MI987

Intel® Q87 / H81 Based Mini-ITX board
Mini ITX Motherboard

USER'S MANUAL

Version 1.0A

Acknowledgments

Table of Contents

Introduction 1

Product Description 1
Checklist 1
MI987 Specifications 2
Board Dimensions 4

Installations 5

Installing the Memory 6
Setting the Jumpers 7
Connectors on MI987 12

BIOS Setup 23

Drivers Installation 45

Intel Chipset Software Installation Utility 46
VGA Drivers Installation 47
Realtek HD Audio Driver Installation 48
LAN Drivers Installation 49
Intel® Management Engine Interface 50
Intel® USB 3.0 Drivers 51

Appendix 53

A. I/O Port Address Map 53
B. Interrupt Request Lines (IRQ) 54
C. Digital I/O Sample Code 55
D. Watchdog Timer Configuration 60

This page is intentionally left blank.

Introduction

Product Description

The MI987 Mini ITX motherboard is based on the latest Intel® Q87 chipset. The platform supports Intel® 4th Generation Core™ DT i7/i5/i3 processors.

The latest Intel® processors provide advanced performance in both computing and graphics quality. This meets the requirement of customers in the gaming, POS, digital signage and server market segment.

The platform supports two SO-DIMM sockets that can accommodate up to 16GB of DDR3-1600 Non-ECC memory. The Intel® 4th Gen. Core™ DT processor integrated HD graphics supports 3 independent displays, Direct X 11.1, OpenGL 3.2, and Open CL 1.2. Display interfaces are for HDMI, DisplayPort and VGA CRT.

With two Gigabit Ethernets, the MI987 Mini ITX board utilizes the dramatic increase in performance provided Intel's latest cutting-edge technology. Expansion is provided by PCIe(4x), one full sized MiniPCIe and two half sized MiniPCIe (One of half-sized MiniPCIe slot is dedicated mSATA only). Onboard connectors support 2x SATAIII, 4x or 6x USB 2.0 depending on the MI987 model and 2x COM ports. The board measures 170mm x 170mm.

Checklist

Your MI987 package should include the items listed below.

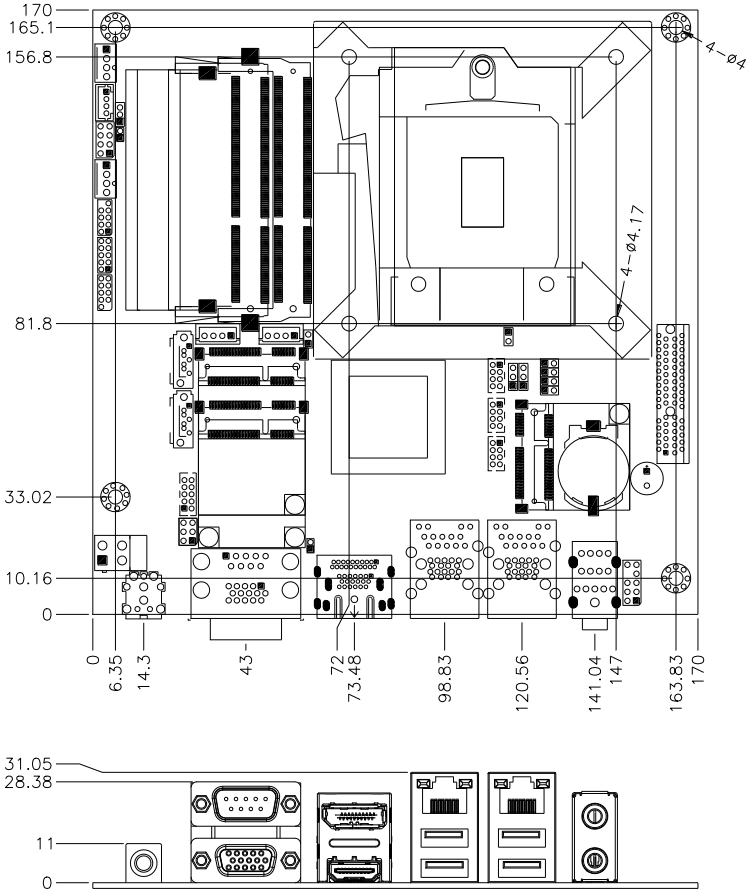
- The MI987 MINI ITX motherboard
- This User's Manual
- 1 CD containing chipset drivers and flash memory utility

MI987 Specifications

Product Name	MI987AF (Q87) (Model name printed on PCB surface) MI987EF (H81)
Form Factor	Mini ITX
CPU Type	- Intel® 4th Generation Core™ DT i7/i5/i3 processors (22nm monolithic) **Recommend to use low power DT CPU @ TDP 35W/45W**
CPU Speed	Up to 2.7GHz
Cache Size	Up to 8MB
CPU Socket	LGA1150 (Socket H3)
Chipset	Intel® Q87 PCH [MI987AF] ; Intel® H81 PCH [MI987EF] Package =23 mm x 22 mm , 0.65 mm ball pitch
BIOS	AMI BIOS
Memory	Intel® 4th Gen Core™ DT processors integrated memory controller, - DDR3-1600 MHz@1.5V - SO-DIMM x 2, Max.=**16**GB (Non-ECC) [Horizontal type]
VGA	Intel® 4th Gen. Core™ DT processor integrated HD Gfx, supports 3 independent displays, Direct X 11.1, OpenGL 3.2, Open CL 1.2 - HDMI x 1 (w/ Level shifter PTN3360D) [Support to 4096 x 2304@24Hz ; 2560 x 1600@60Hz] - DisplayPort x 1 [support to 3820 x 2160@60Hz] - VGA x 1 (Thru PCH) [Support to 1920x1080@60Hz]
LAN	1. Intel® I217LM PHY[MI987AF] or I217V GbE PHY[MI987EF] 2. Intel® I211AT PCIe GbE as 2nd LAN
USB	USB 2.0 host controller [Q87/H81 Integrated] - [MI987AF] : 6 ports via onboard pin header - [MI987EF] : 4 ports thru pin header+ 2 ports @edge I/O connectors - 2 ports via MiniPCIe socket USB 3.0 host controller [Q87 /H81 Integrated] @ edge I/O connectors - [MI987AF] : USB 3.0 x 4 ports - [MI987EF] : USB 3.0 x 2 ports
Serial ATA	Intel® Q87/ H81 PCH built-in SATA controller, - 2x SATA 3.0 (6Gbps) - **2** x mSATA [Q87 PCH Port 3/4 for SATA(3.0) or H81 PCH Port 3/4 for SATA(2.0)]
Storage Device	mSATA x 2 thru full-sized/half-sized Mini-PCIe socket @J12/J13 socket
Audio	Intel® Q87/H81 PCH built-in High Definition Audio controller + Realtek ALC662 w/ 5.1 channels
LPC I/O	Nuvoton NCT5523D (64-pin LQFP [7 mmx7 mm]) - COM #1 (RS232/422/485) support ring-in with power @500 mA (selectable for 5V or 12V) [EXAR SP339EER1 232/422/485 transceiver x 1 for jumper-less] - COM #2 (RS232 only) Hardware Monitor (2 thermal inputs,4 voltage monitor inputs & 2 Fan headers) - CPU FAN x 2 (PWM/DC type, 4-pin connector type, auto-detection by Nuvoton NCT3943S fan controller)
Digital IO	4 in & 4 out
IAMT 9.0	MI987AF only
TPM 1.2	Infineon SLB9655

Expansion Slots	- PCIe **(4x) x1** [Gen 3.0 PEG] - MiniPCIe socket x **2** [Full-sized + Half-sized stack type, Full-sized support PCIe/USB/mSATA, Half-sized support mSATA only] - Half-sized socket x 1 (3rd Mini-PCIe@J16 , support PCIe / USB)
Edge Connectors	DC-Jack x 1(C12135112DC102000P) DB9+DB15 stack connector for COM #1 + CRT DP + HDMI stack connector x 1 Dual USB (3.0) + RJ45 stack connector x 2 ** For MI987EF, USB 3.0 @ CN7 is USB2 only** Audio Jack 2 x 1 for MIC-in / Line-out **Reserved for 3-port connector population**
Onboard Header/Connector	2 ports x SATA III [Blue color] DF-11 2x4 pins pin-header x 3 for 6 ports USB 2.0 (MI987AF) DF-11 2x4 pins pin-header x 2 for 4 ports USB 2.0 (MI987EF) DF-11 2x6 pins pin-header x 1 for front panel audio DF-11 2x5 pins box header x 1 for COM2(RS232) 2x5 pins pin-header x1 for Digital IO 4 pins mini-type header for SATA device power x 2 [JST type] 2 x 4 pins pin header x 1 for front I/O (2.54 pitch)
Watchdog Timer	Yes (256 segments, 0, 1, 2…255 sec/min)
System Voltage	+19V DC-in (Range = Min. +15.2V ~ Max. +22.8V) ** Supplemental voltage: +12V @ 8A max.; +5V @ 8A max. **
Other	- LAN Wakeup - EuP/ErP (MI987EF only, thru Super I/O) - iSMART function(TI MSP430G2433 MCU) - Legacy Free from super I/O - AT24C02C EEPROM [SO8 type] via SMbus
Board Size	170mm x 170mm
Operation System	Windows 7, Windows 8/8.1
Accessories (Optional)	PK1H: C501PK1H009302000P (COM port cable) USB-29: C501USB2908303000P (USB Cable) AUDIO-34: C501AUD3410252000P (Audio Cable) SATA-17: C501SATA170253000P (SATA Cable)

Board Dimensions



Installations

This section provides information on how to use the jumpers and connectors on the MI987 in order to set up a workable system. The topics covered are:

Installing the Memory 6
Setting the Jumpers 7
Connectors on MI987 12

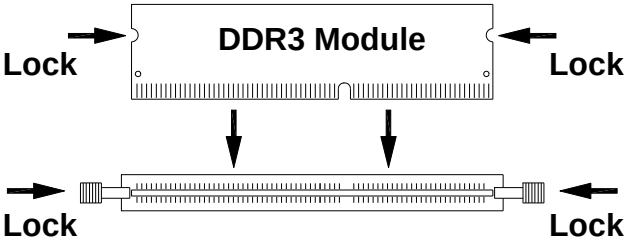
Installing the Memory

The MI987 board supports two DDR3 memory modules for a maximum total of 16GB in DDR3 SODIMM memory type.

Installing and Removing Memory Modules

To install the DDR3 modules, locate the memory slot on the board and perform the following steps:

1. Hold the DDR3 module so that the key of the DDR3 module aligned with that on the memory slot.
2. Gently push the DDR3 module in an upright position until the clips of the slot close to hold the DDR3 module in place when the DDR3 module touches the bottom of the slot.
3. To remove the DDR3 module, press the clips with both hands.

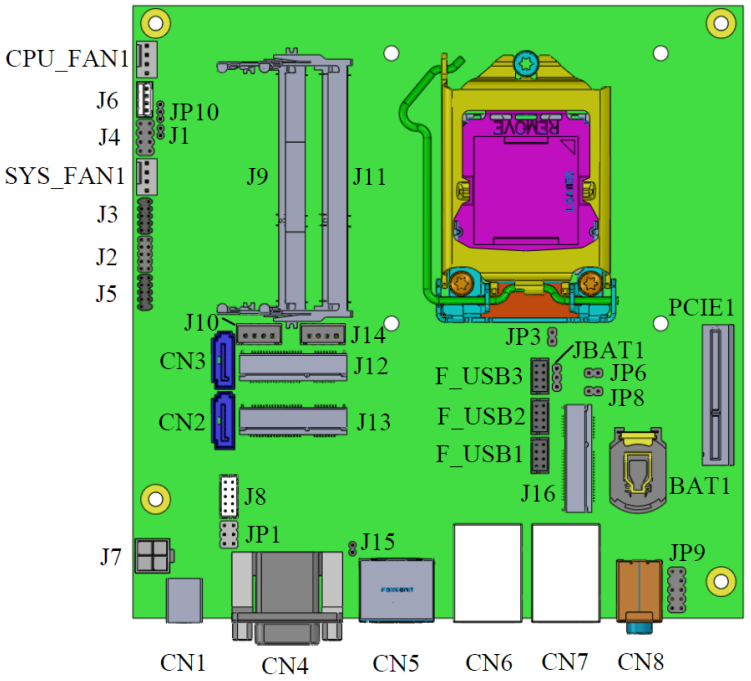


Setting the Jumpers

Jumpers are used on MI987 to select various settings and features according to your needs and applications. Contact your supplier if you have doubts about the best configuration for your needs. The following lists the connectors on MI987 and their respective functions.

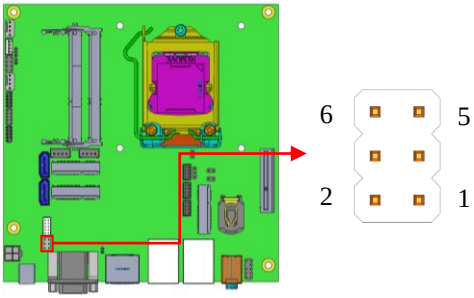
Jumper Locations on MI987 8
JP1: COM1 RS232 RI/+5V/+12V Power Setting 9
JP8: Flash Descriptor Security Override (Factory use only) 9
JBAT1: Clear CMOS Contents 10
J15: DP++ Select 10
JP10: Power On Type 11

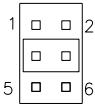
Jumper Locations on MI987



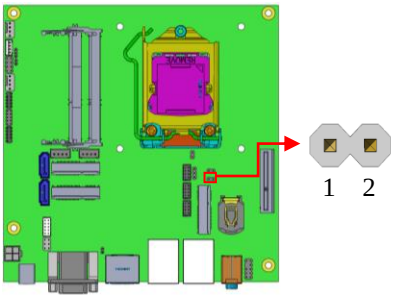
Jumpers on MI987 ..... Page
JP1: COM1 RS232 RI/+5V/+12V Power Setting 9
JP8: Flash Descriptor Security Override (Factory use only) 9
JBAT1: Clear CMOS Contents 10
J15: DP++ Select 10
JP10: Power On Type 11

JP1: COM1 RS232 RI/+5V/+12V Power Setting



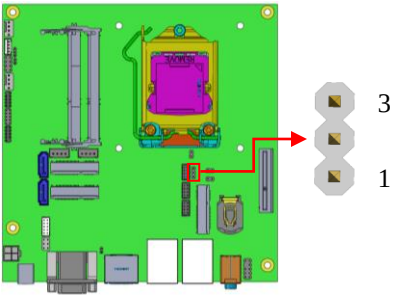
JP1(2.54mm)	Setting	Function
	Pin 1-3 Short/Closed	+12V(500mA)
	Pin 3-4 Short/Closed	RI
	Pin 3-5 Short/Closed	+5V(500mA)

JP8: Flash Descriptor Security Override (Factory use only)



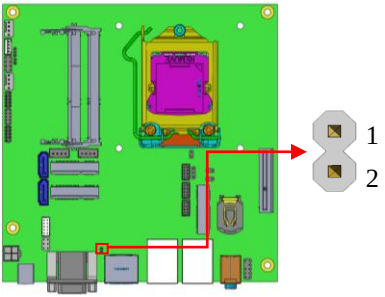
JP8 (2.54mm)	Flash Descriptor Security Override
Open	Disabled (Default)
Close	Enabled

JBAT1: Clear CMOS Contents



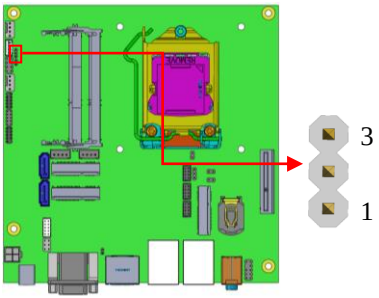
JBAT1(2.54mm)	Setting	Function
	Pin 1-2 Short/Closed	Normal
	Pin 2-3 Short/Closed	Clear CMOS

J15: DP++ Select



JP15(2.0mm)	Function
Short	Display Port
Open	Dongle /DP to DVI / DP to HDMI

JP10: Power On Type

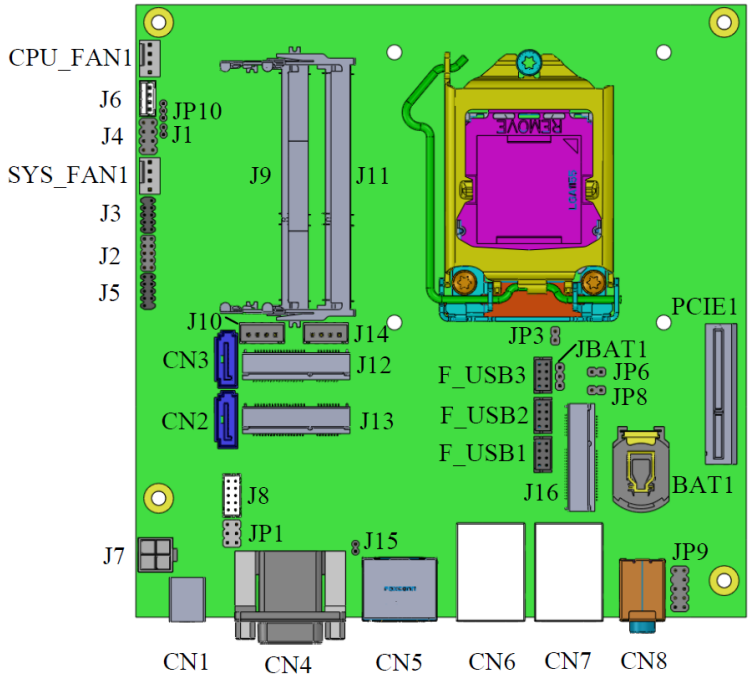


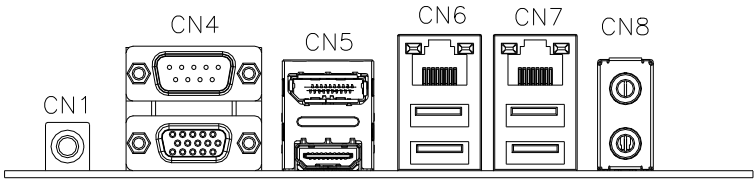
JP10(2.0mm)	Function
1-2	ATX Mode (Default)
2-3	AT Mode

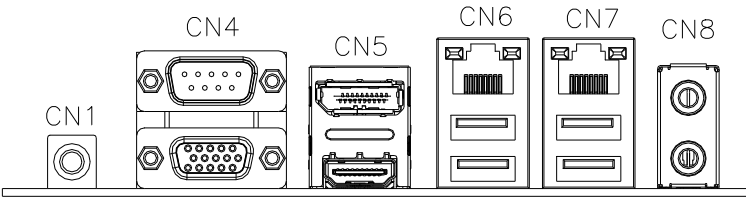
Connectors on MI987

Connector Locations on MI987 13
CN1: DC-IN Connector (+19V DC-in) 14
CN4: CRT/COM1 Connector 14
CN5: DP/HDMI Connector 14
CN6: Gigabit LAN (Intel I217LM or I217V) / USB3.0 14
CN7: Gigabit LAN (Intel I211AT) / USB3.0 (USB2.0 only for MI987EF version) 14
CN8: Audio Connector 14
CN2, CN3: SATA3 Connectors 14
F_USB1/2/3: USB Connectors 15
J2: Digital I/O 15
J4: Front Panel Function Connector 16
J5: SPI Flash Connector (Factory use only) 17
J6: MCU Flash Connector (factory use only) 17
J7: DC-in Jack (19V) 17
J8: COM2 Connector 18
JP9: Audio Front Header 18
J10/J14: HDD Power Connector (Output Only) 19
J12: Mini PCIE/mSATA Connector 19
J13: mSATA Connector 19
J16: Mini PCIE Connector 20
PCIE1: PCIE_x4 Slot 20
CPU_FAN1: CPU Fan Power Connector 20
SYS_FAN1: System Fan1 Power Connector 21

Connector Locations on MI987



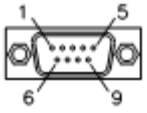




CN1: DC-IN Connector

****+19V DC-in Min+ 15.2V ~ Max +22.8V****

CN4: CRT/COM1 Connector



COM1

Pin #	Signal Name		
	RS-232	**R2-422**	**RS-485**
1	DCD	TX-	DATA-
2	RX	TX+	DATA+
3	TX	RX+	NC
4	DTR	RX-	NC
5	Ground	Ground	Ground
6	DSR	NC	NC
7	RTS	NC	NC
8	CTS	NC	NC
9	RI	NC	NC
10	NC	NC	NC

CN5: DP/HDMI Connector

CN6: Gigabit LAN (Intel I217LM or I217V) / USB3.0

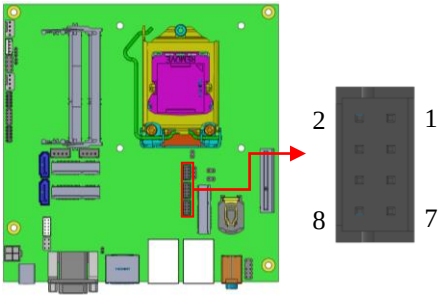
CN7: Gigabit LAN (Intel I211AT) / USB3.0

****Support USB2.0 only for MI987EF version****

CN8: Audio Connector

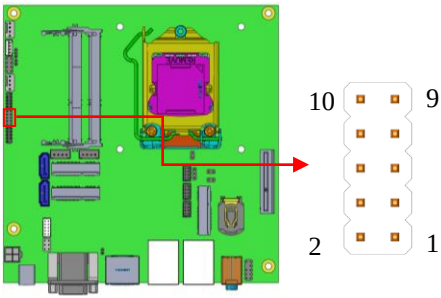
CN2, CN3: SATA3 Connectors

F_USB1/2/3: USB Connectors [HRS DF11-8DP-2DSA(08)]



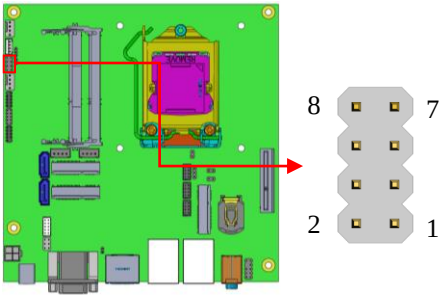
Signal Name	Pin #	Pin #	Signal Name
VCC	1	2	GND
D0-	3	4	D1+
D0+	5	6	D1-
GND	7	8	VCC

J2: Digital I/O (2.0mm)



Signal Name	Pin #	Pin #	Signal Name
GND	1	2	VCC(500mA)
OUT3	3	4	OUT1
OUT2	5	6	OUT0
IN3	7	8	IN1
IN2	9	10	IN0

J4: Front Panel Function Connector (2.54mm)



Pin #	Signal Name
8	GND
7	+5V
6	RESET
5	
4	HDD Active
3	3.3V
2	ATX Power
1	On/OFF

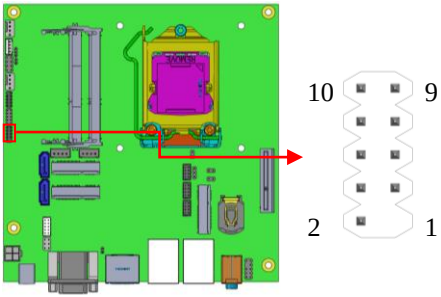
ATX Power ON Switch: Pins 1 and 2
This 2-pin connector is an "ATX Power Supply On/Off Switch" on the system that connects to the power switch on the case. When pressed, the power switch will force the system to power on. When pressed again, it will force the system to power off.

Hard Disk Drive LED Connector: Pins 3 and 4
This connector connects to the hard drive activity LED on control panel. This LED will flash when the HDD is being accessed.

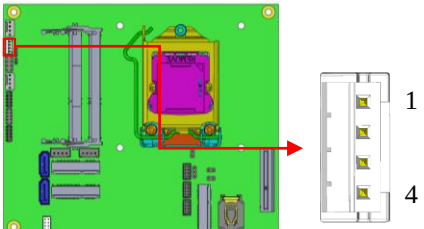
Reset Switch: Pins 5 and 6
The reset switch allows the user to reset the system without turning the main power switch off and then on again. Orientation is not required when making a connection to this header.

Power LED: Pins 7 and 8

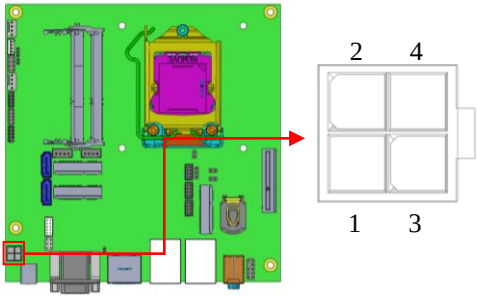
J5: SPI Flash Connector (Factory use only) [2.0mm]



J6: MCU Flash Connector (factory use only)

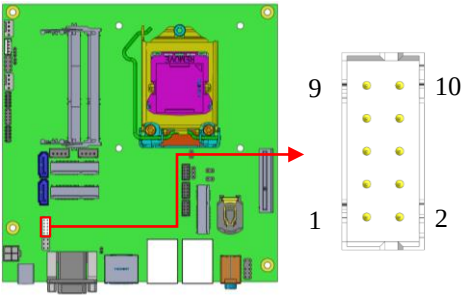


J7: DC-in Jack (19V) [HAOGUO ATX4PT-NY46]



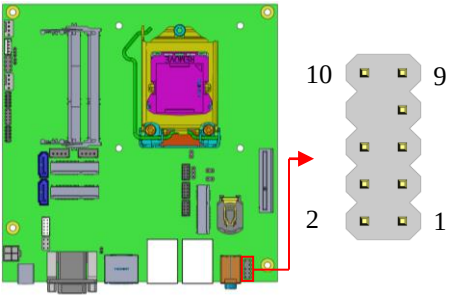
Pin #	Signal Name
1	Ground
2	Ground
3	DC_IN
4	DC_IN

J8: COM2 Connector [HRS DF11-10DP-2DSA(08)]



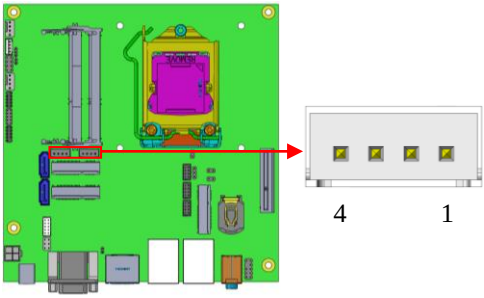
Signal Name	Pin #	Pin #	Signal Name
DCD, Data carrier detect	1	2	RXD, Receive data
TXD, Transmit data	3	4	DTR, Data terminal ready
GND, ground	5	6	DSR, Data set ready
RTS, Request to send	7	8	CTS, Clear to send
RI, Ring indicator	9	10	Not Used

JP9: Audio Front Header (2.54mm)



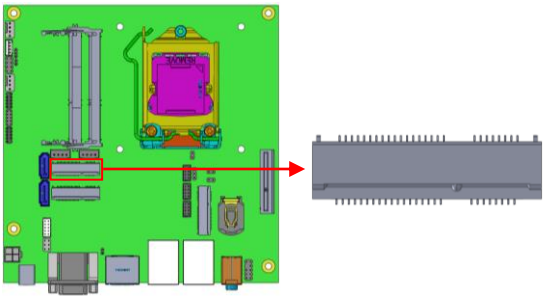
Signal Name	Pin #	Pin #	Signal Name
MIC2_L	1	2	Ground
MIC2_R	3	4	Presence#
Line2_R	5	6	MIC2_ID
Sense	7	8	NC
Line2_L	9	10	Line2_ID

J10/J14: HDD Power Connector *[E-CALL_0110-071-040]*
****Output Only****

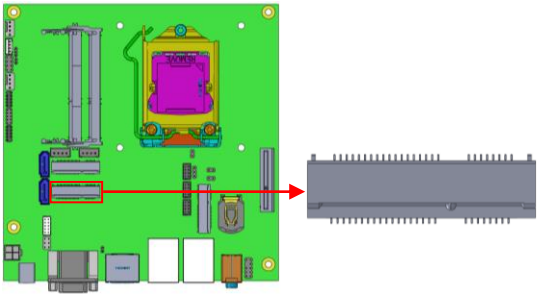


Pin #	Signal Name
1	+5V (2A)
2	Ground
3	Ground
4	+12V(1A)

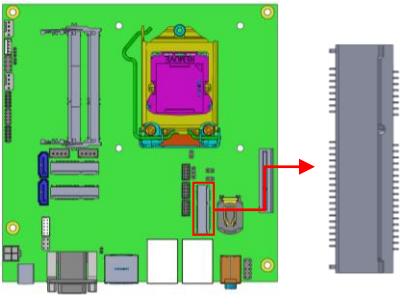
J12: Mini PCIE/mSATA Connector



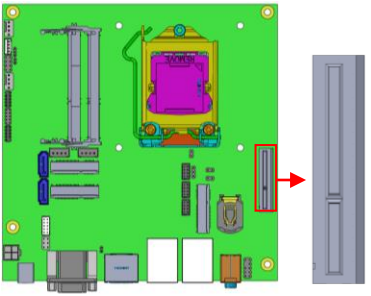
J13: mSATA Connector



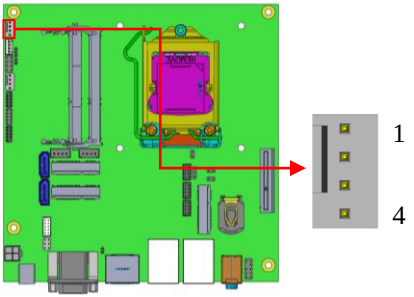
J16: Mini PCIE Connector



PCIE1: PCIE_x4 Slot

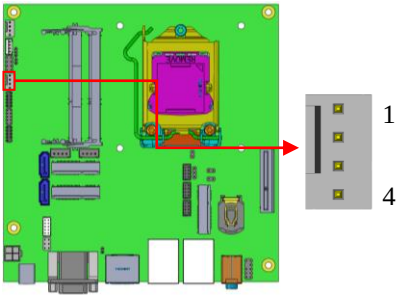


CPU_FAN1: CPU Fan Power Connector



Pin #	Signal Name
1	Ground
2	+12V(1A)
3	Rotation detection
4	Control

SYS_FAN1: System Fan1 Power Connector



Pin #	Signal Name
1	Ground
2	+12V(1A)
3	Rotation detection
4	Control

This page is intentionally left blank.

BIOS Setup

This chapter describes the different settings available in the AMI BIOS that comes with the board. The topics covered in this chapter are as follows:

BIOS Introduction 24
BIOS Setup 24
Advanced Settings 26
Chipset Settings 36
Boot Settings 41
CSM parameters 42
Security Settings 43
Save & Exit Settings 44

BIOS Introduction

The BIOS (Basic Input/Output System) installed in your computer system's ROM supports Intel processors. The BIOS provides critical low-level support for a standard device such as disk drives, serial ports and parallel ports. It also password protection as well as special support for detailed fine-tuning of the chipset controlling the entire system.

BIOS Setup

The BIOS provides a Setup utility program for specifying the system configurations and settings. The BIOS ROM of the system stores the Setup utility. When you turn on the computer, the BIOS is immediately activated. Pressing the <Del> key immediately allows you to enter the Setup utility. If you are a little bit late pressing the <Del> key, POST (Power On Self Test) will continue with its test routines, thus preventing you from invoking the Setup. If you still wish to enter Setup, restart the system by pressing the "Reset" button or simultaneously pressing the <Ctrl>, <Alt> and <Delete> keys. You can also restart by turning the system Off and back On again. The following message will appear on the screen:

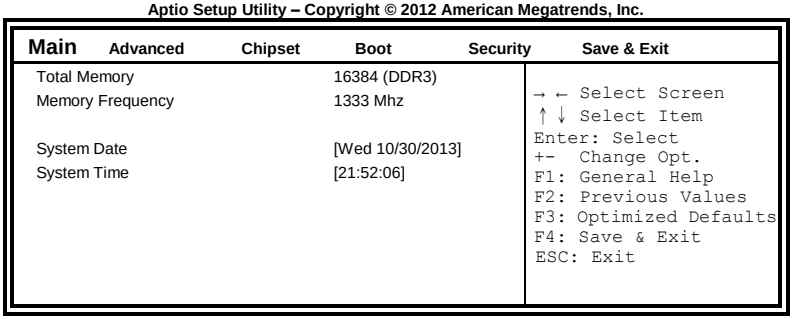
```
Press  <DEL>  to  Enter  Setup
```

In general, you press the arrow keys to highlight items, <Enter> to select, the <PgUp> and <PgDn> keys to change entries, <F1> for help and <Esc> to quit.

When you enter the Setup utility, the Main Menu screen will appear on the screen. The Main Menu allows you to select from various setup functions and exit choices.

Warning: *It is strongly recommended that you avoid making any changes to the chipset defaults. These defaults have been carefully chosen by both AMI and your system manufacturer to provide the absolute maximum performance and reliability. Changing the defaults could cause the system to become unstable and crash in some cases.*

Main Settings



System Date

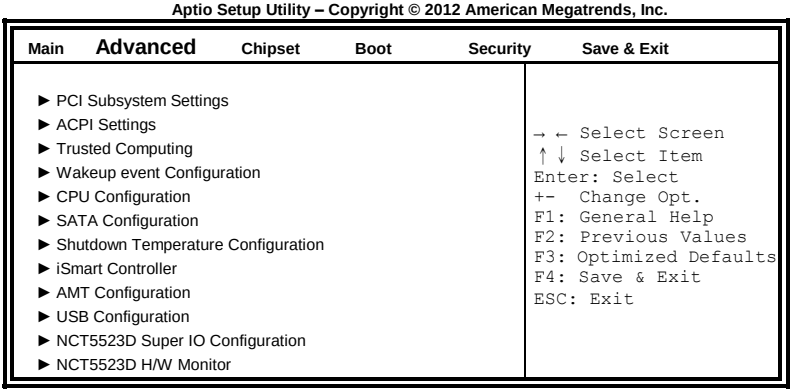
Set the Date. Use Tab to switch between Date elements.

System Time

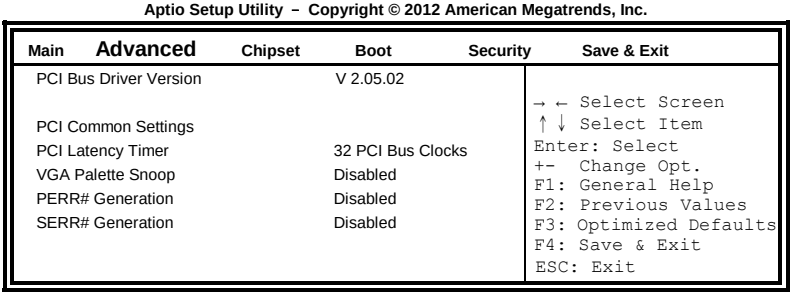
Set the Time. Use Tab to switch between Time elements.

Advanced Settings

This section allows you to configure and improve your system and allows you to set up some system features according to your preference.



PCI Subsystem Settings



PCI Latency Timer

Value to be programmed into PCI Latency Timer Register.

VGA Palette Snoop

Enables or disables VGA Palette Registers Snooping.

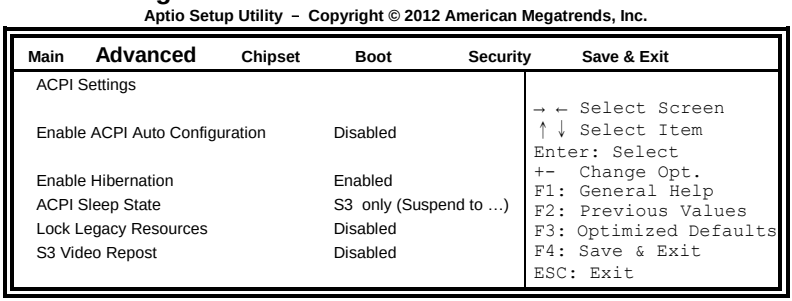
PERR# Generation

Enables or disables PCI device to generate PERR#.

SERR# Generation

Enables or disables PCI device to generate SERR#.

ACPI Settings



Enabled ACPI Auto Configuration

Enables or Disables BIOS ACPI Auto Configuration.

Enable Hibernation

Enables or Disables System ability to Hibernate (OS/S4 Sleep State). This option may be not effective with some OS.

ACPI Sleep State

Select ACPI sleep state the system will enter when the SUSPEND button is pressed.

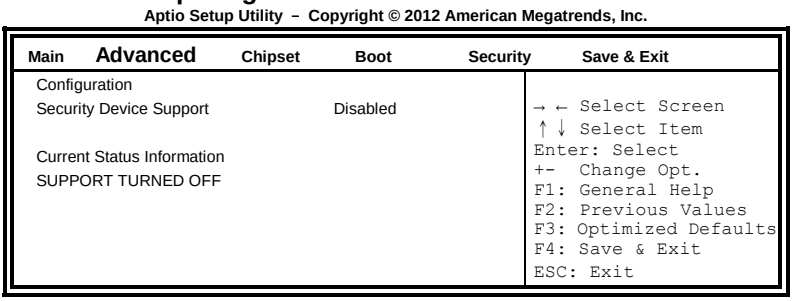
Lock Legacy Resources

Enables or Disables Lock of Legacy Resources.

S3 Video Repost

Enable or Disable S3 Video Repost.

Trusted Computing



Security Device Support

This configuration is supported only with MI987AF. Enables or disables BIOS support for security device. O.S. will not show Security Device. TCG EFI protocol and INT1A interface will not be available.

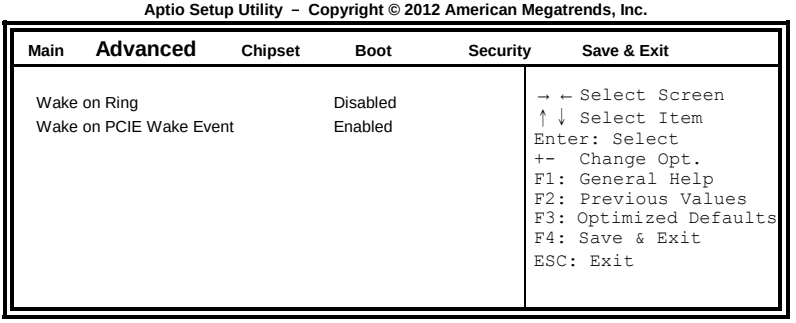
TPM State

Enable/Disable Security Device. NOTE: Your Computer will reboot during restart in order to change State of the Device.

Pending operation

Schedule an Operation for the Security Device. NOTE: Your Computer will reboot during restart in order to change State of Security Device.

Wake up event settings

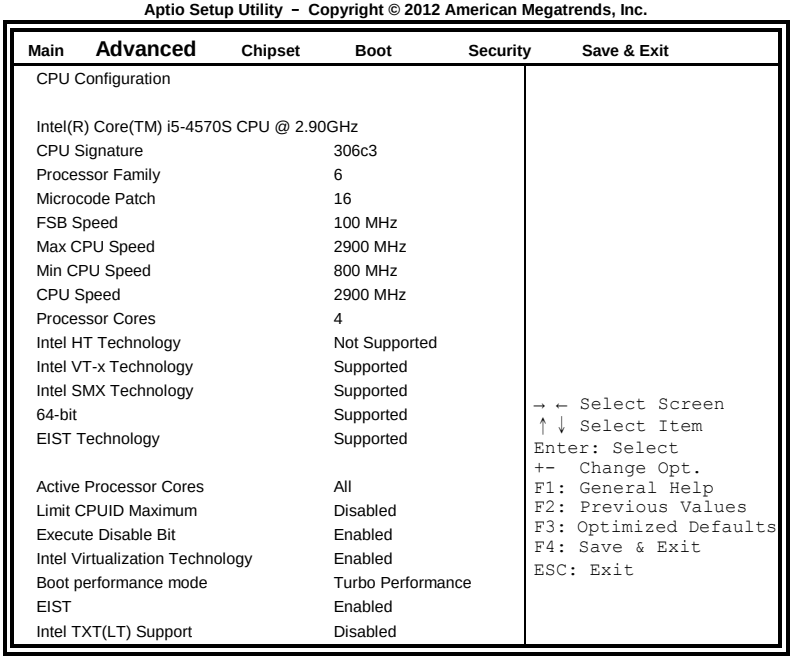


Wake up by Ring / Wake up by PCIE WAKE#

The options are Disabled and Enabled.

CPU Configuration

This section shows the CPU configuration parameters.



Active Processor Cores

Number of cores to enable in each processor package.

Execute Disable Bit

XD can prevent certain classes of malicious buffer overflow attacks when combined with a supporting OS (Windows Server 2003 SP1, SuSE Linux 9.2, RedHat Enterprise 3 Update 3.)

Intel Virtualization Technology

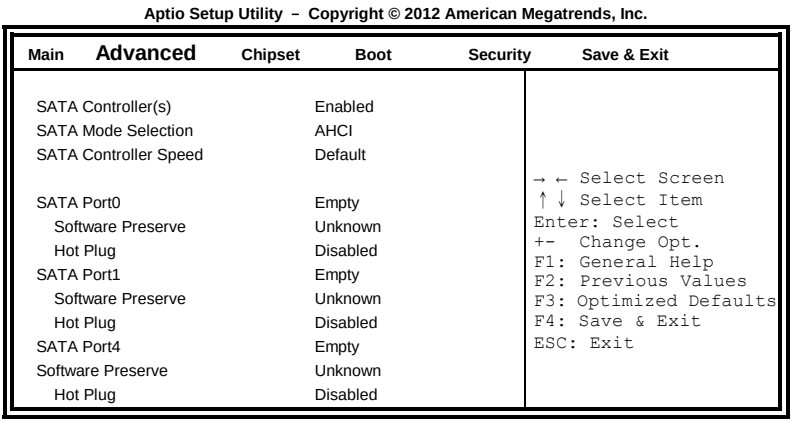
When enabled, a VMM can utilize the additional hardware capabilities provided by Vanderpool Technology.

Boot performance mode
Select the performance state that the BIOS will set before OS handoff.

EIST
Enable/Disable Intel Speedstep

Intel TXT(LT) Support
Enables or Disables Intel (R)TXT (LT) Support.

SATA Configuration
SATA Devices Configuration.



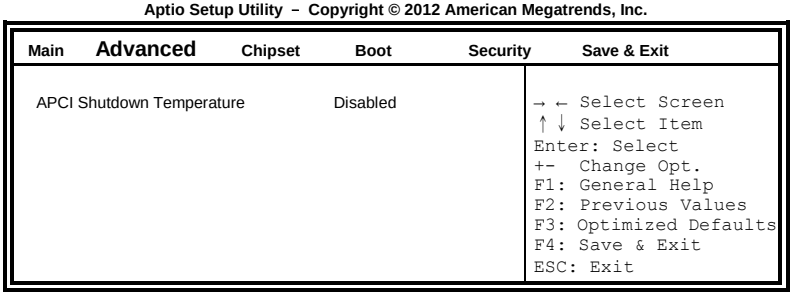
SATA Controller(s)
Enable or disable SATA Device.

SATA Mode Selection
Determines how SATA controller(s) operate.
(1) IDE Mode.
(2) AHCI Mode.
(3) RAID Mode. (MI987AF only)

SATA Controller Speed
Indicates the maximum speed the SATA controller can support.

Hot Plug
Designates this port as Hot Pluggable.

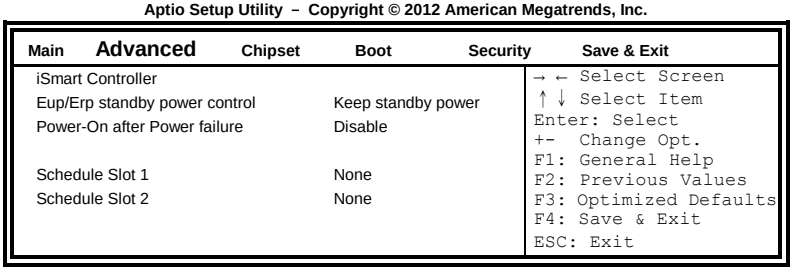
Shutdown Temperature Configuration



ACPI Shutdown Temperature

The default setting is Disabled .

iSmart Controller



Eup/Erp standby power control

This configuration is supported only with MI987EF. Eup/Erp control on S5[Keep standby power] Enable all of the standby power and ignore Eup/Erp specification .[Ethernet Only] Only provide the standby power for Ethernet chip.[Disabled] Shutdown all of the standby power.

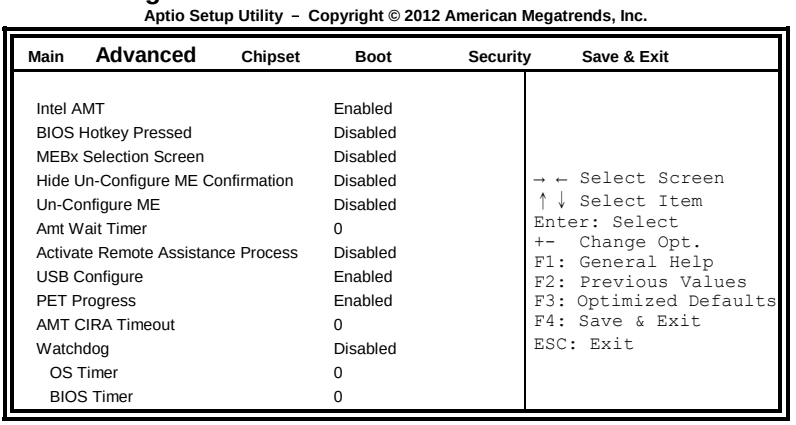
Power-On after Power failure

This field sets the system power status whether *Disable or Enable* when power returns to the system from a power failure situation.

Schedule Slot 1 / 2

Setup the hour/minute for system power on.

AMT Configuration



AMT configuration is supported only with MI987AF (with iAMT function).

Intel AMT

Enable/Disable Intel (R) Active Management Technology BIOS Extension.
Note: iAMT H/W is always enabled. This option just controls the BIOS extension execution. If enabled, this requires additional firmware in the SPI device.

BIOS Hotkey Pressed

OEMFLag Bit 1: Enable/Disable BIOS hotkey press.

AMT Configuration

OEMFLag Bit 2: Enable/Disable MEBx selection screen.

Hide Un-Configure ME Configuration

OEMFLag Bit 6: Hide Un-Configure ME without password Confirmation Prompt.

Un-Configure ME

OEMFLag Bit 15: Un-Configure ME without password.

Amt Wait Timer

Set timer to wait before sending ASF_GET_BOOT_OPTIONS.

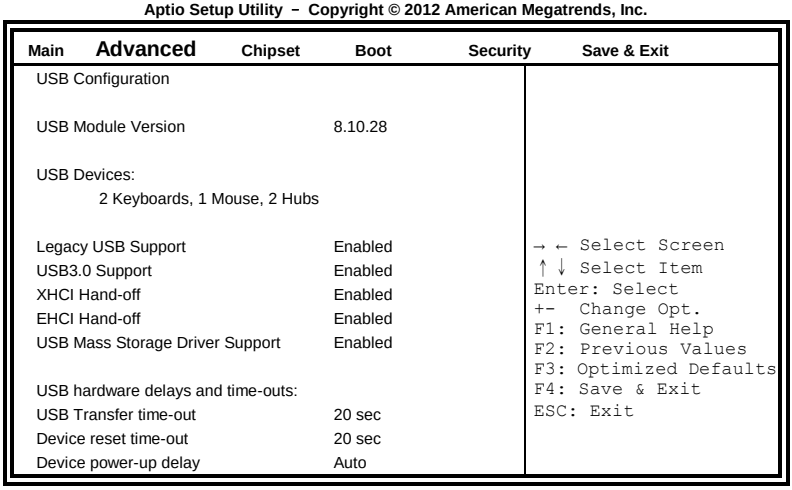
Activate Remote Assistance Process
Trigger CIRA boot.

USB Configure
Enable/Disable USB Configure function.

PET Progress
User can Enable/Disable PET Events progress to receive PET events or not.

Watchdog Timer
Enable/Disable Watchdog Timer.

USB Configuration



Legacy USB Support
Enables Legacy USB support.
AUTO option disables legacy support if no USB devices are connected.
DISABLE option will keep USB devices available only for EFI applications.

USB3.0 Support
Enable/Disable USB3.0 (XHCI) Controller support.

XHCI Hand-off
This is a workaround for OSes without XHCI hand-off support. The XHCI ownership change should be claimed by XHCI driver.

EHCI Hand-off
This is a workaround for OSes without EHCI hand-off support. The XHCI ownership change should be claimed by EHCI driver.

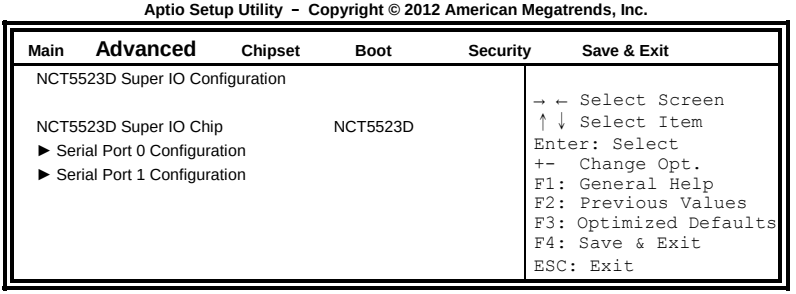
USB Mass Storage Driver Support
Enable/Disable USB Mass Storage Driver Support.

USB Transfer time-out
The time-out value for Control, Bulk, and Interrupt transfers.

Device reset tine-out
USB mass Storage device start Unit command time-out.

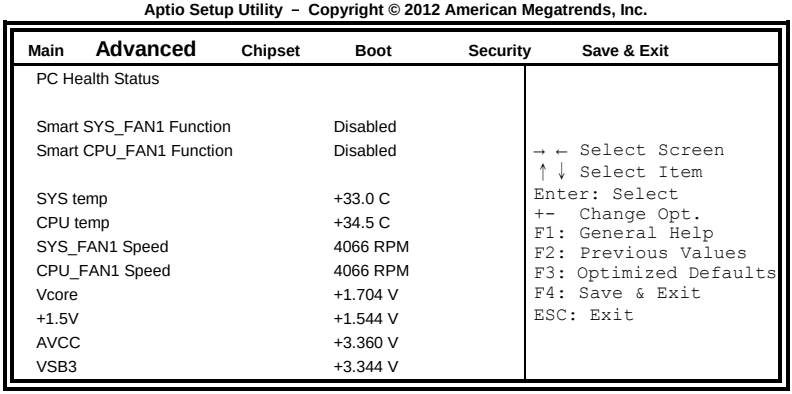
Device power-up delay
Maximum time the device will take before it properly reports itself to the Host Controller. 'Auto' uses default value: for a Root port it is 100ms, for a Hub port the delay is taken from Hub descriptor.

NCT5523D Super IO Configuration



Serial Port Configuration
Set Parameters of Serial Ports. User can Enable/Disable the serial port and Select an optimal settings for the Super IO Device.

NCT5523D H/W Monitor



Smart SYS_FAN1/CPU_FAN1 Function

This field enables or disables the smart fan feature.

Disabled (default)

50 ℃

60 ℃

70 ℃

80 ℃

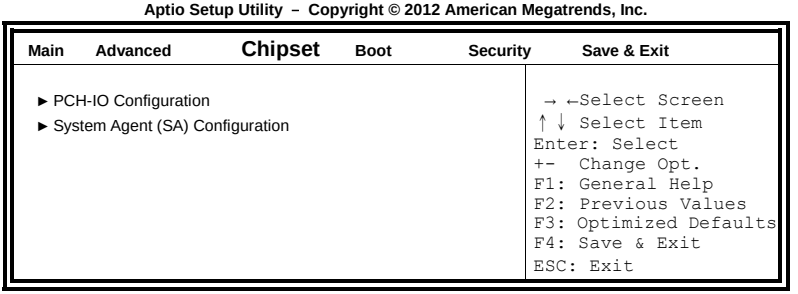
90 ℃

Temperatures/Voltages

These fields are the parameters of the hardware monitoring function feature of the motherboard. The values are read-only values as monitored by the system and show the PC health status.

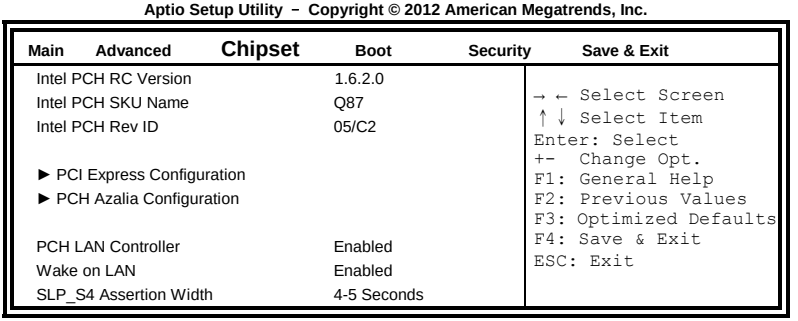
Chipset Settings

This section allows you to configure and improve your system and allows you to set up some system features according to your preference.



PCH-IO Configuration

This section allows you to configure the North Bridge Chipset.



PCH LAN Controller

Enable or disable onboard NIC.

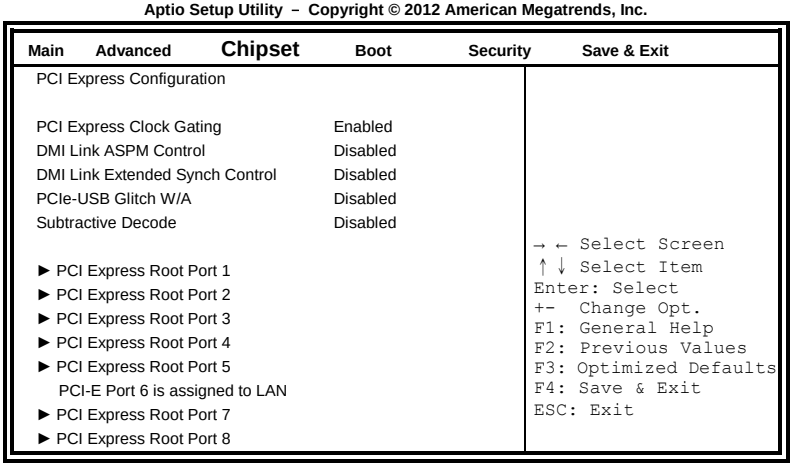
Wake on LAN

Enable or disable integrated LAN to wake the system. (The Wake On LAN cannot be disabled if ME is on at Sx state.)

SLP_S4 Assertion Width

Select a minimum assertion width of the SLP_S4# signal.

PCI Express Configuration



PCI Express Clock Gating

Enable or disable PCI Express Clock Gating for each root port.

DMI Link ASPM Control

The control of Active State Power Management on both NB side and SB side of the DMI Link.

DMI Link Extended Synch Control

The control of Extended Synch on SB side of the DMI Link.

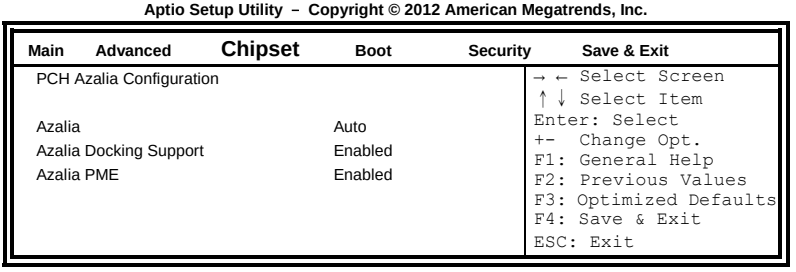
PCIe-USB Glitch W/A

PCIe-USB Glitch W/A for bad USB device(s) connected behind PCIE/PEG port.

Subtractive Decode

Enable or disable PCI Express Subtractive Decode.

PCH Azalia Configuration



Azalia

Control Detection of the Azalia device.
Disabled = Azalia will be unconditionally disabled.
Enabled Azalia will be unconditionally Enabled .
Auto = Azalia will be enabled if present, disabled otherwise.

Azalia Docking Support

Enable or disable Azalia Docking Support of Audio Controller.

Azalia PME

Enable or disable Power Management capability of Audio Controller.

System Agent (SA) Configuration

Aptio Setup Utility – Copyright © 2012 American Megatrends, Inc.

Main	Advanced	Chipset	Boot	Security	Save & Exit
System Agent Bridge Name			Haswell		
System Agent RC Version			1.6.2.0		
VT-d Capability			Supported		
VT-d			Enabled		
► Graphics Configuration					

→ ← Select Screen
↑ ↓ Select Item
Enter: Select
+- Change Opt.
F1: General Help
F2: Previous Values
F3: Optimized Defaults
F4: Save & Exit
ESC: Exit

VT-d

Check to enable VT-d function on MCH.

Graphics Configuration

Aptio Setup Utility – Copyright © 2012 American Megatrends, Inc.

Main	Advanced	Chipset	Boot	Security	Save & Exit
Graphics Configuration					
IGFX VBIOS Version			2170		
IGfx Frequency			700 MHz		
Primary Display			Auto		
Primary PEG			Auto		
Primary PCIE			Auto		
Internal Graphics			Auto		
Aperture Size			256MB		
DVMT Pre-Allocated			32M		
DVMT Total Gfx Mem			256MB		
Gfx Low Power Mode			Enabled		

→ ← Select Screen
↑ ↓ Select Item
Enter: Select
+- Change Opt.
F1: General Help
F2: Previous Values
F3: Optimized Defaults
F4: Save & Exit
ESC: Exit

Primary Display

Select which of IGFX/PEG/PCI graphics device should be Primary Display or select SG for switchable Gfx.

Primary PEG

Select PEG0/PEG1/PEG2/PEG3 Graphics device should be Primary PEG.

Primary PCIE
Select PCIE0/PCIE1/PCIE2/PCIE3/PCIE4/PCIE5/PCIE6PCIE7 Graphics device should be Primary PCIE.

Internal Graphics
Keep IGD enabled based on the setup options.

Aperture Size
Select the Aperture Size.

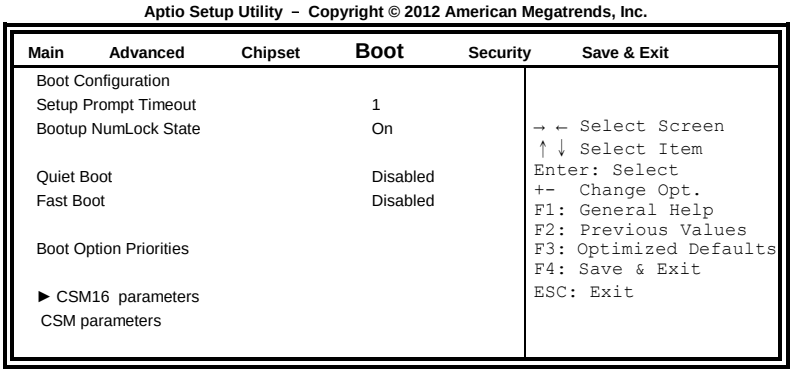
DVMT Pre-Allocated
Select DVMT 5.0 Pre-Allocated (Fixed) Graphics Memory Size used by the Internal Graphics Device.

DVMT Total Gfx Mem
Select DVMT 5.0 Total Graphics Memory Size used by the Internal Graphics Device.

Gfx Low Power Mode
This option is applicable for SFF only.

Boot Settings

This section allows you to configure the boot settings.



Setup Prompt Timeout

Number of seconds to wait for setup activation key.
65535(0xFFFF) means indefinite waiting.

Bootup NumLock State

Select the keyboard NumLock state.

Quiet Boot

Enables or disables Quiet Boot option.

Fast Boot

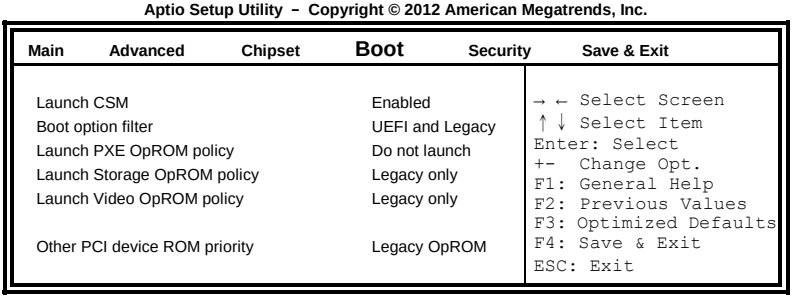
Enables or disables boot with initialization of a minimal set of devices required to launch active boot option. Has no effect for BBS boot options.

Boot Option Priorities

Sets the system boot order.

CSM parameters

This section allows you to configure the boot settings.



Launch CSM

This option controls if CSM will be launched.

Boot option filter

This option controls what devices system can boot to.

Launch PXE OpROM policy

Controls the execution of UEFI and Legacy PXE OpROM.

Launch Storatge OpROM policy

Controls the execution of UEFI and Legacy Storage OpROM.

Launch Video OpROM policy

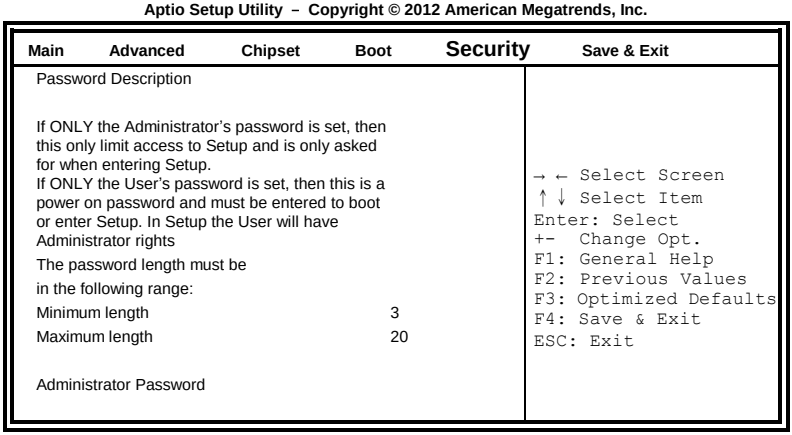
Controls the execution of UEFI and Legacy Video OpROM.

Other PCI device ROM priority

For PCI devices other than Network, Mass storage or Video defines which OpROM to launch.

Security Settings

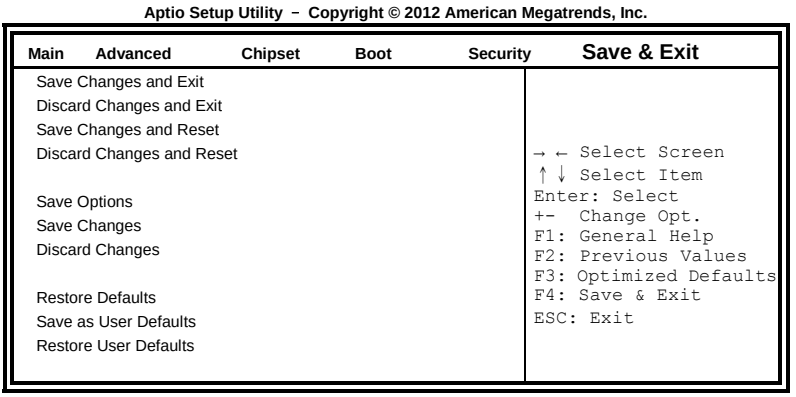
This section allows you to configure and improve your system and allows you to set up some system features according to your preference.



Administrator Password

Set Administrator Password.

Save & Exit Settings



Save Changes and Exit
Exit system setup after saving the changes.

Discard Changes and Exit
Exit system setup without saving any changes.

Save Changes and Reset
Reset the system after saving the changes.

Discard Changes and Reset
Reset system setup without saving any changes.

Save Changes
Save Changes done so far to any of the setup options.

Discard Changes
Discard Changes done so far to any of the setup options.

Restore Defaults
Restore/Load Defaults values for all the setup options.

Save as User Defaults
Save the changes done so far as User Defaults.

Restore User Defaults
Restore the User Defaults to all the setup options.

Drivers Installation

This section describes the installation procedures for software and drivers. The software and drivers are included with the motherboard. If you find the items missing, please contact the vendor where you made the purchase. The contents of this section include the following:

Intel Chipset Software Installation Utility 46
VGA Drivers Installation..... 47
Realtek HD Audio Driver Installation 48
LAN Drivers Installation 49
Intel® Management Engine Interface 50
Intel® USB 3.0 Drivers 51

IMPORTANT NOTE:
After installing your Windows operating system, you must install first the Intel Chipset Software Installation Utility before proceeding with the drivers installation.

Intel Chipset Software Installation Utility

The Intel Chipset Drivers should be installed first before the software drivers to enable Plug & Play INF support for Intel chipset components. Follow the instructions below to complete the installation.

1. Insert the DVD that comes with the board. Click ***Intel*** and then ***Intel(R) 8 Series Chipset Drivers***.

2. Click ***Intel(R) Chipset Software Installation Utility.***

3. When the Welcome screen to the Intel® Chipset Device Software appears, click ***Next*** to continue.

4. Click ***Yes*** to accept the software license agreement and proceed with the installation process.

5. On the Readme File Information screen, click ***Next*** to continue the installation.

6. The Setup process is now complete. Click ***Finish*** to restart the computer and for changes to take effect.

VGA Drivers Installation

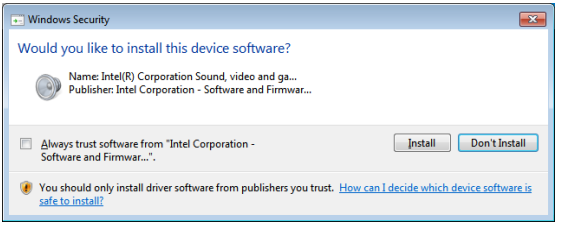
1. Insert the DVD that comes with the board. Click ***Intel*** and then ***Intel(R) 8 Series Chipset Drivers***. Click ***Intel(R) Graphics Driver.***

2. When the Welcome screen appears, click ***Next*** to continue.

3. Click ***Yes*** to to agree with the license agreement and continue the installation.

4. On the Readme File Information screen, click ***Next*** to continue the installation of the Intel® HD Graphics Driver.

5. On the screen shown below, click ***Install*** to continue.



6. On the Setup Progress screen, click ***Next*** to continue.

7. Setup complete. Click ***Finish*** to restart the computer and for changes to take effect.

Realtek HD Audio Driver Installation

1. Insert the DVD that comes with the board. Click ***Intel*** and then ***Intel(R) 8 Series Chipset Drivers***. Click ***Realtek High Definition Audio Driver.***

2. On the Welcome to the InstallShield Wizard screen, click ***Next*** to proceed with and complete the installation process.

3. The InstallShield Wizard Complete. Click ***Finish*** to restart the computer and for changes to take effect.

LAN Drivers Installation

1. Insert the DVD that comes with the board. Click ***Intel*** and then ***Intel(R) 8 Series Chipset Drivers***. Click ***Intel(R) PRO LAN Network Drivers.***

2. Click ***Install Drivers and Software.***

4. When the Welcome screen appears, click ***Next***.

5. Click ***Next*** to to agree with the license agreement.

6. Click the checkbox for **Drivers** in the Setup Options screen to select it and click **Next** to continue.

7. The wizard is ready to begin installation. Click ***Install*** to begin the installation.

8. When InstallShield Wizard is complete, click ***Finish***.

Intel® Management Engine Interface

Follow the steps below to install the Intel Management Engine.

1. Insert the DVD that comes with the board. Click ***Intel*** and then ***Intel(R) 8 Series Chipset Drivers*** and then ***Intel(R) ME 9.0 Drivers***.

2. When the Welcome screen to the InstallShield Wizard for Intel® Management Engine Components, click the checkbox for **Install Intel® Control Center** & click ***Next***.

3. Click ***Yes*** to to agree with the license agreement.

4. When the Setup Progress screen appears, click ***Next***. Then, click ***Finish*** when the setup progress has been successfully installed.

Intel® USB 3.0 Drivers

1. Insert the DVD that comes with the board. Click ***Intel*** and then ***Intel(R) 8 Series Chipset Drivers***. Click ***Intel(R) USB 3.0 Drivers.***

2. When the Welcome screen to the InstallShield Wizard for Intel® USB 3.0 eXtensible Host Controller Driver, click ***Next***.

3. Click ***Yes*** to to agree with the license agreement and continue the installation.

4. On the Readme File Information screen, click ***Next*** to continue the installation of the Intel® USB 3.0 eXtensible Host Controller Driver.

5. When the Setup Progress screen appears, click ***Next***. Setup complete. Click ***Finish*** to restart the computer and for changes to take effect.

This page is intentionally set blank.

Appendix

A. I/O Port Address Map

Each peripheral device in the system is assigned a set of I/O port addresses which also becomes the identity of the device. The following table lists the I/O port addresses used.

Address	Device Description
0000h-001Fh	Direct memory access controller
0000h-001Fh	PCI bus
0040h-0043h	System timer
0050h-0053h	System timer
0070h-0077h	System CMOS/real time clock
0081h-0091h	Direct memory access controller
0093h-009Fh	Direct memory access controller
00C0h-00DFh	Direct memory access controller
00F0h-00F0h	Numeric data processor
02F8h-02FFh	Communications Port (COM2)
03B0h-03BBh	Intel(R) HD Graphics 4600
03C0h-03DFh	Intel(R) HD Graphics 4600
03F8h-03FFh	Communications Port (COM1)
0D00h-FFFFh	PCI bus
E000h-EFFFh	Intel(R) 8 Series/C220 Series PCI Express Root Port #7 - 8C1C
F000h-F03Fh	Intel(R) HD Graphics 4600
F040h-F05Fh	Intel(R) 8 Series/C220 Series SMBus Controller - 8C22
F060h-F07Fh	Intel(R) 8 Series/C220 Series SATA AHCI Controller - 8C02
F0A0h-F0A3h	Intel(R) 8 Series/C220 Series SATA AHCI Controller - 8C02
F0B0h-F0B7h	Intel(R) 8 Series/C220 Series SATA AHCI Controller - 8C02
F0C0h-F0C3h	Intel(R) 8 Series/C220 Series SATA AHCI Controller - 8C02
F0D0h-F0D7h	Intel(R) 8 Series/C220 Series SATA AHCI Controller - 8C02
F0E0h-F0E7h	Intel(R) Active Management Technology - SOL (COM3)

B. Interrupt Request Lines (IRQ)

Peripheral devices use interrupt request lines to notify CPU for the service required. The following table shows the IRQ used by the devices on board.

Level	Function
IRQ0	System Timer
IRQ3	Serial Port #2
IRQ4	Serial Port #1
IRQ 10	Intel(R) 8 Series/C220 Series SMBus Controller - 8C22
IRQ 13	Numeric data processor
IRQ 16	High Definition Audio Controller
IRQ 16	Intel(R) 8 Series/C220 Series USB EHCI #2 - 8C2D
IRQ 16	Intel(R) Management Engine Interface
IRQ 19	Intel(R) 8 Series/C220 Series SATA AHCI Controller - 8C02
IRQ 19	Intel(R) Active Management Technology - SOL (COM3)
IRQ 22	High Definition Audio Controller
IRQ 23	Intel(R) 8 Series/C220 Series USB EHCI #1 - 8C26

C. Digital I/O Sample Code

File of the NCT5523D.H

```
//---------------------------------------------------------------------------
// THIS CODE AND INFORMATION IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY
// KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE
// IMPLIED WARRANTIES OF MERCHANTABILITY AND/OR FITNESS FOR A PARTICULAR
// PURPOSE.
//---------------------------------------------------------------------------
#ifndef __NCT5523D_H
#define __NCT5523D_H                1
//---------------------------------------------------------------------------
#define     NCT5523D_INDEX_PORT         (NCT5523D_BASE)
#define     NCT5523D_DATA_PORT          (NCT5523D_BASE+1)
//---------------------------------------------------------------------------
#define     NCT5523D_REG_LD             0x07
//---------------------------------------------------------------------------
#define NCT5523D_UNLOCK             0x87
#define     NCT5523D_LOCK               0xAA
//---------------------------------------------------------------------------
unsigned int Init_NCT5523D(void);
void Set_NCT5523D_LD( unsigned char);
void Set_NCT5523D_Reg( unsigned char, unsigned char);
unsigned char Get_NCT5523D_Reg( unsigned char);
//---------------------------------------------------------------------------
#endif//__NCT5523D_H
```

File of the MAIN.CPP

```
//---------------------------------------------------------------------------
// THIS CODE AND INFORMATION IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY
// KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE
// IMPLIED WARRANTIES OF MERCHANTABILITY AND/OR FITNESS FOR A PARTICULAR
// PURPOSE.
//---------------------------------------------------------------------------
#include <dos.h>
#include <conio.h>
#include <stdio.h>
#include <stdlib.h>
#include "NCT5523D.H"
//---------------------------------------------------------------------------
int main (void);

void Dio5Initial(void);
void Dio5SetOutput(unsigned char);
unsigned char Dio5GetInput(void);
void Dio5SetDirection(unsigned char);
unsigned char Dio5GetDirection(void);
//---------------------------------------------------------------------------
int main (void)
{
	char SIO;

	SIO = Init_NCT5523D();
	if (SIO == 0)
	{
		printf("Can not detect Nuvoton NCT5523D, program abort.\n");
		return(1);
	}

	Dio5Initial();

	//for GPIO20..27
	Dio5SetDirection(0x0F); //GP20..23 = input, GP24..27=output
	printf("Current DIO direction = 0x%X\n", Dio5GetDirection());

	printf("Current DIO status = 0x%X\n", Dio5GetInput());

	printf("Set DIO output to high\n");
	Dio5SetOutput(0x0F);

	printf("Set DIO output to low\n");
	Dio5SetOutput(0x00);

	return 0;
}
//---------------------------------------------------------------------------
```

```
void Dio5Initial(void)
{
        unsigned char ucBuf;

    ucBuf = Get_NCT5523D_Reg(0x1C);
    ucBuf &= ~0x02;
    Set_NCT5523D_Reg(0x1C, ucBuf);

        Set_NCT5523D_LD(0x07);                                  //switch to logic device 7
        //enable the GP2 group
        ucBuf = Get_NCT5523D_Reg(0x30);
        ucBuf |= 0x04;
        Set_NCT5523D_Reg(0x30, ucBuf);
}
//---------------------------------------------------------------------------
void Dio5SetOutput(unsigned char NewData)
{
        Set_NCT5523D_LD(0x07);                                  //switch to logic device 7
        Set_NCT5523D_Reg(0xE1, NewData);
}
//---------------------------------------------------------------------------
unsigned char Dio5GetInput(void)
{
        unsigned char result;

        Set_NCT5523D_LD(0x07);                                  //switch to logic device 7
        result = Get_NCT5523D_Reg(0xE1);
        return (result);
}
//---------------------------------------------------------------------------
void Dio5SetDirection(unsigned char NewData)
{
        //NewData : 1 for input, 0 for output
        Set_NCT5523D_LD(0x07);                                  //switch to logic device 7
        Set_NCT5523D_Reg(0xE8, NewData);
}
//---------------------------------------------------------------------------
unsigned char Dio5GetDirection(void)
{
        unsigned char result;

        Set_NCT5523D_LD(0x07);                                  //switch to logic device 7
        result = Get_NCT5523D_Reg(0xE8);
        return (result);
}
//---------------------------------------------------------------------------
```

File of the NCT5523D.CPP

```
//---------------------------------------------------------------------------
// THIS CODE AND INFORMATION IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY
// KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE
// IMPLIED WARRANTIES OF MERCHANTABILITY AND/OR FITNESS FOR A PARTICULAR
// PURPOSE.
//---------------------------------------------------------------------------
#include "NCT5523D.H"
#include <dos.h>
//---------------------------------------------------------------------------
unsigned int NCT5523D_BASE;
void Unlock_NCT5523D (void);
void Lock_NCT5523D (void);
//---------------------------------------------------------------------------
unsigned int Init_NCT5523D(void)
{
        unsigned int result;
        unsigned char ucDid;

        NCT5523D_BASE = 0x4E;
        result = NCT5523D_BASE;

        ucDid = Get_NCT5523D_Reg(0x20);
        if (ucDid == 0xC4)                                      //NCT5523D??
        {       goto Init_Finish;    }

        NCT5523D_BASE = 0x2E;
        result = NCT5523D_BASE;

        ucDid = Get_NCT5523D_Reg(0x20);
        if (ucDid == 0xC4)                                      //NCT5523D??
        {       goto Init_Finish;    }

        NCT5523D_BASE = 0x00;
        result = NCT5523D_BASE;

Init_Finish:
        return (result);
}
//---------------------------------------------------------------------------
void Unlock_NCT5523D (void)
{
        outportb(NCT5523D_INDEX_PORT, NCT5523D_UNLOCK);
        outportb(NCT5523D_INDEX_PORT, NCT5523D_UNLOCK);
}
//---------------------------------------------------------------------------
void Lock_NCT5523D (void)
{
        outportb(NCT5523D_INDEX_PORT, NCT5523D_LOCK);
}
//---------------------------------------------------------------------------
```

```
void Set_NCT5523D_LD( unsigned char LD)
{
        Unlock_NCT5523D();
        outportb(NCT5523D_INDEX_PORT, NCT5523D_REG_LD);
        outportb(NCT5523D_DATA_PORT, LD);
        Lock_NCT5523D();
}
//----------------------------------------------------------------------------
void Set_NCT5523D_Reg( unsigned char REG, unsigned char DATA)
{
        Unlock_NCT5523D();
        outportb(NCT5523D_INDEX_PORT, REG);
        outportb(NCT5523D_DATA_PORT, DATA);
        Lock_NCT5523D();
}
//----------------------------------------------------------------------------
unsigned char Get_NCT5523D_Reg(unsigned char REG)
{
        unsigned char Result;
        Unlock_NCT5523D();
        outportb(NCT5523D_INDEX_PORT, REG);
        Result = inportb(NCT5523D_DATA_PORT);
        Lock_NCT5523D();
        return Result;
}
//----------------------------------------------------------------------------
```

D. Watchdog Timer Configuration

The WDT is used to generate a variety of output signals after a user programmable count. The WDT is suitable for use in the prevention of system lock-up, such as when software becomes trapped in a deadlock. Under these sorts of circumstances, the timer will count to zero and the selected outputs will be driven. Under normal circumstance, the user will restart the WDT at regular intervals before the timer counts to zero.

SAMPLE CODE:

```
File of the NCT5523D.H
//-----------------------------------------------------------------------------
//
// THIS CODE AND INFORMATION IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY
// KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE
// IMPLIED WARRANTIES OF MERCHANTABILITY AND/OR FITNESS FOR A PARTICULAR
// PURPOSE.
//
//-----------------------------------------------------------------------------
#ifndef __NCT5523D_H
#define __NCT5523D_H                1
//-----------------------------------------------------------------------------
#define       NCT5523D_INDEX_PORT           (NCT5523D_BASE)
#define       NCT5523D_DATA_PORT            (NCT5523D_BASE+1)
//-----------------------------------------------------------------------------
#define       NCT5523D_REG_LD               0x07
//-----------------------------------------------------------------------------
#define NCT5523D_UNLOCK             0x87
#define       NCT5523D_LOCK                 0xAA
//-----------------------------------------------------------------------------
unsigned int Init_NCT5523D(void);
void Set_NCT5523D_LD( unsigned char);
void Set_NCT5523D_Reg( unsigned char, unsigned char);
unsigned char Get_NCT5523D_Reg( unsigned char);
//-----------------------------------------------------------------------------
#endif//__NCT5523D_H
```

File of the MAIN.CPP.

```
//---------------------------------------------------------------------------
//
// THIS CODE AND INFORMATION IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY
// KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE
// IMPLIED WARRANTIES OF MERCHANTABILITY AND/OR FITNESS FOR A PARTICULAR
// PURPOSE.
//
//---------------------------------------------------------------------------
#include <dos.h>
#include <conio.h>
#include <stdio.h>
#include <stdlib.h>
#include "NCT5523D.H"
//---------------------------------------------------------------------------
int main (void);

void WDTInitial(void);
void WDTEnable(unsigned char);
void WDTDisable(void);

//---------------------------------------------------------------------------
int main (void)
{
	char SIO;

	SIO = Init_NCT5523D();
	if (SIO == 0)
	{
		printf("Can not detect Nuvoton NCT5523D, program abort.\n");
		return(1);
	}

	WDTInitial();

	WDTEnable(10);

	WDTDisable();

	return 0;
}
//---------------------------------------------------------------------------
void WDTInitial(void)
{
	unsigned char bBuf;
	Set_NCT5523D_LD(0x08);                          //switch to logic device 8
	bBuf = Get_NCT5523D_Reg(0x30);
	bBuf &= (~0x01);
	Set_NCT5523D_Reg(0x30, bBuf);                   //Enable WDTO
}
//---------------------------------------------------------------------------
```

```
void WDTEnable(unsigned char NewInterval)
{
	unsigned char bBuf;

	Set_NCT5523D_LD(0x08);                          //switch to logic device 8
	Set_NCT5523D_Reg(0x30, 0x01);                   //enable timer

	bBuf = Get_NCT5523D_Reg(0xF0);
	bBuf &= (~0x08);
	Set_NCT5523D_Reg(0xF0, bBuf);                   //count mode is second

	Set_NCT5523D_Reg(0xF1, NewInterval);            //set timer
}
//-------------------------------------------------------------------------
void WDTDisable(void)
{
	Set_NCT5523D_LD(0x08);                          //switch to logic device 8
	Set_NCT5523D_Reg(0xF1, 0x00);                   //clear watchdog timer
	Set_NCT5523D_Reg(0x30, 0x00);                   //watchdog disabled
}
//-------------------------------------------------------------------------
```

File of the NCT5523D.CPP

```
//---------------------------------------------------------------------------
//
// THIS CODE AND INFORMATION IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY
// KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE
// IMPLIED WARRANTIES OF MERCHANTABILITY AND/OR FITNESS FOR A PARTICULAR
// PURPOSE.
//
//---------------------------------------------------------------------------
#include "NCT5523D.H"
#include <dos.h>
//---------------------------------------------------------------------------
unsigned int NCT5523D_BASE;
void Unlock_NCT5523D (void);
void Lock_NCT5523D (void);
//---------------------------------------------------------------------------
unsigned int Init_NCT5523D(void)
{
        unsigned int result;
        unsigned char ucDid;

        NCT5523D_BASE = 0x4E;
        result = NCT5523D_BASE;

        ucDid = Get_NCT5523D_Reg(0x20);
        if (ucDid == 0xC4)                                      //NCT5523D??
        {       goto Init_Finish;    }

        NCT5523D_BASE = 0x2E;
        result = NCT5523D_BASE;

        ucDid = Get_NCT5523D_Reg(0x20);
        if (ucDid == 0xC4)                                      //NCT5523D??
        {       goto Init_Finish;    }

        NCT5523D_BASE = 0x00;
        result = NCT5523D_BASE;

Init_Finish:
        return (result);
}
//---------------------------------------------------------------------------
void Unlock_NCT5523D (void)
{
        outportb(NCT5523D_INDEX_PORT, NCT5523D_UNLOCK);
        outportb(NCT5523D_INDEX_PORT, NCT5523D_UNLOCK);
}
//---------------------------------------------------------------------------
void Lock_NCT5523D (void)
{
        outportb(NCT5523D_INDEX_PORT, NCT5523D_LOCK);
}
//---------------------------------------------------------------------------
```

```
void Set_NCT5523D_LD( unsigned char LD)
{
	Unlock_NCT5523D();
	outportb(NCT5523D_INDEX_PORT, NCT5523D_REG_LD);
	outportb(NCT5523D_DATA_PORT, LD);
	Lock_NCT5523D();
}
//----------------------------------------------------------------------------
void Set_NCT5523D_Reg( unsigned char REG, unsigned char DATA)
{
	Unlock_NCT5523D();
	outportb(NCT5523D_INDEX_PORT, REG);
	outportb(NCT5523D_DATA_PORT, DATA);
	Lock_NCT5523D();
}
//----------------------------------------------------------------------------
unsigned char Get_NCT5523D_Reg(unsigned char REG)
{
	unsigned char Result;
	Unlock_NCT5523D();
	outportb(NCT5523D_INDEX_PORT, REG);
	Result = inportb(NCT5523D_DATA_PORT);
	Lock_NCT5523D();
	return Result;
}
//----------------------------------------------------------------------------
```