10.8 Restore on Power Loss

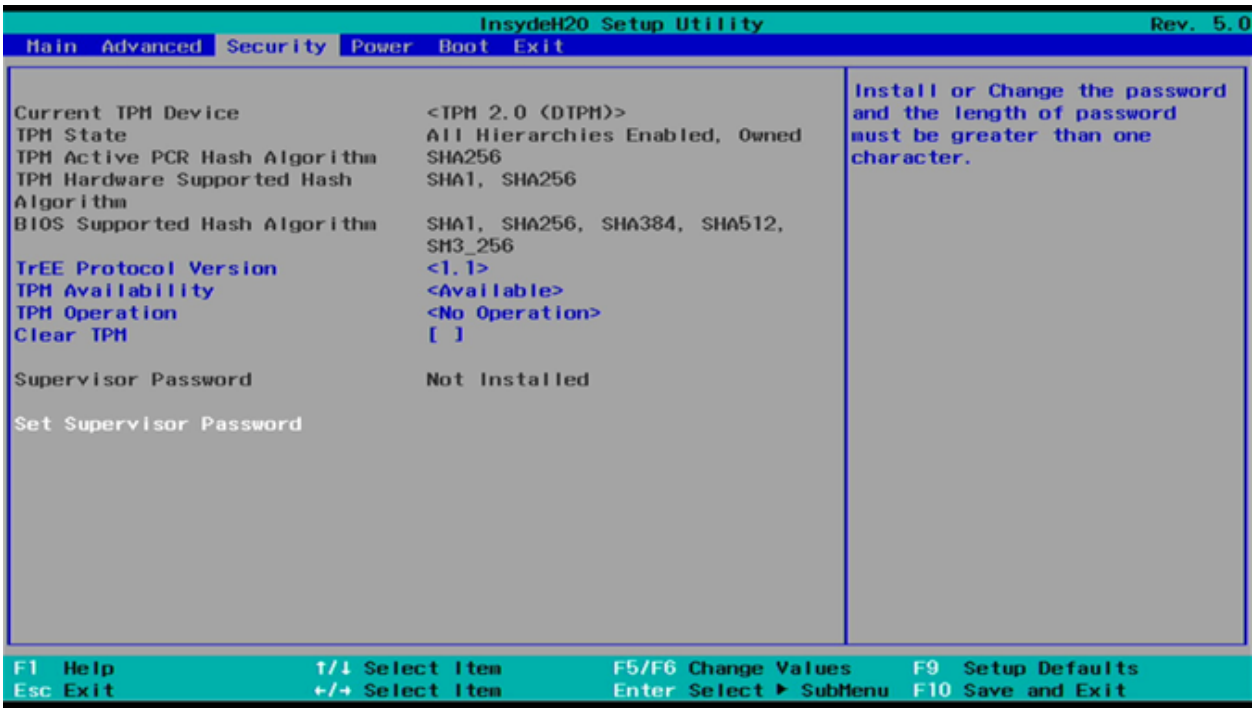




Toggle to configure the desired behavior after power loss. The available options are:

- **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.

10.9 Security



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

10.9.1 Supervisor Password

To set up a **Supervisor password**:

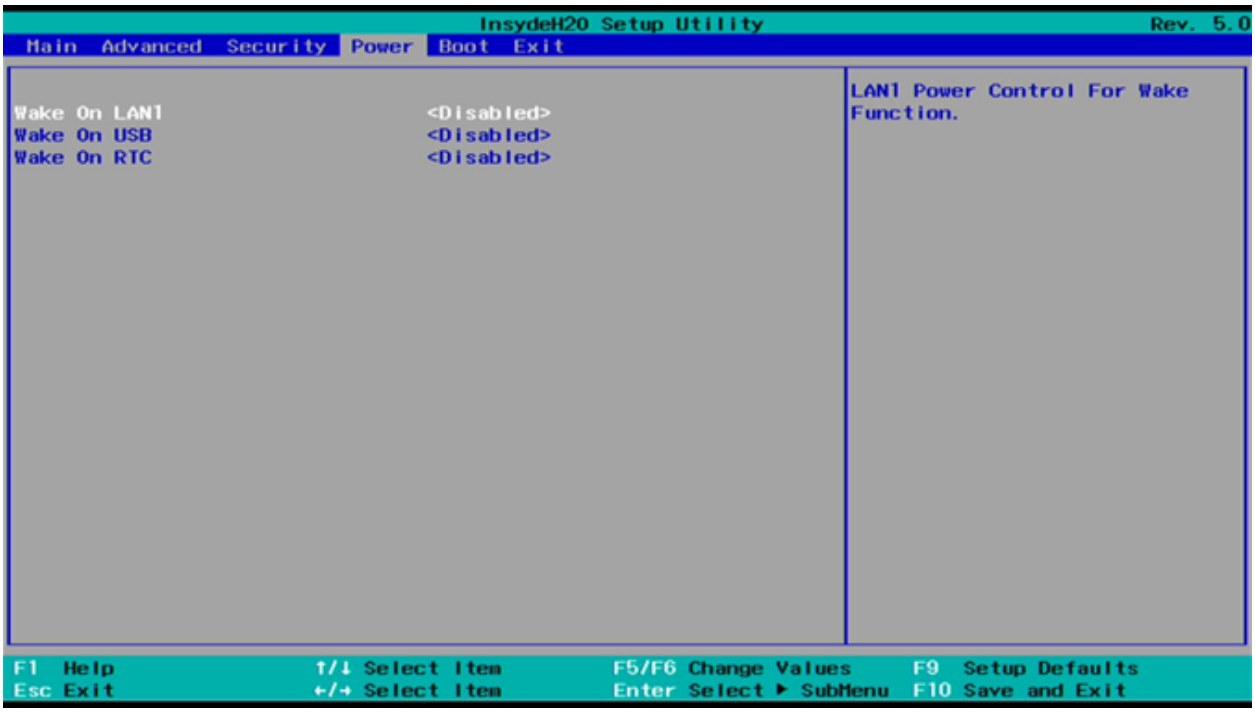
1. Select **Supervisor Password**. A “Create New Password” dialog will pop up.
2. Enter your desired password (must be between 3 and 10 characters).
3. Press `Enter` to submit.

10.9.2 Security Advisory

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

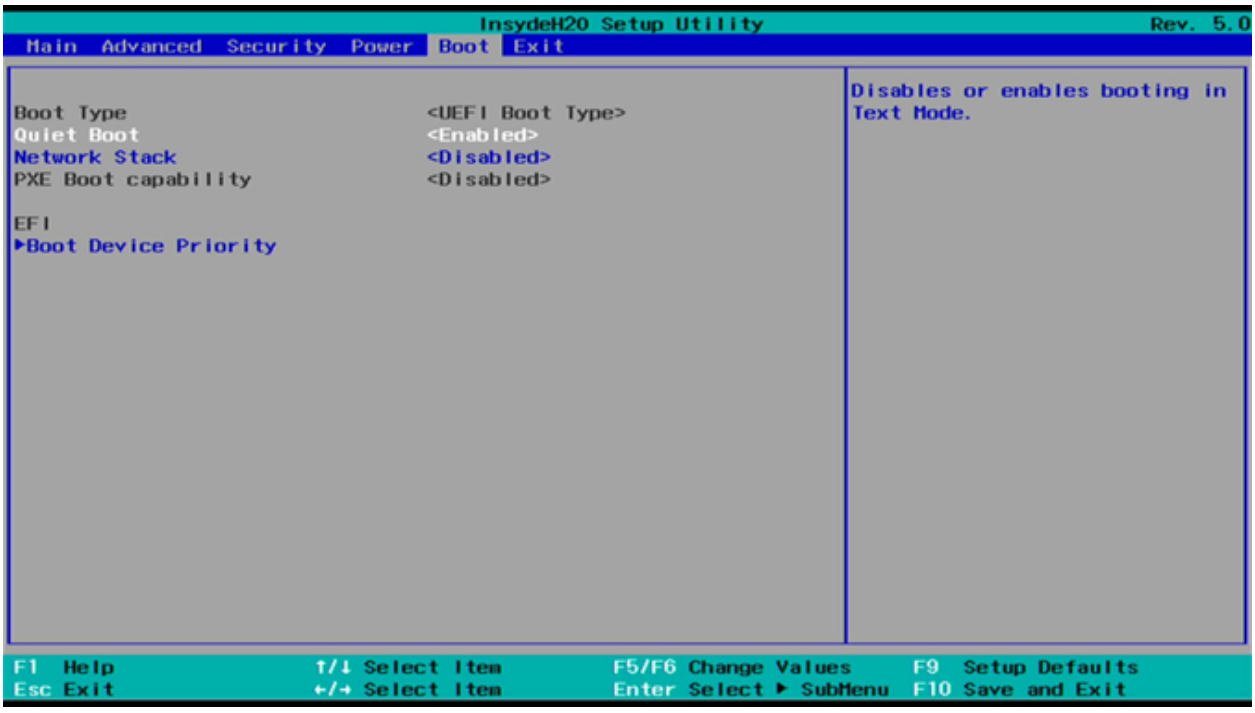
1. Create an **Admin Password** in the **BIOS -> Security** section.
2. Deactivate all unnecessary boot media in the **BIOS -> Boot** section.

10.10 Power



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

10.11 Boot



10.11.1 Boot Type

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

10.11.2 Quiet Boot

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

10.11.3 Network Stack

Enable this option if you need PXE functionality.

- **Default**: Disabled

10.11.4 PXE Boot Capability

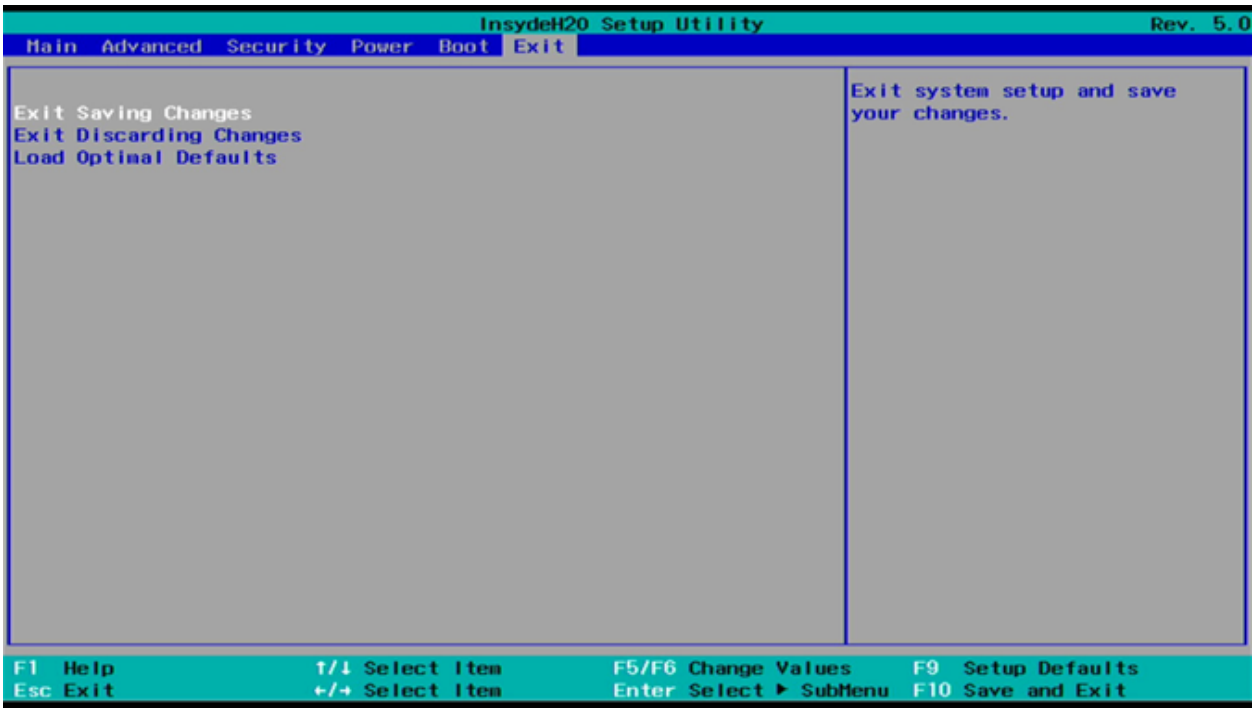
This item determines the protocol used during PXE boot:

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

10.11.5 EFI

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

10.12 Exit



10.12.1 Exit Saving Changes

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

10.12.2 Exit Discarding Changes

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

10.12.3 Load Optimal Defaults

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

11 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.

12 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.

13 Appendix B: F75111N DIO & Watchdog Device

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

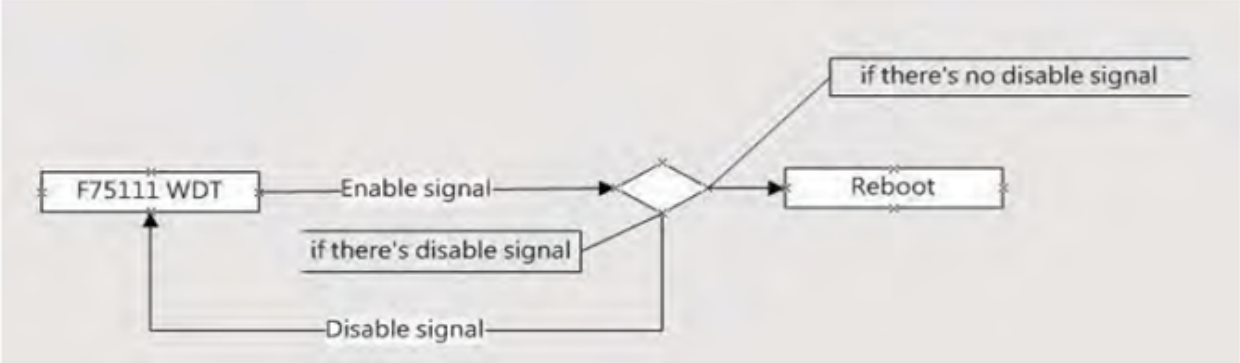
13.1 Watchdog Timer under DOS

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

13.1.1 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.



13.1.2 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:

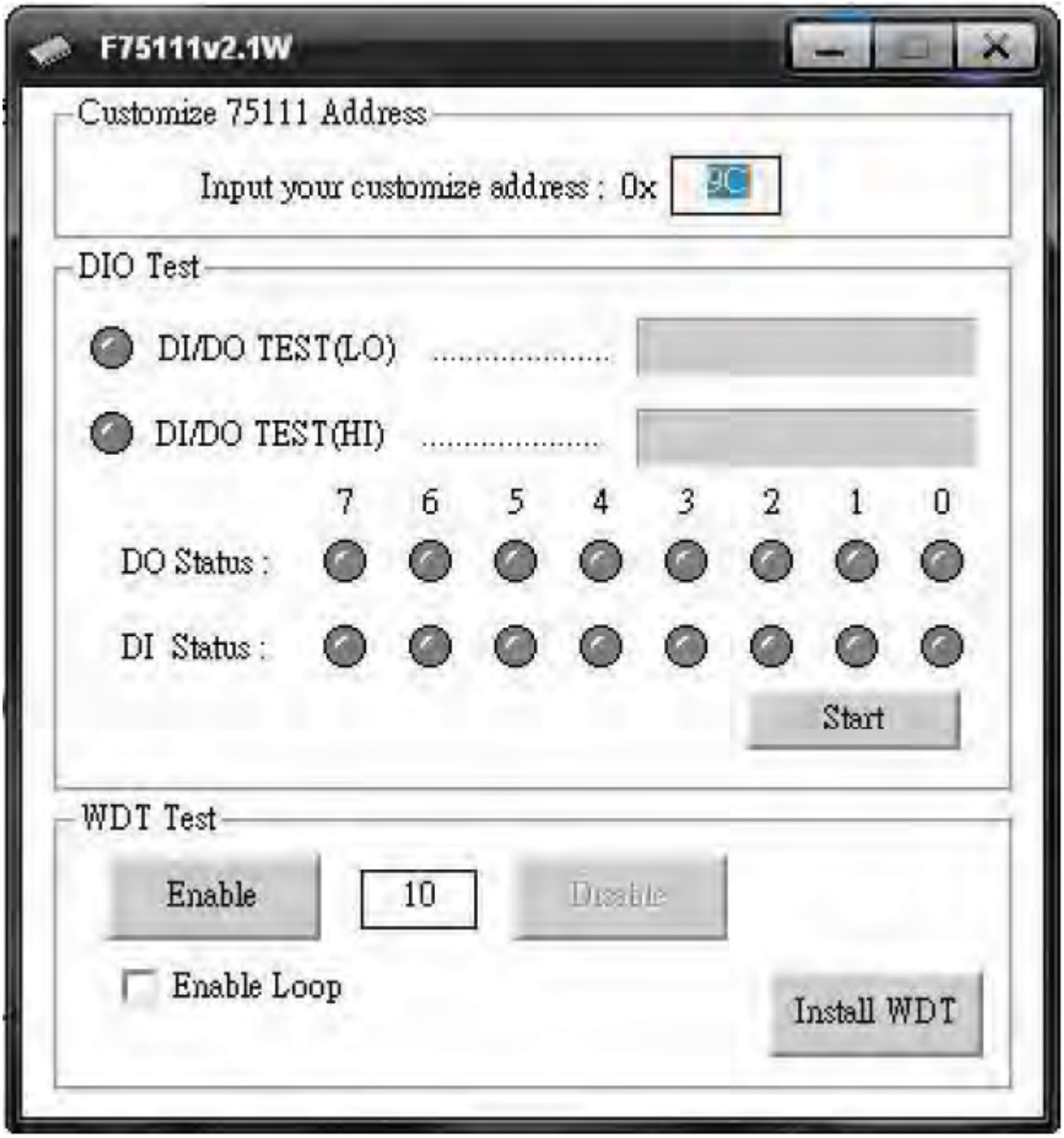
```
void pause(int time) {
    asm mov ah,0h; // Ah = 00 Read System Time Counter
    asm int 1ah;   // Read time from Time Counter and store it in DX register
    asm add dx, time;
    asm mov bx, dx;
label:
    asm int 1ah;
    asm cmp bx, dx;
    asm jne label;
}
```

13.2 Watchdog Timer and DIO under Windows:

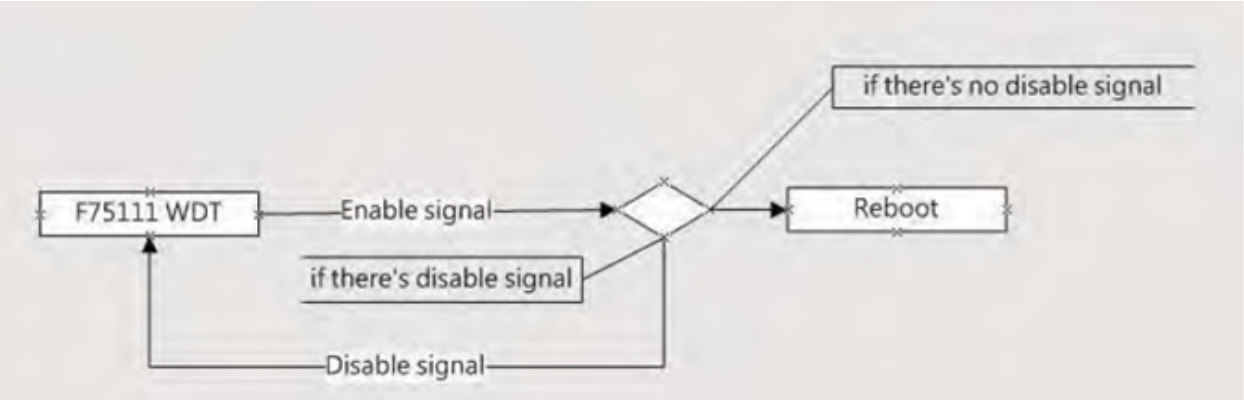
The necessary software resources for programming the watchdog timer can be accessed from the Driver Download section:

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

13.2.1 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.

13.2.2 Introduction:

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

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

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

13.2.4 Enable/Disable WDT:

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

13.2.5 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
}
```

13.2.6 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
}
```

13.2.7 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
}
```

13.2.8 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;
}
```

13.2.9 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
}
```

13.2.10 Disable Watchdog:

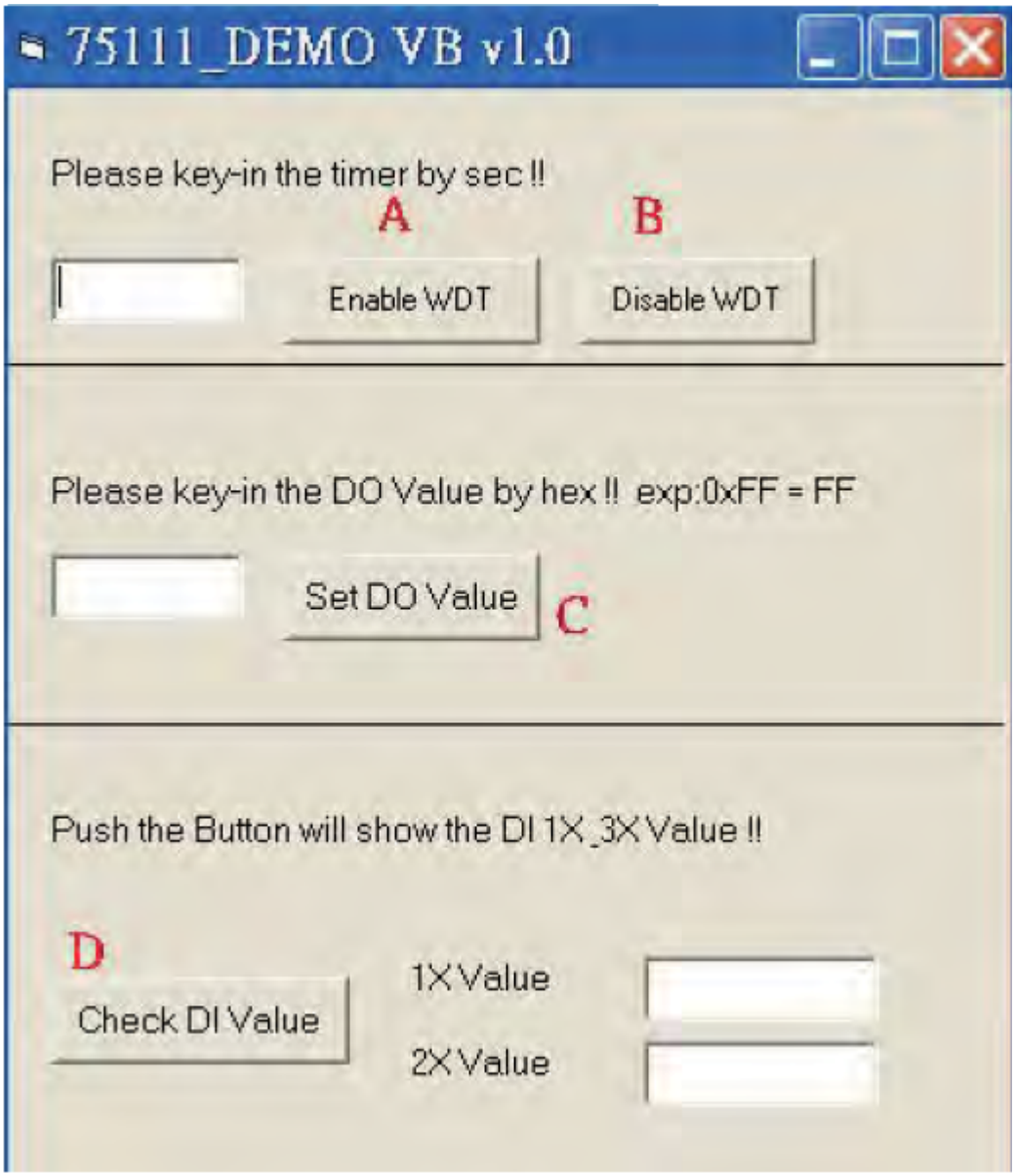
```
void F75111_SetWDTDisable() {
    WriteByte(F75111_INTERNAL_ADDR, WDT_CONFIGURATION, 0x00); // Disable Watchdog
}
```

13.3 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

13.3.1 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.

13.3.2 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
```

13.4 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

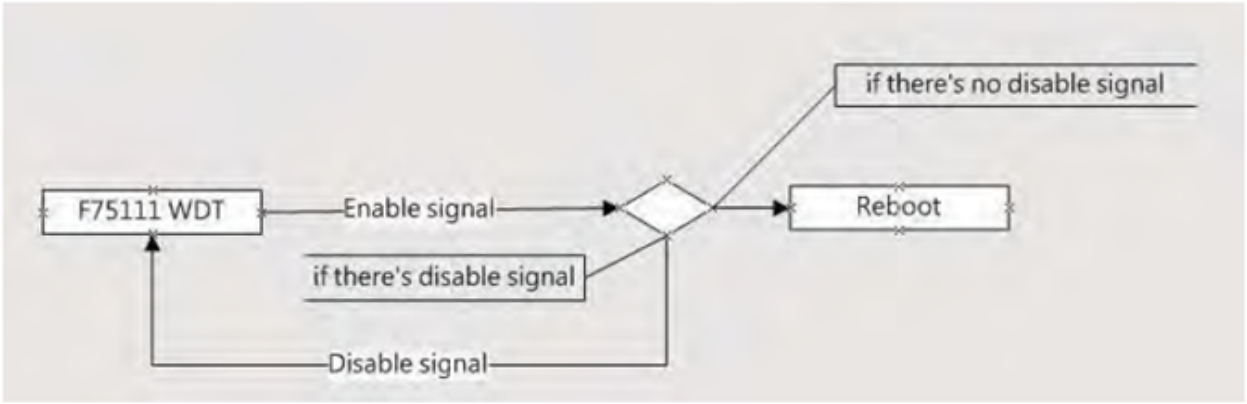
13.4.1 How to Compile the Source Code

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`.

13.4.2 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.

13.4.3 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
}
```