

Arrakis MK3 Series

Version:
v1.0.0

Date:
10.07.2024



Contents

1	Copyright	3
2	CE Notice	4
3	Safety Instructions	5
4	Product Introduction	6
4.1	About Arrakis MK3	6
4.2	Product Specification	7
5	System Information	8
5.1	System Drawing	8
5.2	Mainboard Block Diagram	10
5.3	System	11
6	Mainboard Information	12
6.1	Mainboard Layout View	12
6.2	List of Jumpers	13
6.3	Jumper Setting Description	13
6.4	JSB1: CMOS Data Clear	13
6.5	JVL1: LCD panel power select	14
6.6	JVC1/JVC2: COM1/2 Ring Indicator and power select	15
7	Connectors	16
7.1	List of Connectors	16
7.2	DC Power In	17
7.3	CMOS Battery holder	17
7.4	COM Ports	18
7.5	LAN Ports	19
7.6	USB Ports	19
7.7	mPCIe MiniExpressCard	19
8	BIOS	21
8.1	Introduction	21
8.2	Enter BIOS	21
8.3	BIOS menu and function keys	22
8.4	BIOS Help	23
8.5	Menu	24
8.6	Advanced	25
8.7	Security	32
8.8	Power	33
8.9	Boot	34
8.10	Exit	35
9	Driver Installation	36
10	Appendix A Power consumption	37
11	Appendix B F75111N DIO & Watchdog device	38
11.1	Watchdog timer under DOS	38
11.2	Watchdog Timer and DIO under Windows	39

11.3	IO Device: F75111 VB6 under Windows	41
11.4	SDK Function Introduction	42
11.5	Watchdog Timer and DIO under Linux	43

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 CE Notice

This device complies with the requirements of the CE directive.

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 9V 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 60 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 MK3

Thank you for choosing Welotec ARRAKIS MK3. ARRAKIS MK3 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.

ARRAKIS MK3 is powered by a 1.6/2.0 GHz Quadcore Intel Atom® Processor E3950, and its standard Version has a SoDIMM socket for 1866MHz DDR3L Memory.

It is capable to support 1x HDMI and 1x Display Port Displays with the resolution up to 4K. The system features mSATA storage, one FullSize mPCIe socket and, with optional Expansion module, 2x FullSize mPCIe sockets with USB and 2 external SIM Sockets. Four USB 3.0, dual 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.

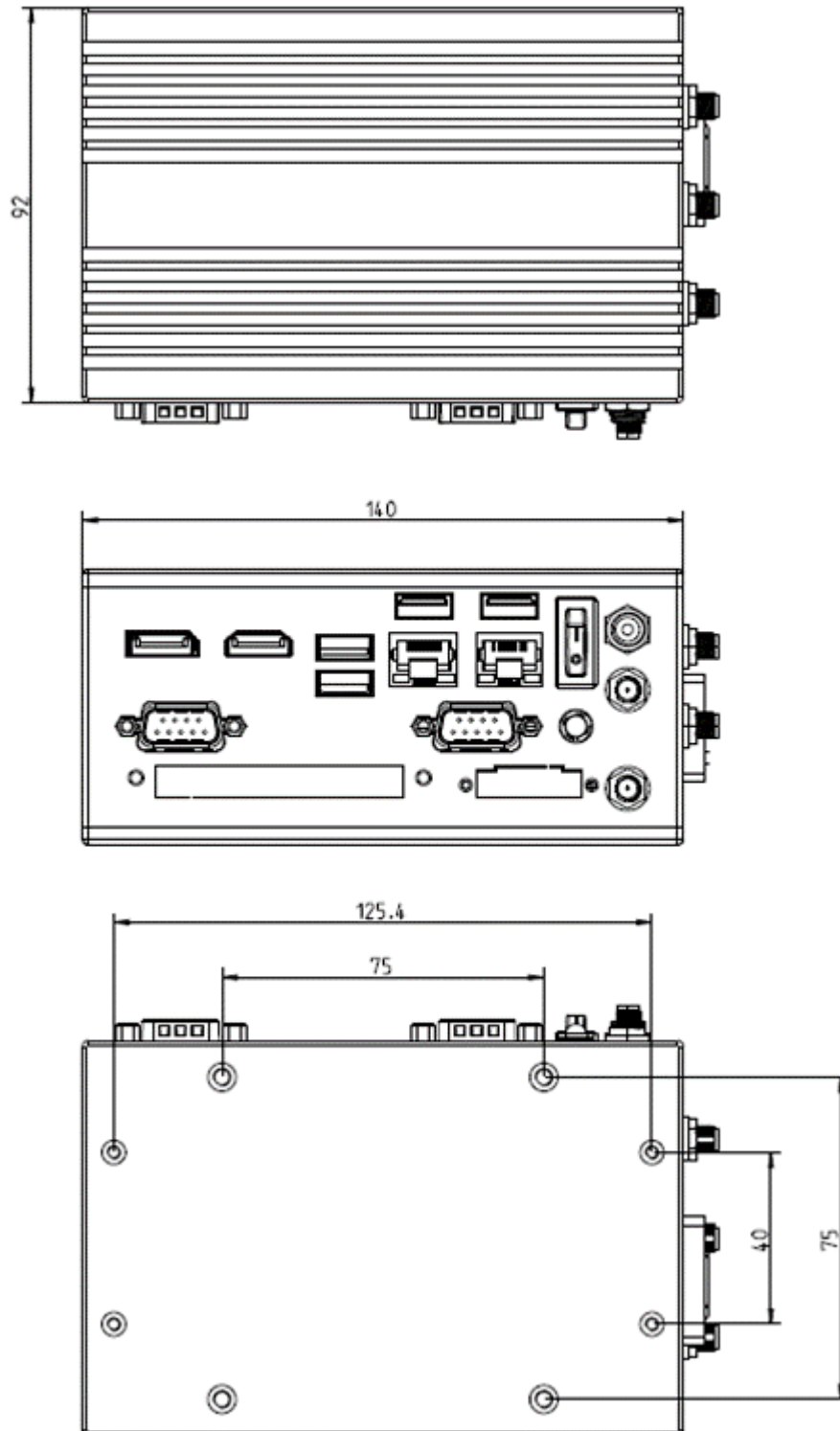
Combining the above-mentioned features with its aluminium housing and dust-proof design makes ARRAKIS MK3 a perfect industrial embedded system to operate in environments with harsh and critical conditions.

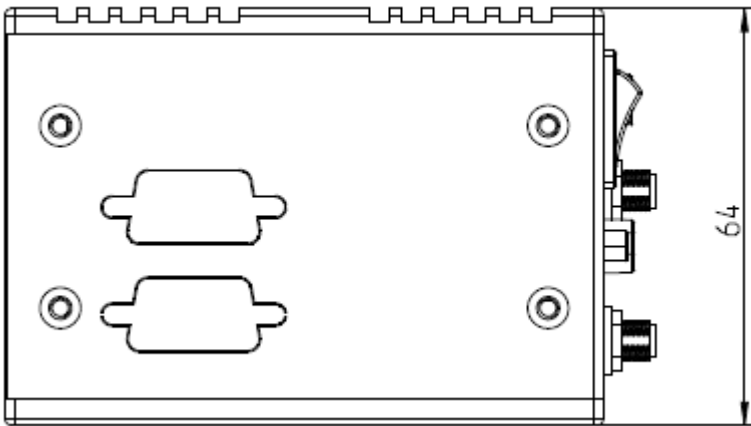
4.2 Product Specification

Technical Details		
Processor	CPU	Intel Atom® Quadcore Processor E3950 1.6/2.0 GHz (Std.)
Memory	RAM	1866 MHz DDR3L SoDIMM Socket, up to 8GB
Display	Max. Resolution	DP + HDMI, max. 4K
Storage		x1 2.5" SATA 3.0 drive bay (option)
		x1 Fullsize mSATA connector
I/O Physical	Displayport	x 1
	HDMI	x 1
	LAN	x 2 RJ45
	USB 3.0	x 4
	RS232/422/482	x 2 (optional 2 additional RS232/422/485)
Networking	LAN	x2 (10/100/1000 Base-T), Intel i210IT LAN chip
	WLAN	(Optional, mPCIe)
	WWAN	(Optional, 4G/5G, USB)
Other Expansion	SIM Slot	Other Expansion SIM Slot x2 push-push Type SIM Slot (only if equipped with 4G/5G)
Others	Others	Others Others Line in/out, Digital I/O, CAN and others (optional)
	Watch dog Timer	System Reset, Programmable via Software from 1 to 255 Seconds
Environment	Temperature	Environment Temperature -20° to 70° C Operation
		-20° to 80° C Storage
	Relative Humidity	5% to 95% non-condensing
Power	Power Supply	9-36V 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		All Windows 10 versions, Ubuntu Linux, others on request
Physical Characteristics	Material/Color	Aluminum / Steel, Silver
	Dimension (mm)	64 x 140 x 92 mm
Regulatory Compliance		CE

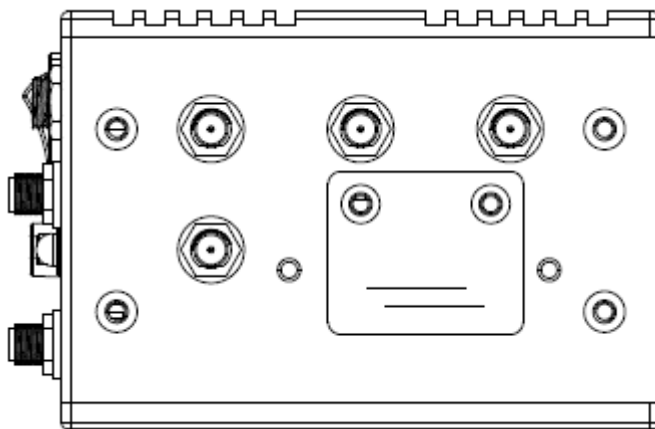
5 System Information

5.1 System Drawing

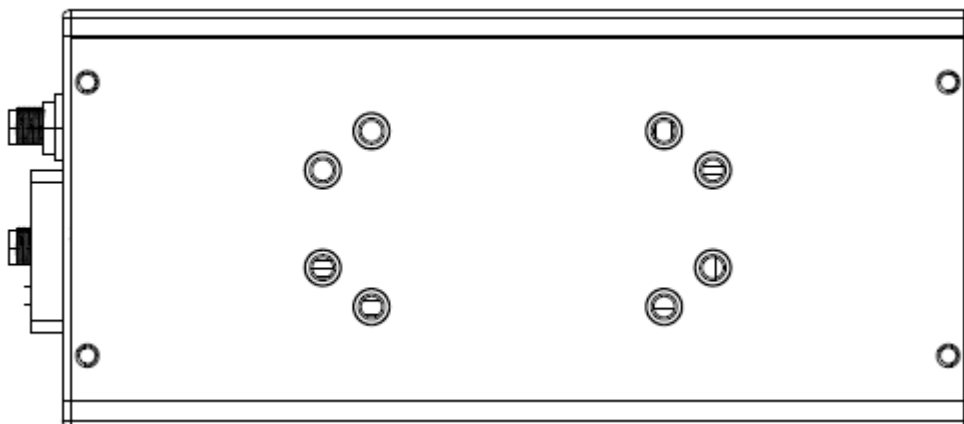




Bottom side



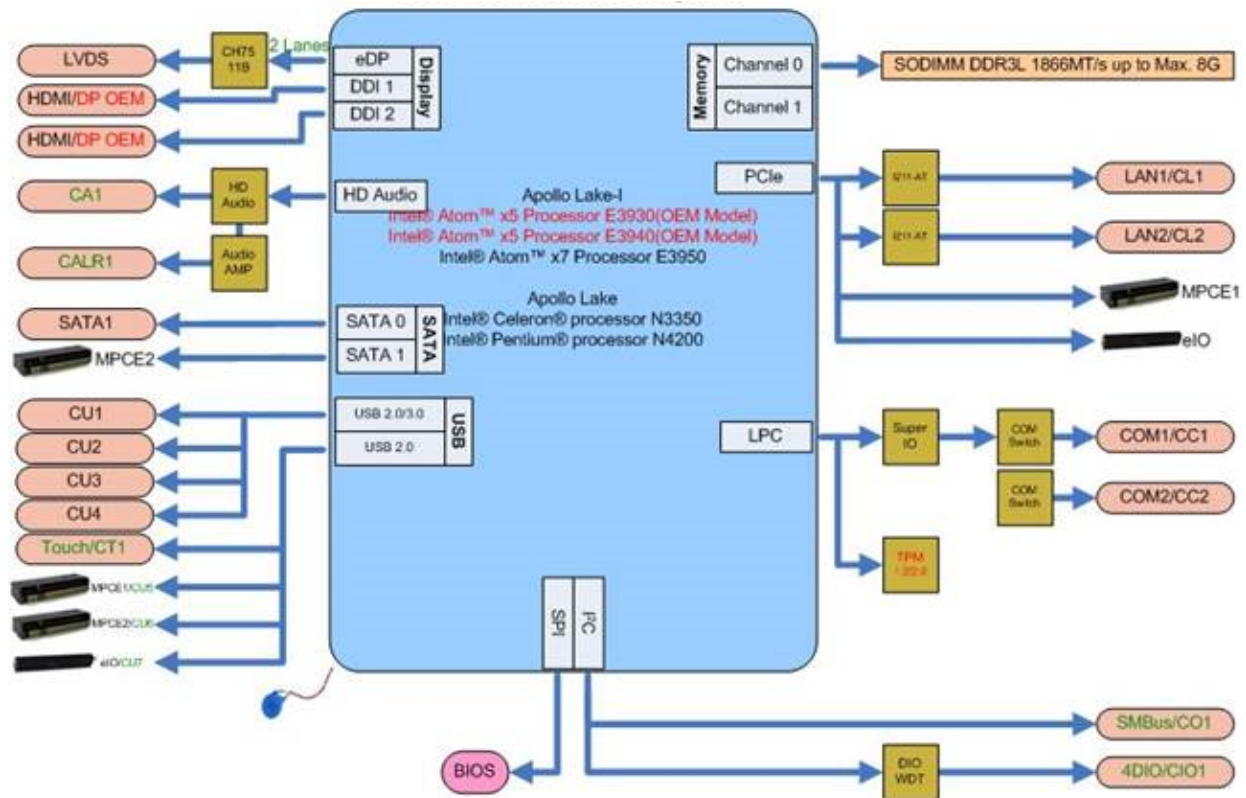
Top side



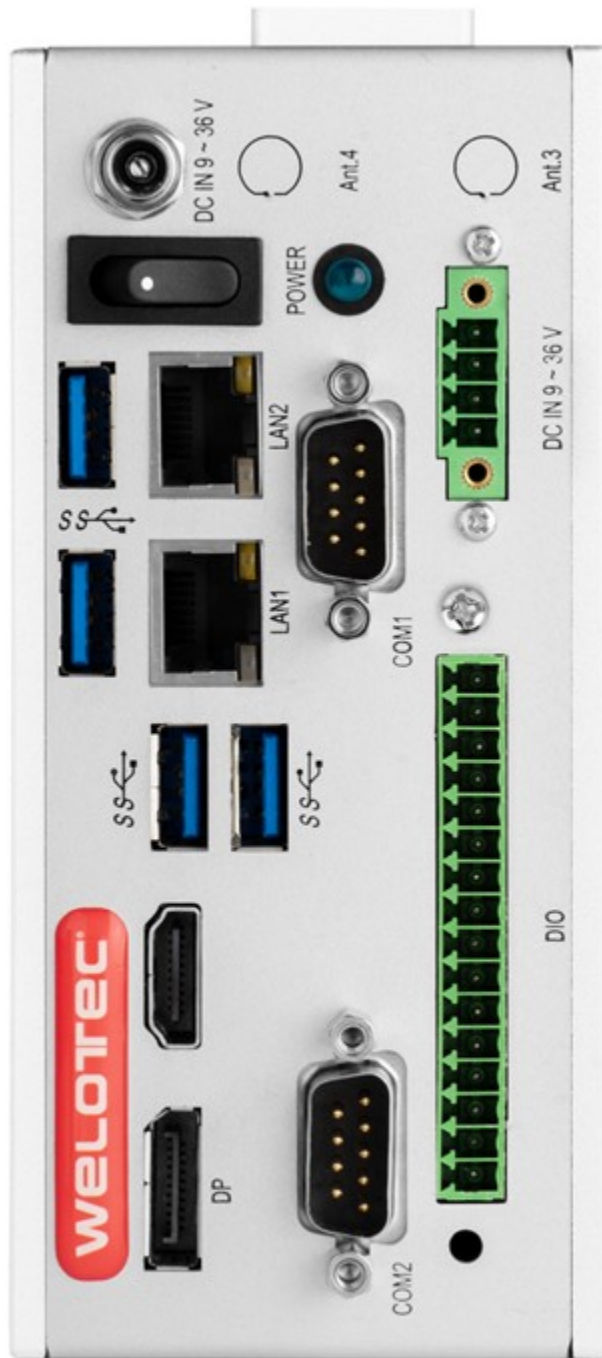
Rear side

5.2 Mainboard Block Diagram

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

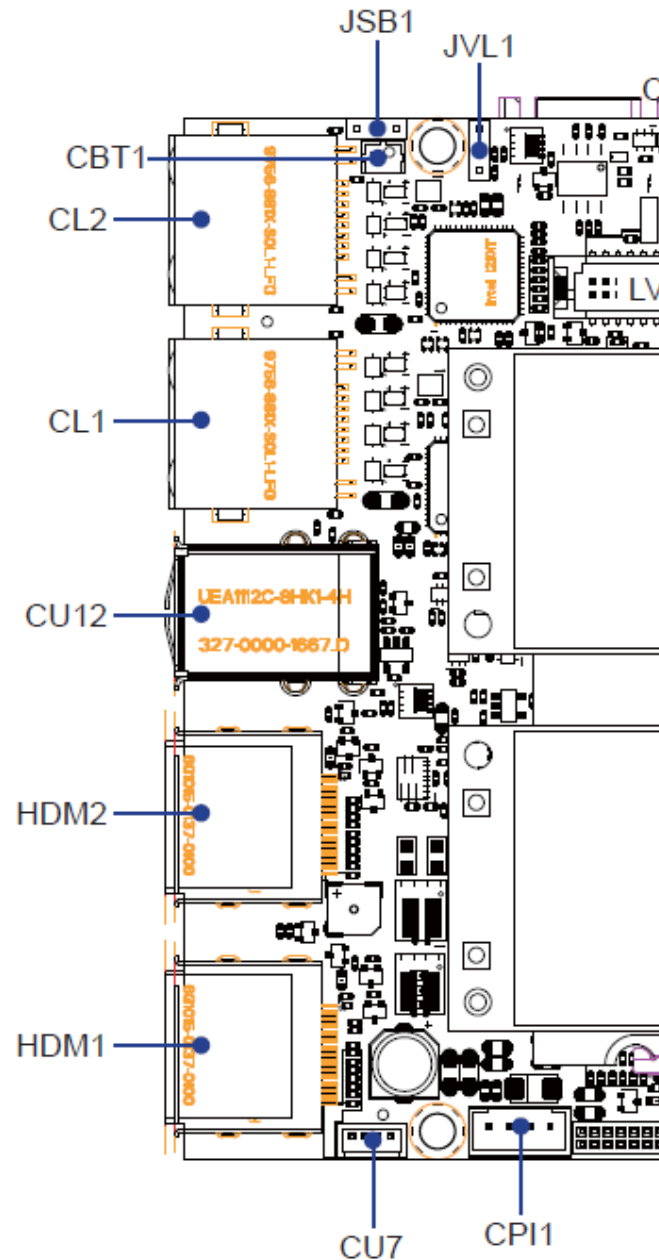


5.3 System



6 Mainboard Information

6.1 Mainboard Layout View



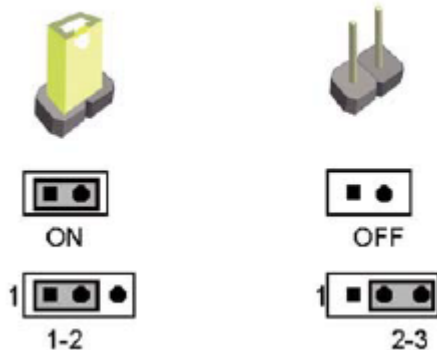
The following layout shows the connectors location on the mainboard.

6.2 List of Jumpers

JSB1: CMOS DATA clear JVL1: LCD Panel power select JVC1/JVC2: COM1/2 RI & power select

6.3 Jumper Setting Description

A jumper is ON as a closed circuit with a plastic cap covering two pins. A jumper is OFF as an open circuit without the plastic cap. Some jumpers have three pins, labelled 1, 2, and 3. You could connect either pin 1 and 2 or 2 and 3. The below figure shows the examples of different jumper settings in this manual.



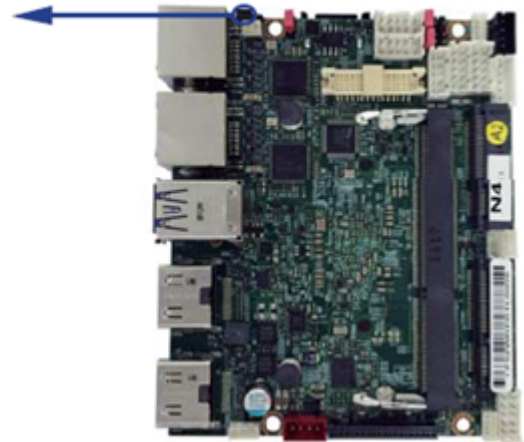
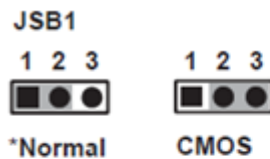
All jumpers already have its default setting with the plastic cap inserted as ON, or without the plastic cap as OFF. The default setting may be referred in this manual with a " * " symbol.

6.4 JSB1: CMOS Data Clear

A battery must be used to retain the motherboard configuration in CMOS RAM. Close Pin1 and pin 2 of JSB1 to store the CMOS data. To clear the CMOS, follow the procedures below:

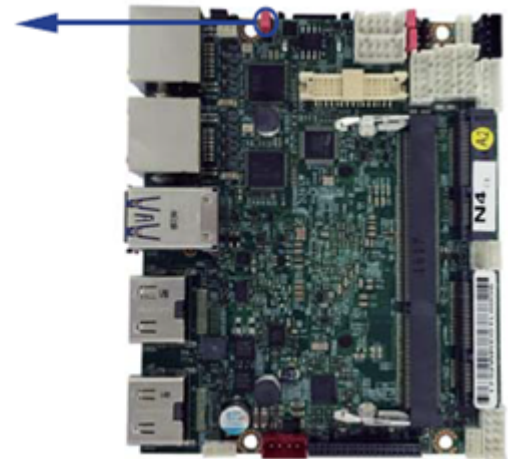
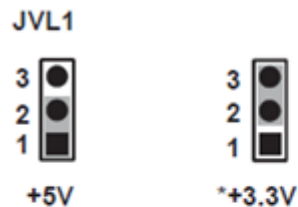
1. Turn off the system and disconnect the power
2. Remove DC 12V power cable from DC 12V power connector
3. Locate JSB1 and close pin 1-2 for few seconds
4. Return to default setting by close pin 1-2
5. Connect DC 12V power cable back to DC 12V Power connector

Do not clear CMOS unless you are a) troubleshooting or b) forgot the BIOS Password



JSB1	Description
*1-2	Normal set
2-3	CMOS data clear

6.5 JVL1: LCD panel power select



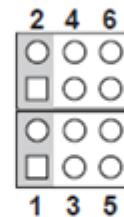
JVL1	Description
1-2	+5V
*2-3	+3.3V

6.6 JVC1/JVC2: COM1/2 Ring Indicator and power select

JVC1/JVC2	DESCRIPTION
*1-2	RI
3-4	+5V
5-6	+12V



JVC2



JVC1

*RI Signal



+5V



+12V

7 Connectors

This chapter provides all necessary information of the peripheral's connections, switches and indicators. Always turn the system off and disconnect power you install the peripherals.

7.1 List of Connectors

BAT1: Li 3V battery holder
CA3: Line-out/Line-in/Mic-in/SPDIF-out 2x5 pin (2.0mm) Wafer
CALR1: Amplifier Line-out Right/Left channel 4pin (1.25mm) wafer
CC1: COM1 DB9 Connector
CC2: COM 2x5pin (2.0mm) wafer
CBT1: Li 3V Battery 1x2pin (1.25mm) Wafer
CA3: Line-out/Line-in/Mic-in 2x5 pin (2.0mm) Wafer
CALR1: Amplifier 1x4 pin (1.25mm) Wafer
CC1: COM 2x5pin (2.0mm) wafer
CC2: COM 2x5pin (2.0mm) wafer
CFP1: Front Panel connector 2x5pin (2.0mm) wafer
CIO1: DIO 2x5pin (2.0mm) Wafer
CL1: LAN port 1 RJ45 Connector
CL2: LAN port 2 RJ45 Connector
CL11: LAN port 1 RJ45 2x4pin (2.0mm) wafer(option)
CL21: LAN port 2 RJ45 2x4pin (2.0mm) wafer(option)
CO1: I2C Bus 1x4pin (1.25mm) Wafer
CPI1: DC 12V-IN 1x4pin (2.0mm) Red wafer connector
CPO1: +12V/+5V power output 1x4pin (2.0mm) Black wafer connector
CU12: Dual USB3.0 Type A connector
CU3: USB2.0 Type A connector
CU4: USB3.0 Type A connector
CU5: USB 2.0 port 1x4pin (1.25mm) Wafer
CU6: USB 2.0 port 1x4pin (1.25mm) Wafer
CU7: USB 2.0 port 1x4pin (1.25mm) Wafer
LVDS1: LVDS 2x15pin (1.25mm) connector
CPP1: BKL power 1x5pin (2.0mm) Wafer
CT1: TOUCH 2x5pin (2.0mm) Wafer
SATA1: One SATA connector 7pin
MPCE1: Full size mini card port 1 sockets 52pin

MPCE2: Full size mini card port 2 sockets 52pin

HDMI1: HDMI connector

HDMI2: Display Port connector

EIO1: eIO port 2x20pin (1.27mm) Wafer

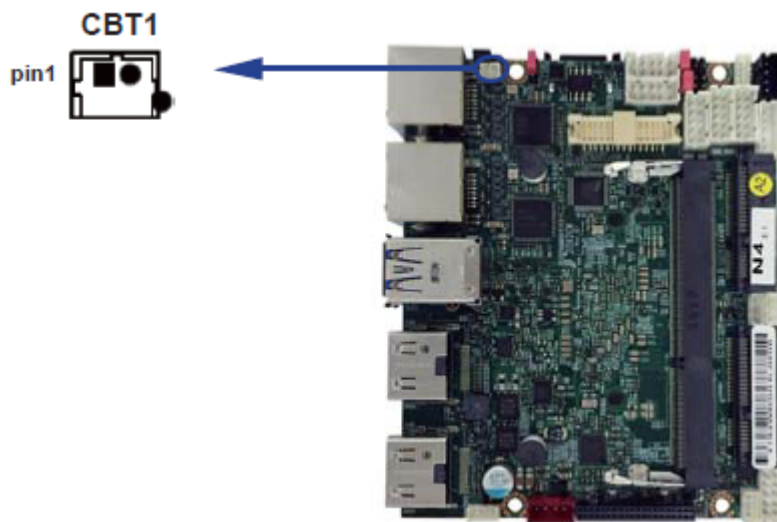
This chapter provides all necessary information of the peripheral's connections, switches and indicators. Always power off the board before you install the peripherals.

7.2 DC Power In



Pin	Description
0 (left)	9 – 36V+
1,2 (middle)	External Power Switch
3 (right)	Ground

7.3 CMOS Battery holder



PIN NO.	DESCRIPTION
1	GND
2	+3V

CBT1: 3V Battery hold 2pin CBT1: Battery use Li 3V / 40mAh (CR1220)

Note: Power consumption w/o connected Power Adapter: 2.7uA Power consumption with connected Power Adapter: 0.1uA

7.4 COM Ports

RS232 mode:

PIN NO.	Description	PIN NO.	Description
1	DCD	6	DSR
2	RXD	7	RTS
3	TXD	8	CTS
4	DTR	9	RI/Voltage
5	GND		

RS422 mode:

PIN NO.	Description	PIN NO.	Description
1	RS422 TX-	6	NC
2	RS422 TX+	7	NC
3	RS422 RX+	8	NC
4	RS422 RX-	9	NC
5	GND		

RS485 mode:

PIN NO.	Description	PIN NO.	Description
1	RS485 TX-	6	NC
2	RS485 TX+	7	NC
3	NC	8	NC
4	NC	9	NC
5	GND		

7.5 LAN Ports

PIN NO.	Description	PIN NO.	Description
1	TR0-/TX+	5	TR2-/NC
2	TR0+/TX-	6	TR2+/RX-
3	TR1-/RX+	7	TR3-/NC
4	TR1+/NC	8	TR3+/NC

7.6 USB Ports

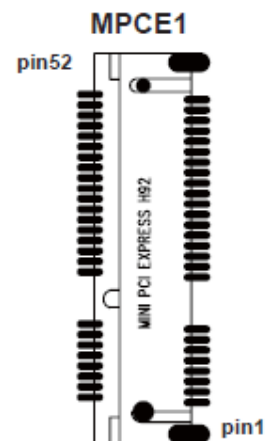
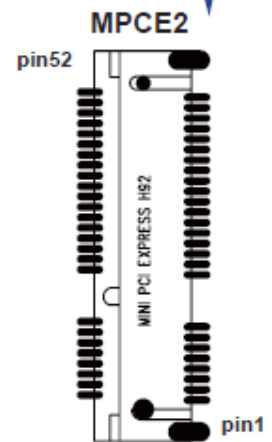
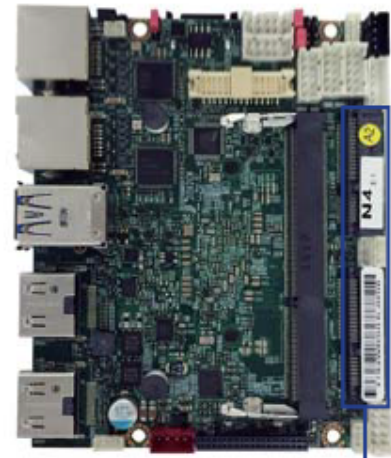
PIN NO.	Description
1	+5V
2	DATA -
3	DATA +
4	GND

7.7 mPCIe MiniExpressCard

The Arrakis MK3 offers a total of 2 on Board mPCIe Connectors and 2 mPCIe Connectors on the optional SIM/4G Card carrier module.

1. Fullsize mPCIe: usually unavailable and occupied by mSATA SSD
2. Fullsize mPCIe: supports USB, mSATA & PCIe connections
3. Two Fullsize mPCIe: only installed on demand, or with 4G Modem pre-installed. Usually one is occupied with 4G Modem, only USB devices are supported.

PIN NO.	Description	PIN NO.	Description
1	NC	2	+3.3V
3	NC	4	GND
5	NC	6	+1.5V
7	NC	8	NC
9	GND	10	NC
11	PCIe-CLK-	12	NC
13	PCIe-CLK+	14	NC
15	GND	16	NC
KEY			
17	NC	18	GND
19	NC	20	NC
21	GND	22	Reset
23	PCIe-RX-/mSATA-RX+	24	+3.3V
25	PCIe-RX+/mSATA-RX-	26	GND
27	GND	28	+1.5V
29	GND	30	SMB-CLK
31	PCIe-TX-/mSATA-TX-	32	SMB-DATA
33	PCIe-TX+/mSATA-TX+	34	GND
35	GND	36	USB-DATA-
37	GND	38	USB-DATA+
39	+3.3V	40	GND
41	+3.3V	42	NC
43	GND	44	NC
45	NC	46	NC
47	NC	48	+1.5V
49	NC	50	GND
51	mSATA/PCIe detect	52	+3.3V



8 BIOS

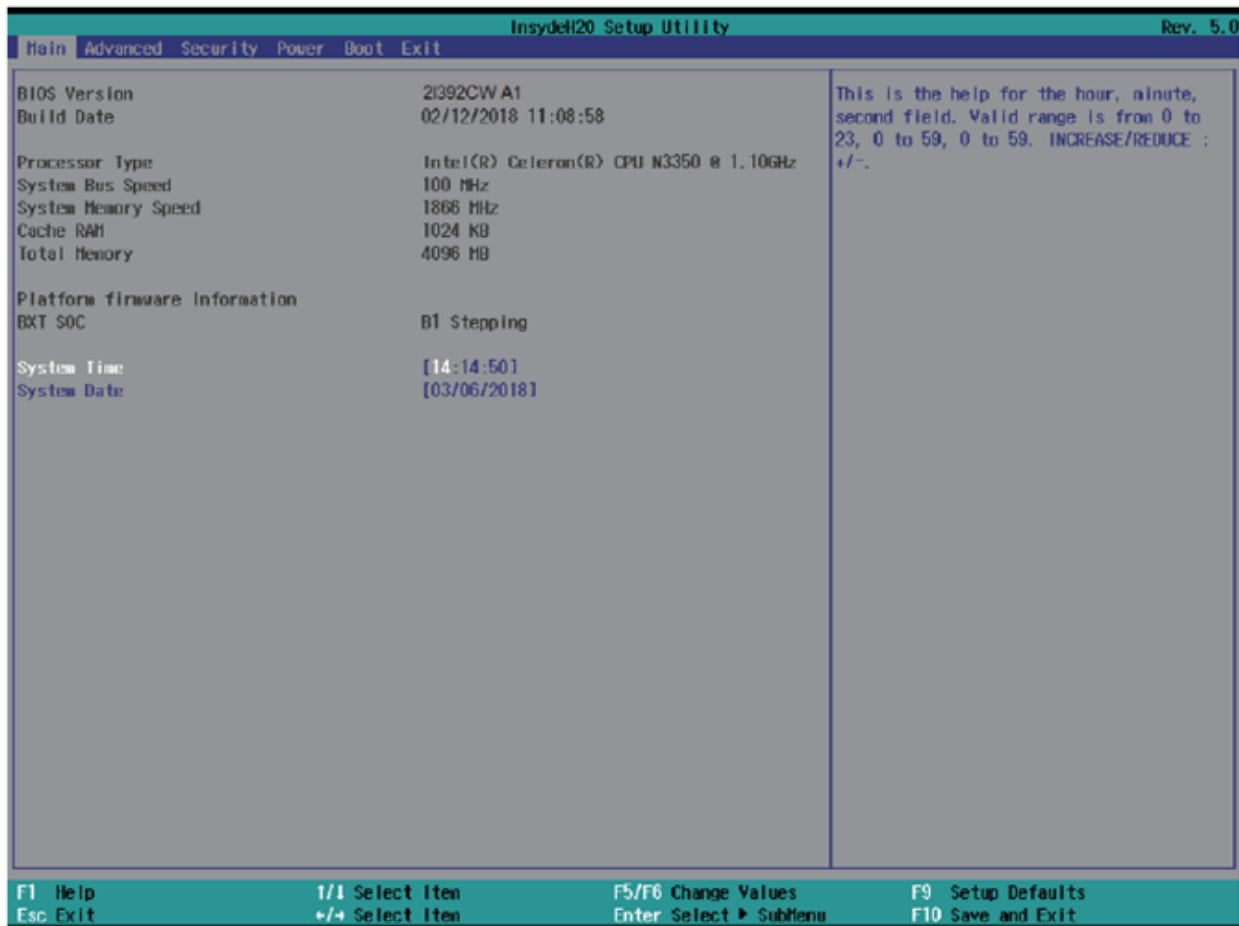
8.1 Introduction

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.

8.2 Enter BIOS

Power on the computer and press 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.

8.3 BIOS menu and function keys

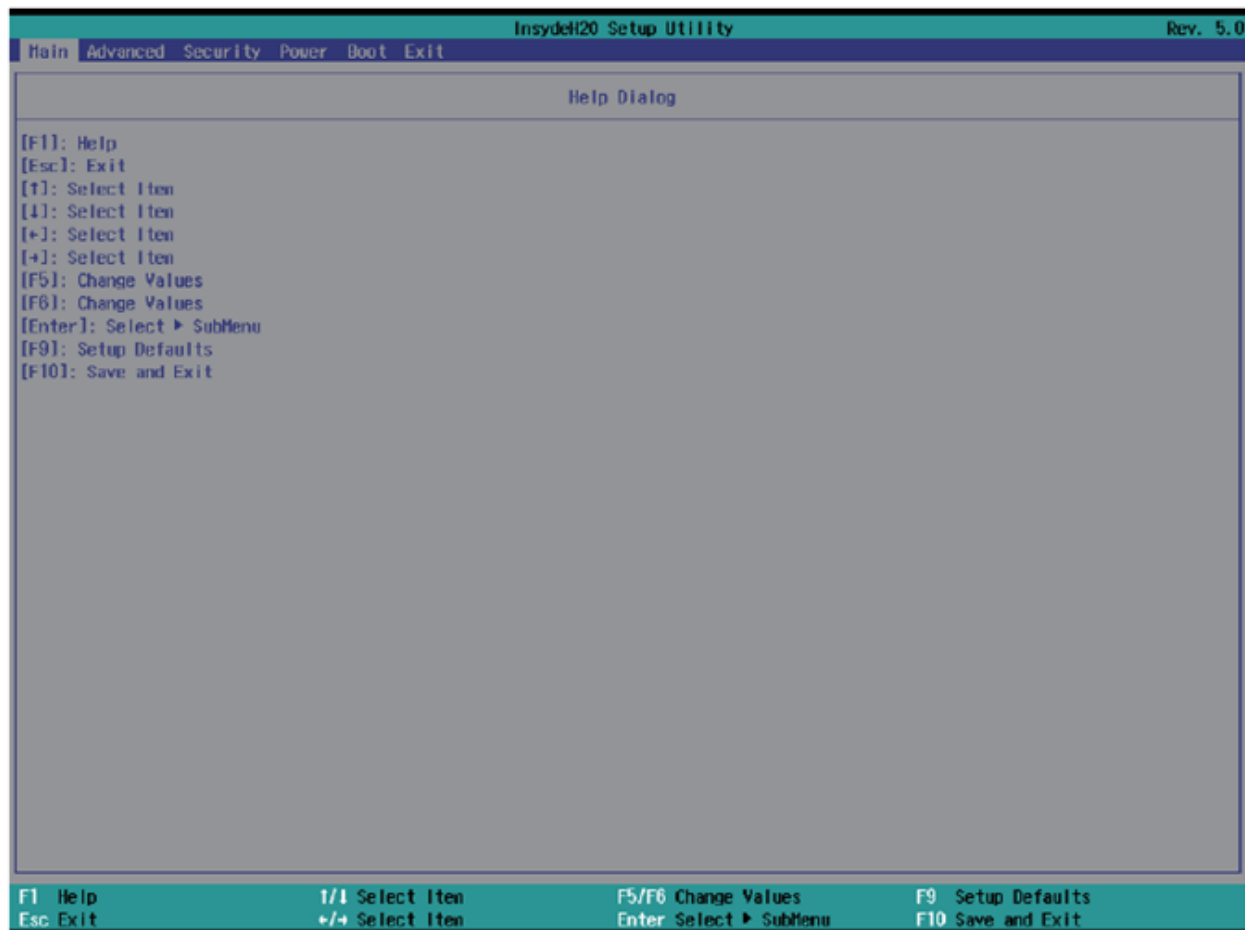


In the above BIOS Setup main menu of, you can see several options. We will explain these options step by step in the following pages of this chapter, but let us first see a short description of the function keys you may use here:

- 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.

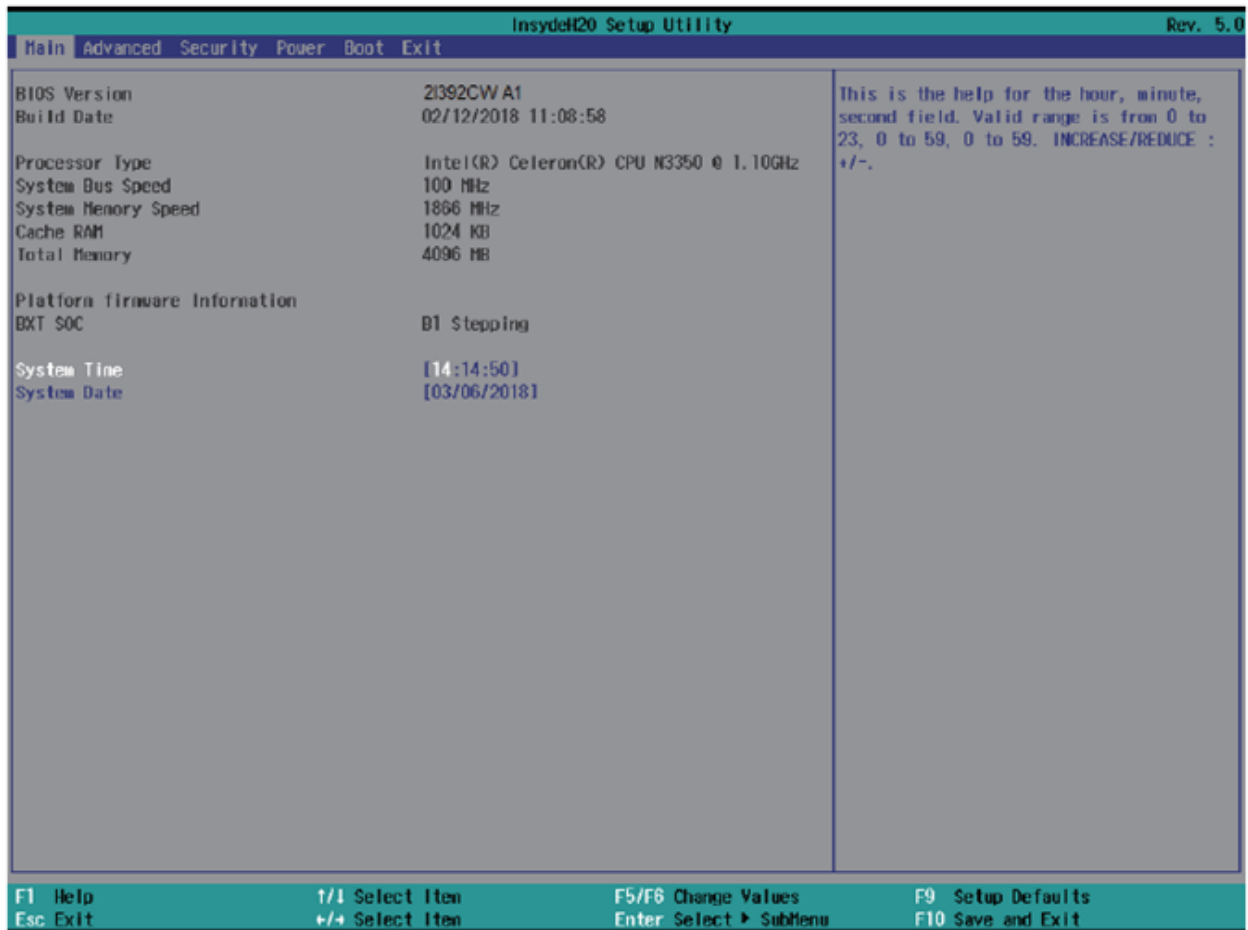
8.4 BIOS Help



Status Page Setup Menu/ Option Page Setup Menu

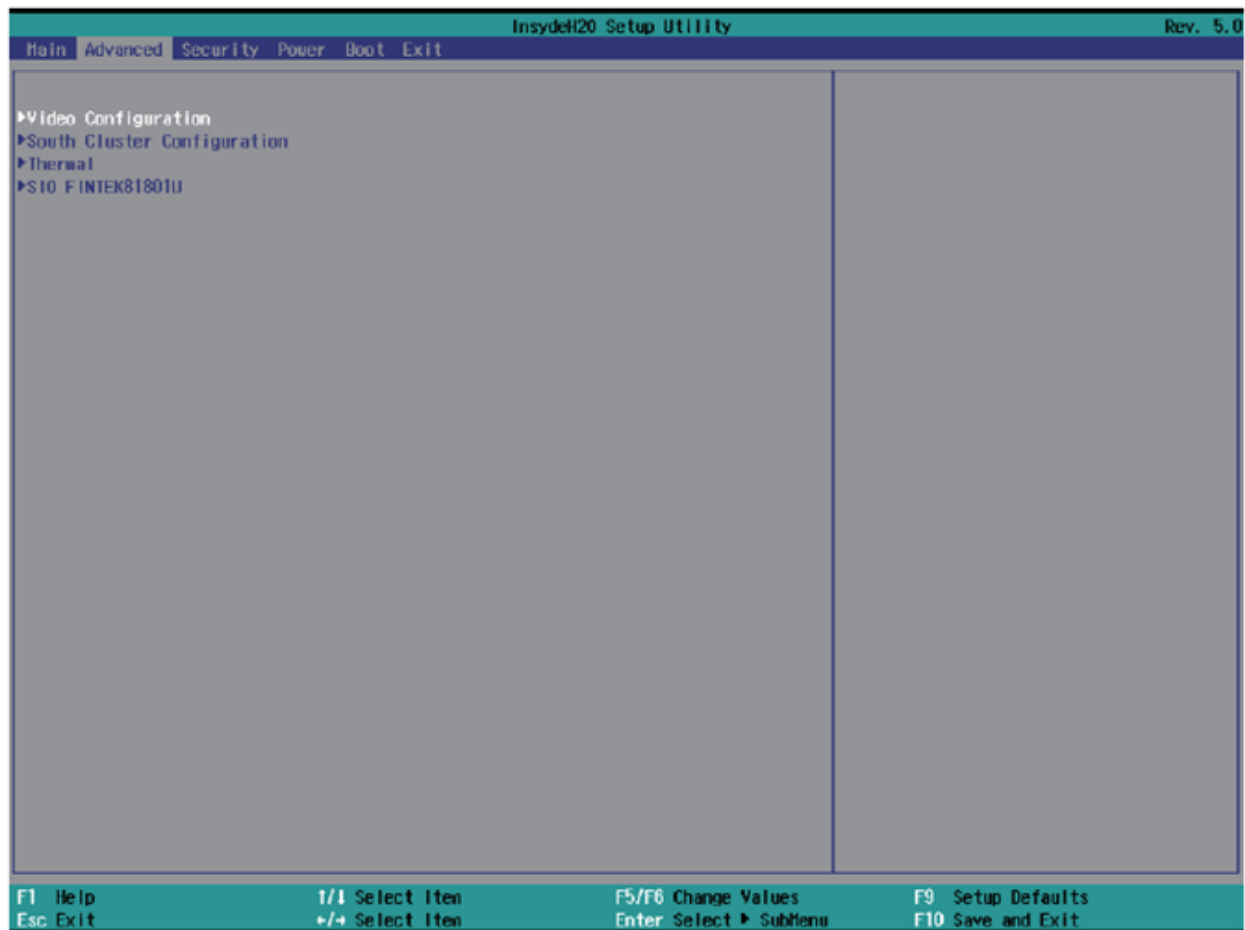
Press F1 to pop up a help window that describes the appropriate keys to use and the possible selections for the highlighted item. To exit the Help Window, press <Esc>.

8.5 Menu

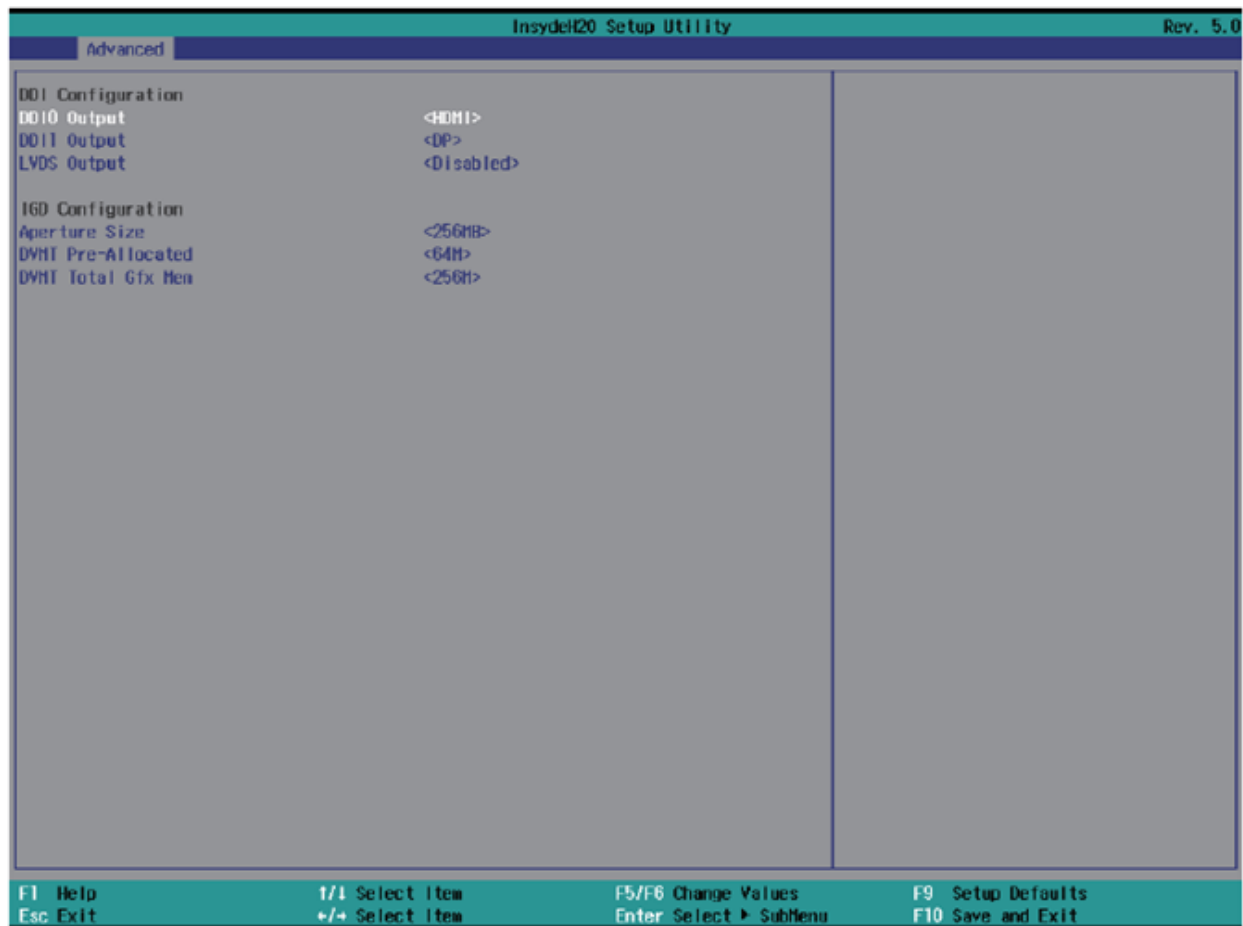


Main menu screen includes some basic system information. Highlight the item and then use the <+> or <-> and numerical keyboard keys to select the value you want in each item. System Date Set the Date. Please use [Tab] to switch between data elements. System Time Set the Time. Please use [Tab] to switch between data elements.

8.6 Advanced



8.6.1 Video Configuration



Aperture Size

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

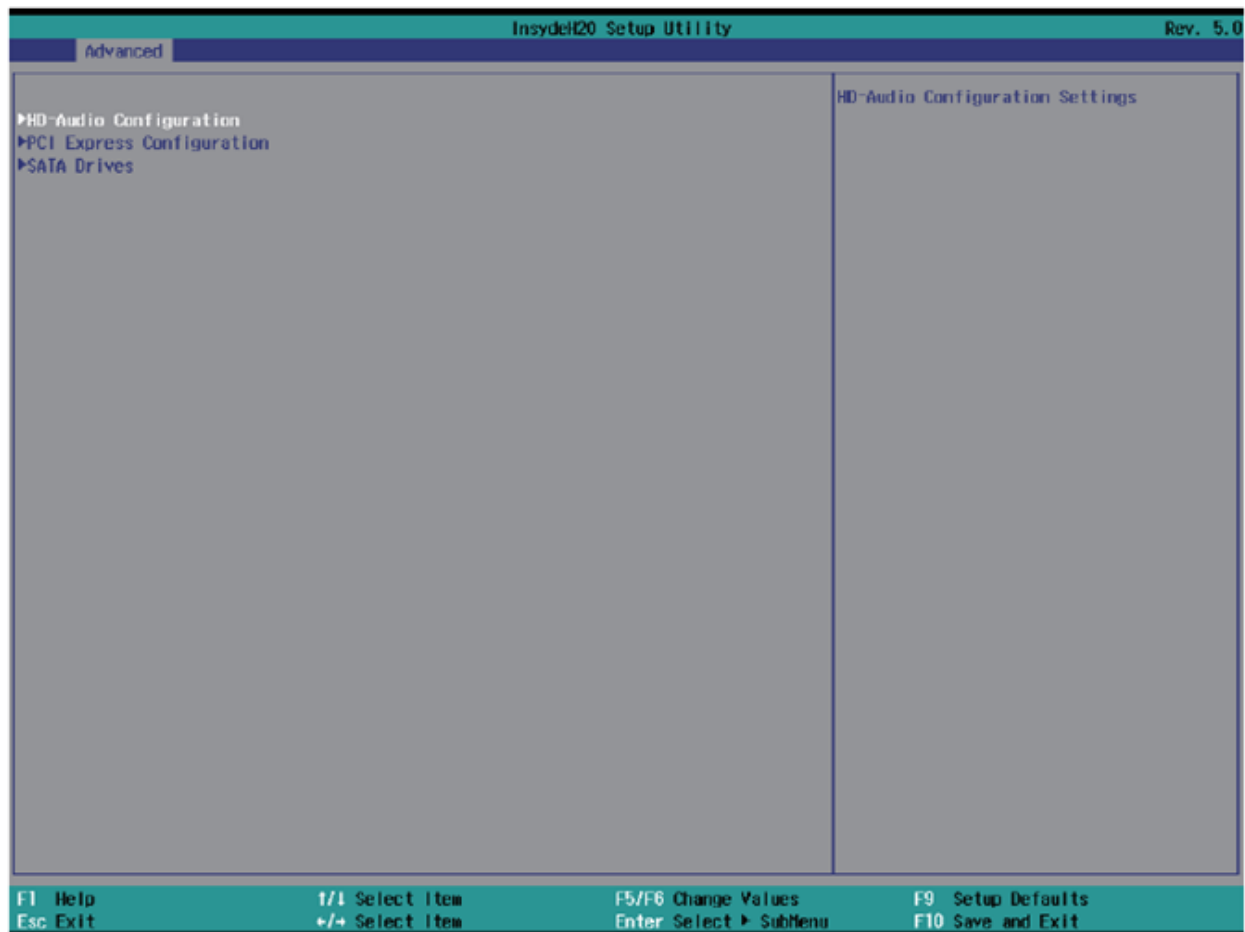
IGD - DVMT Pre-Allocated

Use this item to 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

IGD - DVMT Total Gfx Mem

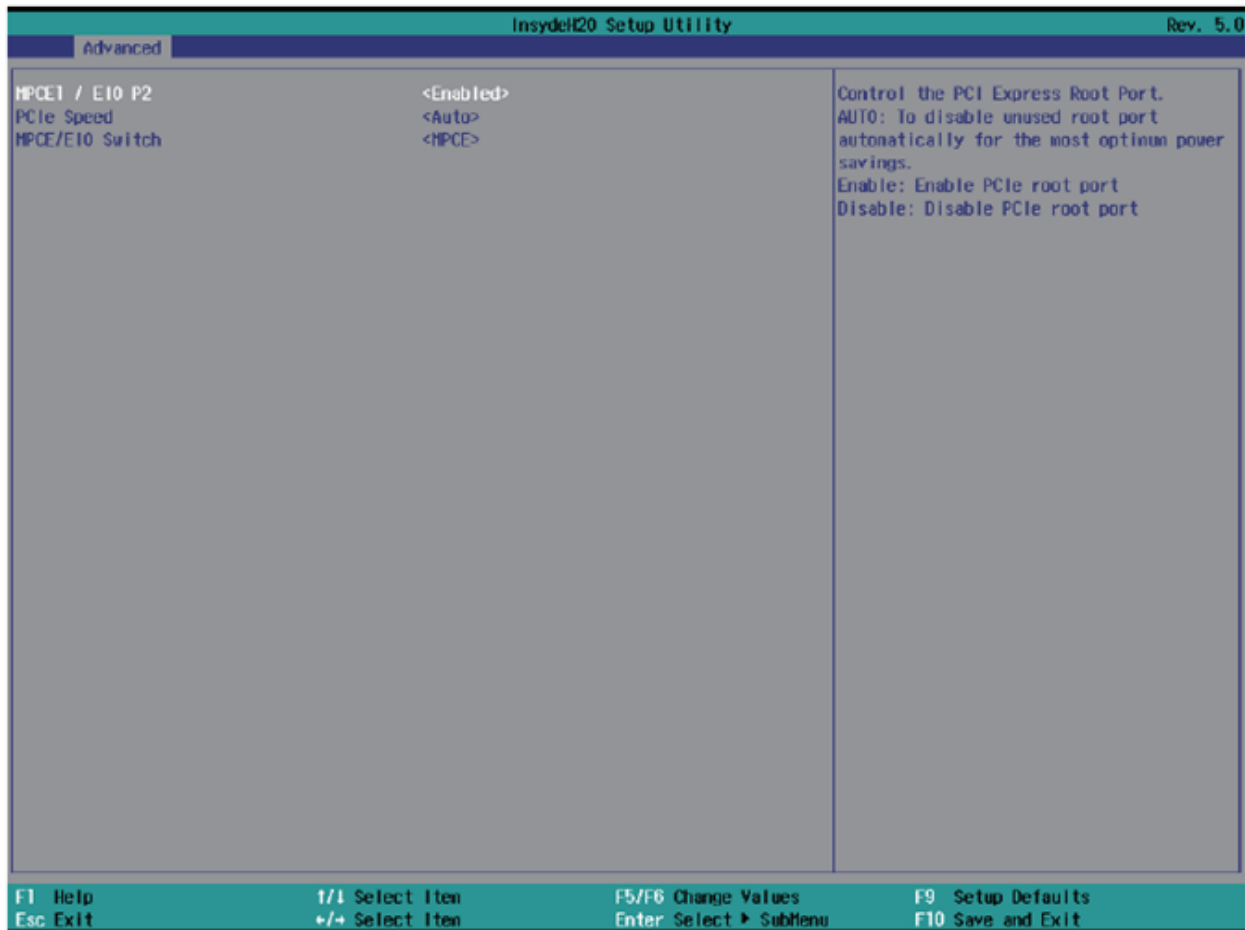
Use this item to select DVMT 5.0 total graphics memory size used by the internal graphics device. The optional settings are: 128M, 256M (default), MAX

8.6.2 HD-Audio Configuration



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

8.6.3 PCI Express Configuration



MPCE1 / EIO P2

The optional settings are: Disabled, Enabled (default) for MPCE1.

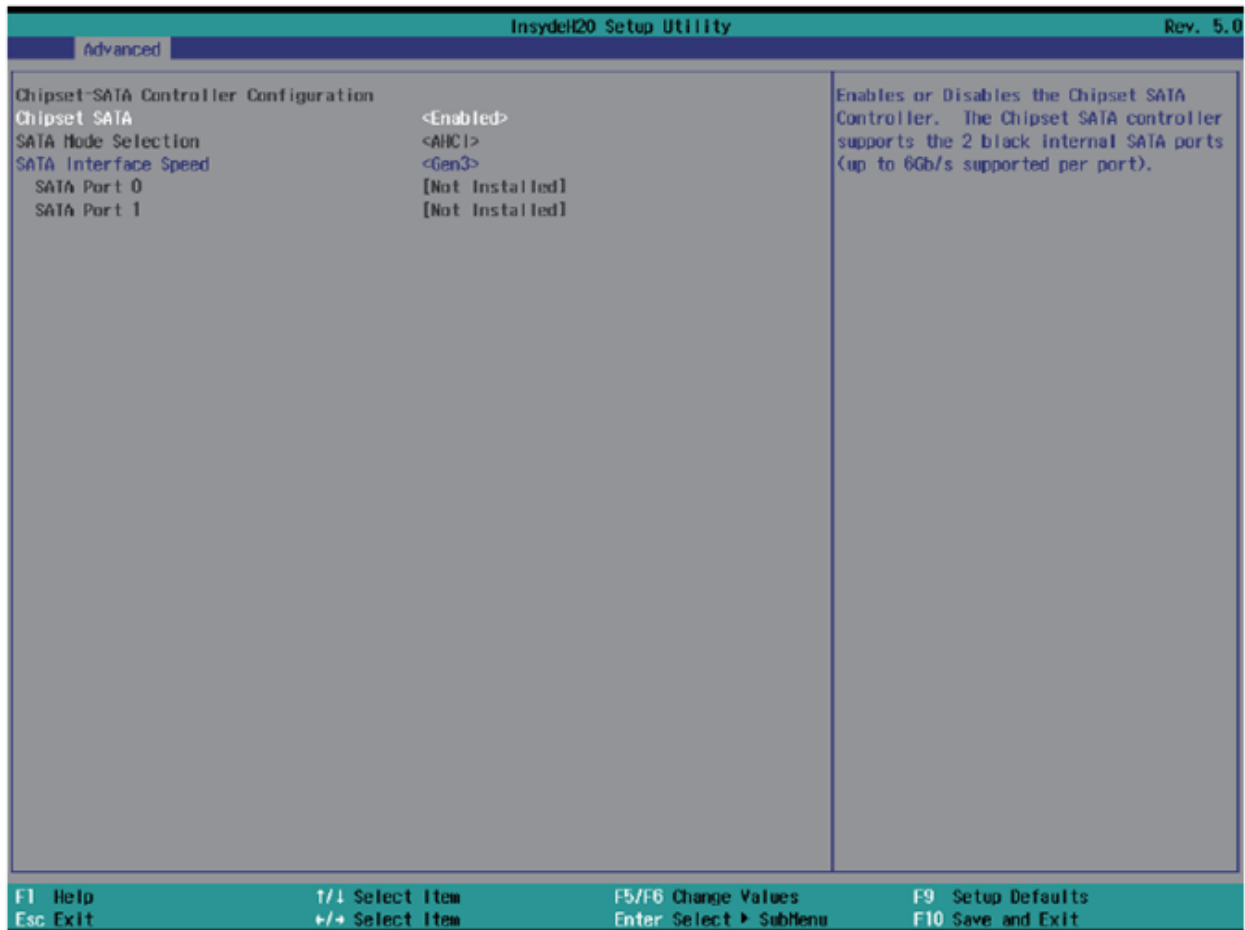
PCIe speed

This item that setting PCIe speed: Auto (default), Gen1, Gen2

MPCE / EIO Switch

This item to switch the PCIe signal to MPCE1 (default) or EIO (internal expansion port for OEM I/O or function boards)

8.6.4 SATA Drives Configuration



Chipset SATA

Use this item to Enable or Disable SATA function. The optional settings are: Enabled (default) or Disabled

SATA Mode Select

This Item is just for Info, the Arrakis MK3 will always work in ACHI mode.

SATA Interface Speed

This item will determine the SATA speed. The optional settings are: Gen1, Gen2, Gen3 (default)

8.6.5 Thermal



Thermal Configuration Parameters

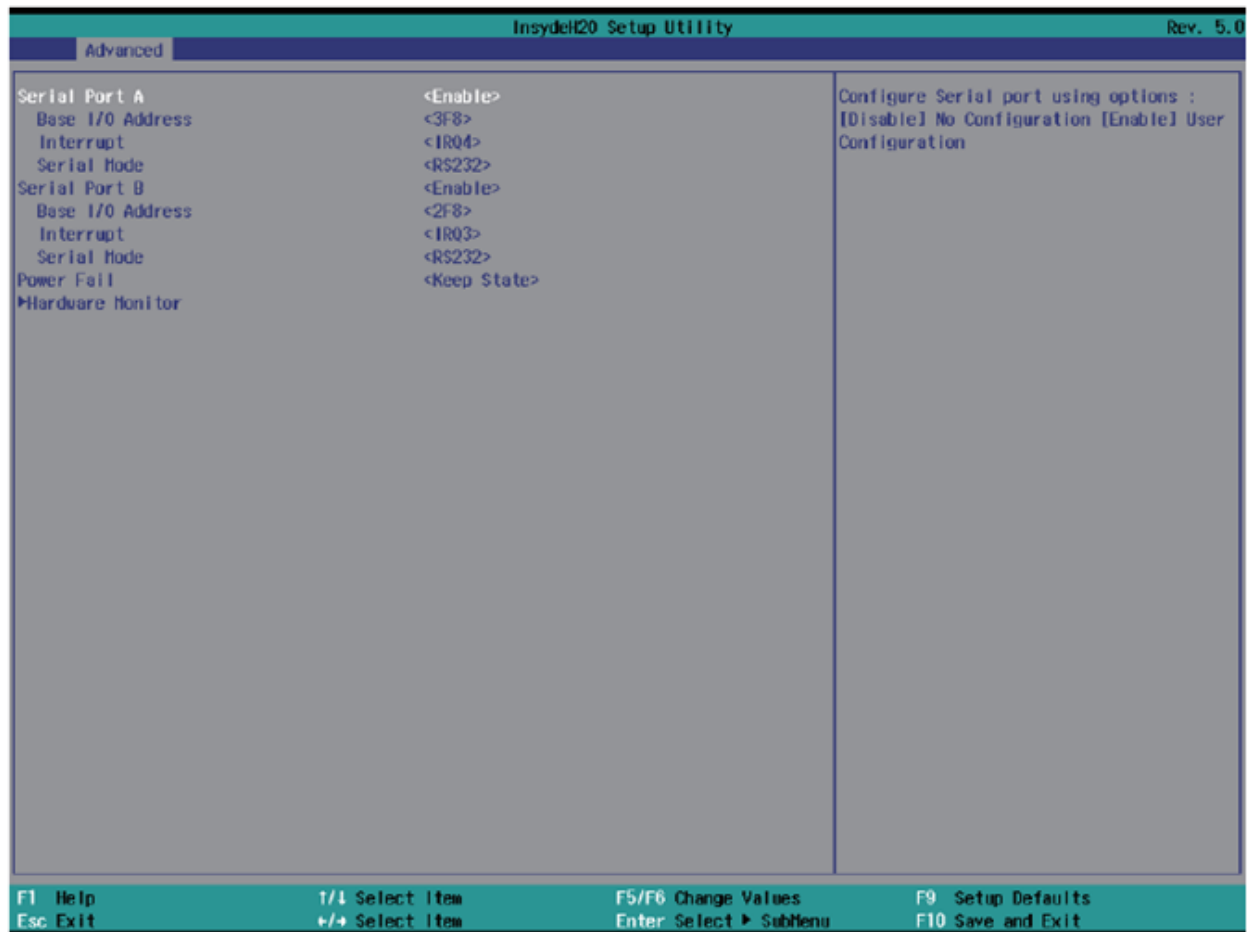
This Value controls the temperature of the ACPI Critical Trip Point, the point in which the OS will shutdown the system.

Critical Trip point is the shutdown temperature, the default value is 110°

The CPU frequency will auto reduce when CPU temperature arrived at the **passive Trip point**.

The default of the passive trip point is 105°

8.6.6 SIO FINETEK 81801U



Serial Port A/B

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

Serial Port A/B 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.

Power Failure

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

[Keep 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.

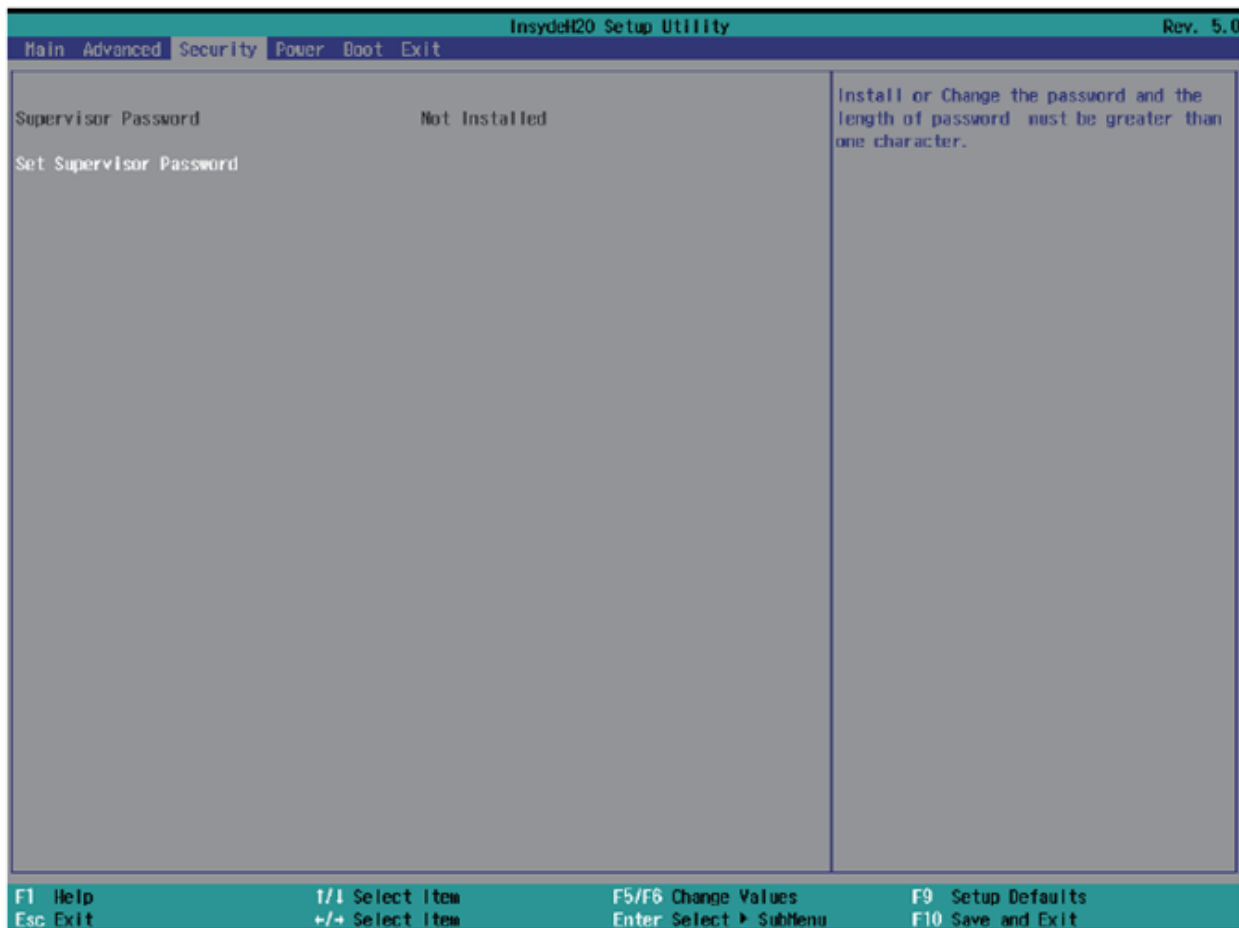
Hardware Monitor

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.

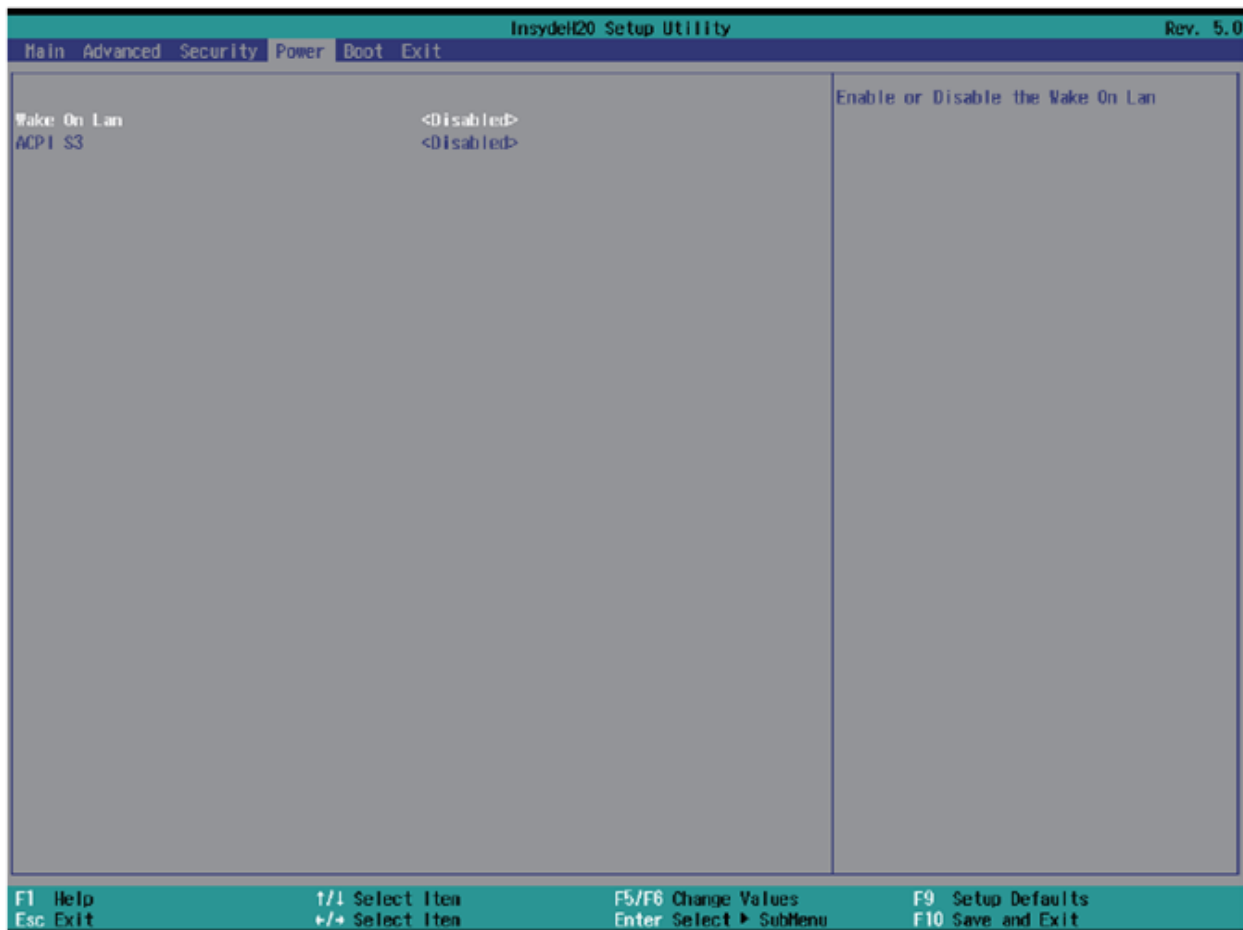
8.7 Security



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.

8.8 Power



Wake on LAN

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

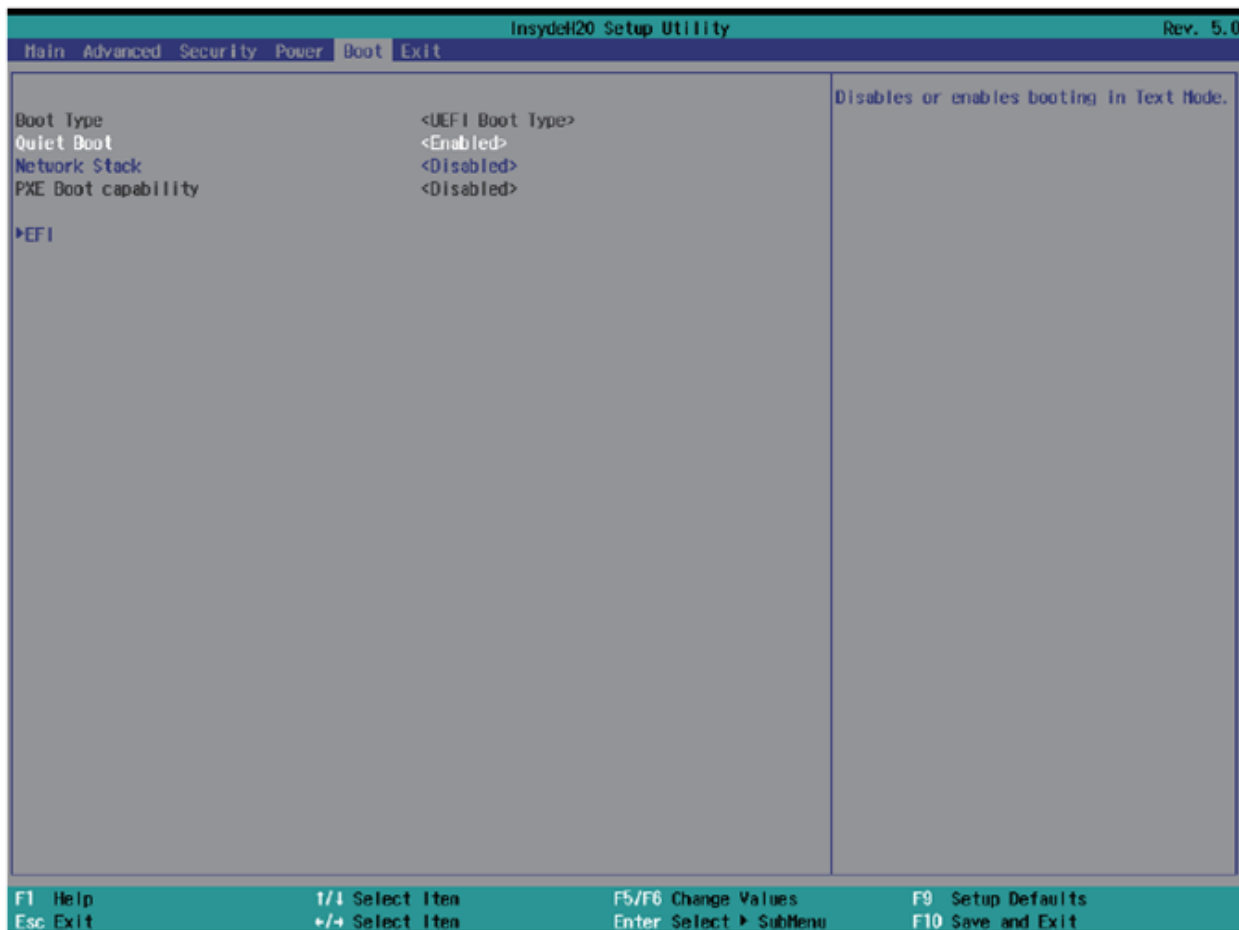
The optional settings: S3, S5, S3 / S5, Disabled (default)

ACPI S3

Select ACPI sleep state (S3) support.

The optional settings: Enabled or Disabled (default)

8.9 Boot



Boot type

The Arrakis MK3 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

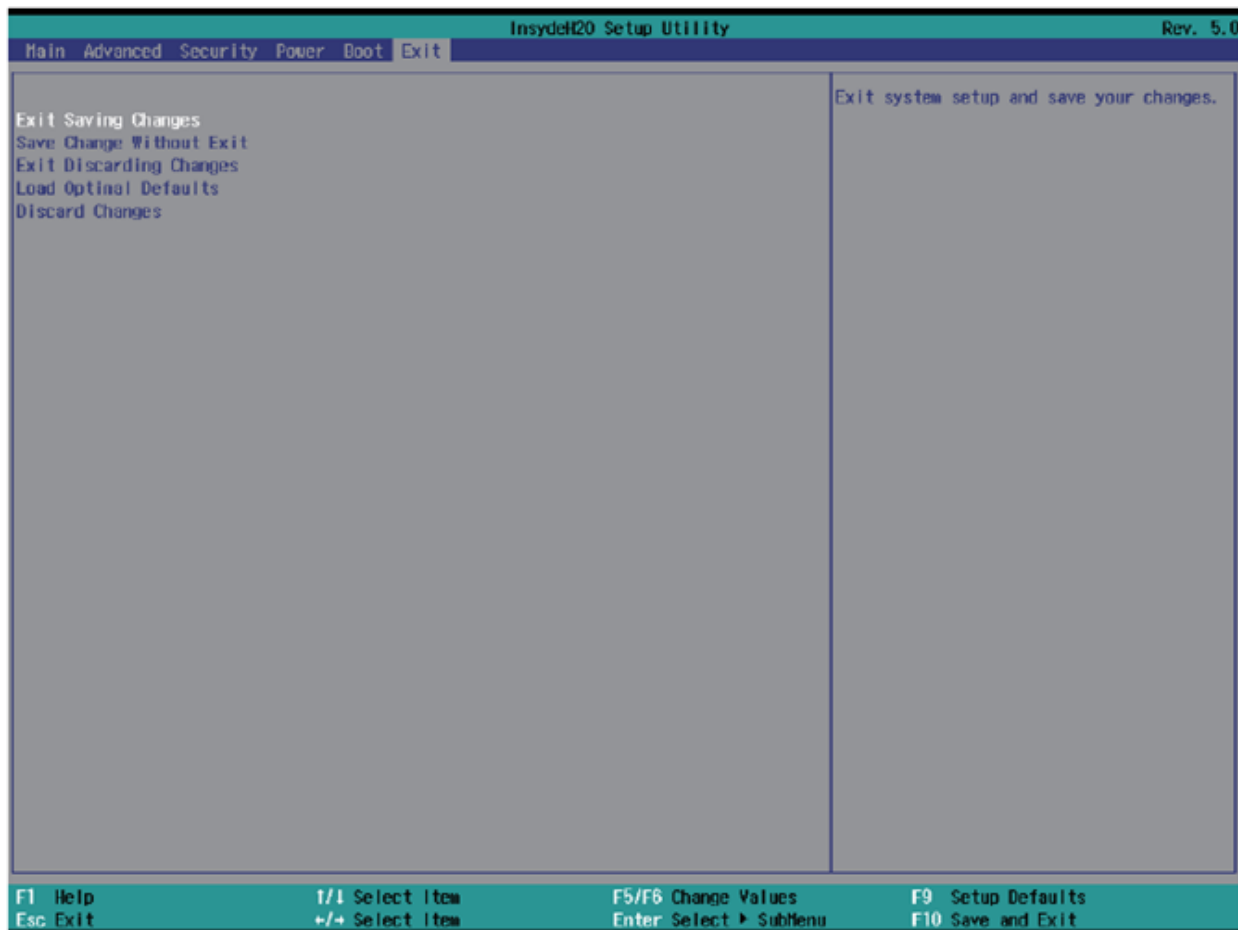
The item determines what protocol operation in PXE, these values are: Disabled (default), UEFI: IPv4, UEFI: IPv6

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.

8.10 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.

9 Driver Installation

The Arrakis MK3 is usually shipped with an Operating System preinstalled (recommended)

In case you have chose to purchase an Arrakis MK3 without preinstalled operating system or need to reinstall, you can download all available System drivers from this address:



[Download Service](#)

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

10 Appendix A Power consumption

Item	Specification
CPU	E3950
RAM	DDR3L 4GB 1866MHz
Operating System	Windows 10 IoT 2019 LTSC
Test Program	Passmark
mSATA	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.

11 Appendix B F75111N DIO & Watchdog device

The Arrakis MK3 is available with optional DIO Ports, this chapter introduces programming.

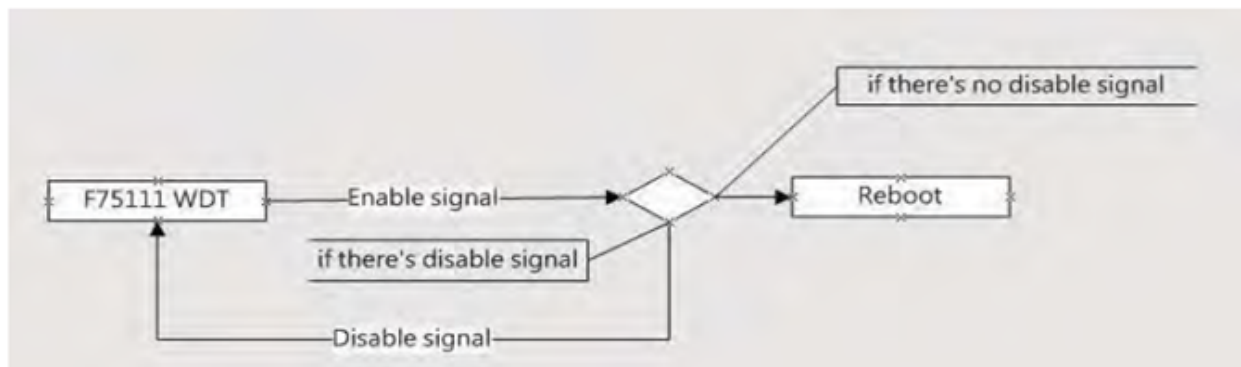
11.1 Watchdog timer under DOS

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:

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

Or WriteI2CByte(I2CADDR, WDT_TIMER_CTL, 0x00);

Or

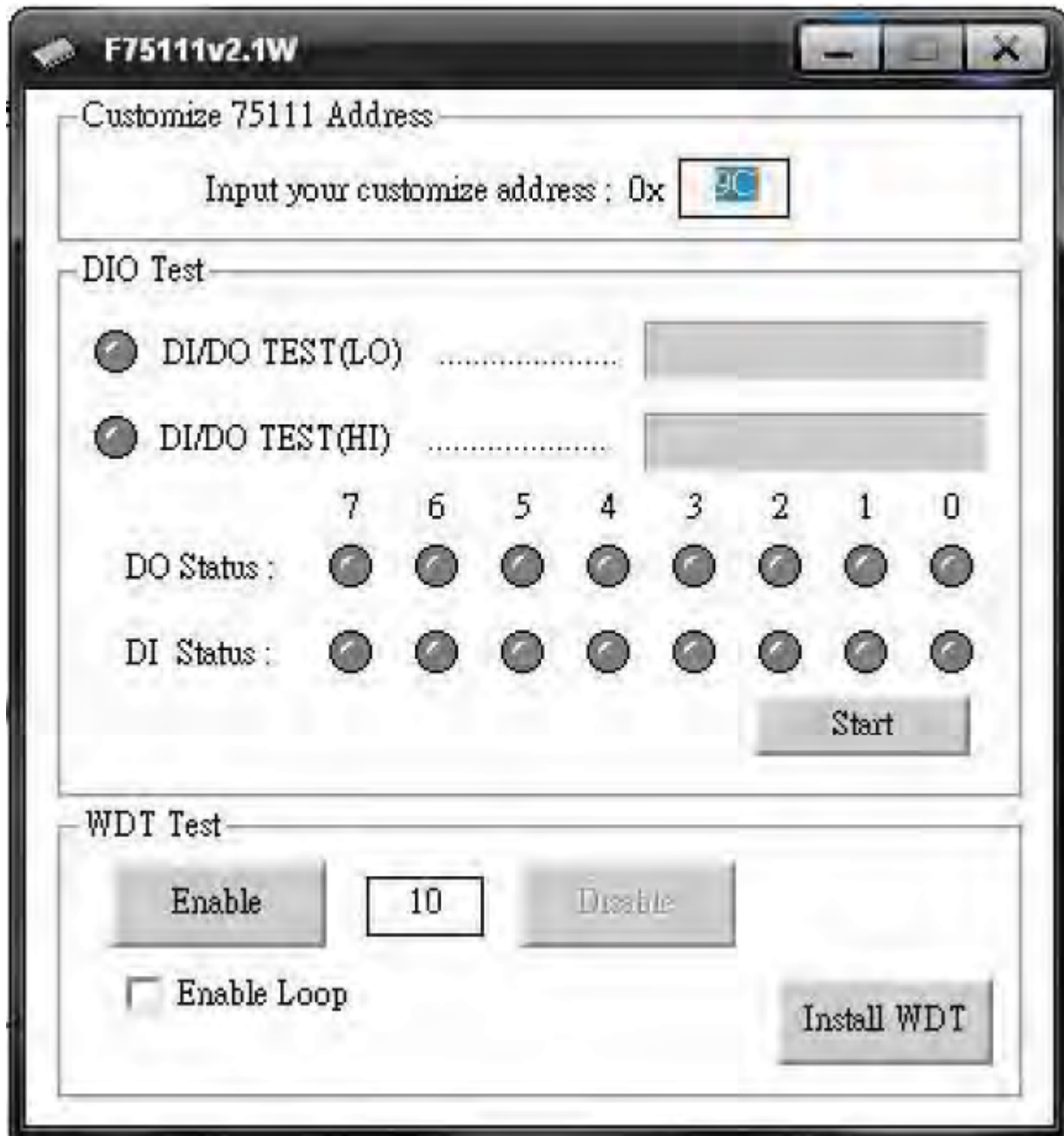
```
void pause(int time) {
    asm mov ah, 0h; // Ah = 00 Read System Time Counter
    asm int 1ah;    // read time from Time Counter and store it in DX register
    asm add dx, time;
    asm mov bx, dx;
label:
    asm int 1ah;
    asm cmp bx, dx;
    asm jne label;
}
```

11.2 Watchdog Timer and DIO under Windows

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

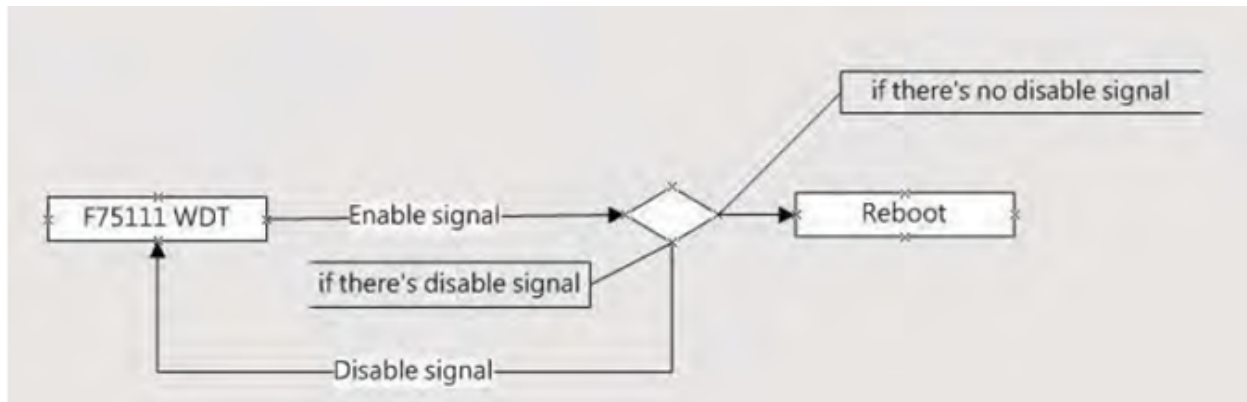
Source file: F75111_DIOSrc.rar Binary file: F75111_DemoBin.rar USERNAME & PASSWORD: sf

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

- 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 byte-Value) 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;
  
```

(continues on next page)

(continued from previous page)

```

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 configure
}

```

Disable WatchDog:

```

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

```

11.3 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!!

11.4 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 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
```

11.5 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

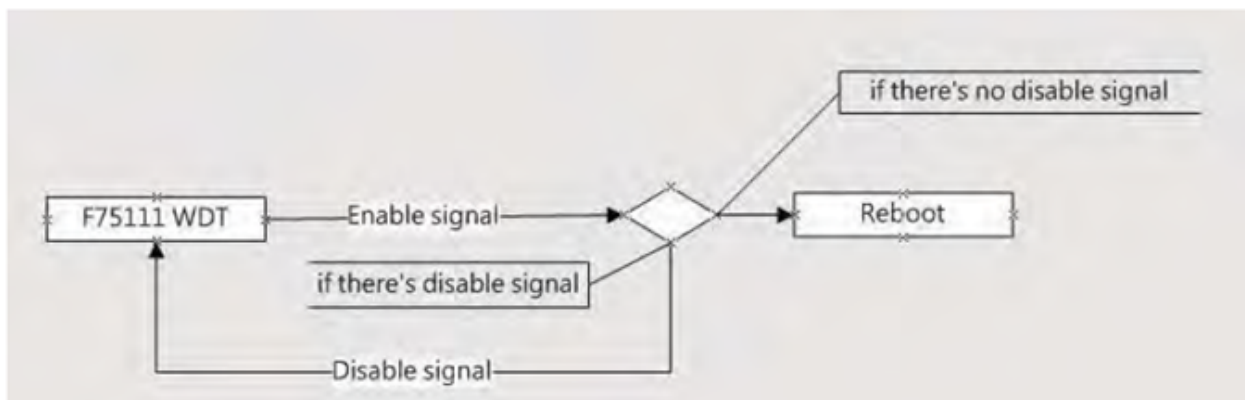
How to compile source code:

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
 2. make
 3. 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 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
    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;
}
```

(continues on next page)

(continued from previous page)

```
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 configure
}
```

Disable WatchDog:

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