

# Arrakis Pico MK3 Series

Version:  
v1.01

Date:  
12.12.2024



# Contents

<b>1</b>	<b>Copyright</b>	<b>2</b>
<b>2</b>	<b>Regulatory Compliances</b>	<b>3</b>
2.1	CE and UKCA Notice . . . . .	3
2.2	FCC PART 15 VERIFICATION STATEMENT . . . . .	4
2.3	ICES-003 ISSUE 7 VERIFICATION STATEMENT . . . . .	4
<b>3</b>	<b>Safety Instructions</b>	<b>5</b>
<b>4</b>	<b>Product Introduction</b>	<b>6</b>
4.1	About Arrakis Pico MK3 . . . . .	6
4.2	Product Specifications . . . . .	6
<b>5</b>	<b>System Information</b>	<b>8</b>
5.1	System Drawing . . . . .	9
5.2	Mainboard Block Diagram . . . . .	10
<b>6</b>	<b>I/O Ports</b>	<b>11</b>
6.1	Power Input and Power Switch Connector . . . . .	11
6.2	COM and DI/DO Connector . . . . .	11
6.3	2-4 Digital Input / Output / Watchdog Timer . . . . .	12
6.4	Antennae Connector Position . . . . .	12
<b>7</b>	<b>Interfaces and Connections</b>	<b>13</b>
7.1	Arrakis Pico Mk3 Series . . . . .	14
<b>8</b>	<b>Radio Modules (only relevant with optional LTE/WiFi Modules)</b>	<b>15</b>
8.1	LTE . . . . .	15
8.2	UMTS . . . . .	15
<b>9</b>	<b>BIOS</b>	<b>16</b>
9.1	Introduction . . . . .	16
9.2	Enter BIOS . . . . .	16
9.3	BIOS Menu and Function Keys . . . . .	17
9.4	BIOS Help . . . . .	18
9.5	Main Menu . . . . .	19
9.6	Advanced . . . . .	20
9.7	Security . . . . .	27
9.8	Power . . . . .	28
9.9	Boot . . . . .	29
9.10	Exit . . . . .	30
<b>10</b>	<b>Driver Installation</b>	<b>31</b>
<b>11</b>	<b>Appendix A: Power Consumption</b>	<b>32</b>
<b>12</b>	<b>Appendix B: F75111N DIO &amp; Watchdog Device</b>	<b>33</b>
12.1	Watchdog Timer under DOS . . . . .	33
12.2	Watchdog Timer and DIO under Windows: . . . . .	34
12.3	IO Device: F75111 VB6 under Windows . . . . .	38
12.4	Watchdog Timer and DIO under Linux . . . . .	40

# 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

This device complies with the requirements of the CE directive and UKCA regulations.

**Low Voltage Directive 2014/35/EU + Electrical Equipment Safety Regulations 2016 (SI 2016 No 1101)**

- EN 62368-1:2014

**EMC Directive 2014/30/EU + Electromagnetic Compatibility Regulations 2016**

- BS EN 50121-4:2016+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 50121-4:2016+A1:2019/IEC 62236-4:2018
- EN 61000-6-4:2007+A1:2011/IEC 61000-6-4:2010
- EN 61000-4-2:2009 and IEC 61000-4-2:2008
- EN IEC 61000-4-3:2020 and IEC 61000-4-3:2020
- EN 61000-4-4:2012 and IEC 61000-4-4:2012
- EN 61000-4-5:2014+A1:2017 and IEC 61000-4-5:2014+A1:2017
- EN 61000-4-6:2014+AC:2015 and IEC 61000-4-6:2013
- EN 61000-4-8:2010 and IEC 61000-4-8:2010 and IEC 61000-4-8:2009
- EN 55032:2015+AC: 2016
- EN 55035:2017
- EN 61000-3-2:2014
- EN 61000-3-3:2013
- EN 61000-4-3:2006+A1
- EN 61000-4-11:2004+A1:2017

**RoHS 2 Directive 2011/65/EU & 2015/863/EU + RoHS 2 Directive 2020 No. 1647**

- RoHS 2 Directive 2011/65/EU & 2015/863/EU
  - Exemption(s) used:
    - \* 6a, 6b, 6c



## 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:

- 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

# 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 MK3

Thank you for choosing Welotec Arrakis Pico MK3. The Arrakis Pico MK3 is an ultra-compact, fanless embedded system designed for industrial computing applications. It features industrial-grade components, making it suitable for operation in harsh and critical environments.

Powered by a 1.6/1.8 GHz Quad-core Intel Atom® Processor E3940, the standard version comes with 8GB LP-DDR3 Memory on board. The system supports a 4K resolution through an HDMI port and features M.2 NVMe storage, an additional M.2 socket, and an external Nano SIM Socket. It includes three USB 3.0 ports, one USB 2.0 port, three Gigabit-LAN ports, one Digital In and Out each, support for 4G/5G WWLAN, and one COM port.

The quick mounting design allows for both wall mount and DIN rail mount options. Its aluminum housing and dust-proof design make the Arrakis Pico MK3 an ideal choice for industrial environments.

## 4.2 Product Specifications

## 4.2.1 Technical Details

Feature	Specification	Details
<b>Processor</b>	CPU	Intel Atom® Quad-core E3940, 1.6/1.8 GHz (Standard)
<b>Memory</b>	RAM	8GB LP-DDR3
<b>Display</b>	Max. Resolution	HDMI, up to 4K resolution
<b>Storage Options</b>		Full-size M.2 2242 bay NVMe
<b>I/O Ports</b>	HDMI	1 port
	LAN	3 RJ45 Gigabit Ethernet ports
	USB 3.0	3 ports
	USB 2.0	1 port
	Serial Ports	1 RS232/422/485 (TX/RX only)
	Digital I/O	1 DI, 1 DO
	<b>Networking</b>	Ethernet
	WLAN	Optional, via USB/PCIe
	WWAN	Optional 4G/5G via USB
<b>Expansion</b>	SIM Slot	1 push-push type Nano-SIM slot
<b>Additional</b>	Watchdog Timer	Programmable from 1 to 255 seconds
<b>Environmental</b>	Operating Temperature	-20° to 60° C
	Storage Temperature	-20° to 80° C
	Humidity	5% to 95% non-condensing
<b>Power</b>	Supply	12-24V DC, 4-pin terminal block and DC jack
	Adapter	Optional 60W, 12V/5A external desktop power adapter, 40W, 24V/1.7A DIN rail power adapter, CR1220 CMOS battery
<b>Mounting</b>	Options	Optional DIN Rail Mounting Kits
<b>Operating System</b>	Compatibility	All Windows 10 versions, Ubuntu Linux, others upon request
<b>Physical Build</b>	Material/Color	Aluminum / Steel, Silver
	Dimensions	130 x 90 x 30 mm
<b>Compliance</b>	Regulatory	CE/FCC



## 5 System Information

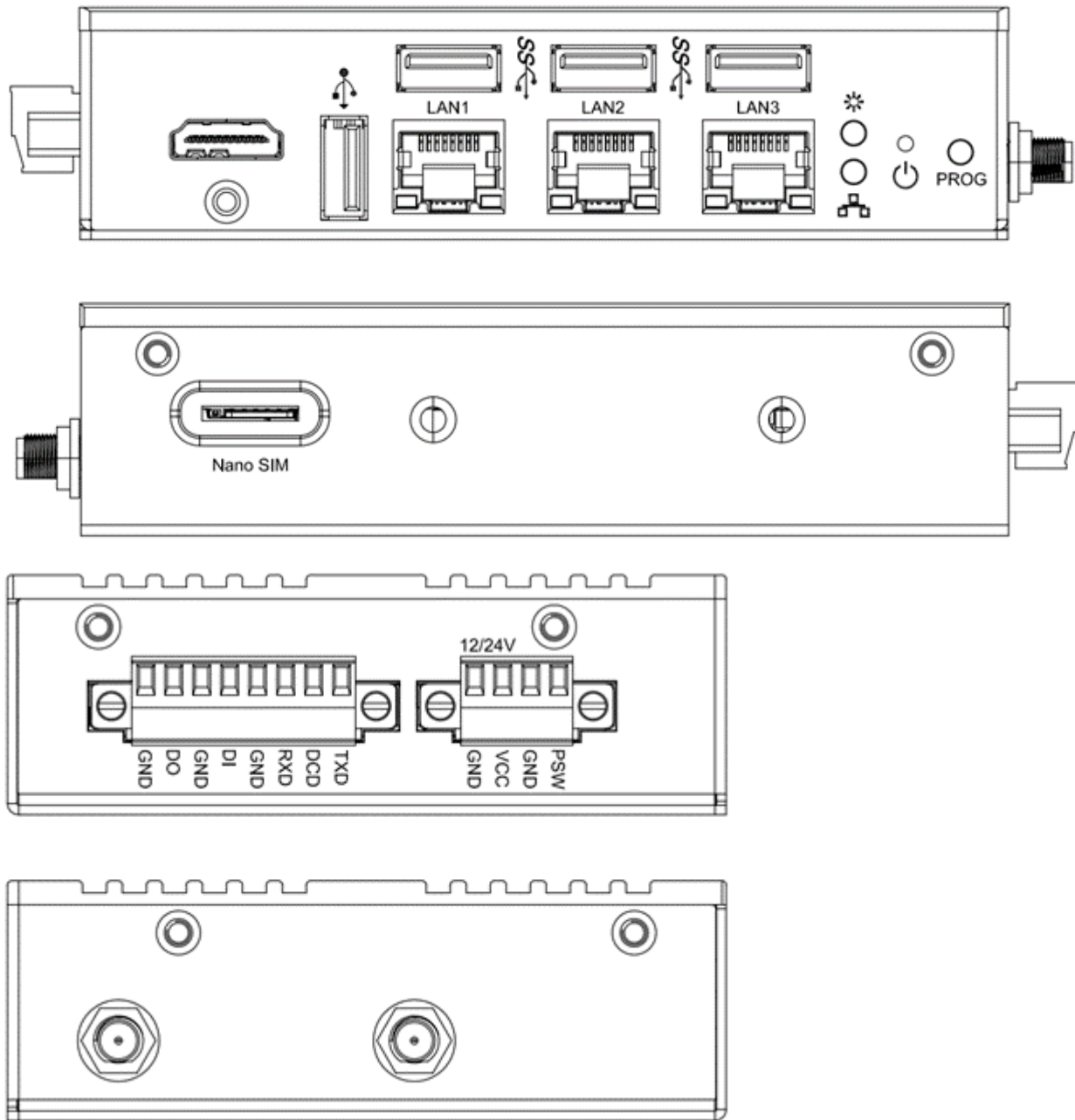
**Attention:** When opening the chassis make sure to slide the chassis top to the rear. Lifting the top up may shear the SIM Slot from the PCB.



Being a powerful, yet small fanless system, the Arrakis Pico Mk3 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 Mk3 when mounted on a DIN-Rail 100mm headroom above the Arrakis Pico Mk3 when mounted horizontally. The heatsink should be on top.

## 5.1 System Drawing

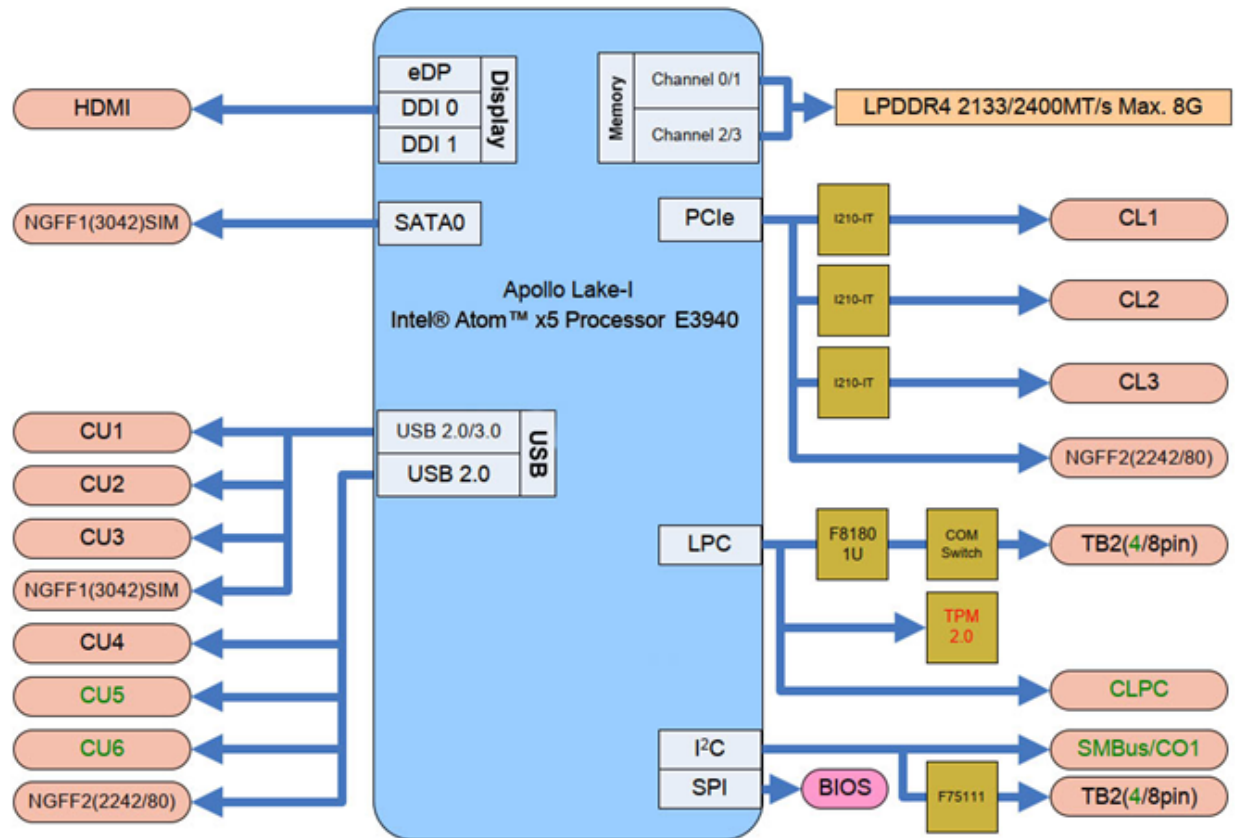


**Attention: Do not connect Voltage to the PSW Pin on the Power Connector!**

Power Connector Pin-Out	Description
GND	Ground
VCC	12-24V
GND	Ground
PSW	External Power Switch Connector

## 5.2 Mainboard Block Diagram

This block diagram illustrates the relationships among all interfaces and modules on the mainboard.



# 6 I/O Ports

## 6.1 Power Input and Power Switch Connector

Do not connect Voltage to the PSW Pin on the Power Connector!

Power Connector Pin-Out	
GND	Ground
VCC	12-24V
GND	Ground
PSW	External Power Switch Connector

The External Power Switch Connector can be used to connect an additional On/Off Switch to the Arrakis Pico MK3.

## 6.2 COM and DI/DO Connector

COM1 can be set to RS232 or RS485 through the BIOS, with the default being RS232.

### 6.2.1 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	

### 6.2.2 TB2-TB 8pin connector (3.5mm) (RS485 Mode)

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

## 6.3 2-4 Digital Input / Output / Watchdog Timer

### 6.3.1 TB2-TB 8pin connector (3.5mm)

PIN NO.	DESCRIPTION	PIN NO.	DESCRIPTION
1		5	DI
2		6	GND
3		7	DO
4		8	GND

**Note:**

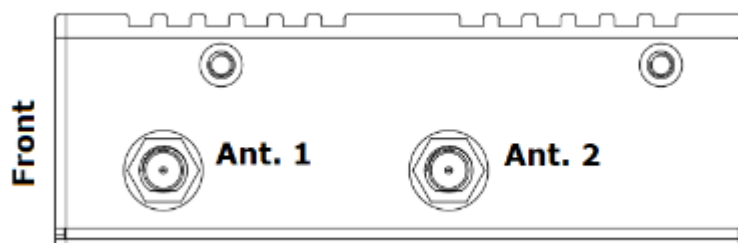
1. F75111N-1 I2C bus address is 0x9C.
2. DI input range is +12V or +24V.
3. DI pin default pull-up is 10K $\Omega$  to +3.3V.
4. DO output voltage rail is from DC-in (+12V or +24V) with a current limit of max 2A.

For F75111N I2C watchdog timer device:

- Input low Voltage (VIL): +0.8V Max
- Input High Voltage (VIH): +2V Min
- Output low Current (IOL): 10mA (Min) at VOL=0.4V
- Output High Current (IOH): -10mA (Min) at VOL=2.4V
- Watchdog Timer value: 0~255 sec

The system will issue a reset when the WDT is enabled and the hardware countdown reaches zero. The reset timer has a 10~20% tolerance depending on the temperature.

## 6.4 Antennae Connector Position

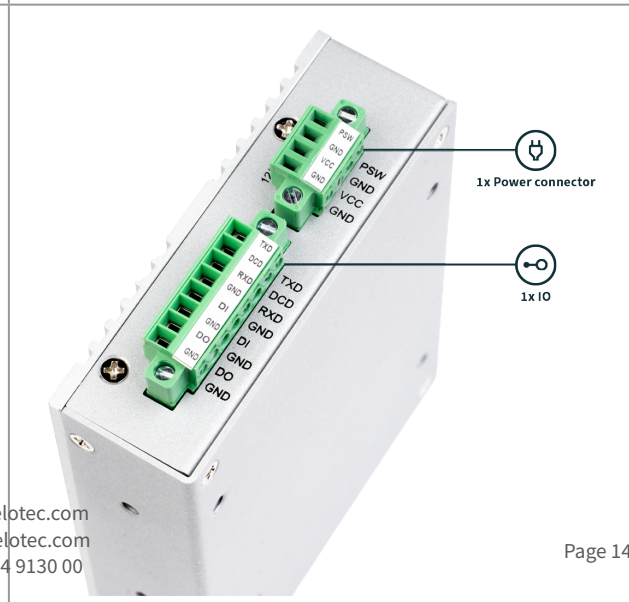
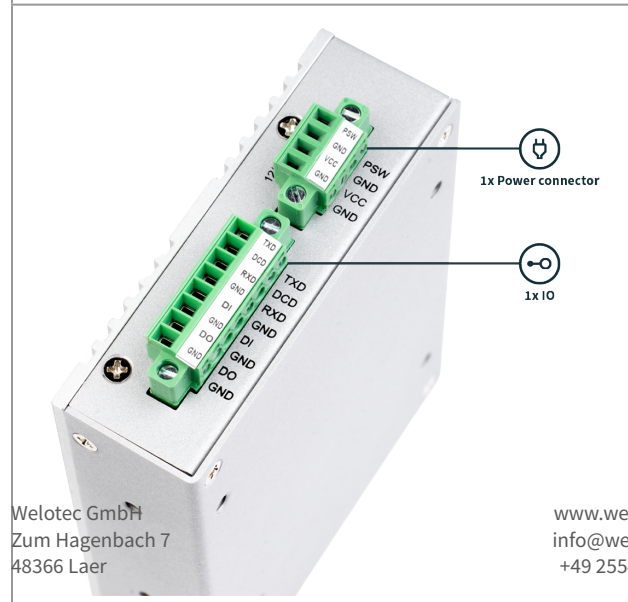
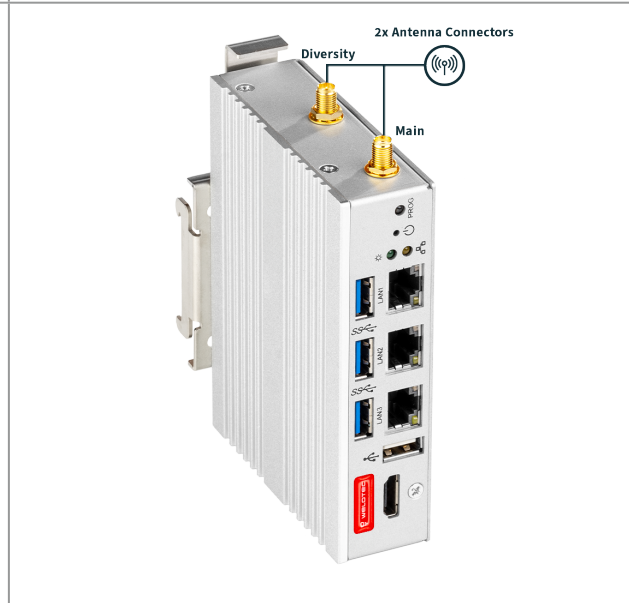
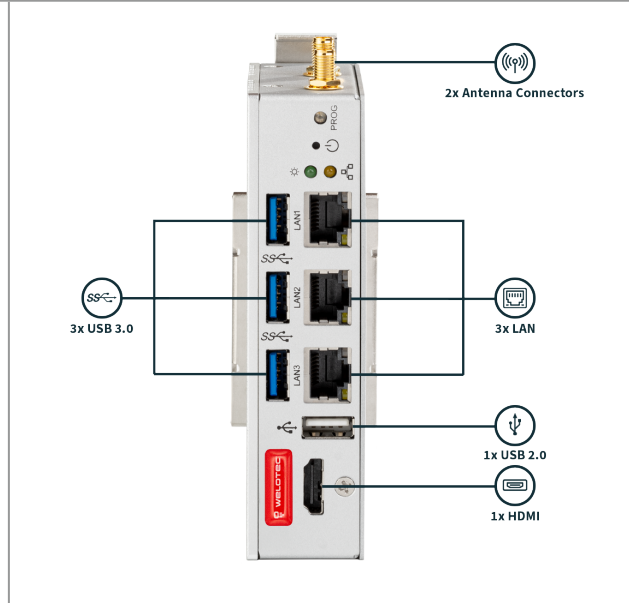
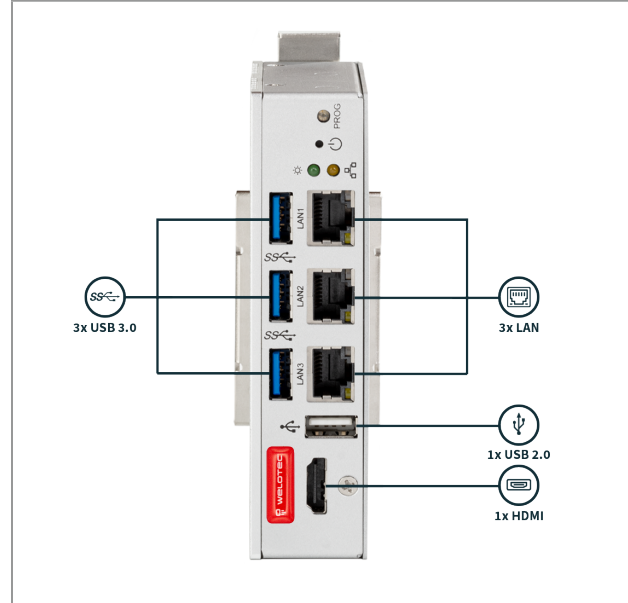


- Ant. 1: LTE Main Connector
- Ant. 2: LTE Diversity or GNSS, depending on customer configuration

# 7 Interfaces and Connections

# 7.1 Arrakis Pico Mk3 Series

Arrakis Pico Mk3 | Arrakis Pico Mk3 (with optional Radio Module)



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

## 8.1 LTE

Band	Frequency Range Down	Frequency Range Up	Max Transmission Power
Band 1	2110 MHz - 2170 MHz	1920 MHz - 1980 MHz	199 mW
Band 3	1805 MHz - 1880 MHz	1710 MHz - 1785 MHz	199 mW
Band 7	2620 MHz - 2690 MHz	2500 MHz - 2570 MHz	199 mW
Band 8	925 MHz - 960 MHz	880 MHz - 915 MHz	199 mW
Band 20	791 MHz - 821 MHz	832 MHz - 862 MHz	199 mW

## 8.2 UMTS

Band	Frequency Range Down	Frequency Range Up	Max Transmission Power
Band 1	2110 MHz - 2170 MHz	1920 MHz - 1980 MHz	251 mW
Band 8	925 MHz - 960 MHz	880 MHz - 915 MHz	251 mW

### 8.2.1 Notes

- **Down:** Refers to the downlink frequency range.
- **Up:** Refers to the uplink frequency range.
- **Max Transmission Power:** Maximum power at which the device transmits.



# 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).

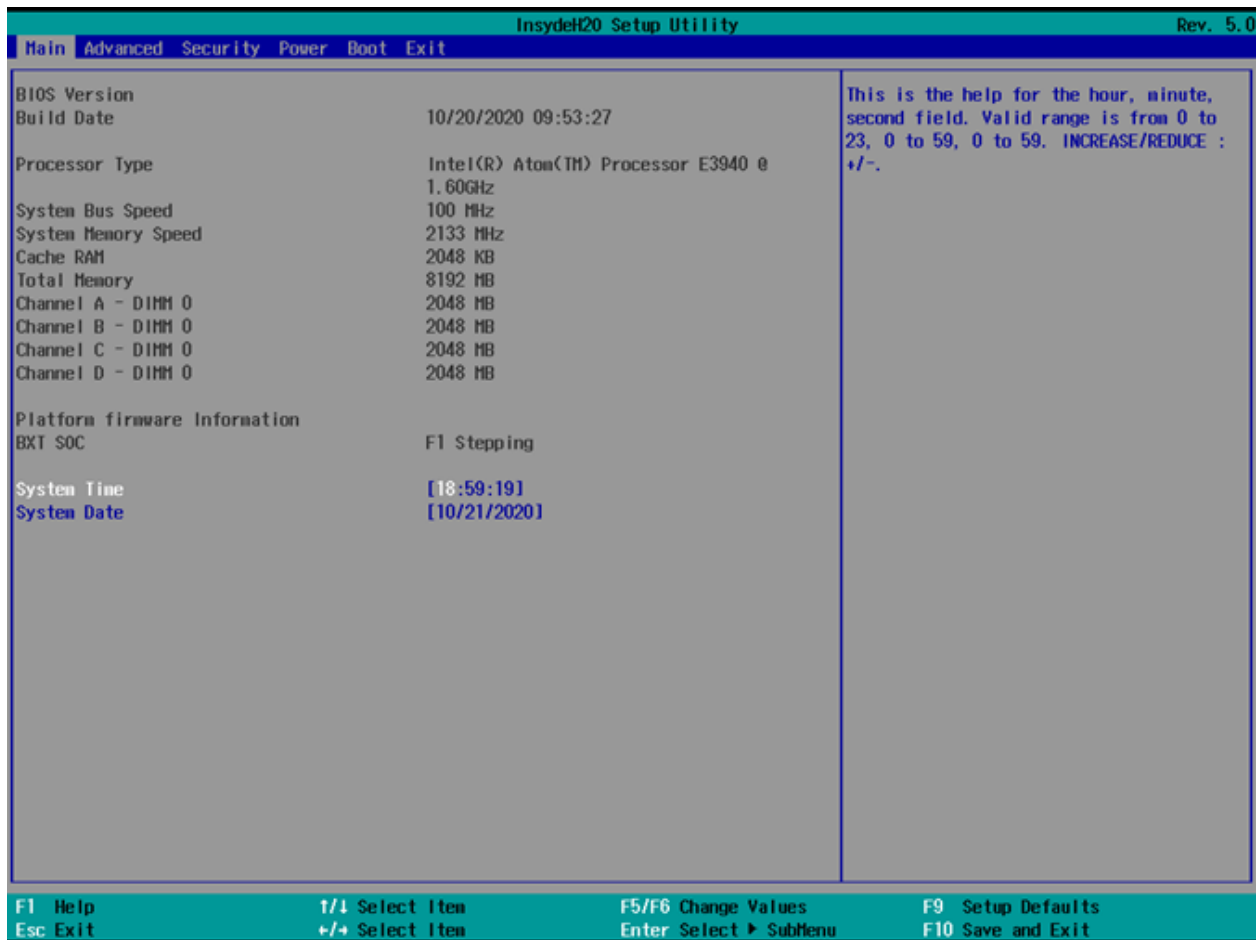
Since the BIOS is the sole channel for hardware and software communication, it is crucial for system stability and optimal performance. In the BIOS Setup main menu, you can see several options. These options will be explained in detail down below. First, let's look at the function keys you may use here:

- 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.

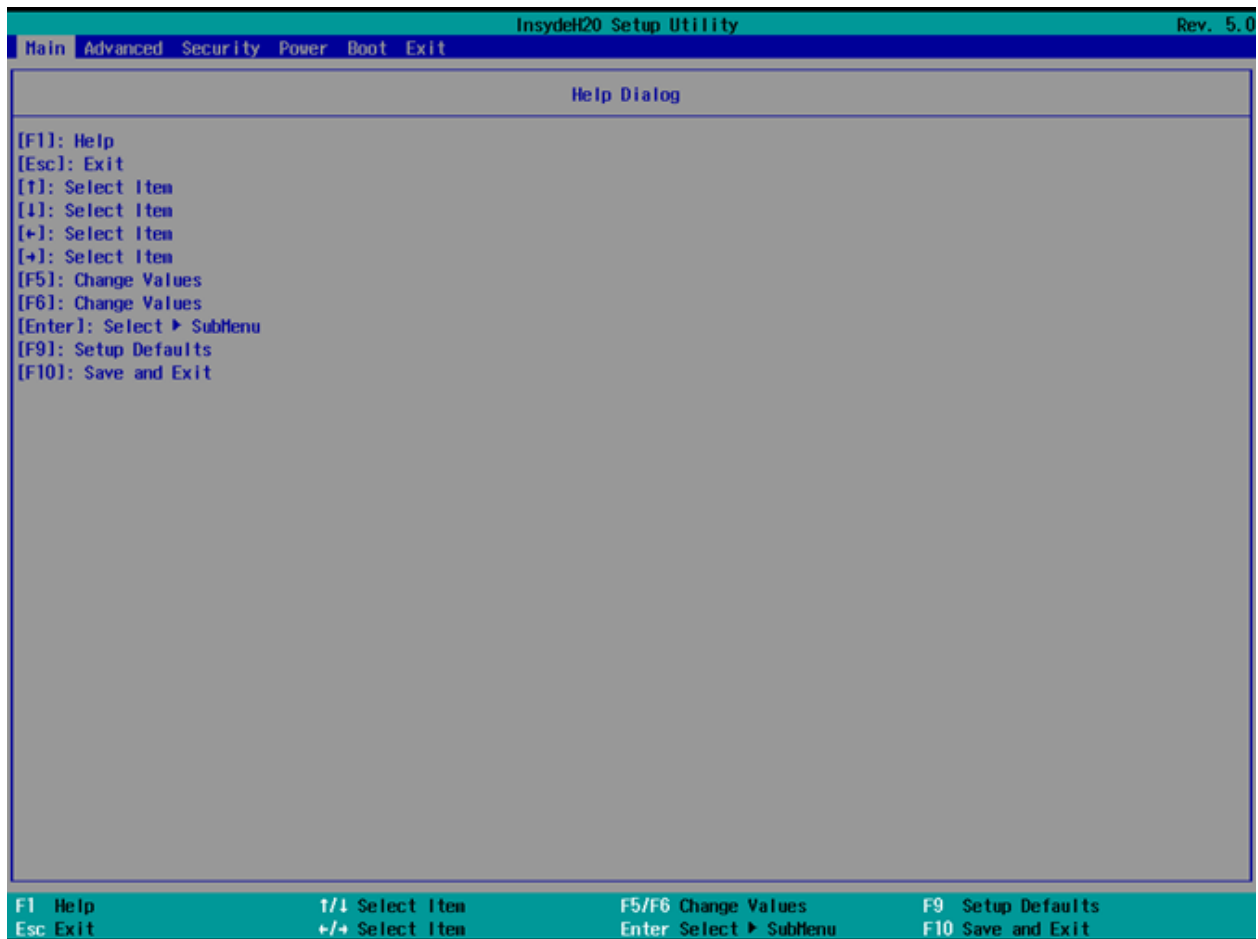
There are six menu bars on top of the BIOS screen:

- **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.

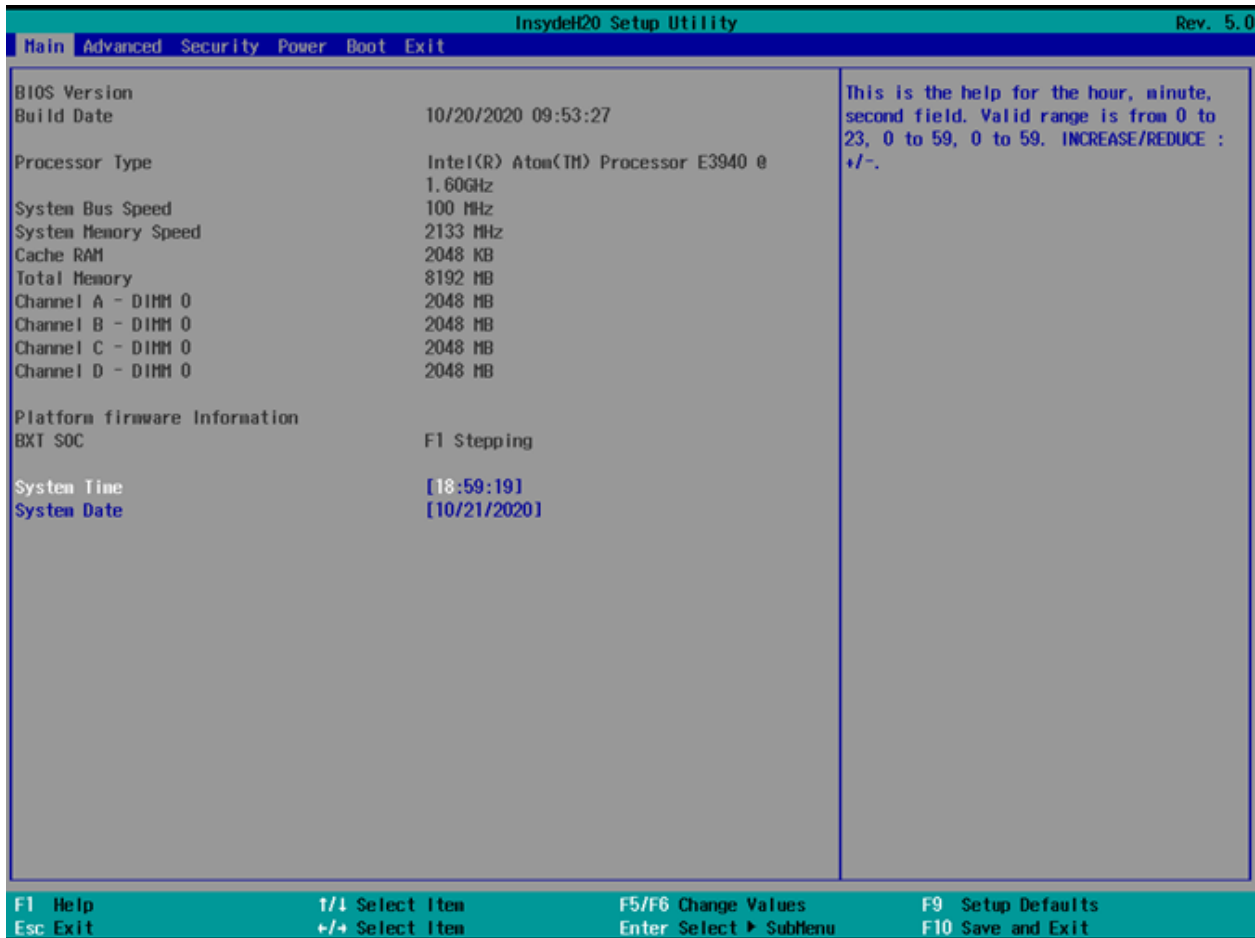
## 9.4 BIOS Help



### 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

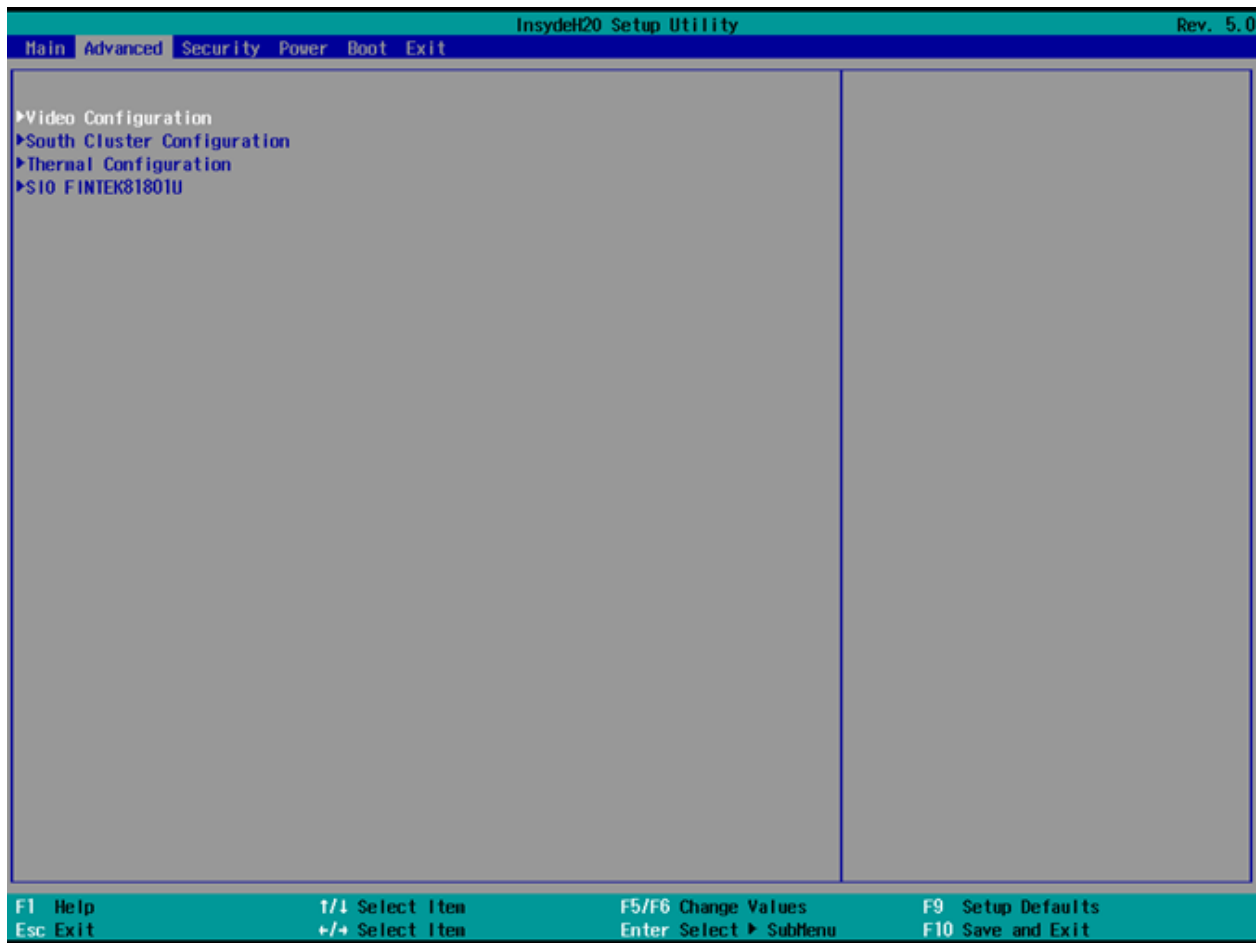


InsydeH20 Setup Utility		Rev. 5.0	
Main   Advanced   Security   Power   Boot   Exit			
BIOS Version		This is the help for the hour, minute, second field. Valid range is from 0 to 23, 0 to 59, 0 to 59. INCREASE/REDUCE : +/-.	
Build Date	10/20/2020 09:53:27		
Processor Type	Intel(R) Atom(TM) Processor E3940 @ 1.60GHz		
System Bus Speed	100 MHz		
System Memory Speed	2133 MHz		
Cache RAM	2048 KB		
Total Memory	8192 MB		
Channel A - DIMM 0	2048 MB		
Channel B - DIMM 0	2048 MB		
Channel C - DIMM 0	2048 MB		
Channel D - DIMM 0	2048 MB		
Platform firmware Information			
BXT SOC	F1 Stepping		
System Time	[18:59:19]		
System Date	[10/21/2020]		
F1 Help	t/1 Select Item	F5/F6 Change Values	F9 Setup Defaults
Esc Exit	+/- Select Item	Enter Select > SubMenu	F10 Save and Exit

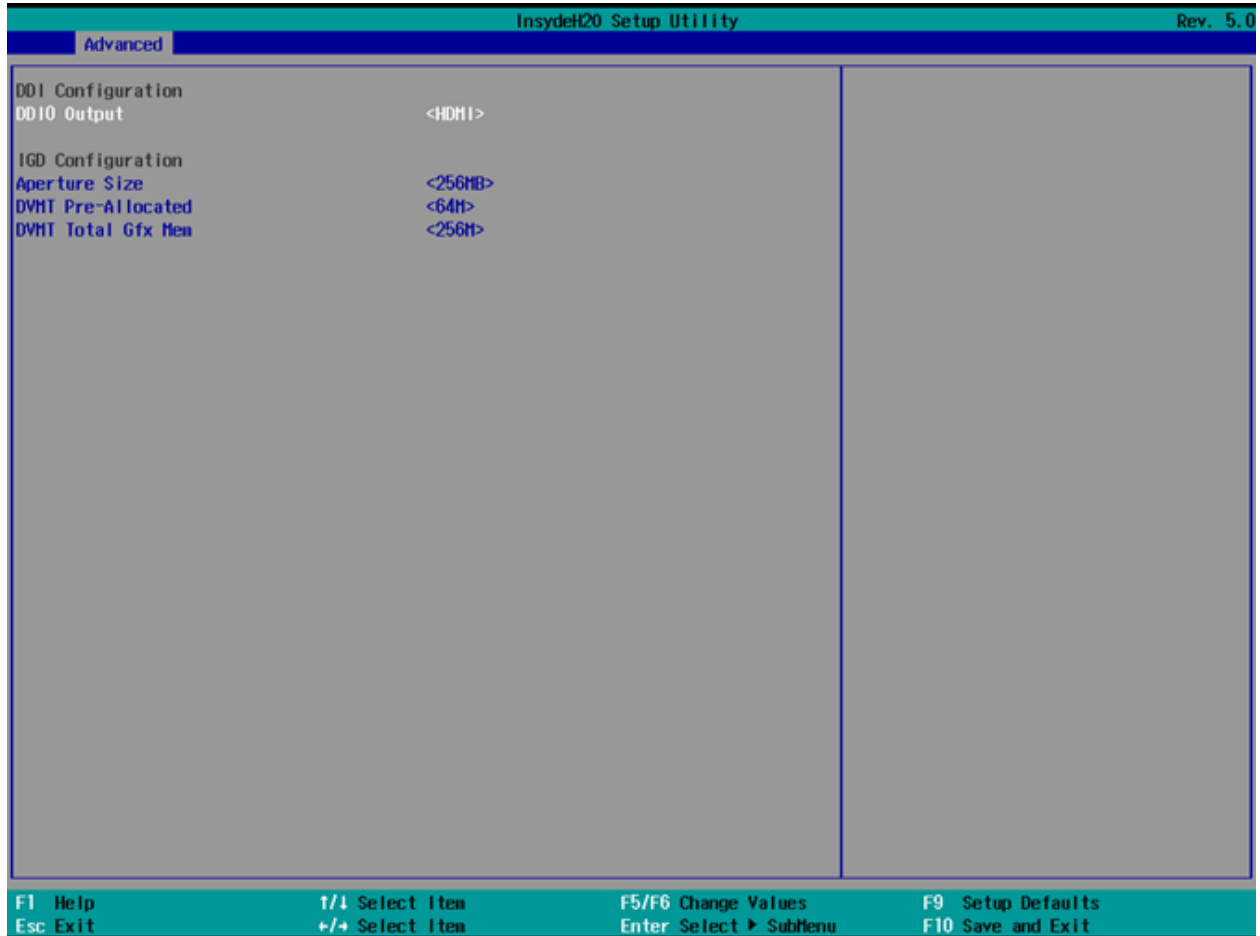
The Main menu screen includes some basic system information. Highlight the item and then use the + or – keys and numerical keyboard keys to select the value you want in each item.

- **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 Video Configuration:

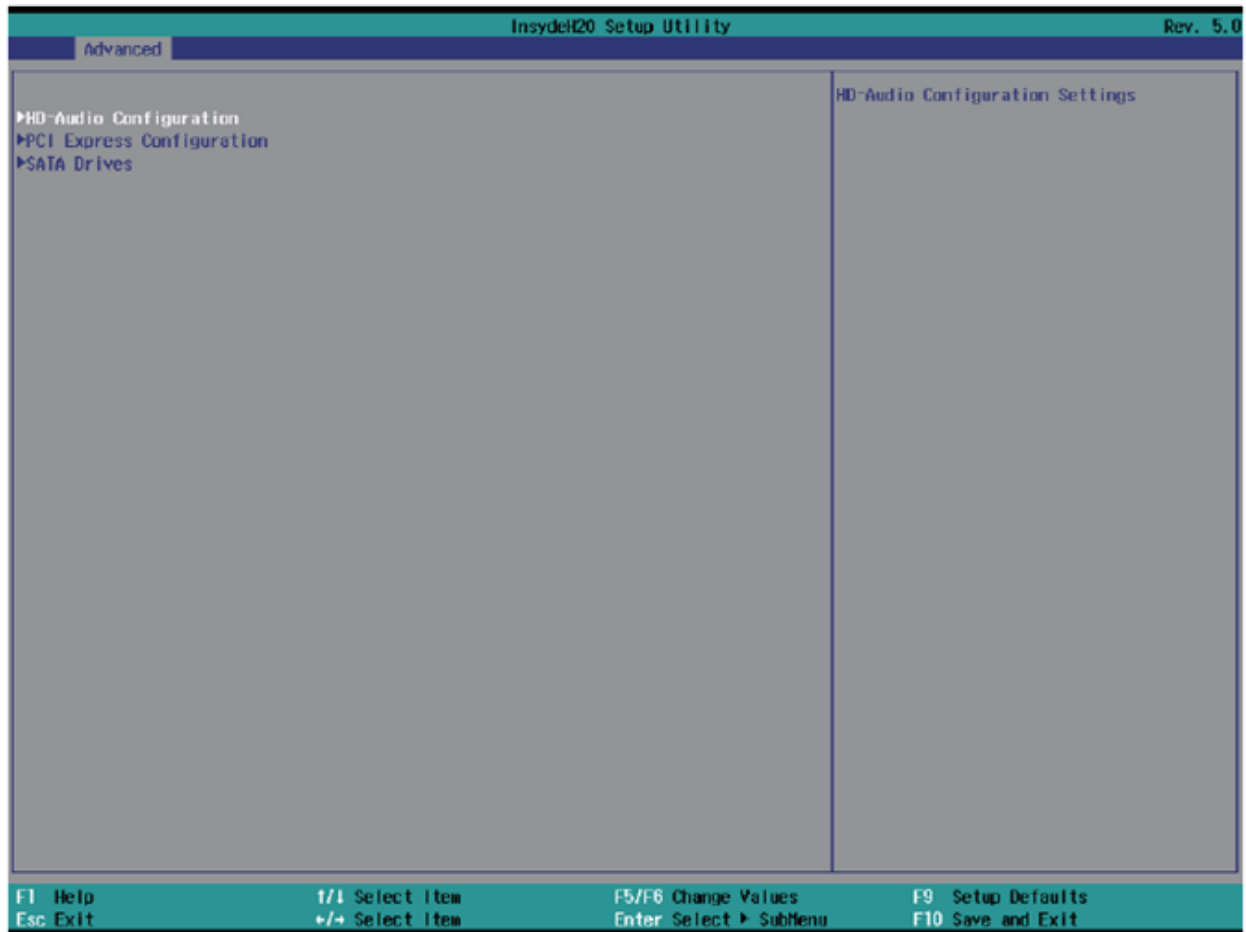


**Aperture Size** The optional settings are: 128MB, 256MB (default), 512MB.

**DVMT Pre-Allocated** Select DVMT 5.0 pre-allocated (fixed) graphics memory size used by the internal graphics device. The optional settings are: 64 (default), 128, 256, 512MB.

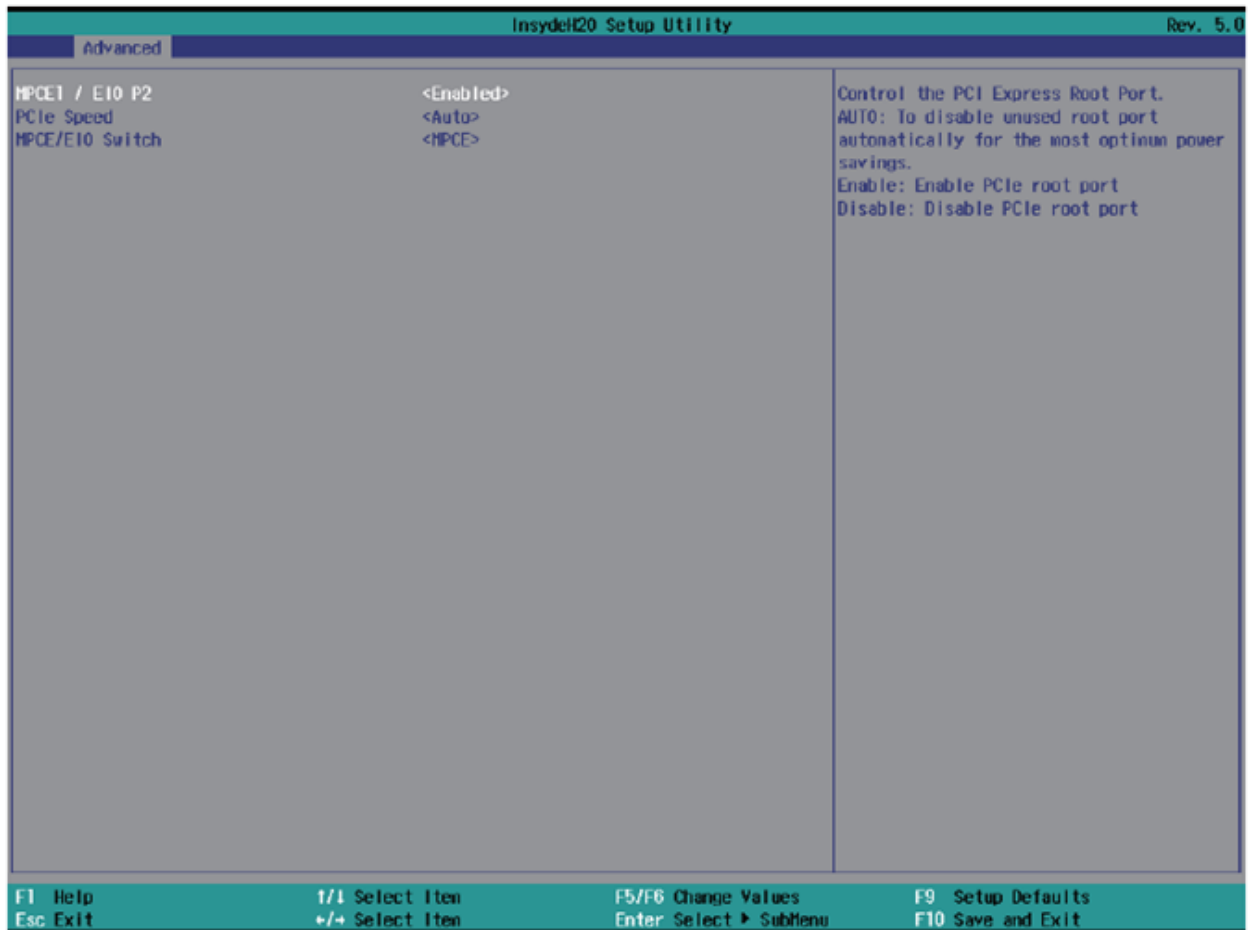
**DVMT Total Gfx Mem** Select DVMT 5.0 total graphics memory size used by the internal graphics device. The optional settings are: 128MB, 256MB (default), MAX.

## 9.6.2 HD-Audio Configuration:



**HD-Audio Support** The optional settings are: Enabled (default), Disabled.

## 9.6.3 PCI Express Configuration:



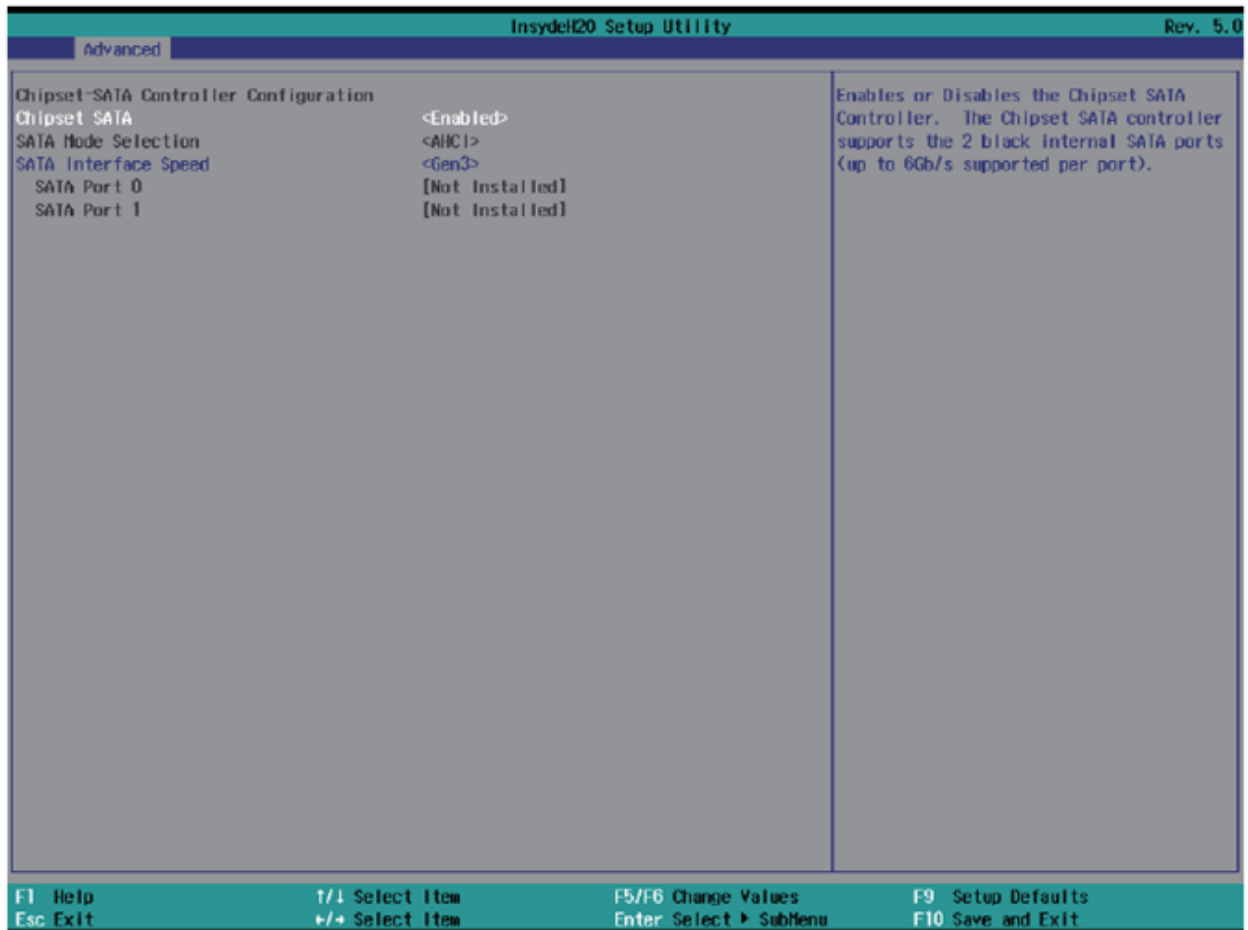
**MPCE1 / EIO P2** The optional settings are: Disabled, Enabled (default) for MPCE1.

**PCIe Speed** Set PCIe speed: Auto (default), Gen1, Gen2.

**MPCE / EIO Switch** Switch the PCIe signal to MPCE1 (default) or EIO (internal expansion port for OEM I/O or function boards).



## 9.6.4 SATA Drives Configuration:

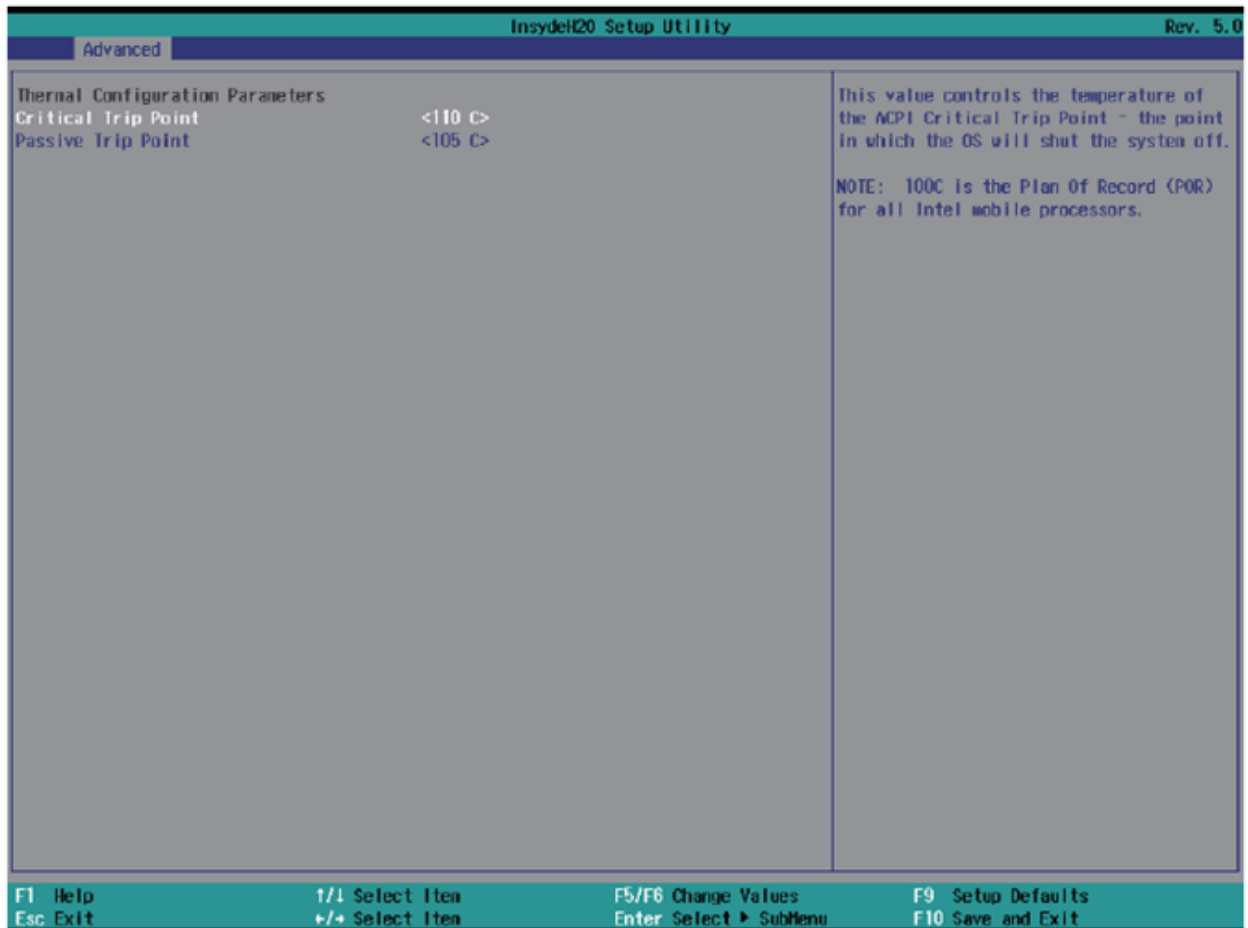


**Chipset SATA** Enable or Disable SATA function. The optional settings are: Enabled (default), Disabled.

**SATA Mode Select** This item is for information; the Arrakis MK3 always operates in AHCI mode.

**SATA Interface Speed** Determine the SATA speed. The optional settings are: Gen1, Gen2, Gen3 (default).

## 9.6.5 Thermal:



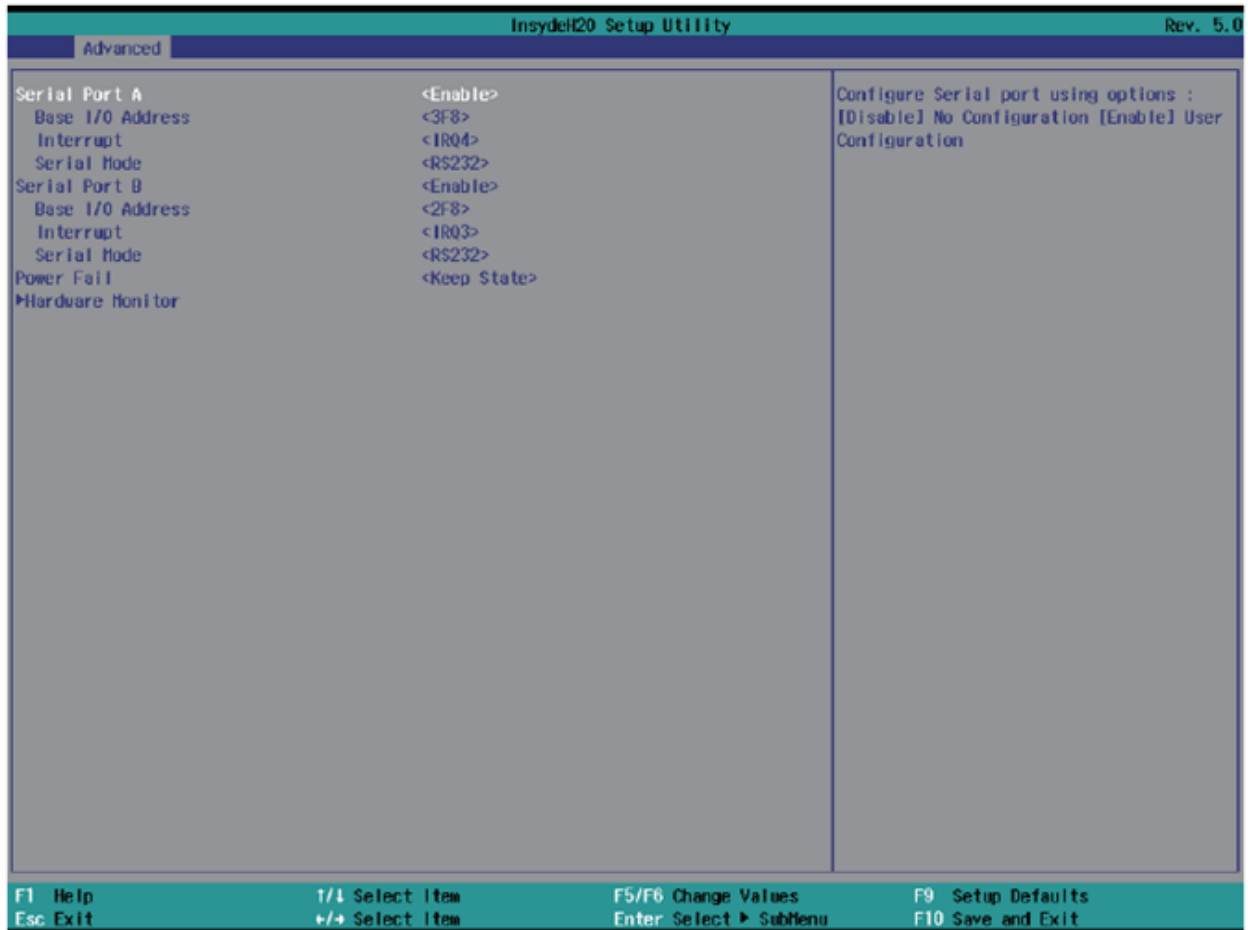
### Thermal Configuration Parameters

This value controls the temperature of the ACPI Critical Trip Point, the point at which the OS will shut down the system.

**Critical Trip Point** The shutdown temperature, with a default value of 110°C.

The CPU frequency will automatically reduce when the CPU temperature reaches the passive Trip Point, which has a default value of 105°C.

## 9.6.6 SIO FINETEK 81801U:



**Serial Port A/B Enable** or disable serial port (COM1 or COM2). The default is Enabled.

**Serial Port A/B Base IO Address / Interrupt** Select an optimal setting for the super IO device. The optional settings are:

- IO=3F8h; IRQ=3,4
- IO=3E8h; IRQ=3,4
- IO=2E8h; IRQ=3,4
- IO=2F8h; IRQ=3,4

The default for port A is IO=3F8h; IRQ=4. The default for port B is IO=2F8h; IRQ=3.

### Serial Mode

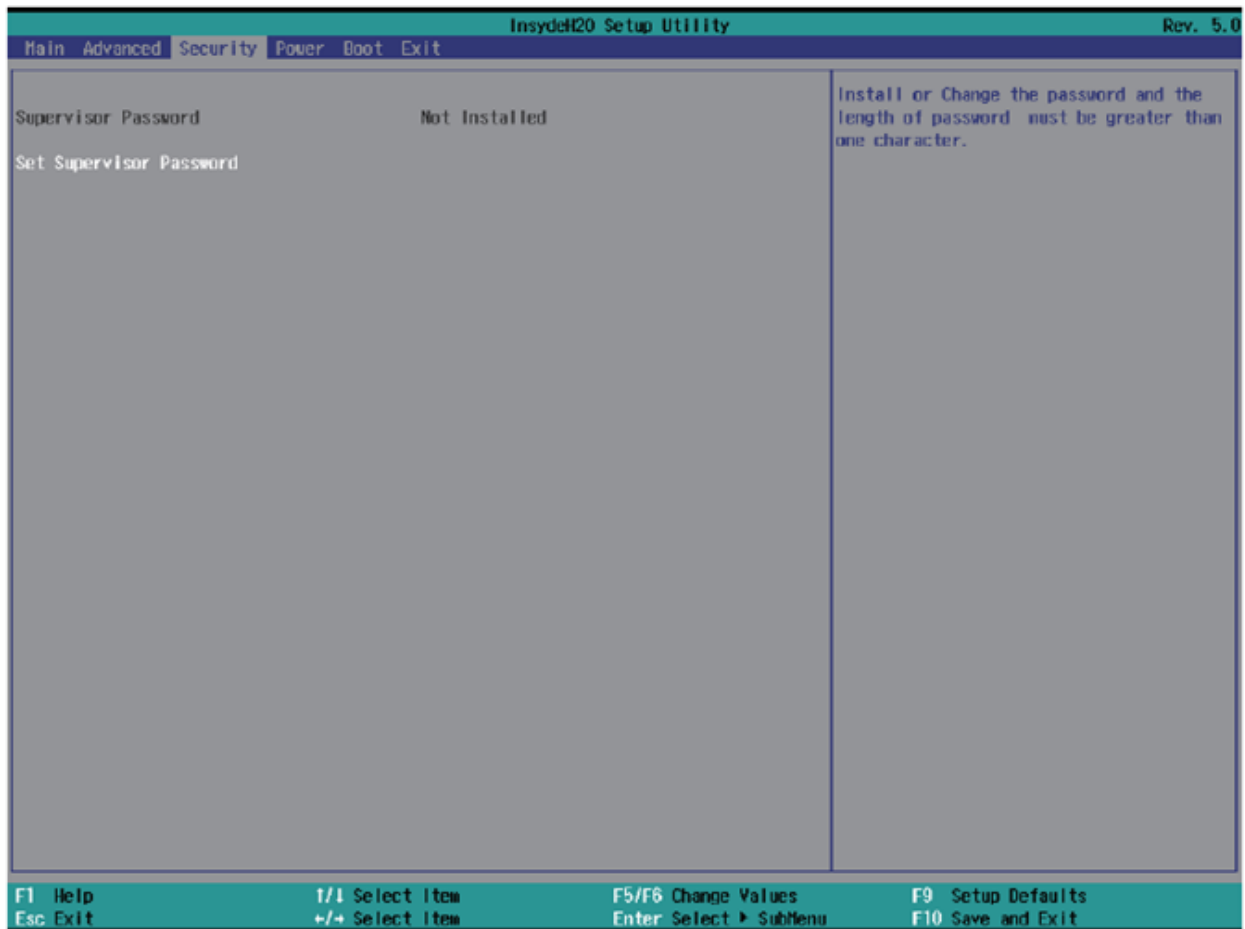
- RS232 driver (default): When hardware is set to RS232/RS422 mode, select RS232 driver.
- RS485 driver: When hardware is set to RS485 mode, select RS485 driver. This enables the auto flow function for RS485.

**Power Failure** Specify whether your system will reboot after a power failure or interrupt occurs.

- **Keep state:** Restores the system to the status before the power failure or interrupt occurred (default).
- **Always on:** Leaves the computer in the power-on state.
- **Always off:** Leaves the computer in the power-off state.

**Hardware Monitor** Displays system voltage and temperature messages. The voltage shows three types: VCC3, VCORE, VNN. The temperature is measured via a separate sensor, not by the CPU.

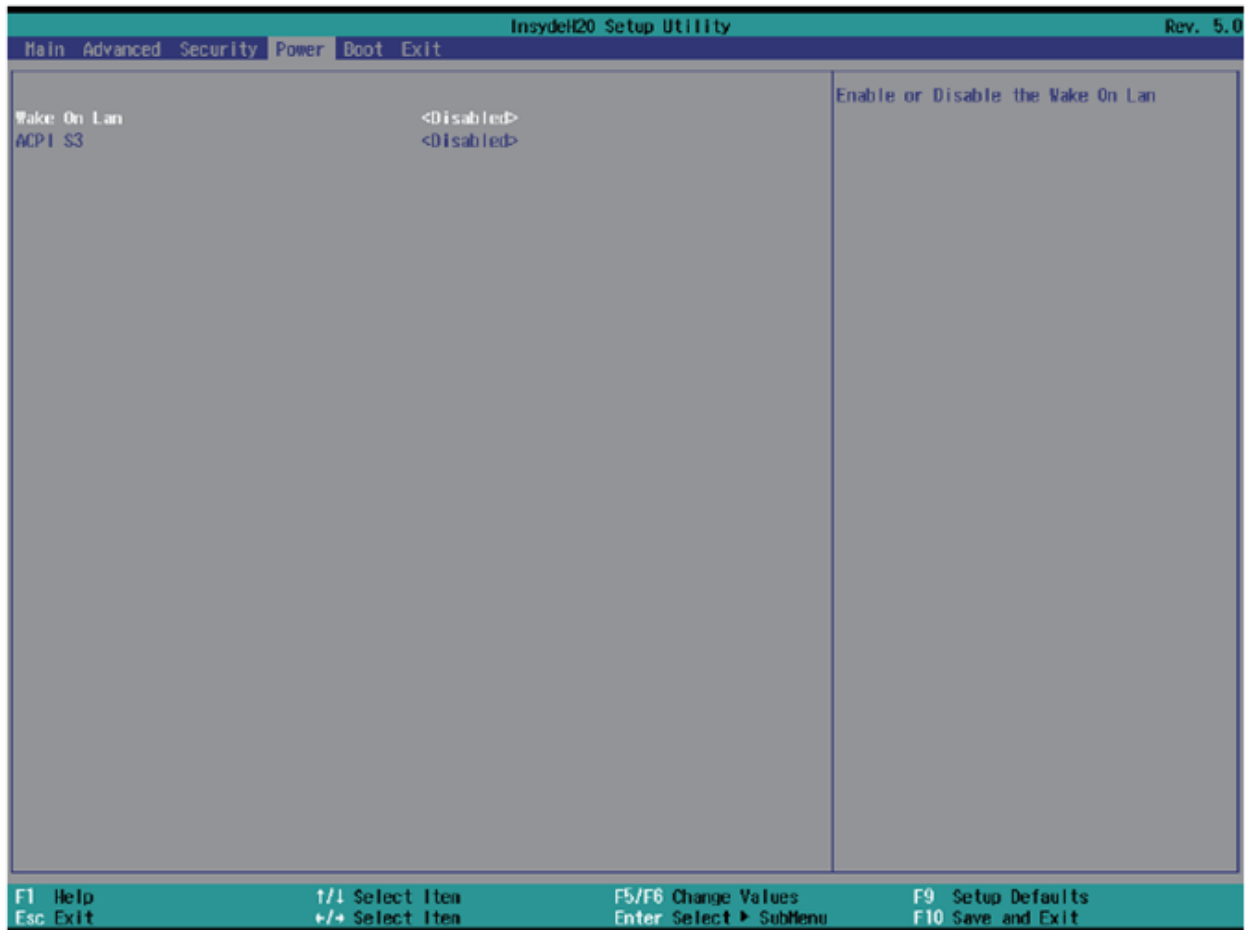
## 9.7 Security



**Supervisor Password** To set up a Supervisor password:

1. Select Supervisor Password. A “Create New Password” dialog will appear.
2. Enter a password that is between 3 and 10 characters long.
3. Press Enter to submit.

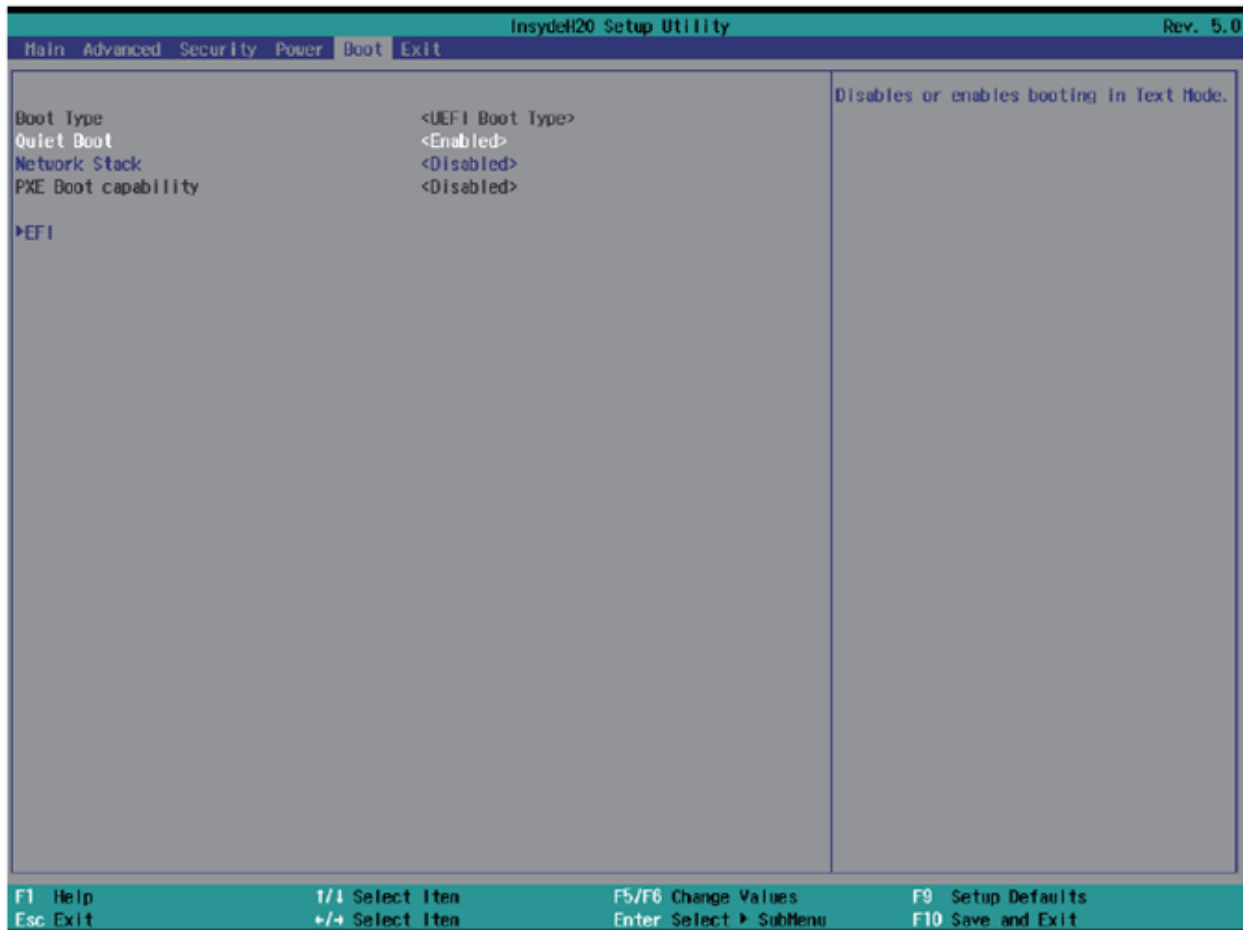
## 9.8 Power



**Wake on LAN** Wake On LAN from LAN1 when the system is in S3 or S5 state, or both. The optional settings are: S3, S5, S3/S5, Disabled (default).

**ACPI S3** Select ACPI sleep state (S3) support. The optional settings are: Enabled, Disabled (default).

## 9.9 Boot



**Boot Type** The Arrakis MK3 is a UEFI Boot-only system.

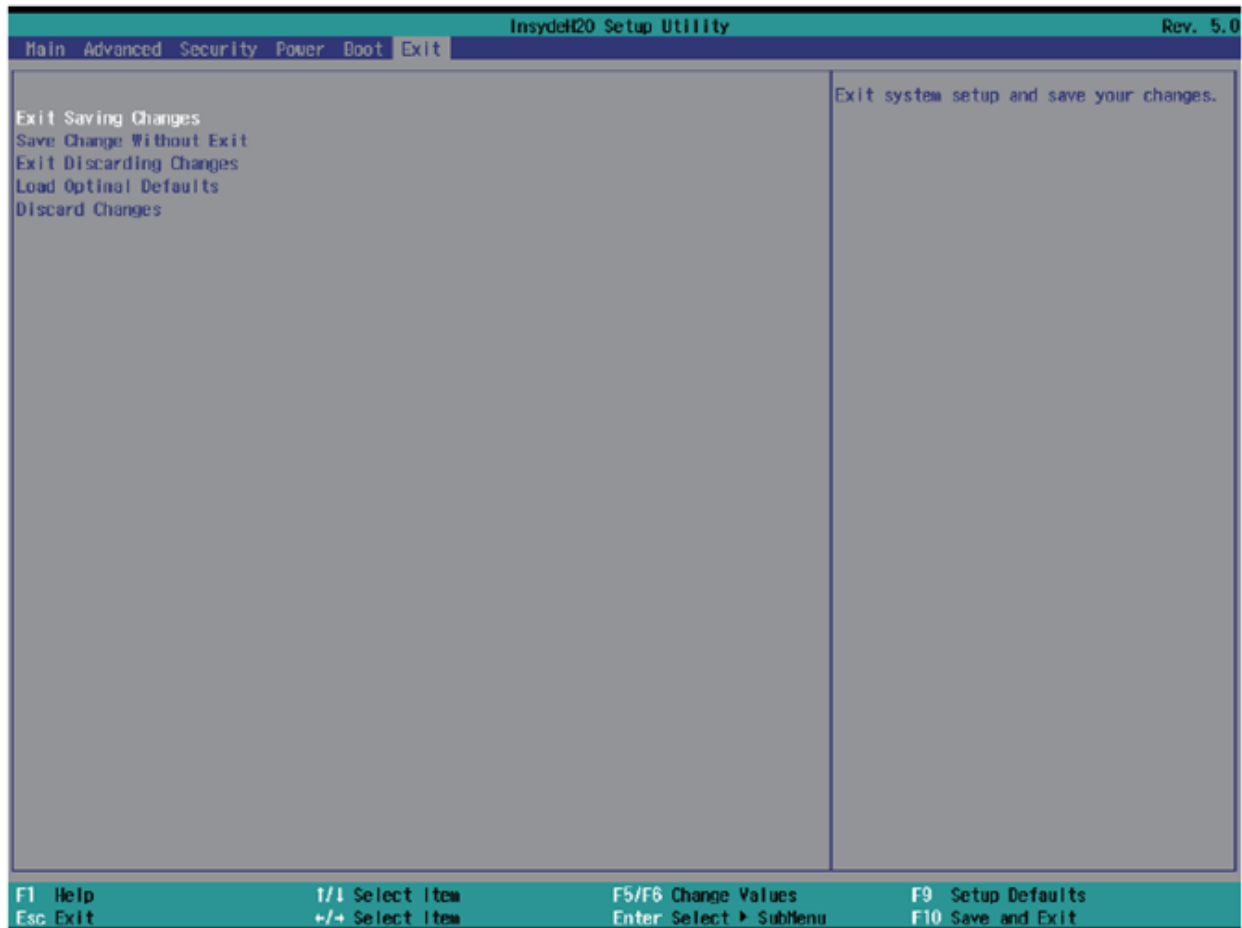
**Quiet Boot** The optional settings are: Enabled (default), Disabled.

**Network Stack** Enable if using the PXE function; otherwise, disable (default).

**PXE Boot Capability** This item determines the protocol operation in PXE. The values are: Disabled (default), UEFI: IPv4, UEFI: IPv6. This item is not modifiable when the above item is disabled.

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

## 9.10 Exit



**Exit Saving Changes** Allows the user to reset the system after saving the changes.

**Save Change Without Exit** Allows the user to save the changes without restarting.

**Exit Discard Changes** Allows the user to restart the system without saving the changes.

**Load Optimal Default** Restores the optimal default settings for all the setup options.

**Discard Changes** Cancels all the setup options without saving.

# 10 Driver Installation

The Arrakis MK3 typically comes with a preinstalled Operating System (recommended).

If you have opted for an Arrakis MK3 without a preinstalled operating system or need to reinstall it, you can download all available system drivers from the following link:



[Download Link](#)

To install the drivers, please run the driver installation programs and follow the on-screen instructions.



# 11 Appendix A: Power Consumption

Item	Specification
CPU	Intel E3940
RAM	LP-DDR3 8GB
Operating System	Windows 10 IoT 2019 LTSC
Test Program	Passmark
mSATA	64GB

**Note: Results are for reference only!**

Voltage	Power Off	Start Up Max.	Start Up Stable	Burn-In Max.	Shut Down
12V	0.14A	0.95A	0.62A	1.10A	0.82A
24V	0.09A	0.50A	0.32A	0.57A	0.42A

**Note: Power consumption varies depending on options and software configurations.**

# 12 Appendix B: F75111N DIO & Watchdog Device

The Arrakis MK3 includes optional DIO Ports. This chapter provides an introduction to programming these ports.

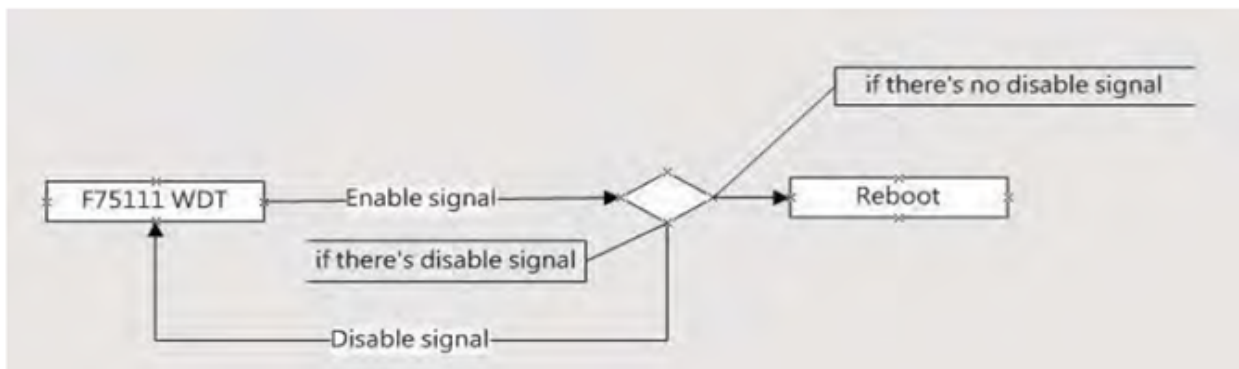
## 12.1 Watchdog Timer under DOS

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:

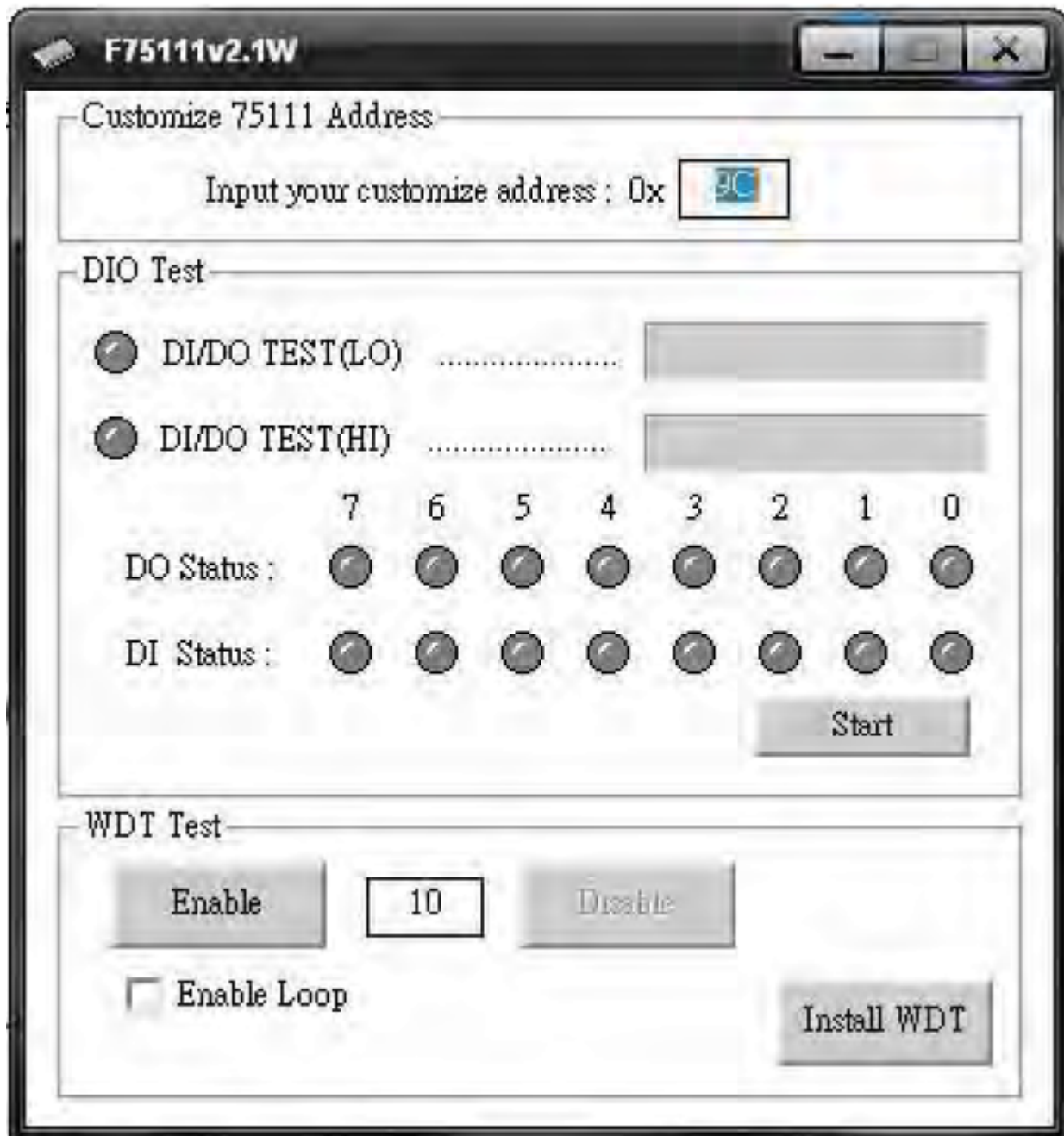
```
void pause(int time) {
    asm mov ah,0h; // Ah = 00 Read System Time Counter
    asm int 1ah; // Read time from Time Counter and store it in DX register
    asm add dx, time;
    asm mov bx, dx;
label:
    asm int 1ah;
    asm cmp bx, dx;
    asm jne label;
}
```

## 12.2 Watchdog Timer and DIO under Windows:

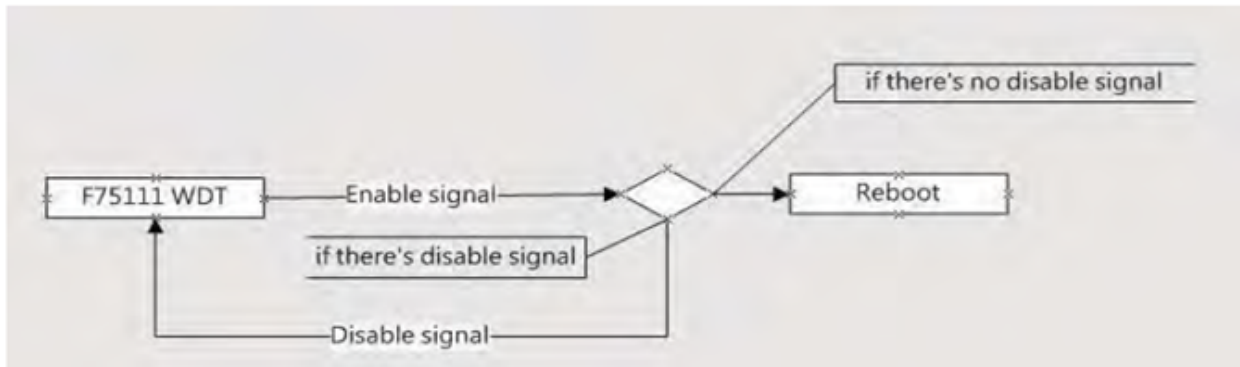
The necessary software resources for programming the watchdog timer can be accessed from the Driver Download section:

- 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:

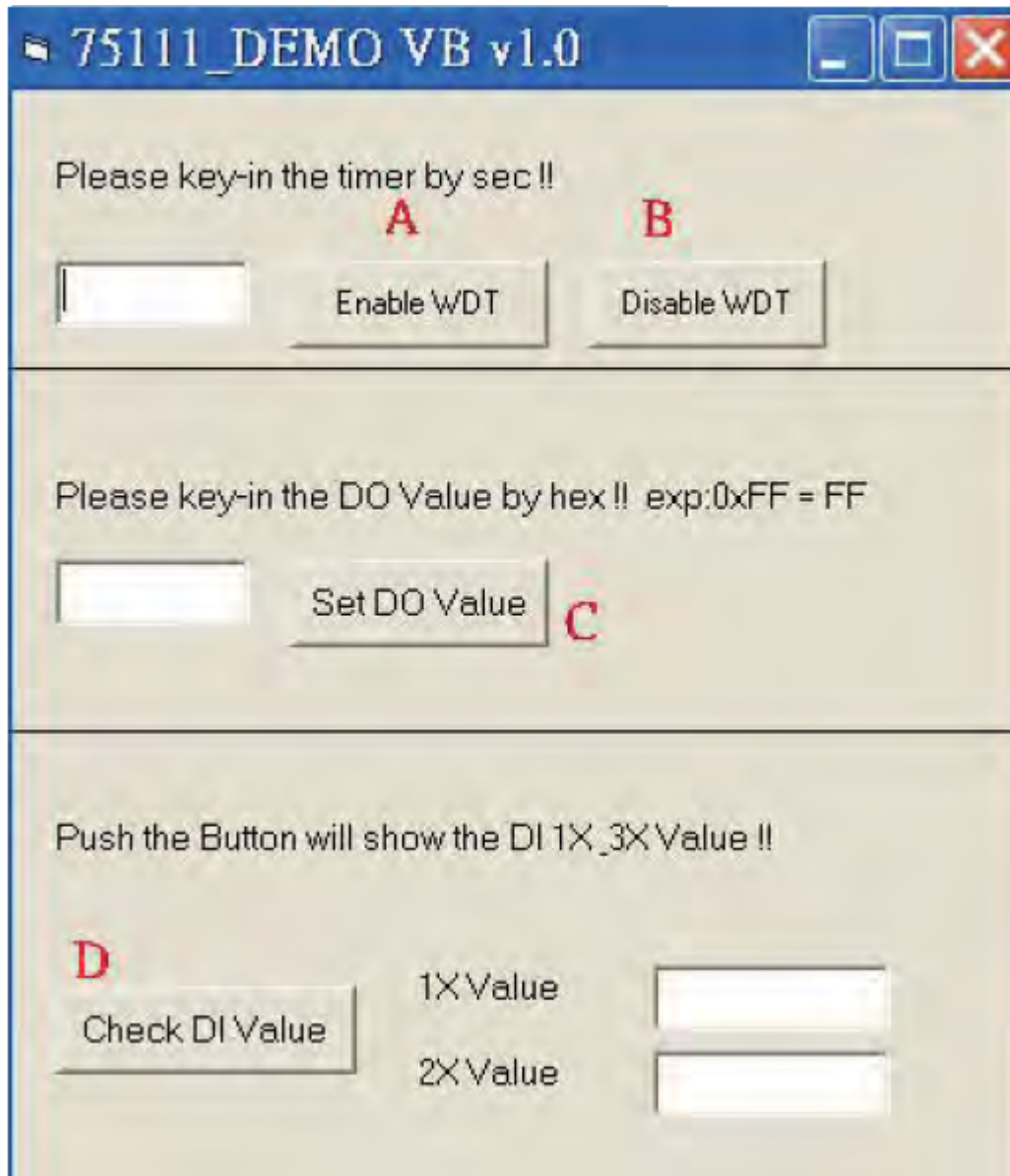
```
void F75111_SetWDTDisable() {
    WriteByte(F75111_INTERNAL_ADDR, WDT_CONFIGURATION, 0x00); // Disable Watchdog
}
```

# 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”:

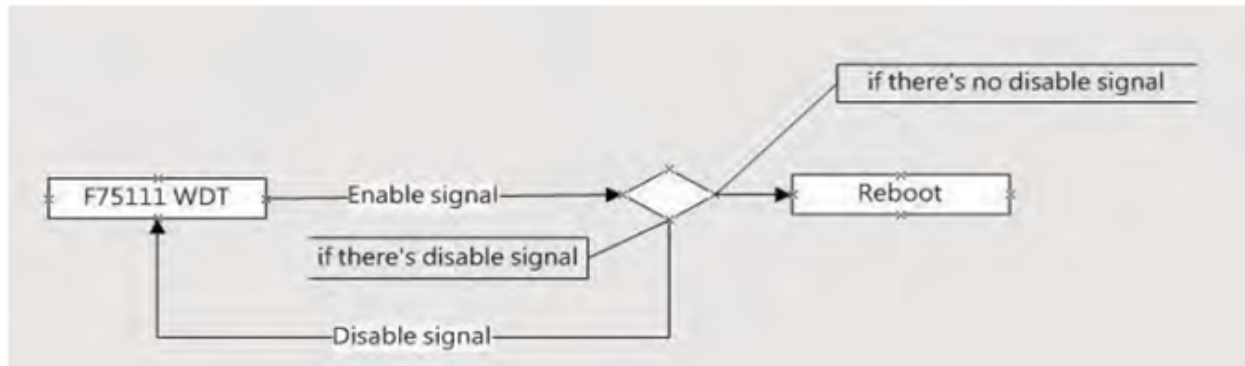
- Navigate to the F75111 directory: `cd F75111`.
- Compile the source: `make`.
- Execute the binary file: `src/f75111`.

## 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
}
```