

# Welotec Arrakis Mk4

## MANUAL

Version 0.1



Welotec GmbH  
Zum Hagenbach 7, D-48366 Laer  
T: +49 (0)2554/9130-00  
F: +49 (0)2554/9130-10  
[info@welotec.com](mailto:info@welotec.com)

## Table of Contents

|          |   |    |
|----------|---|----|
| <b>1</b> | <b>Copyright</b> .....                                    | 3  |
| <b>2</b> | <b>Regulatory Compliances</b> .....                       | 3  |
| 2.1      | CE Notice .....   | 3  |
| 2.2      | FCC Notice .....  | 3  |
| <b>3</b> | <b>Safety Instructions</b> .....                          | 4  |
| <b>4</b> | <b>Product Introduction</b> .....                         | 5  |
| 4.1      | About Arrakis Mk4 .....                                   | 5  |
| 4.2      | Product Specification .....                               | 5  |
| <b>5</b> | <b>System Information</b> .....                           | 6  |
| 5.1      | System Drawing .....                                      | 6  |
| 5.2      | Mainboard Block Diagram .....                             | 8  |
| 5.3      | System.....   | 9  |
| 5.4      | Antenna Configurations.....                               | 9  |
| <b>6</b> | <b>BIOS</b> .....   | 10 |
| 6.1      | Enter BIOS .....  | 10 |
| 6.2      | BIOS menu and function keys .....                         | 11 |
| 6.3      | BIOS Help .....   | 12 |
| 6.4      | Menu .....  | 13 |
| 6.5      | Advanced.....   | 14 |
| 6.6      | Security .....  | 19 |
| 6.7      | Power .....   | 20 |
| 6.8      | Boot.....   | 21 |
| 6.9      | Exit.....   | 22 |
| <b>7</b> | <b>Driver Installation</b> .....                          | 23 |
| <b>8</b> | <b>Appendix A Power consumption</b> .....                 | 24 |
| <b>9</b> | <b>Appendix B F75111N DIO &amp; Watchdog device</b> ..... | 25 |
| 9.1      | Watchdog timer under DOS: .....                           | 25 |
| 9.2      | Watchdog Timer and DIO under Windows: .....               | 25 |
| 9.3      | IO Device: F75111 VB6 under Windows: .....                | 29 |
| 9.4      | SDK Function Introduction: .....                          | 31 |
| 9.5      | Watchdog Timer and DIO under Linux: .....                 | 32 |

# 1 Copyright

## **Copyright and Trademarks, 2017 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 Notice

This device complies with the requirements of the CE directive.

## 2.2 FCC Notice

This device complies with the requirements of the FCC regulations

### 3 Safety Instructions

Please read these instructions carefully and keep these instructions for future references.

1. Disconnect this equipment from Power outlet before cleaning. Do not use liquid or sprayed detergent for cleaning. Use moisture sheet or cloth for cleaning.
2. Keep this equipment from humidity.
3. Make sure power cord is laid in such a way that people cannot step on it. Do not place anything over the power cord.
4. All cautions and warning on the equipment should be noted.
5. If the equipment is not used for a long time, disconnect the equipment from main power to avoid being damaged by transient over voltage.
6. **Prolonged usage with less than 12V may damage the PSU or destroy the Mainboard**
7. Never pour any liquid into opening as this could cause fire or electrical shock.
8. If one of the following situations arises, have the equipment checked by service personnel.
  - a. The power cord or plug is damaged.
  - b. Liquid has penetrated the equipment.
  - c. The equipment has been exposed to moisture in a condensation Environment.
  - d. The equipment does not work well, or it is unable to get it working on following this user manual.
  - e. The equipment has dropped and damaged.
9. Do not leave this equipment in an environment unconditioned, storage temperature below -20 degrees or above 70 degrees in Celsius for longer periods of time, it may damage the equipment.
10. Unplug the power cord when doing any service or adding optional kits.
11. Lithium Battery Caution:
  - a. Danger or explosion can happen if the battery is incorrectly replaced. Replace only the original or equivalent type recommended by the manufacturer. Dispose used batteries according to the manufacturer's instructions.
  - b. Do not remove the cover and ensure no user serviceable components are present inside. Take the unit to the service center for service and repair.

## 4 Product Introduction

### 4.1 About Arrakis Mk4

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

The Arrakis Mk4 is powered by a 1.5/3.0 GHz Quadcore Intel Atom® x6413E Processor, and its standard Version has a SoDIMM socket for 3200MHz DDR4 Memory.

It is capable to support 1x HDMI and 1x Display Port Displays with the resolution up to 4K. The system features NVMe storage, one M.2 3042 socket and Nano SIM Slot, with optional Expansion module, 2x FullSize mPCIe sockets with USB and 2 external SIM Sockets. Three USB 3.0, one USB 2.0, triple 2.5 Gigabit-LAN, and options to choose between CAN-Bus, Digital I/O, WLAN and WWLAN. It also supports maximum of four COM ports.

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

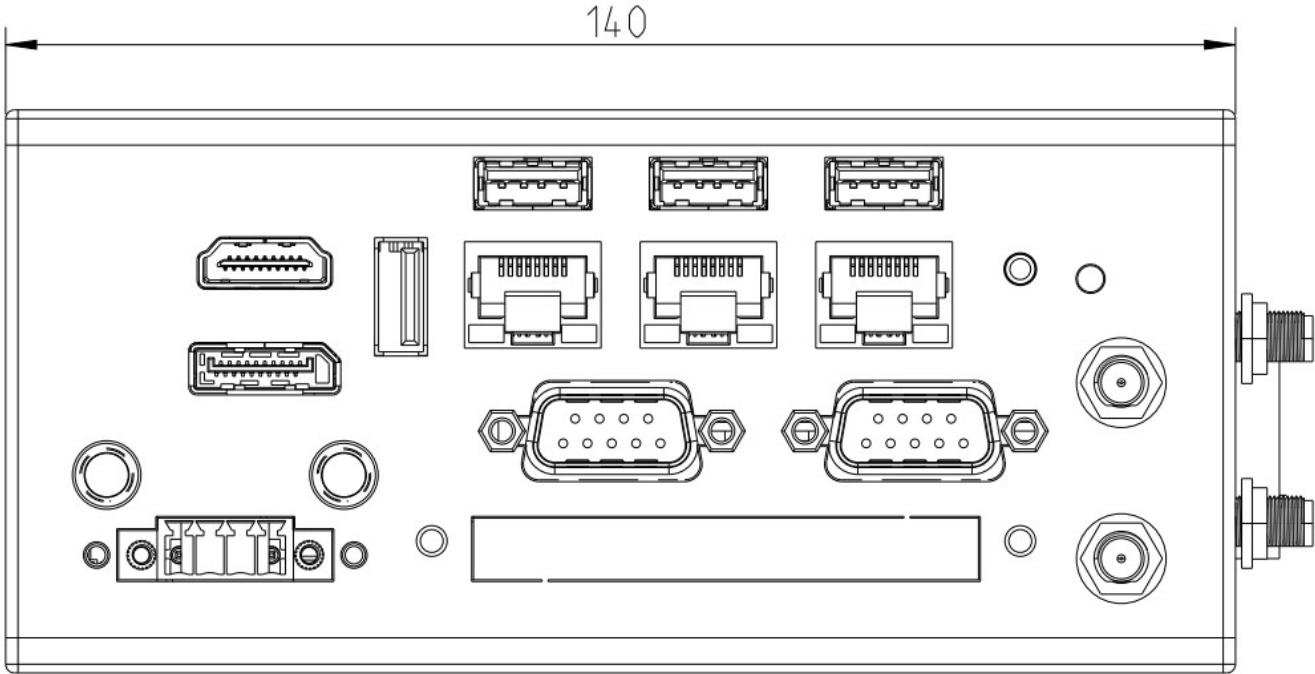
Combining the above-mentioned features with its aluminium housing and dust-proof design makes Arrakis Mk4 a perfect industrial embedded system to operate in environments with harsh and critical conditions.

### 4.2 Product Specification

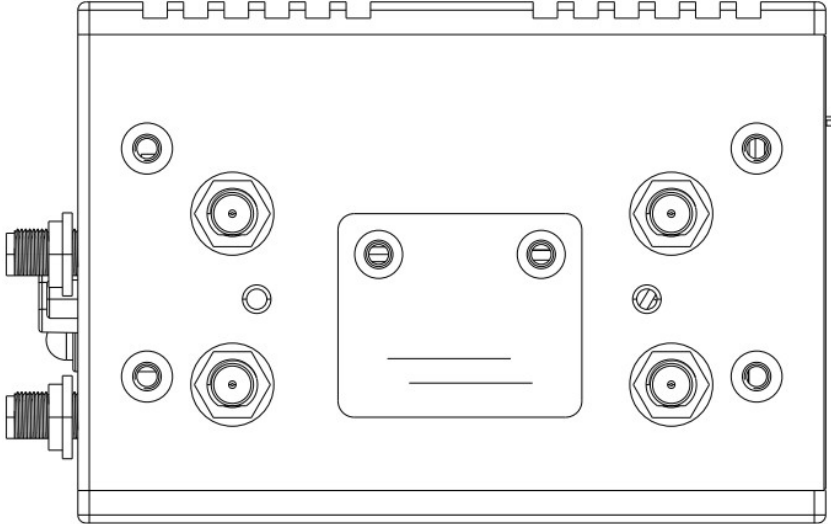
| Technical Details        |                        |  |
|--------------------------|------------------------|--|
| Processor                | CPU                    | Intel Atom® x6413E Processor 1.5/3.0 GHz (Std.)                    |
| Memory                   | RAM                    | 3200 MHz DDR4 SoDIMM Socket, up to 32GB                            |
| Display                  | Max. Resolution        | DP + HDMI, max. 4K   |
| Storage                  |                        | x1 NVMe B+M Key, 2x PCIe 3.0 Lanes                                 |
|                          |                        | x1 SATA DOM Connector  |
| I/O Physical             | Displayport            | x 1  |
|                          | HDMI                   | x 1  |
|                          | LAN                    | x 3 RJ45   |
|                          | USB 3.0                | x 3  |
|                          | USB 2.0                | X 1  |
|                          | RS232/422/482          | x 2 (optional 2 additional RS232/422/485)                          |
| Networking               | LAN                    | x3 (10/100/1000/2500 Base-T), Intel i225/226-IT LAN chip           |
|                          | WLAN                   | (Optional, mPCIe)  |
|                          | WWAN                   | (Optional, 4G/5G)  |
| Other Expansion          | SIM Slot               | x1 push-push Type Nano SIM Slot (plus 2x Micro SIM Slots optional) |
| Others                   | Others                 | Digital I/O, CAN and others optional)                              |
|                          | Watch dog Timer        | System Reset, Programmable via Software from 1 to 255 Seconds      |
| Environment              | Temperature            | -20° to 70° C Operation<br>-20° to 80° C Storage                   |
|                          | Relative Humidity      | 5% to 95% non-condensing   |
| Power                    | Power Supply           | 12-36V DC Terminal block type 4 pin and DC jack type               |
|                          | Power Adapter          | Optional 60W,24V/5A external power adapter, CR1220 CMOS Battery    |
| Mounting                 | DIN Rail Mounting Kits | (Optional)   |
| Compatible OS            |                        | Windows 10/11, Ubuntu Linux, others on request                     |
| Physical Characteristics | Material/ Color        | Aluminum / Steel, Silver   |
|                          | Dimension (mm)         | 64 x 140 x 92 mm   |
| Regulatory Compliance    |                        | CE, FCC  |

## 5 System Information

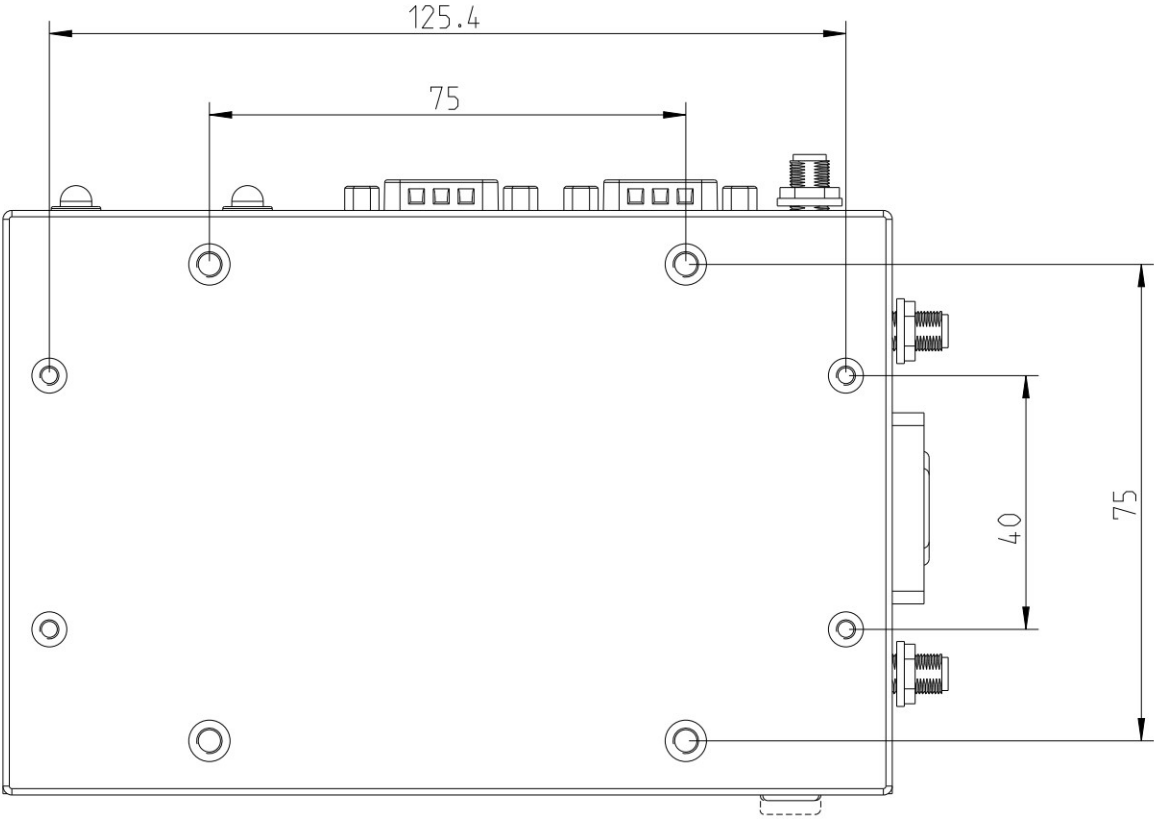
### 5.1 System Drawing



Top site

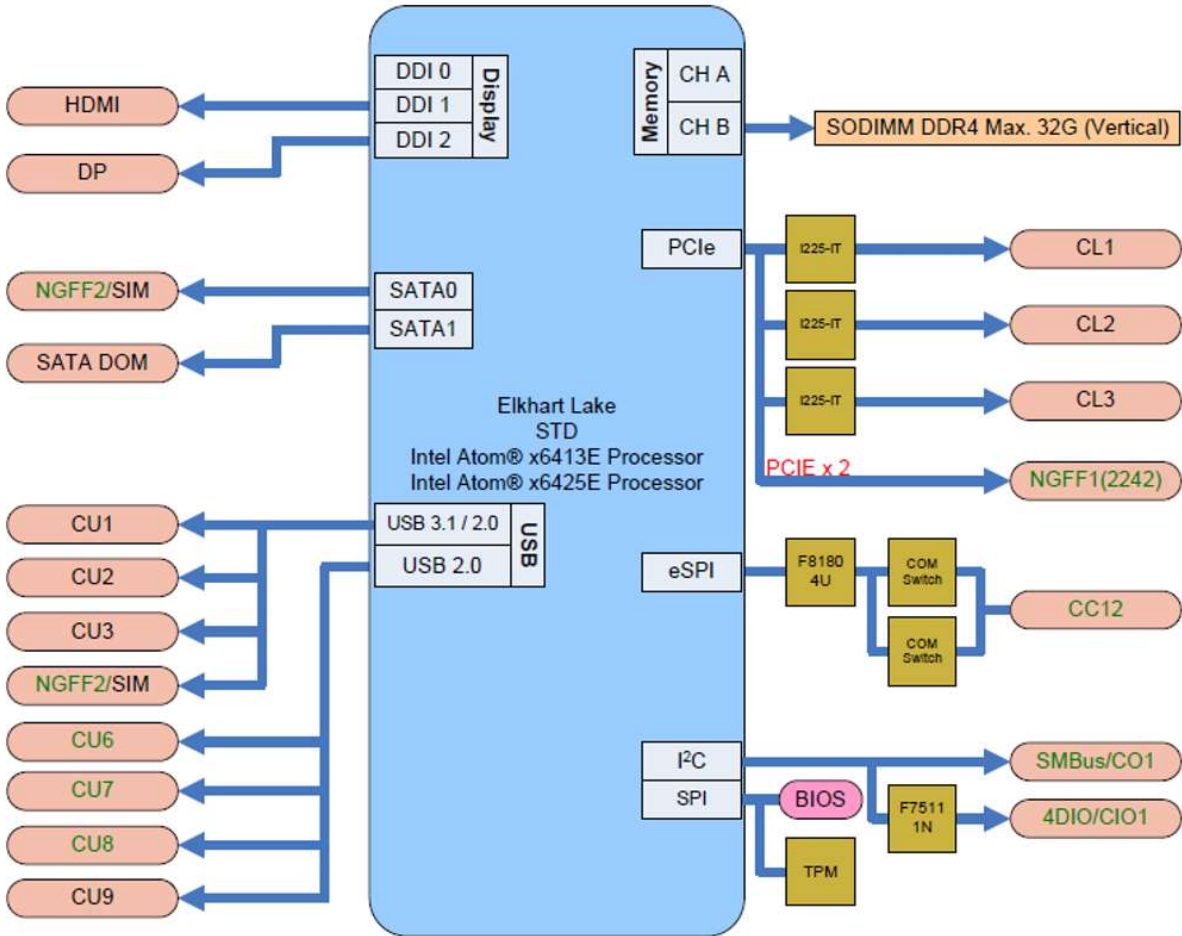


Rear site



### 5.2 Mainboard Block Diagram

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





5.3 System



5.4 Antenna Configurations

|      | ANT1 | ANT2       | ANT3      | ANT4 | ANT5 | ANT6      |
|------|------|------------|-----------|------|------|-----------|
| LTE  |      |            | Diversity | Main | GNSS |           |
| 5G   |      | Diversity* | Diversity | Main | GNSS | Diversity |
| WiFi | X    | (X)*       |           |      |      |           |

\*with 5G Antenna allocation is dependent on customer requirements

## 6 BIOS

### **Introduction:**

The BIOS is a program located in the Flash Memory on the motherboard. This program is a bridge between motherboard and operating system. When you start the computer, the BIOS program gains control. The BIOS first executes an auto-diagnostic test called POST (Power on Self Test) for all the necessary hardware, it detects the entire hardware devices and configures the parameters of the hardware synchronization. After these tasks are completed, BIOS will give control of the computer back to operating system (OS).

Since the BIOS is the only channel for hardware and software to communicate with, it is the key factor of system stability and of ensuring your system performance at best.

In the BIOS Setup main menu, you can see several options. We will explain these options in the following pages. First, let us see 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 to exit the BIOS Setup menu after you complete the setup of BIOS parameters.

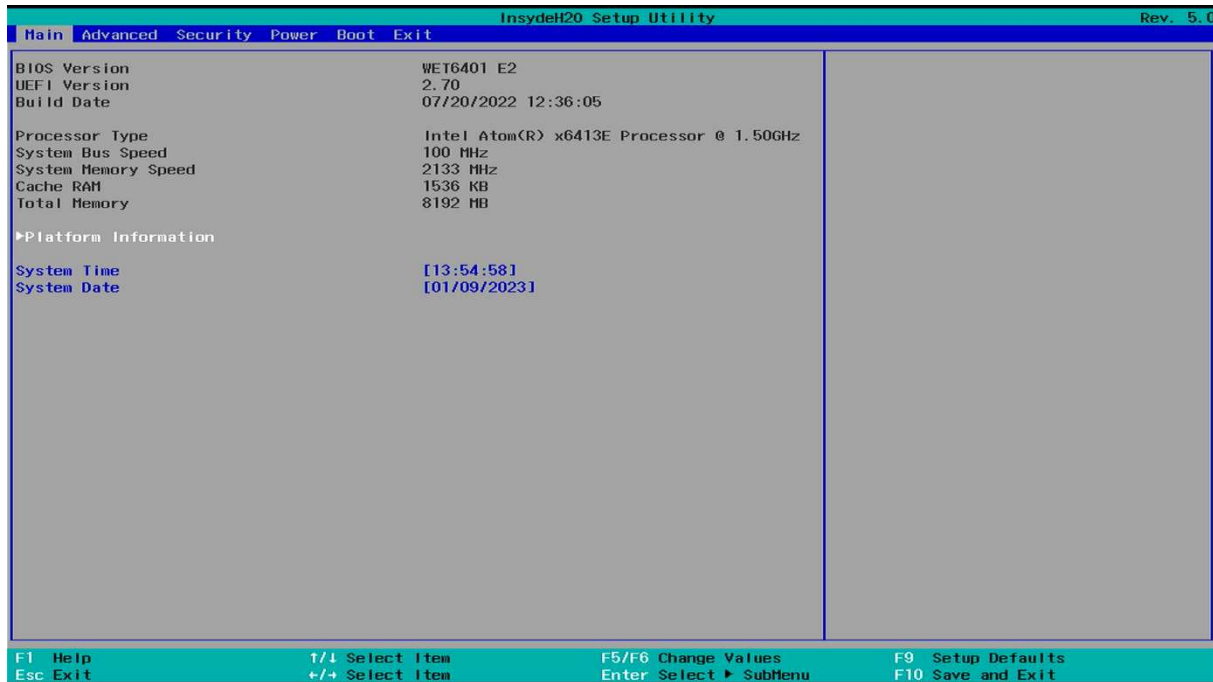
Press Page Up/Page Down or +/- keys to modify the BIOS parameters for the active option.

### **6.1 Enter BIOS**

Power on the computer and press <Del> key immediately to enter Setup.

If the message disappears before your 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.

## 6.2 BIOS menu and function keys



In the above BIOS Setup main menu of, you can see several options.

We will explain these options step by step in the following pages of this chapter, but let us first see a short description of the function keys you may use here:

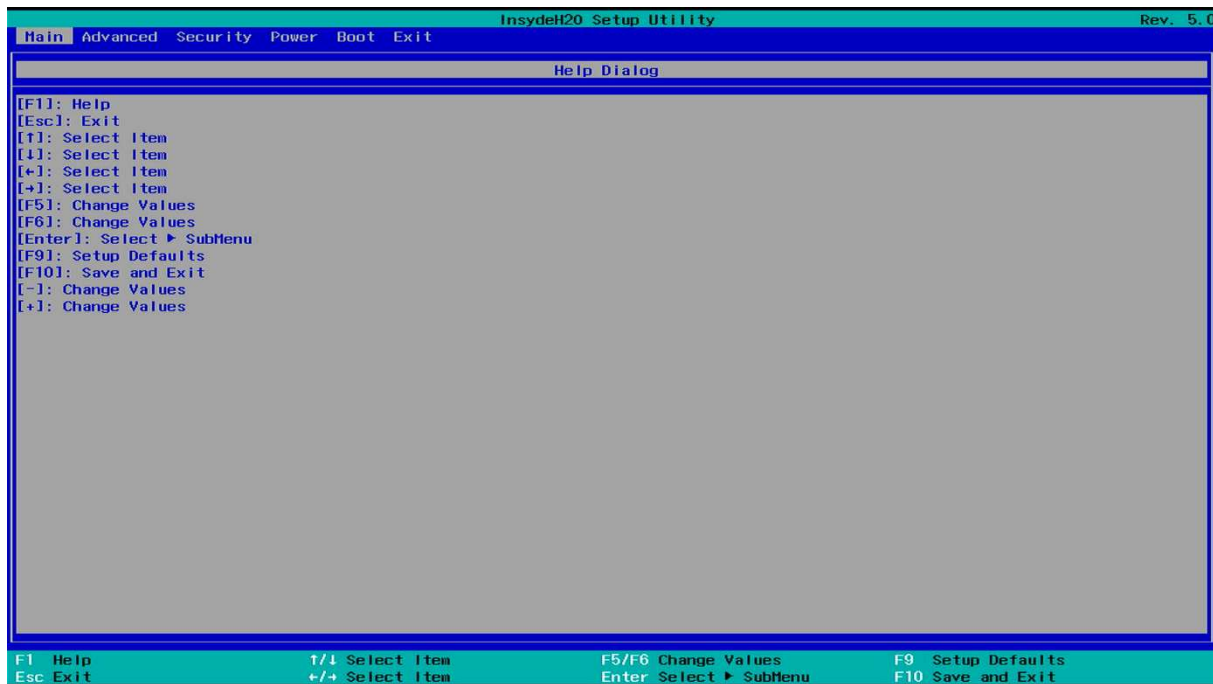
- Press ←→ (left, right) to select screen;
- Press ↑↓ (up, down) to choose, in the main menu, the option you want to confirm or to modify.
- Press <Enter> to select.
- Press <+>/<-> keys when you want to modify the BIOS parameters for the active option.
- [F1]: General help.
- [F2]: Previous value.
- [F3]: Optimized defaults.
- [F4]: Save & Reset.
- Press <Esc> to quit the BIOS Setup.

There are six menu bars on top of 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 setting, loading and exit options.

The selected one is highlighted.

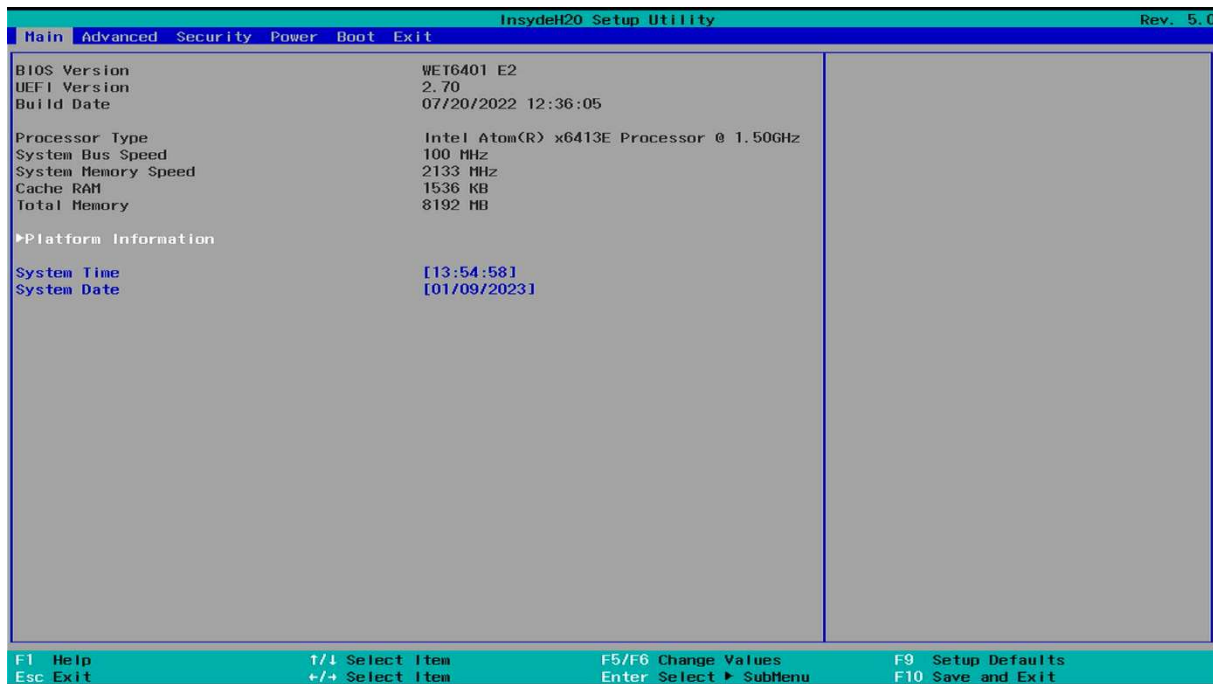
## 6.3 BIOS Help



### Status Page Setup Menu/ Option Page Setup Menu

Press F1 to pop up 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>.

## 6.4 Menu



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

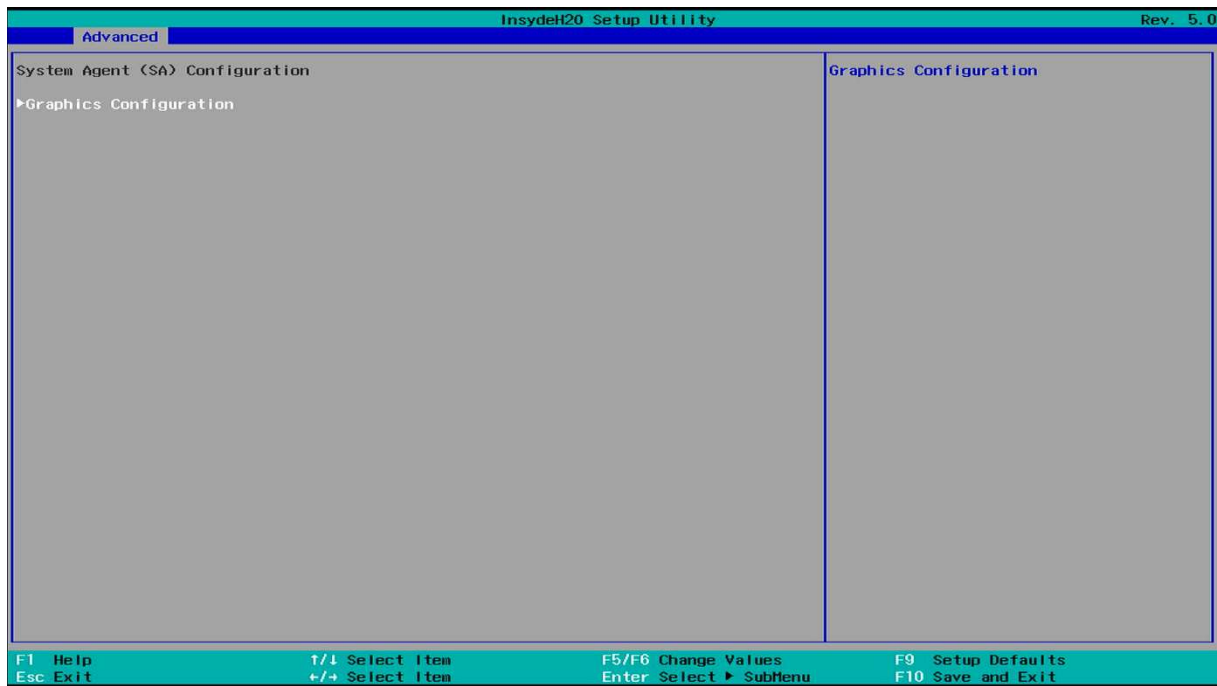
### System Date

Set the Date. Please use [Tab] to switch between data elements.

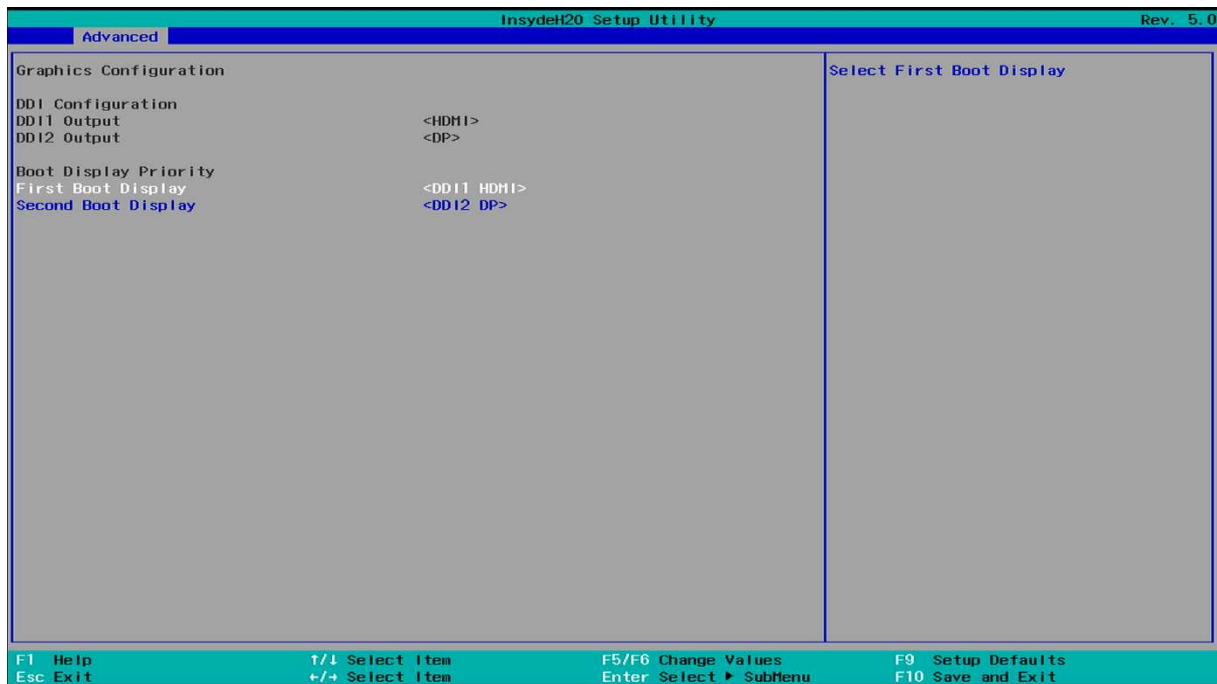
### System Time

Set the Time. Please use [Tab] to switch between data elements.

## 6.5 Advanced



## Video Configuration:



### Graphic Configuration

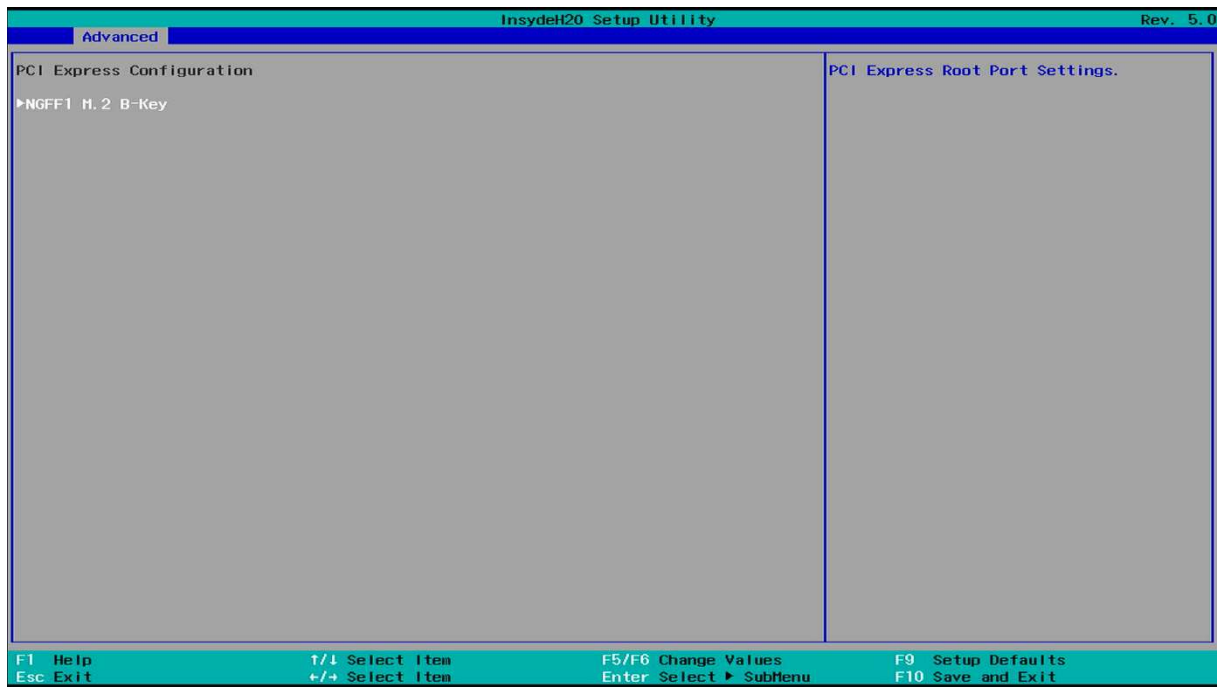
#### First Boot Display

To select First Boot Display priority, there are eDP, DDI1 HDMI, DDI2 HDMI, default is HDMI.

#### Second Boot Display

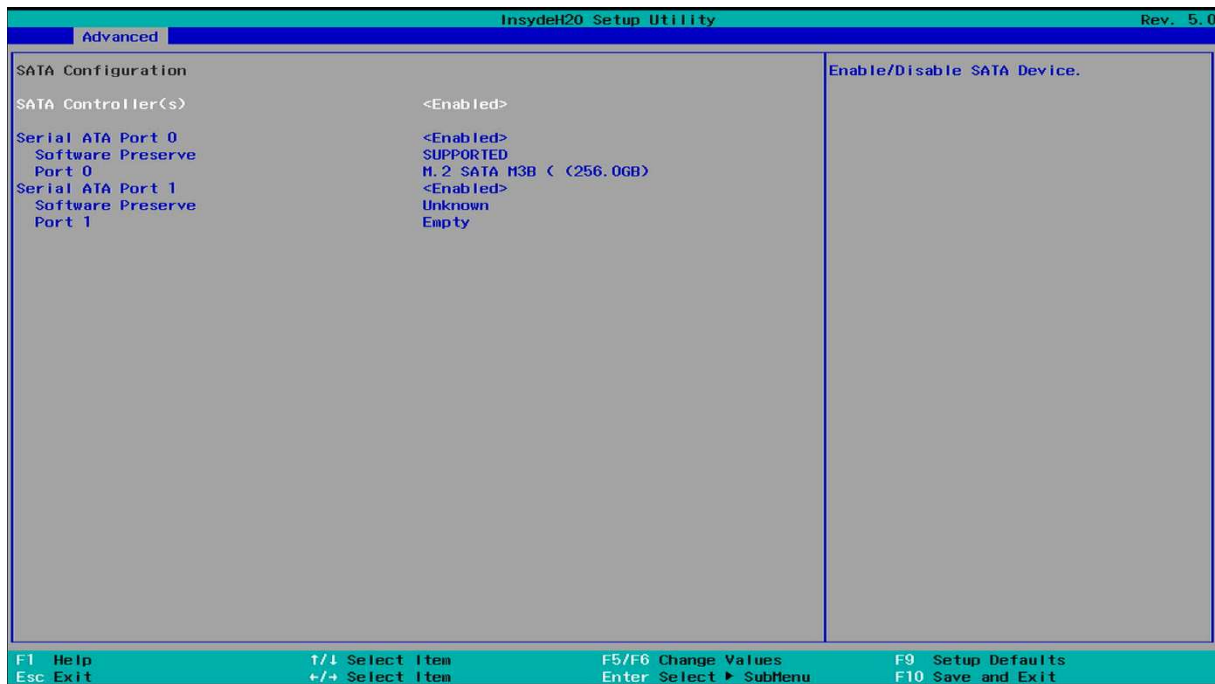
To select Second Boot Display priority, there are DDI1 HDMI, DDI2 HDMI, default is DP

## PCI Express Configuration:



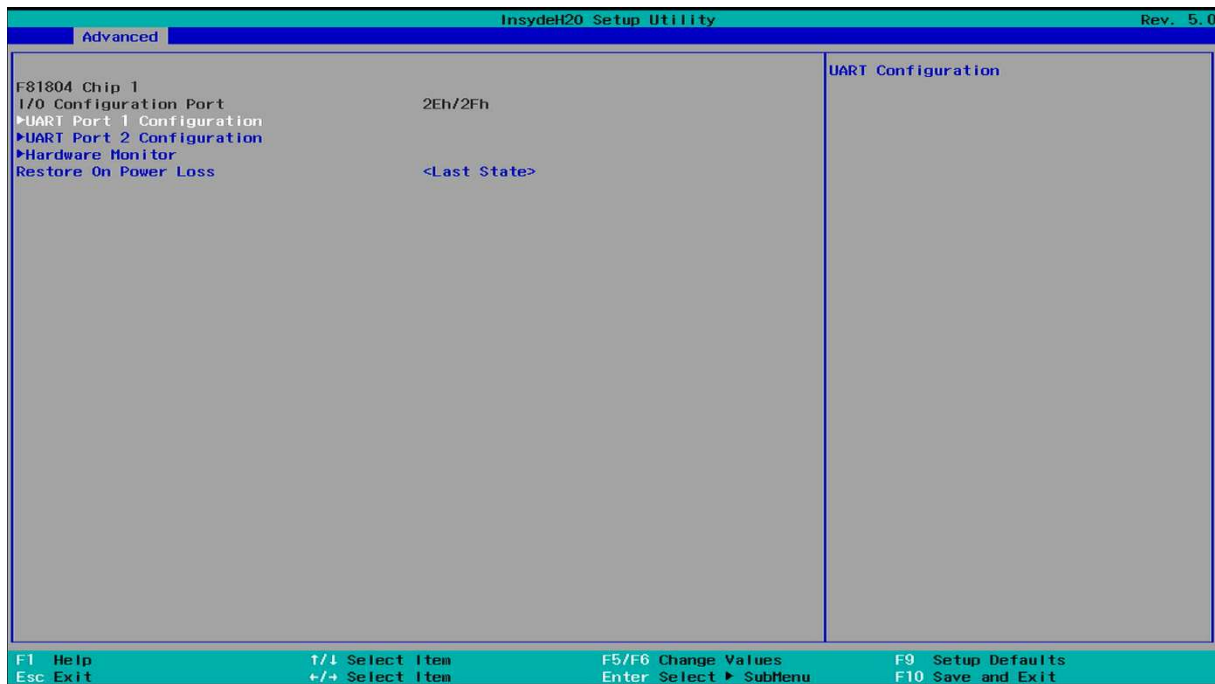


### SATA Drives Configuration:



To select SATA1 & NGFF1 M.2 SATA device enabled or not.

## SIO FINETEK 81804:



### Serial Port 1/2

Use this item to enable or disable serial port (COM1 or COM2). Enabled is the default.

### Serial Port 1/2 Base IO Address / Interrupt

Use this item to select an optimal setting for super IO device.

These optional settings are:

IO=3F8h; IRQ=3,4

IO=3E8h; IRQ=3,4

IO=2E8h; IRQ=3,4

IO=2F8h; IRQ=3,4

The port A default value as IO=3F8h; IRQ=4

The port B default value as IO=2F8h; IRQ=3

### Serial Mode

RS232 driver (default): When hardware is set to RS232/RS422 mode, please enter to RS232 driver.

RS485 driver: When hardware set to RS485 mode, please enter to RS485 driver. It enables the auto flow function for RS485.

### Restore on Power Loss

This item specifies whether your system will reboot after a power failure or interrupt occurs.

[Last state] Restores the system to the status before 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.

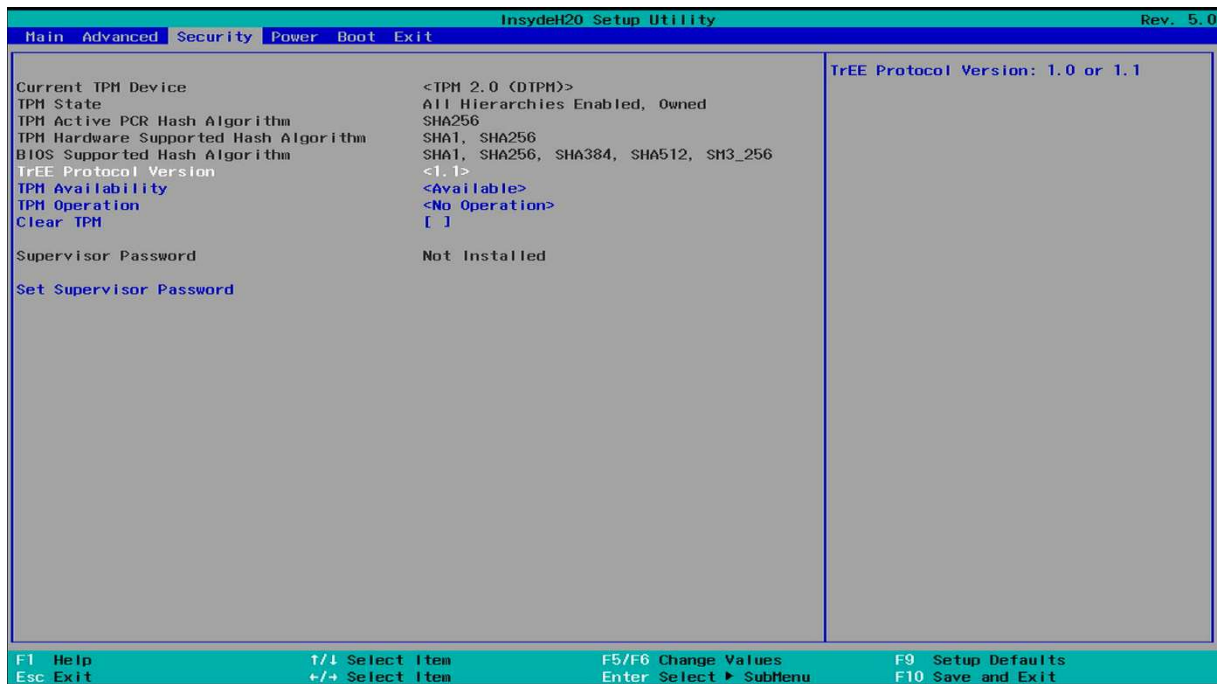
### Hardware Monitor

These are system voltage and temperature message.

The voltage shows the 3 type: VCC3, VCORE, VNN

The Temperature is measured via separate sensor, not by the CPU.

## 6.6 Security

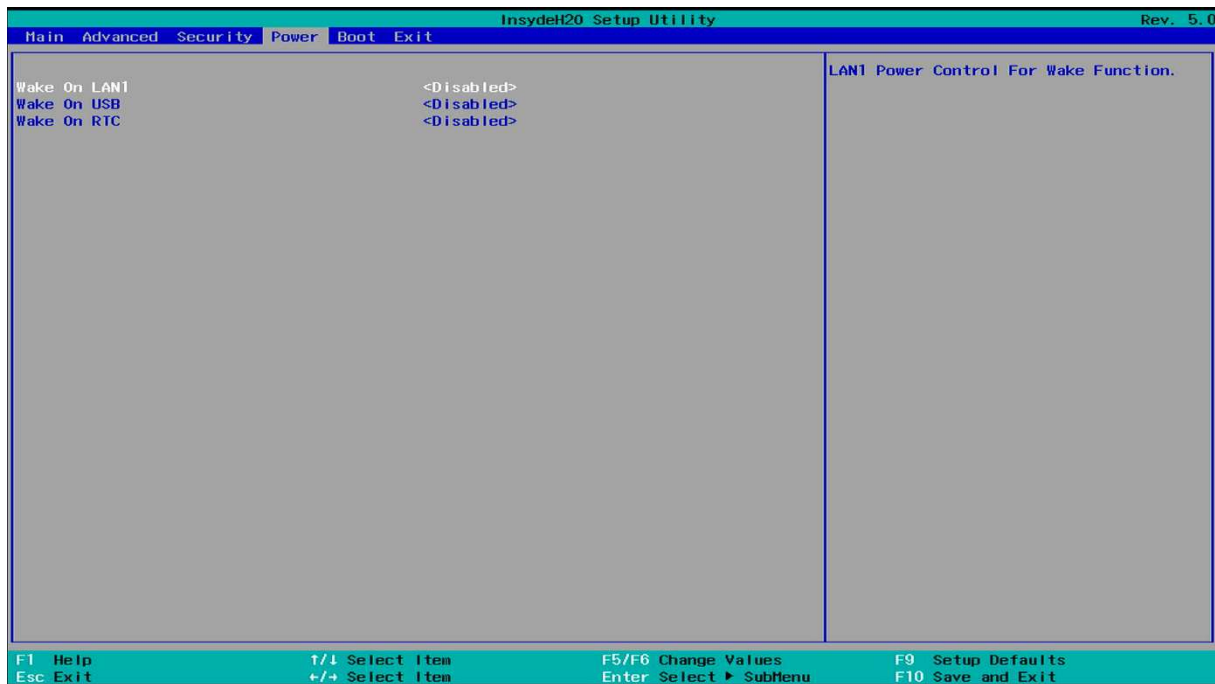


### Supervisor Password

To set up a Supervisor password

1. Select Supervisor Password.  
The screen then pops up an Create New Password dialog.
2. Enter your desired password that is no less than 3 characters and no more than 10 characters.
3. Hit [Enter] key to submit.

## 6.7 Power



### Wake on LAN

Wake On LAN from LAN1 when system in S3 or S5 state or both.

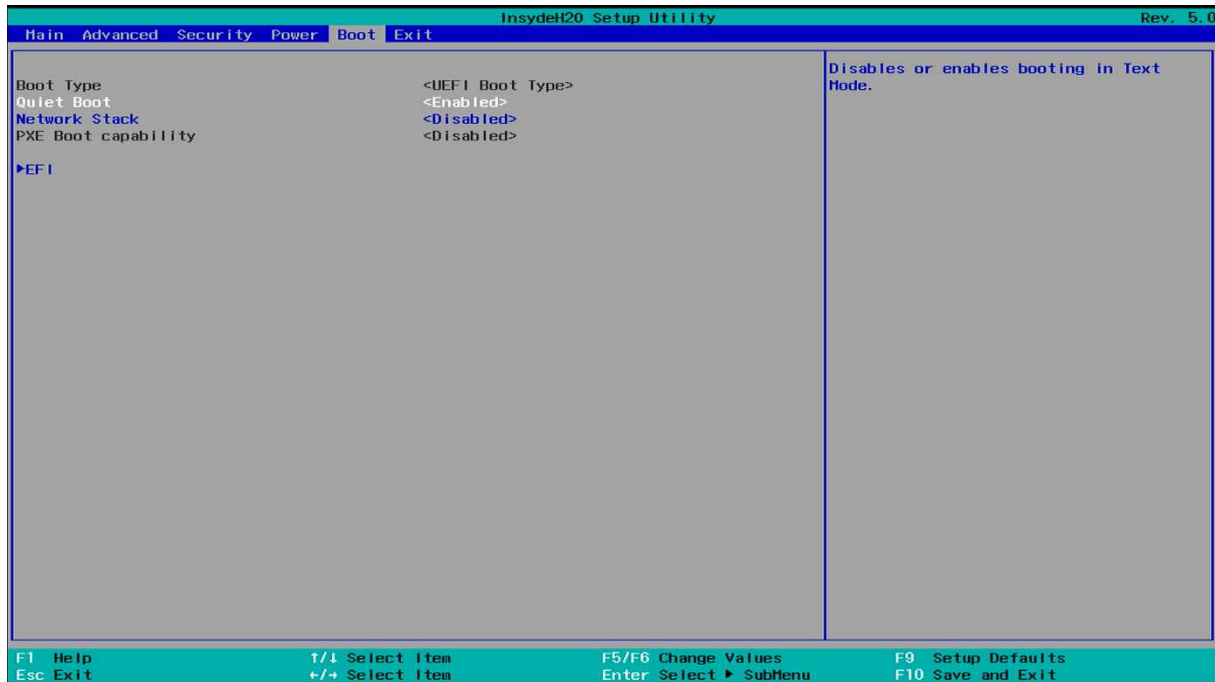
The optional settings: S3, S5, S3 / S5, Disabled (default)

### ACPI S3

Select ACPI sleep state (S3) support.

The optional settings: Enabled or Disabled (default)

## 6.8 Boot



### Boot type

The Arrakis Mk4 is a UEFI Boot only system

### Quiet Boot

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

### Network Stack

If use PXE function need enable otherwise disable (default)

### PXE Boot capability

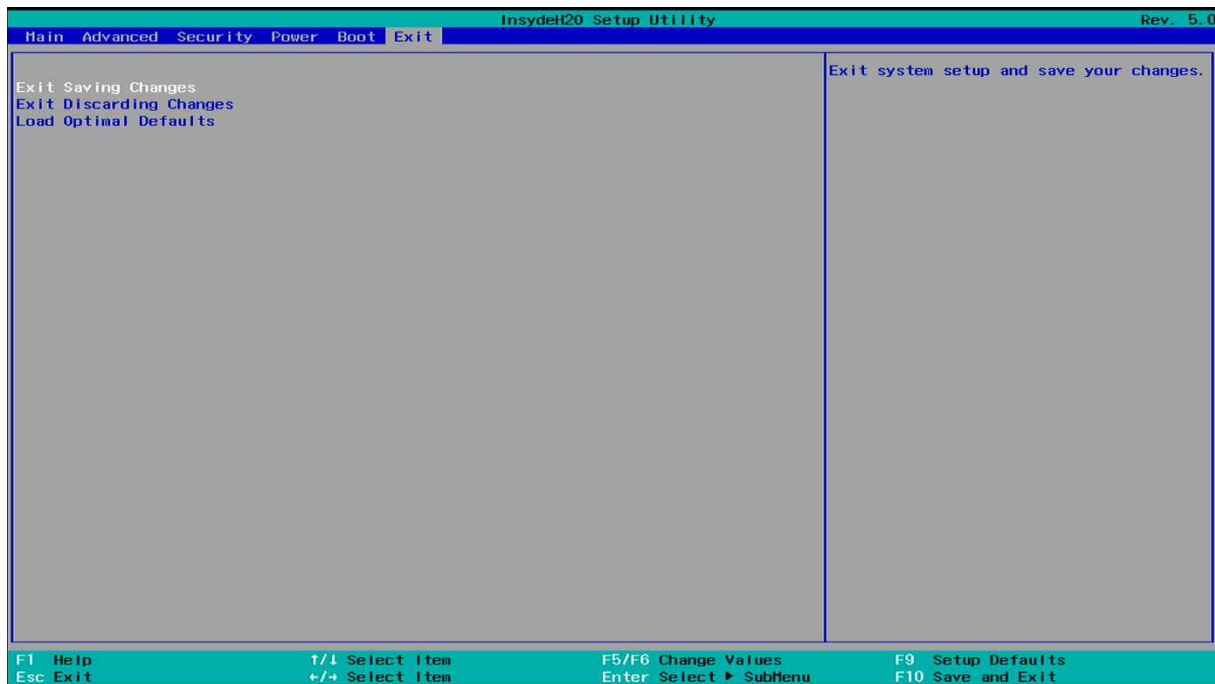
The item determines what protocol operation in PXE, these values are: Disabled (default), UEFI: IPv4, UEFI: IPv6

This is not modifying item. When above item settings disable.

### EFI

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

## 6.9 Exit



### **Exit Saving Changes**

This item allows user to reset the system after saving the changes.

### **Save Change Without Exit**

This item allows user to saving the changes but doesn't restart.

### **Exit Discard Changes**

This item allows user restart the system but no saving the changes

### **Load Optimal Default**

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

### **Discard Changes**

Use this item to cancel all the setup options.

## 7 Driver Installation

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

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



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

## 8 Appendix A Power consumption

| Item             | Specification            |
|------------------|--------------------------|
| CPU              | Atom® x6413E             |
| RAM              | DDR4 8GB 2400MHz         |
| Operation System | Windows 10 IoT 2019 LTSC |
| Test Program     | Passmark                 |
| NVMe             | 64GB                     |

**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     |

**The Power Consumption depends on options and Software.**



## 9 Appendix B F75111N DIO & Watchdog device

The Arrakis Mk4 is available with optional DIO Ports, this chapter introduces programming.

### 9.1 Watchdog timer under DOS:

You can download the Sources in the Driver Download Folder in Chapter 9 in the DIO folder.

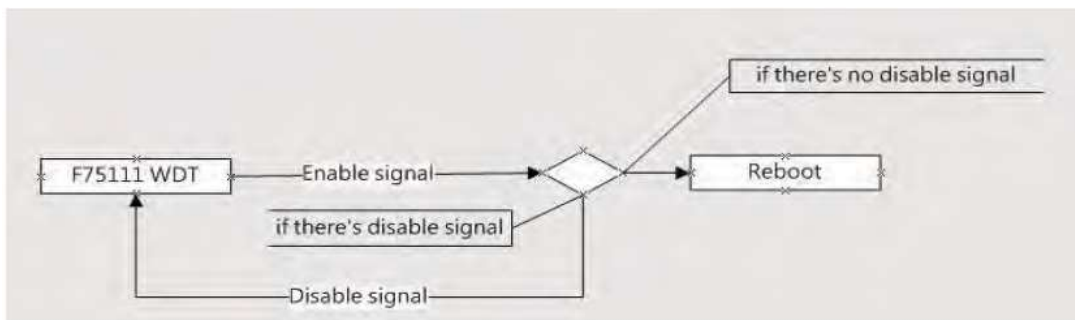
Source file: F75111\_Dos\_Src.rar

Binary file: F75111\_Dos\_Bin.rar

USERNAME & PASSWORD: sf

How to use this Demo Application:

1. Boot Ms-Dos Operating System
2. execute "75WDT.EXE" binary file
3. Input 1 to Enable WDT timer or input 0 to Disable it.
4. Input the numbers of second for chip countdown and Reset Computer



#### Introduction:

How to use the Watchdog Time Demo in different ways:

```

Writel2CByte(I2CADDR, CONFIG, 0x03); //Set Watch Dog Timer function
Writel2CByte(I2CADDR, WDT_TIMER, timer); //Set Watch Dog Timer range from 0-255.
Writel2CByte(I2CADDR, WDT_TIMER_CTL, 0x73); //Enable Watch Dog Timer in second and pulse mode
  
```

Or

```

Writel2CByte(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;
}
  
```

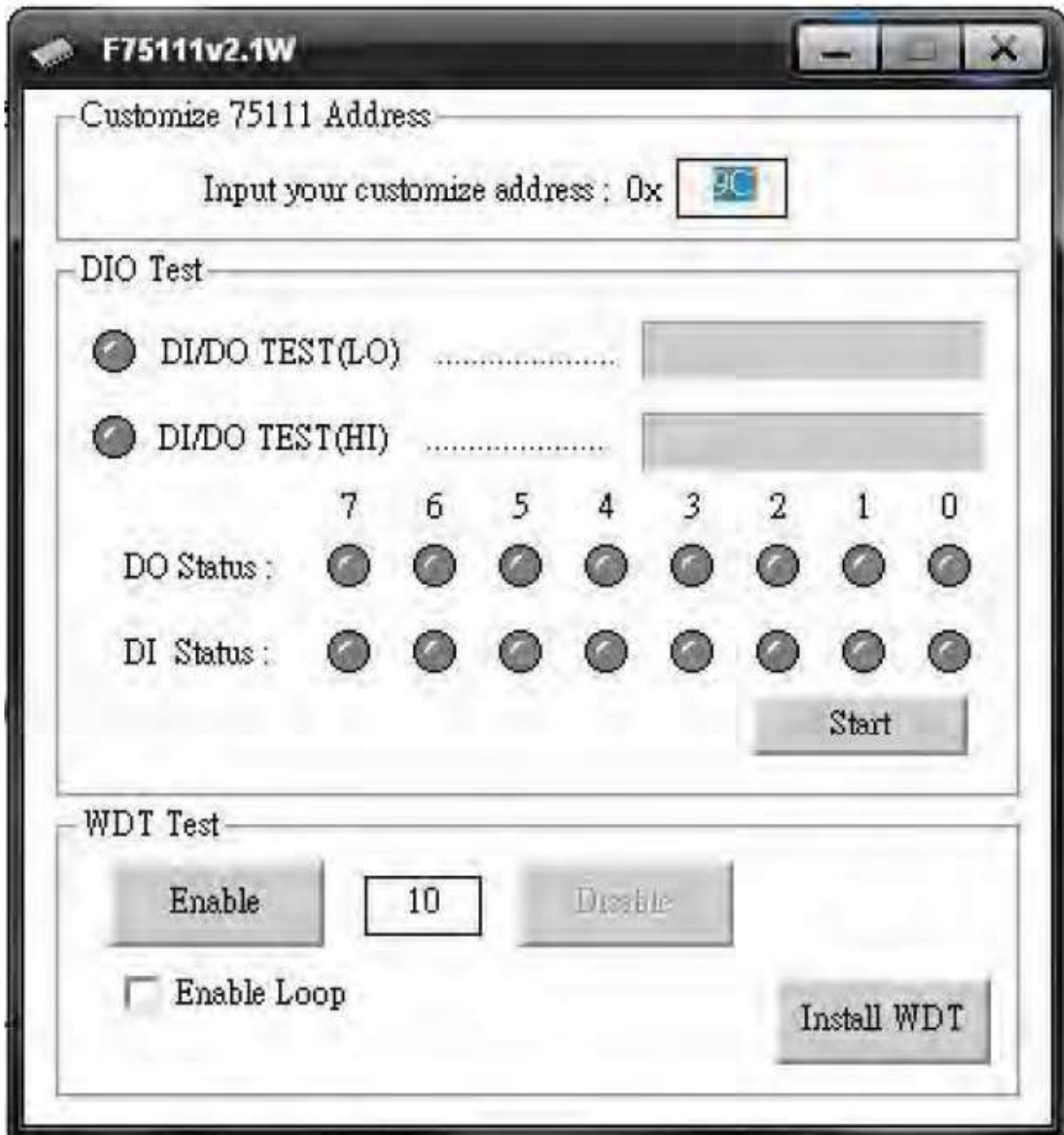
### 9.2 Watchdog Timer and DIO under Windows:


You can download the Sources in the Driver Download Folder in Chapter 9 in the DIO folder.

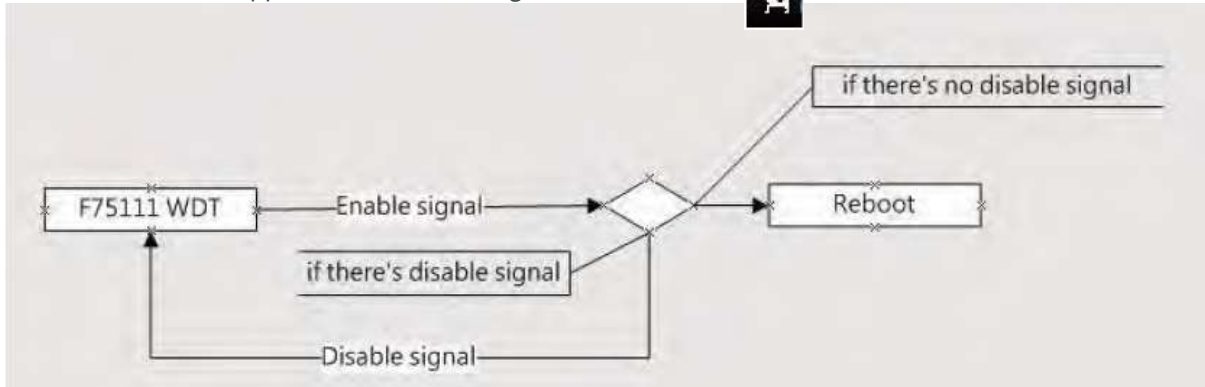
Source file: F75111\_DIOSrc.rar

Binary file: F75111\_DemoBin.rar  
USERNAME & PASSWORD: sf

How to use the Demo Application:



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



F75111 will send "F75111\_SetWDTEnable(BYTE byteTimer)" including a parameter "timer", if there's no disable signal (F75111\_SetWDTDisable()) to stop it before timer countdown to 0, System will reboot.

if there's disable signal received, resent Enable WDT signal to prevent a reboot loop.

#### **Introduction:**

#### **Initial Internal F75111 port address (0x9c)**

Define GPIO1X, GPIO2X, GPIO3X to input or output and Enable WDT function pin

#### **Set F75111 DI/DO (sample code as below Get Input value/Set output value)**

DO: InterDigitalOutput(BYTE byteValue))

DI: InterDigitalInput()

#### **Enable/Disable WDT:**

Enable: F75111\_SetWDTEnable (BYTE byteTimer)

Disable: F75111\_SetWDTDisable ()

#### **PULSE mode:**

#### **Sample to setting GP33, 32, 31, 30 output 1mS low pulse signal.**

```

{
this->Write_Byte(F75111_INTERNAL_ADDR, GPIO3X_PULSE_CONTROL, 0x00); //This is setting low pulse
output
this->Write_Byte(F75111_INTERNAL_ADDR, GPIO3X_PULSE_WIDTH_CONTROL, 0x01); //This selects the pulse
width to 1mS
this->Write_Byte(F75111_INTERNAL_ADDR, GPIO3X_CONTROL_MODE, 0x0F); //This is setting the GP33, 32,
31, 30 to output function.
this->Write_Byte(F75111_INTERNAL_ADDR, GPIO3X_Output_Data , 0x0F); //This is setting the GP33, 32, 31,
30 output data.
}
  
```

#### **Sample to setting GP33, 32, 31, 30 output 1mS low pulse signal.**

```
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
    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 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;
}
```

#### 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 confi gure
}
```

#### Disable WatchDog:

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

### 9.3 IO Device: F75111 VB6 under Windows:

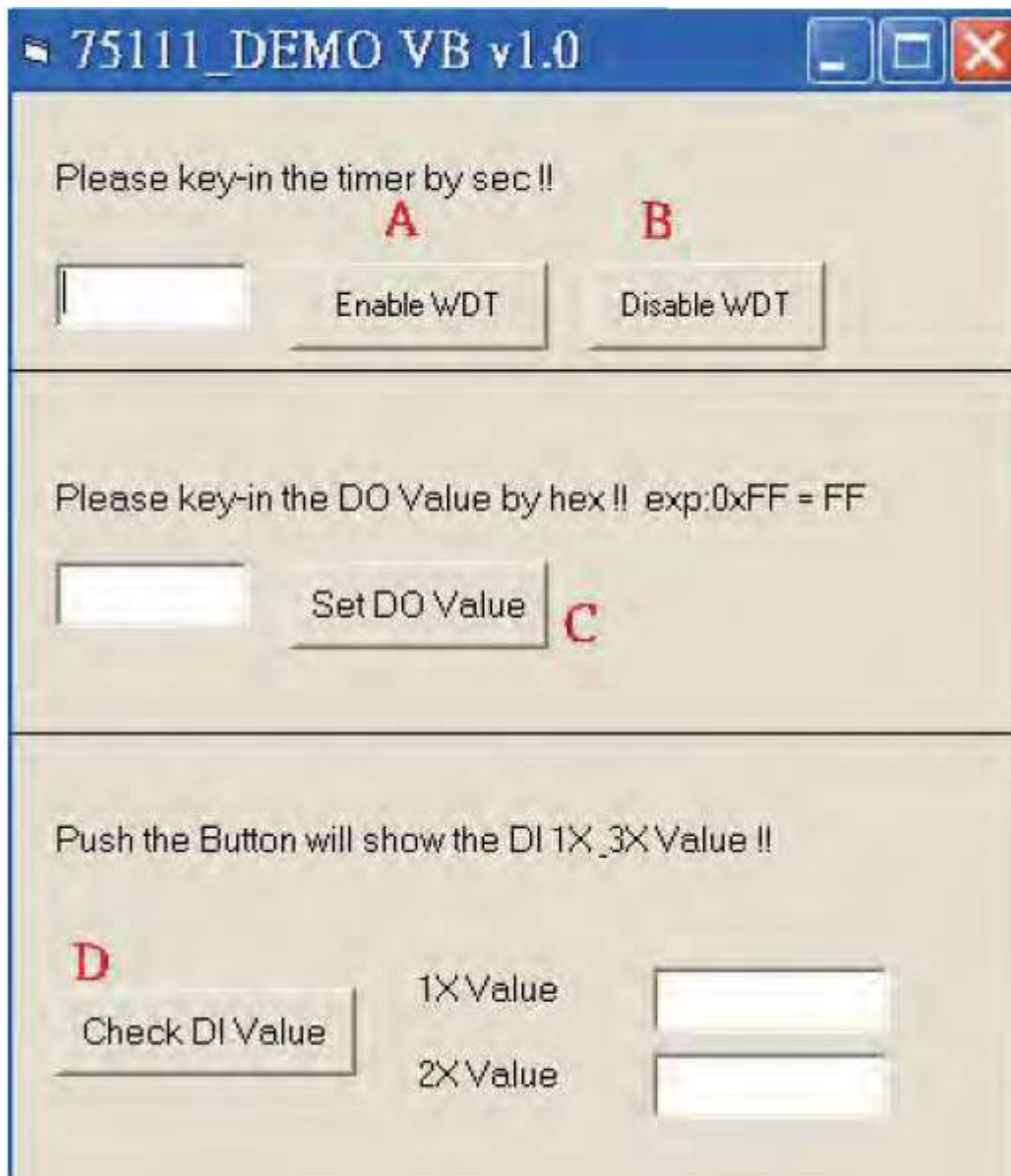
You can download the Sources in the Driver Download Folder in Chapter 9 in the DIO folder.

Source file: 75111\_VB\_v10.rar

Binary file: 75111\_VB\_Src.rar111\_DemoBin.rar

USERNAME & PASSWORD: sf

**How to use this Demo Application:**



- A. Function - Enable WDT timer, Key-in the value by seconds then system will reboot after value which you key-in in left text box!
- B. Function - Disable WDT timer, Push down the button then WDT timer value will be clear !!
- C. Function - Set DO Value, Key-in the DO value by hex then push the button !!
- D. Function - Check DI Value, the right side two text box will display DI 1X & 2X Value when you push down the button!!

## 9.4 SDK Function Introduction:

### Function EnableWDT:

```
Function EnableWDT(timer As Integer)
Call Writel2CByte(&H3, &H3)
Call Writel2CByte(&H37, timer)
Call Writel2CByte(&H36, &H73)
End Function
```

### Function DisableWDT:

```
Function DisableWDT()
Call Writel2CByte(&H36, &H0)
End Function
```

### Function SetDOValue:

```
Function SetDOValue(dovalue As Integer)
Call Writel2CByte(&H23, &H0)
Call Writel2CByte(&H20, &HFF)
Call Writel2CByte(&H2B, &HFF)
Call Writel2CByte(&H21, dovalue)
End Function
```

### Function SetDOValue:

```
Function SetDOValue(dovalue As Integer)
Call Writel2CByte(&H23, &H0)
Call Writel2CByte(&H20, &HFF)
Call Writel2CByte(&H2B, &HFF)
Call Writel2CByte(&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
```

## 9.5 Watchdog Timer and DIO under Linux:

You can download the Sources in the Driver Download Folder in Chapter 9 in the DIO folder.

Source file: F75111v2.0L.tar.gz

Binary file: F75111v2.0LBin.tar.gz

USERNAME & PASSWORD: sf

### How to compile source code:

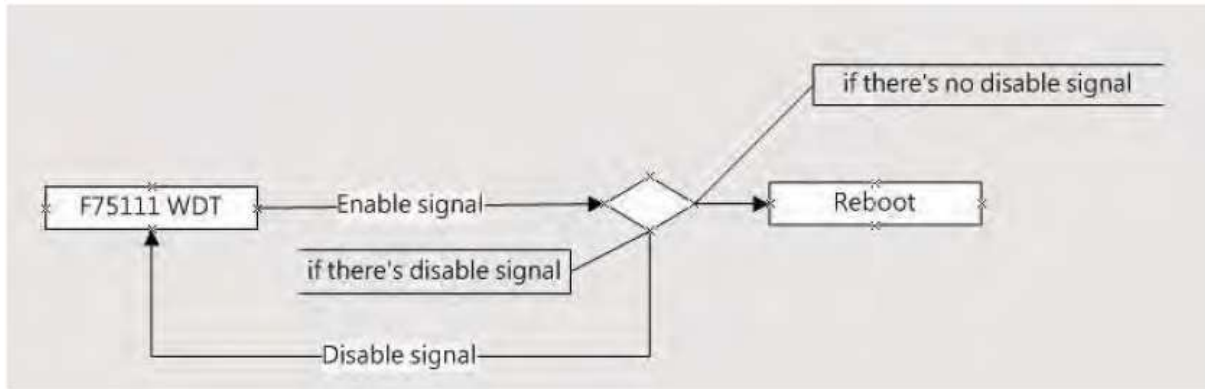
1. Compile source code with Code::Blocks  
download and install the Code::Block with command "apt-get install codeblocks"  
Open an exist project(F75111.cbp) in Code::Blocks, click the compile button  
( add an option 'pkg-config --libs gtk+-2.0 gthread-2.0' in "Project->Build Option->Linker Setting->Other linker option")
2. Compile source code with "make"  
1.cd F75111  
1.make  
1.src/f75111 // execute the binary file

### How to use this Demo Application:



1. Press the "Start" button to test DIO function
2. Press the "Enable" button to test WDT function
3. Press the "Disable" button to disable WDT
4. Check the "Enable Loop" box and press "Enable" to do WDT loop test
5. Press "Install" to set the system to autorun this application when booting, press "Uninstall" to remove this application when booting.
6. If WDT enable, system icon will be blinking.





F75111 will send "F75111\_SetWDTEnable(BYTE byteTimer)" including a parameter "timer", if there's no disable signal (F75111\_SetWDTDisable()) to stop it before timer countdown to 0, System will reboot.

If there's disable signal received, resent Enable WDT signal to prevent a reboot loop.

### Introduction:

#### IO function in the file SMBus.c:

```

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

### 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 confi gure
}
```

### Disable WatchDog:

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