ET950

Intel® QM87

COM-Express Module

USER'S MANUAL

Version 1.0

Acknowledgments

AMI is a registered trademark of American Megatrends Inc.
PS/2 is a trademark of International Business Machines Corporation.
Intel and Intel® Ivy Bridge DC Mobile Processor are registered trademarks of Intel Corporation.
Microsoft Windows is a registered trademark of Microsoft Corporation.
All other product names or trademarks are properties of their respective owners.

Table of Contents

This page is intentionally left blank.

Introduction

Product Description

The ET950 COM-Express Module is based on the latest Intel® QM87 chipset with pin-out Type 6 that fully complies with the PICMG (PCI Industrial Computer Manufactures Group) COM.0 R2.0 specification.. The platform supports 3rd generation Intel® Core processor family with BGA packing and feature an integrated dual-channel DDR3 memory controller as well as a graphics core.

The QM87 platform is made with 22-nanometer technology that supports Intel's first processor architecture to unite the CPU and the graphics core on the transistor level. The ET950 COM-Express Module utilizes the dramatic increase in performance provided this Intel's latest cutting-edge technology. Measuring 125mm x 95mm, the ET950 offers fast 6Gbps SATA , USB3.0 and DisplayPort. ET950 features Intel Active Management Technology 8.0.

ET950F FEATURES:

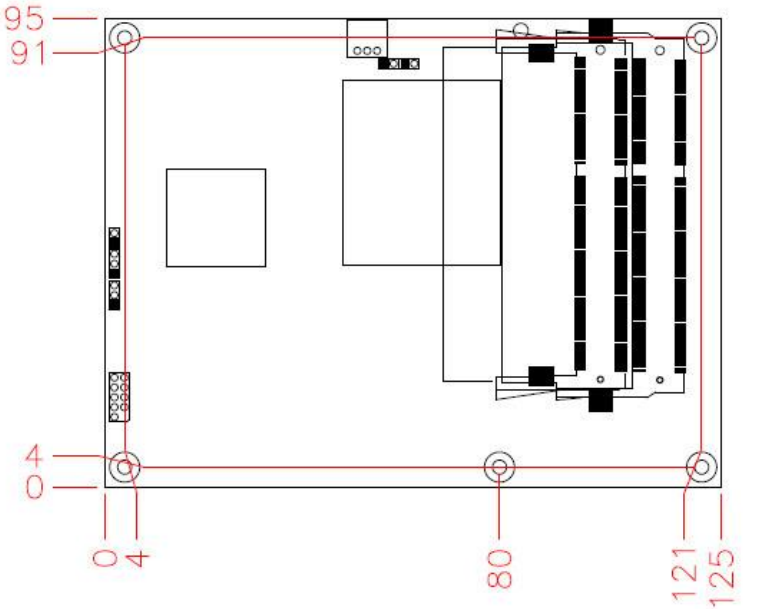
- Supports Intel® 3rd Generation Core i7/i5/i3 QC/DC mobile processors
- Two DDR3 SoDIMM, 1066/1333/1600MHz, Max. 16GB memory
- Intel® PCI-Express Gigabit LAN
- Integrated Graphics for VGA/DisplayPort/LVDS displays
- 2x SATA 2.0, 2x SATA 3.0, 8x USB 2.0, USB 3.0 (4 ports)
- 1x PEG (x16), 7x PCI-EX1

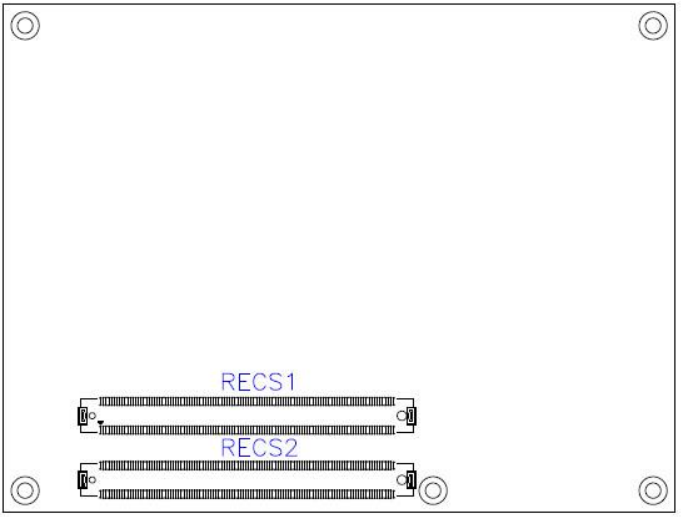
Checklist

Your ET950 package should include the items listed below.

- The ET950 COM-Express Module
- This User's Manual
- 1 CD containing chipset drivers and flash memory utility
- 1 heat sink

Board Dimensions





Installations

This section provides information on how to use the jumpers and connectors on the ET950 in order to set up a workable system. The topics covered are:

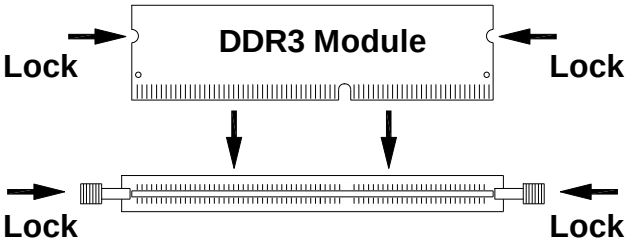
Installing the Memory

The ET950 board supports two DDR3 memory socket for a maximum total memory of 16GB in DDR3 SO-DIMM memory type.

Installing and Removing Memory Modules

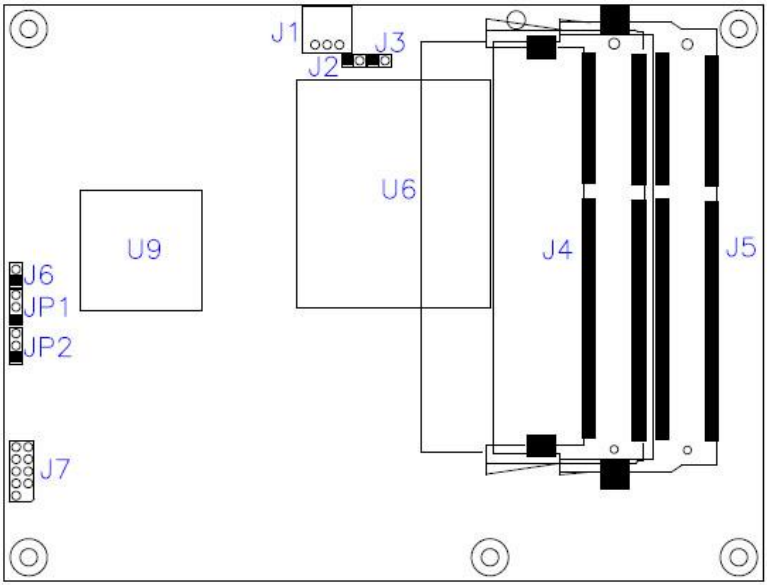
To install the DDR3 modules, locate the memory slot on the board and perform the following steps:

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 ET950 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 ET930 and their respective functions.



J7: SPI Flash connector (Factory use only)

J6: Flash Descriptor Security Override (Factory use only)

J6	Flash Descriptor Security Override
Open	Disabled (Default)
Close	Enabled

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

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:

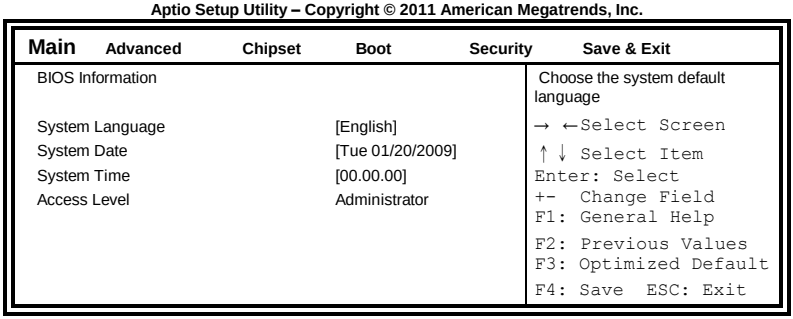
```
Press  <DEL>  to  Enter  Setup
```

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 Language

Choose the system default language.

System Date

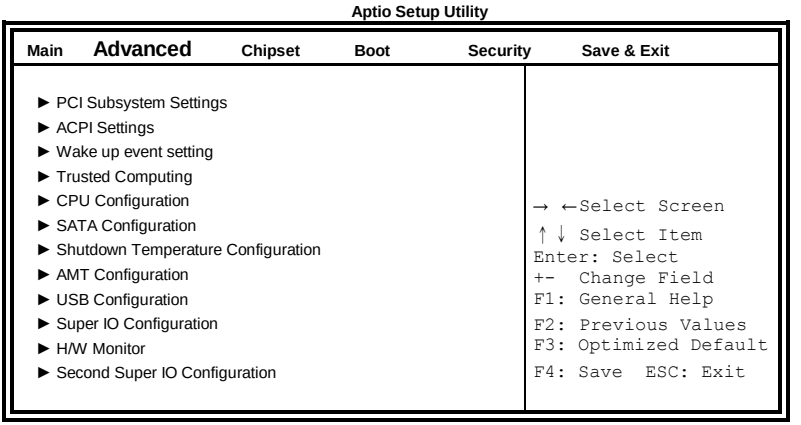
Set the Date. Use Tab to switch between Data elements.

System Time

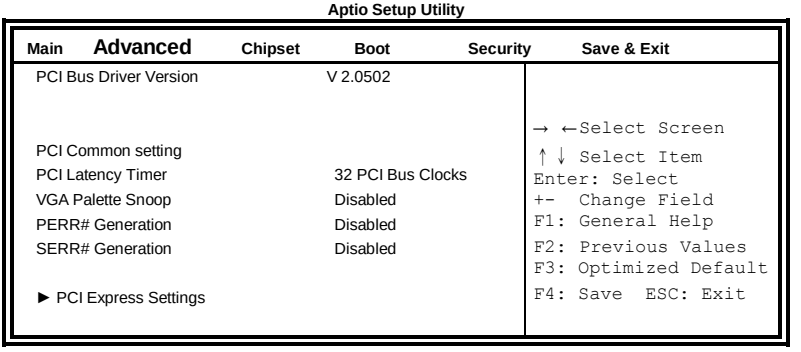
Set the Time. Use Tab to switch between Data elements.

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 Express Settings

Change PCI Express devices settings.

PCI Express Settings

Aptio Setup Utility

Main	Advanced	Chipset	Boot	Security	Save & Exit
PCI Express Device Register Settings					
Relaxed Ordering			Disabled		
Extended Tag			Disabled		
No Snoop			Enabled		
Maximum Payload			Auto		
Maximum Read Request			Auto		
PCI Express Link Register Settings					
ASPM Support			Disabled		
WARNING: Enabling ASPM may cause some PCI-E devices to fail					
Extended Synch			Disabled		
Link Training Retry			5		
Link Training Timeout (uS)			100		
Unpopulated Links			Keep Link ON		
Restore PCIE Registers			Disabled		

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

Relaxed Ordering
Enables or disables PCI Express Device Relaxed Ordering.

Extended Tag
If ENABLED allows device to use 8-bit Tag field as a requester.

No Snoop
Enables or disables PCI Express Device No Snoop option.

Maximum Payload
Set Maximum Payload of PCI Express Device or allow System BIOS to select the value.

Maximum Read Request
Set Maximum Read Request Size of PCI Express Device or allow System BIOS to select the value.

ASPM Support
Set the ASPM Level: Force L0s – Force all links to L0s State: AUTO – BIOS auto configure : DISABLE – Disables ASPM.

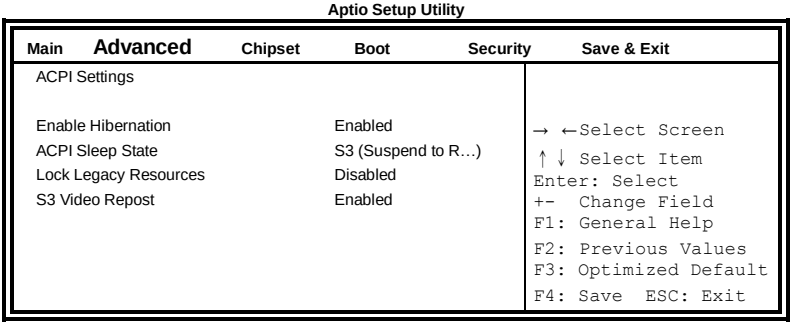
Extended Synch
If ENABLED allows generation of Extended Synchronization patterns.

Link Training Retry
Defines number of Retry Attempts software will take to retrain the link if previous training attempt was unsuccessful.

Link Training Timeout (uS)
Defines number of Microseconds software will wait before polling 'Link Training' bit in Link Status register. Value range from 10 to 1000 uS.

Unpopulated Links
In order to save power, software will disable unpopulated PCI Express links, if this option set to 'Disable Link'.

ACPI Settings



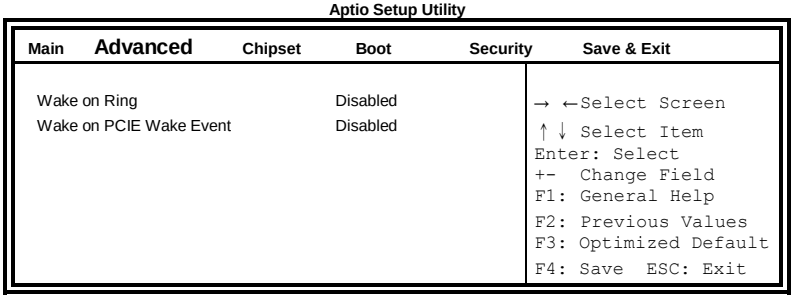
Enable Hibernation
Enables or Disables System ability to Hibernate (OS/S4 Sleep State). This option may be not effective with some OS.

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

Lock Legacy Resources
Enabled or Disabled Lock of Legacy Resources.

S3 Video Repost
Enable or disable S3 Video Repost.

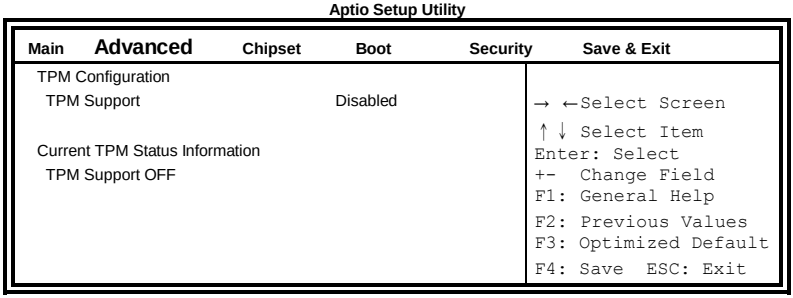
Wake up event settings



Wake on PCIE PME Wake Event

The options are Disabled and Enabled.

Trusted Computing



TPM Support

This configuration is supported only with MB970VF. Enables or Disables TPM support. O.S. will not show TPM. Reset of platform is required.

Security Device Support

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.

CPU Configuration

This section shows the CPU configuration parameters.

Aptio Setup Utility

Main	Advanced	Chipset	Boot	Security	Save & Exit

CPU Configuration		
Intel® Core ™ i7-4770EQ CPU @ 2.40GHz		
CPU Signature	306C3	
Processor Family	6	
Microcode Patch	16	
FSB Speed	100 MHz	
Max CPU Speed	2400 MHz	
Min CPU Speed	800 MHz	
CPU Speed	2500 MHz	
Processor Cores	4	
Intel HT Technology	Supported	
Intel VT-x Technology	Supported	
Intel SMX Technology	Supported	
64-bit	Supported	
EIST Technology	Supported	
CPU C3 State	Supported	→ ←Select Screen
CPU C6 State	Supported	↑ ↓ Select Item
CPU C7 State	Supported	Enter: Select
		+- Change Field
Active Processor Cores	All	F1: General Help
Intel Virtualization Technology	Disabled	F2: Previous Values
EIST	Enabled	F3: Optimized Default
Turbo Mode	Enabled	F4: Save ESC: Exit

Active Processor Cores

Number of cores to enable in each processor package.

Intel Virtualization Technology

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

EIST

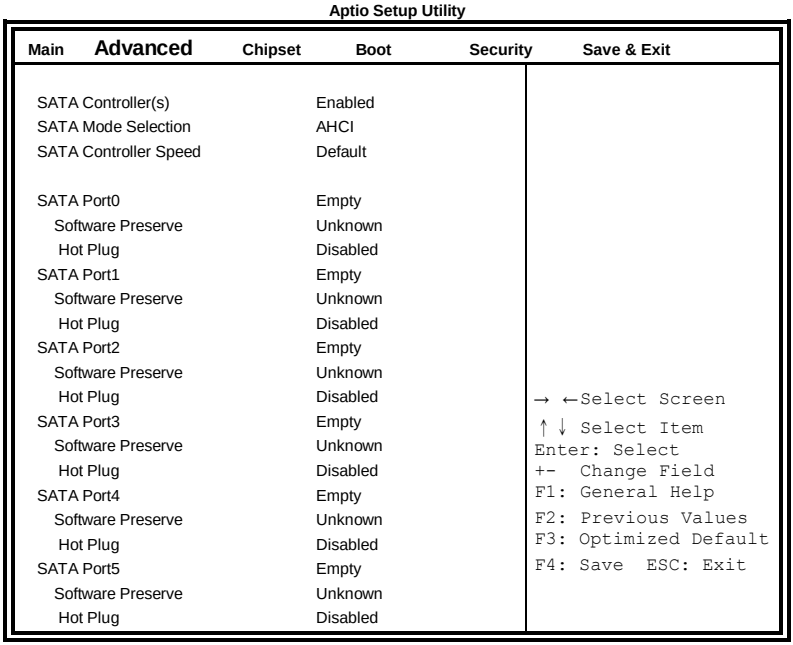
Enabled / Disabled Intel Speedstep.

Turbo Mode

Enabled / Disabled Turbo Mode.

SATA Configuration

SATA Devices Configuration.



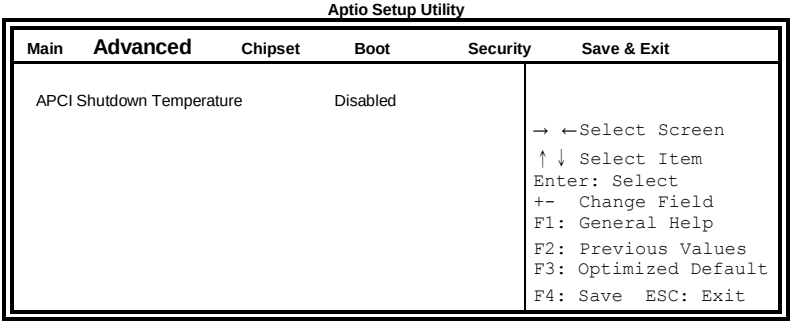
SATA Controller(s)

Enable / Disable Serial ATA Controller.

SATA Mode Selection

(1) IDE Mode.
(2) AHCI Mode.
(3) RAID Mode.

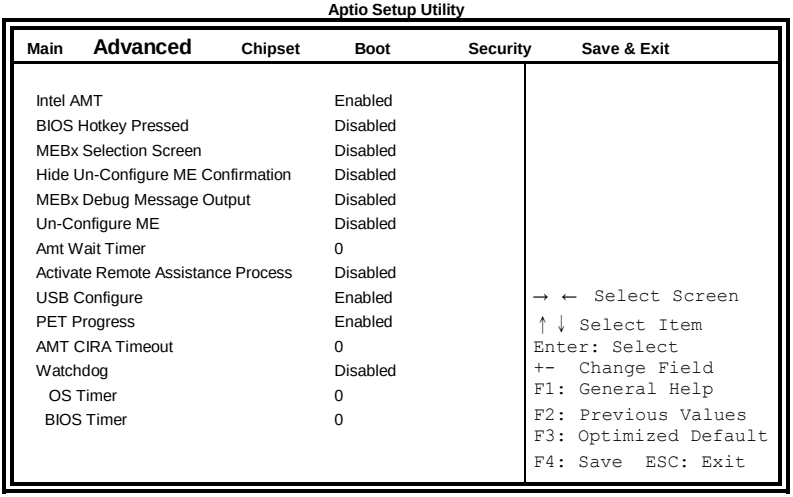
Shutdown Temperature Configuration



ACPI Shutdown Temperature

The default setting is Disabled

AMT Configuration



AMT Configuration

This configuration is supported only with MB980VF (with iAMT function). Options are Enabled and Disabled.
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.

Unconfigure ME

This configuration is supported only with MB980VF (with iAMT function). Perform AMT/ME unconfigure without password operation.

Amt Wait Timer

Set timer to wait before sending ASF_GET_BOOT_OPTIONS.

Activate Remote Assistance Process

Trigger CIRA boot.

PET Progress

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

Watchdog Timer

This configuration is supported only with MB980VF (with iAMT function). Enable/Disable Watchdog Timer.

USB Configuration

Aptio Setup Utility

Main	Advanced	Chipset	Boot	Security	Save & Exit

USB Configuration		
USB Module Version	8.10.28	
USB Devices:		
1Driver,1 Keyboard ,2 Hubs		
Legacy USB Support	Enabled	
USB3.0 Support	Enabled	
XHCI Hand-off	Enabled	
EHCI Hand-off	Enabled	
USB Mass Storage Driver Support	Enabled	
USB hardware delays and time-outs:		→ ←Select Screen
USB Transfer time-out	20 sec	↑ ↓ Select Item
Device reset tine-out	20 sec	Enter: Select
Device power-up delay	Auto	+- Change Field
		F1: General Help
Mass Storage Devices:		F2: Previous Values
USB FLASH DRIVE PMAP	Auto	F3: Optimized Default
		F4: Save ESC: Exit

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

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

USB Mass Storage Driver Support

Enable/Disable USB Mass Storage Driver Support.

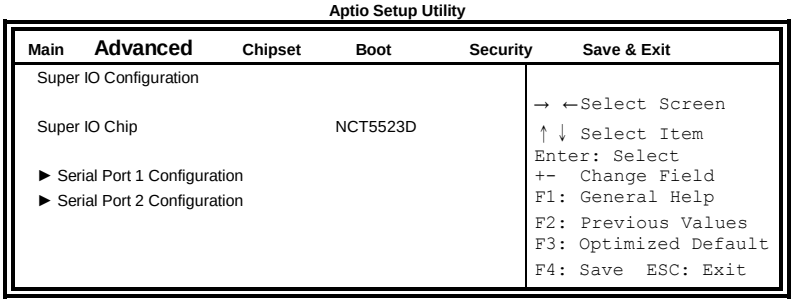
USB Transfer time-out

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

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

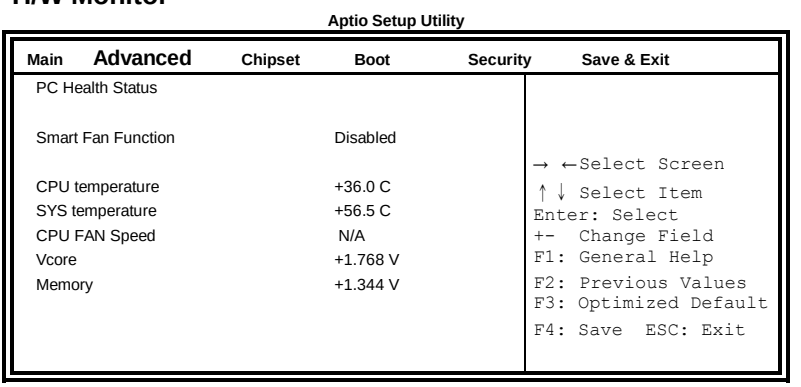
Device power-up delay
Maximum time the device will take before it properly reports itself to the Host Controller. 'Auto' uses default value: for a Root port it is 100ms, for a Hub port the delay is taken from Hub descriptor.

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.

H/W Monitor



Temperatures/Voltages

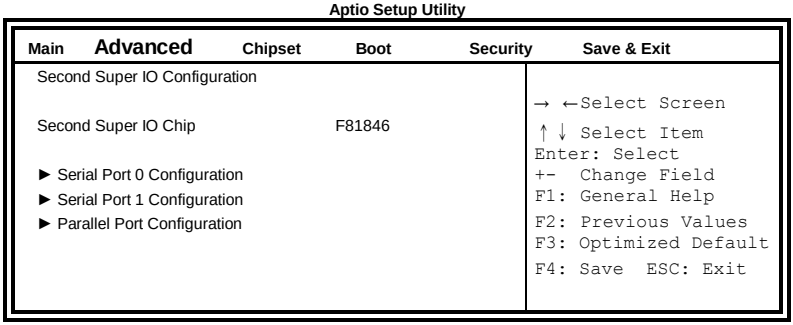
These fields are the parameters of the hardware monitoring function feature of the motherboard. The values are read-only values as monitored by the system and show the PC health status.

Smart Fan Function

This field enables or disables the smart fan feature. At a certain temperature, the fan starts turning. Once the temperature drops to a certain level, it stops turning again.

Second Super IO Configuration

System Second Super IO Chip Parameters.



Serial Port 0 Configuration

Set Parameters of Serial Port 0(COMA).

Serial Port 1 Configuration

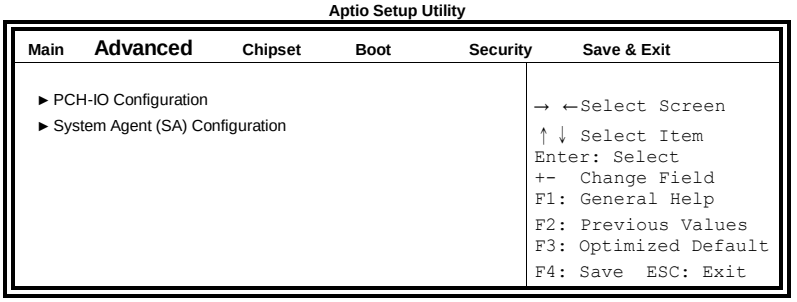
Set Parameters of Serial Port 1(COMB)

Parallel Port Configuration

Set Parameters of Parallel Port (LPT/LPTE)

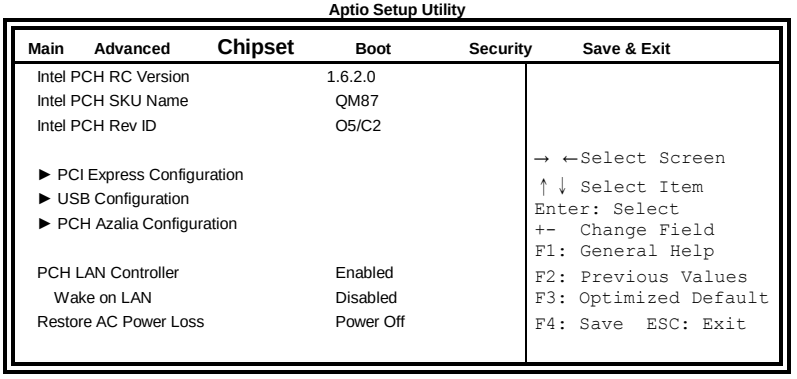
Chipset Settings

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.

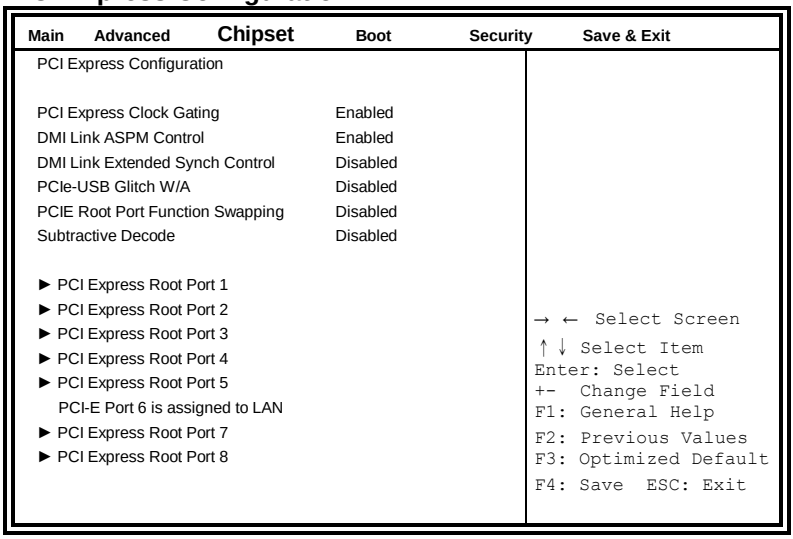
Wake on LAN

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

Restore AC Power Loss

Select AC power state when power is re-applied after a power failure.

PCI Express Configuration



PCI Express Clock Gating

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

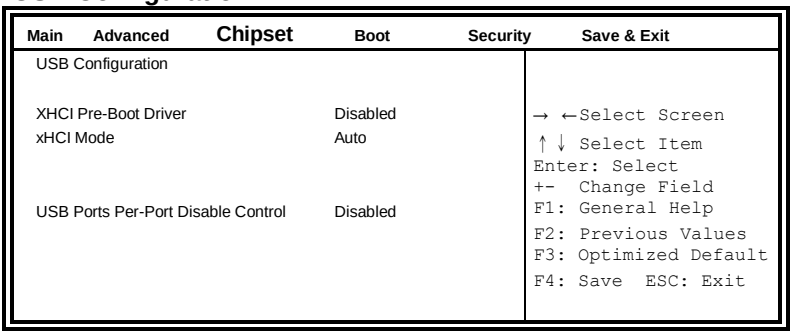
DMI Link ASPM Control

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

PCIe-USB Glitch W/A

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

USB Configuration



USB Precondition

Precondition work on USB host controller and root ports for faster enumeration.

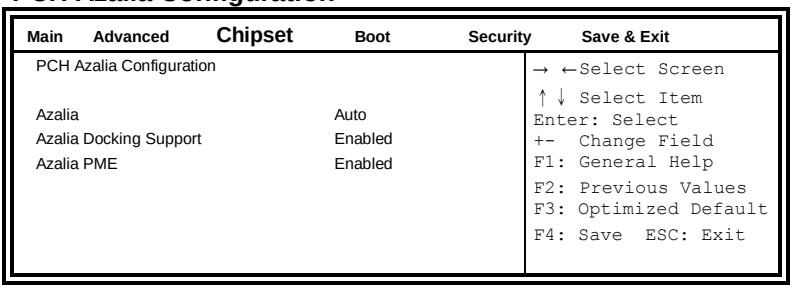
xHCI Mode

Mode of operation of xHCI controller.

USB Ports Per-Port Disable Control

Control each of the USB ports (0~13) disabling.

PCH Azalia Configuration



Azalia

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

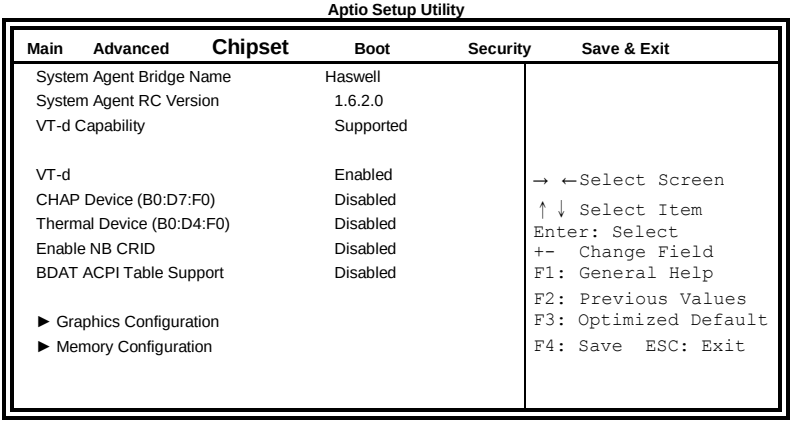
Azalia Docking Support

Enable or Disable Azalia Docking Support of Audio Controller.

Azalia PME

Enable or Disable power Management capability of Audio Controller.

System Agent (SA) Configuration



VT-d

Check to enable VT-d function on MCH.

Enable NB CRID

Enable or disable NB CRID WorkAround.

BDAT ACPI Tabled Support

Enables Support for the BDAT ACPI table.

Graphics Configuration

Aptio Setup Utility

Main	Advanced	Chipset	Boot	Security	Save & Exit

Graphics Configuration	
IGFX VBIOS Version	2179
IGfx Frequency	800 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
Primary IGFX Boot Display	VBIOS Default
LVDS / EDP Control	Disabled
Gfx Low power mode	Enabled
Panel Power Enabled	Disabled

→ ←Select Screen
↑ ↓ Select Item
Enter: Select
+- Change Field
F1: General Help
F2: Previous Values
F3: Optimized Default
F4: Save 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 PEGO/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.

DVMT Total Gfx Mem

Select DVMT 5.0 total graphics memory size used by the internal graphics device.

Primary IGFX Boot Display (LCD Control)

Select the Video Device that will be activated during POST. This has no effect if external graphics present. Secondary booty display selection will appear based on your selection. VGA modes will be supported only on primary display.

Gfx Low Power Mode

This option is applicable for SFF only.

Panel Power Enabled

Enabled / Disabled forcing of Panel Power in the BIOS.

Memory Configuration

Aptio Setup Utility

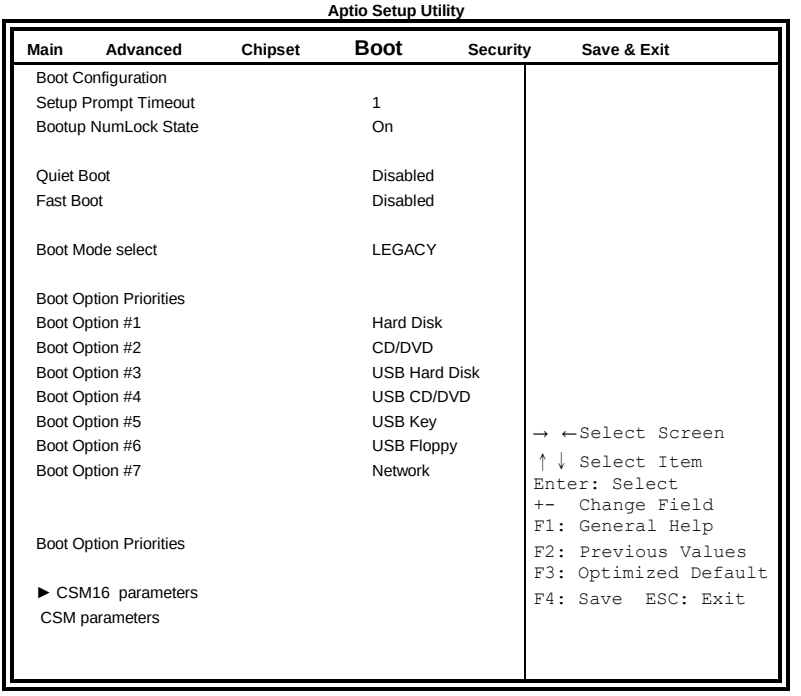
Main	Advanced	Chipset	Boot	Security	Save & Exit

Memory Information	
Memory Frequency	1600 MHz
Total Memory	2048 MB (DDR3)
Memory Voltage	1.35V
DIMM#0	Not Present
DIMM#2	2048 MB (DDR3)
CAS Latency (tCL)	11
Minimum delay time	
CAS to RAS (tRCDmin)	11
Row Precharge (tRPmin)	11
Active to Precharge (tRASmin)	28

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

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/Disables Quiet Boot option.

Fast Boot

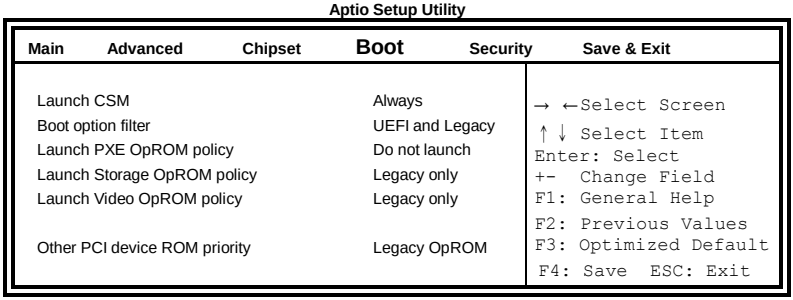
Enables/Disables boot with initialization of a minimal set of devices required to launch active boot option. Has no effect for BBS boot options.

Boot Option Priorities

Sets the system boot order.

CSM parameters

This section allows you to configure the boot settings.



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

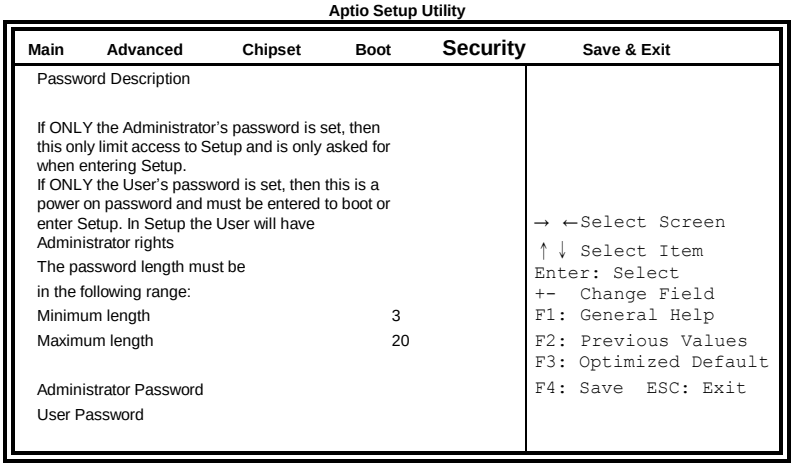
Controls the execution of UEFI and Legacy Video OpROM.

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.



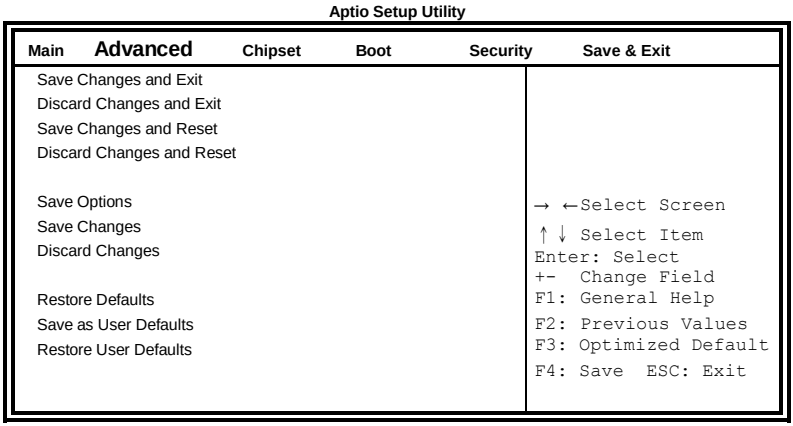
Administrator Password

Set Setup Administrator Password.

User Password

Set User 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:

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.

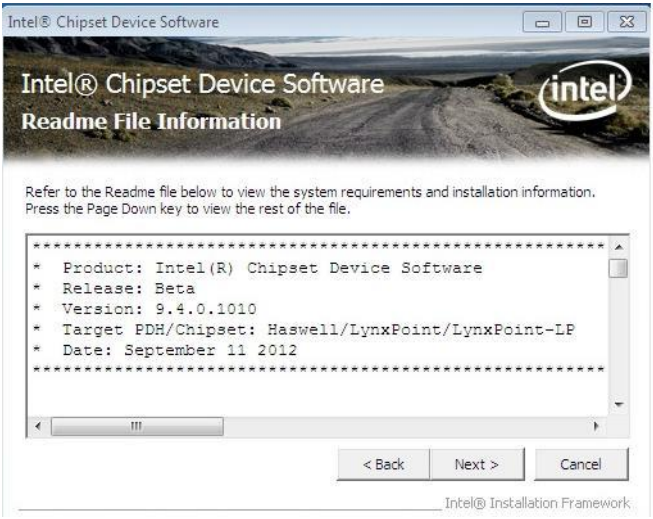
1. Insert the DVD that comes with the board. Click ***Intel*** and then ***Intel(R) 8 Series Chipset Drivers***.

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.

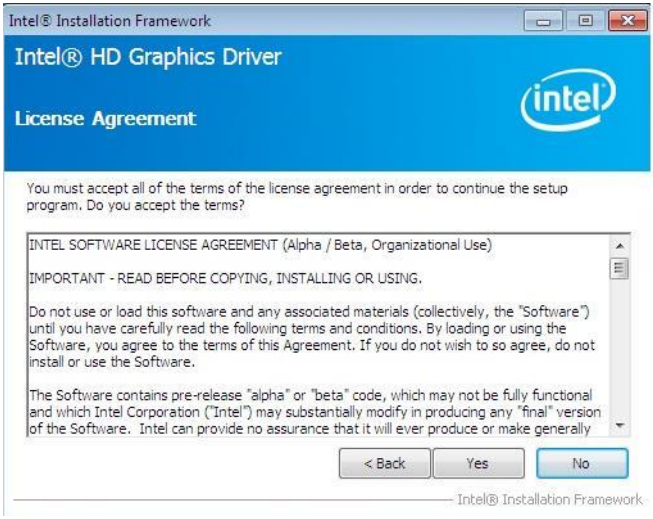
VGA Drivers Installation

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

2. Click ***Intel(R) Core(TM) i3/i5/i7 Graphics Driver.***

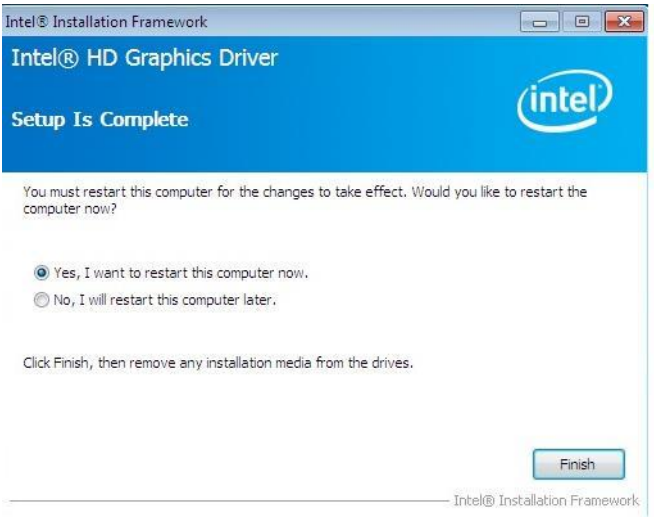
3. When the Welcome screen appears, click ***Next*** to continue.

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



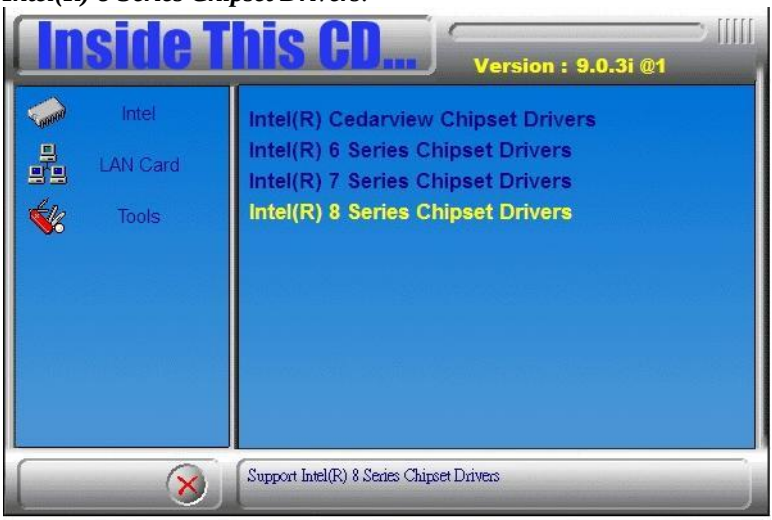
5. On the screen shown below, click ***Install*** to continue.

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

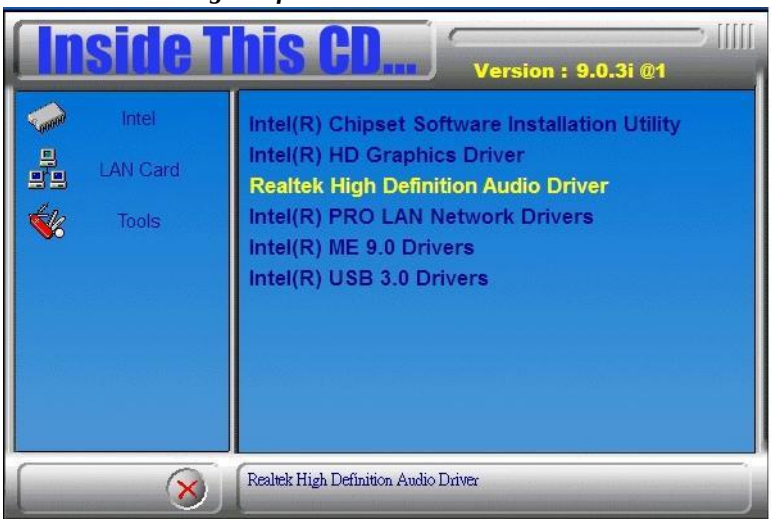


Realtek HD Audio Driver Installation

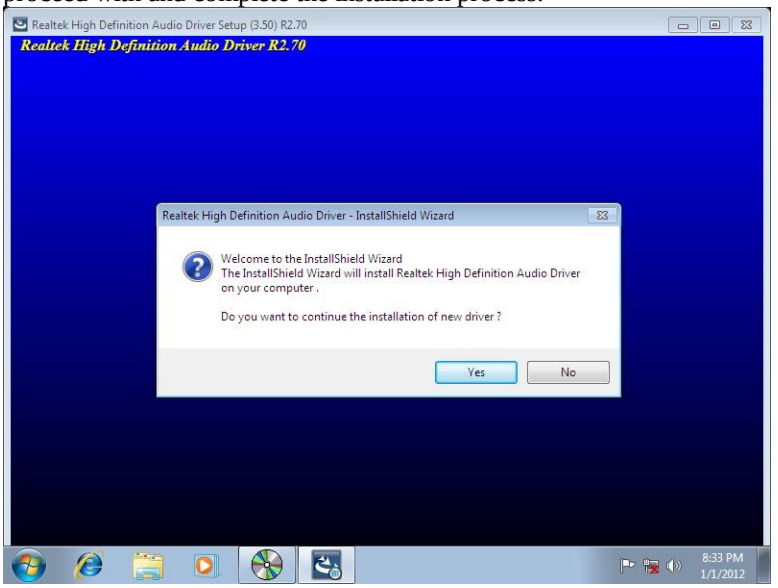
1. Insert the DVD that comes with the board. Click ***Intel*** and then ***Intel(R) 8 Series Chipset Drivers***.



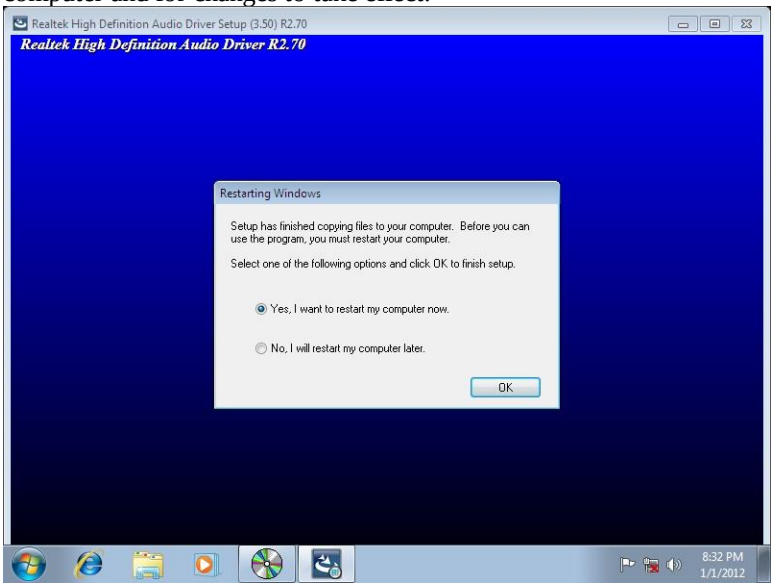
2. Click ***Realtek High Definition Audio Driver.***



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

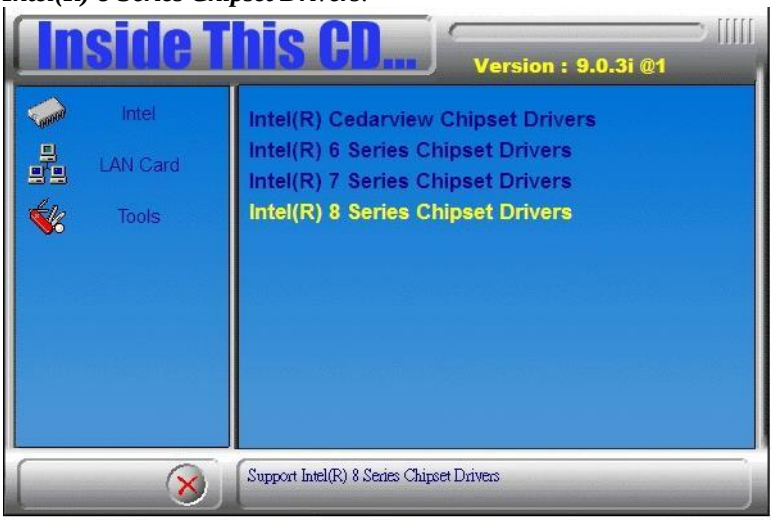


4. The InstallShield Wizard Complete. Click ***Finish*** to restart the computer and for changes to take effect.



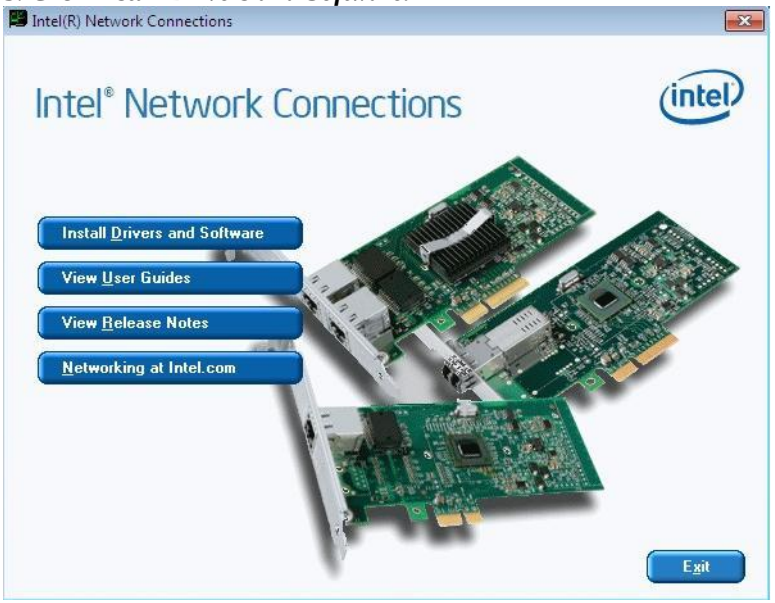
LAN Drivers Installation

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



2. Click ***Intel(R) PRO LAN Network Driver.***

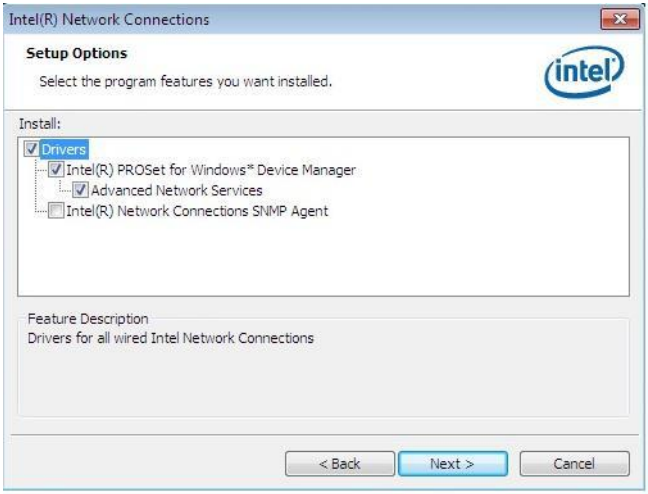
3. Click ***Install Drivers and Software.***



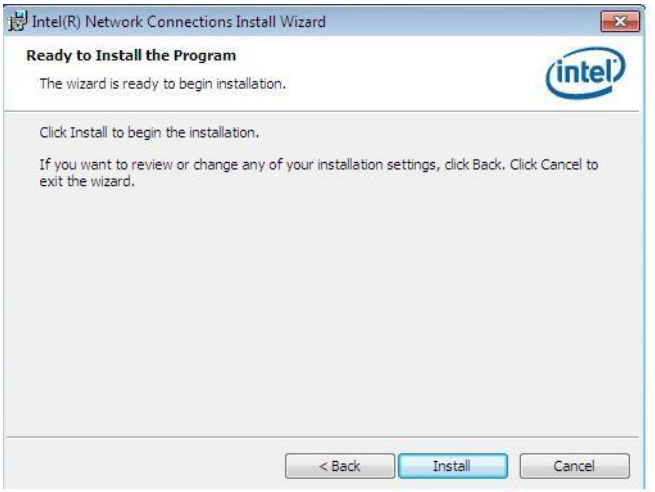
4. When the Welcome screen appears, click ***Next***.

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

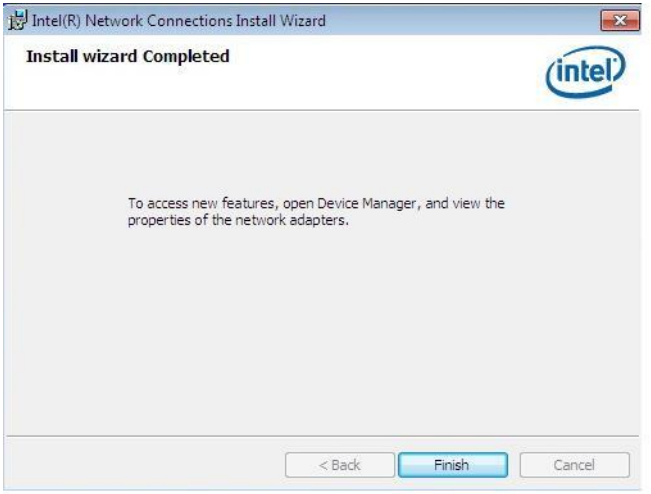
6. Click the checkbox for **Drivers** in the Setup Options screen to select it and click **Next** to continue.



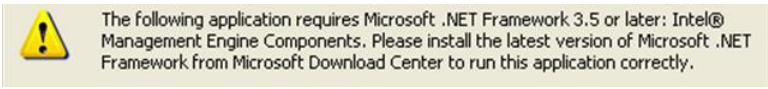
7. The wizard is ready to begin installation. Click ***Install*** to begin the installation.



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



Intel® Management Engine Interface

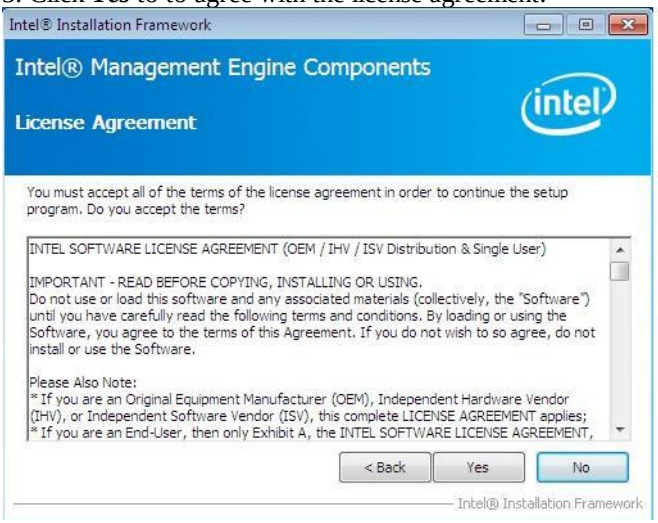


Follow the steps below to install the Intel Management Engine.

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

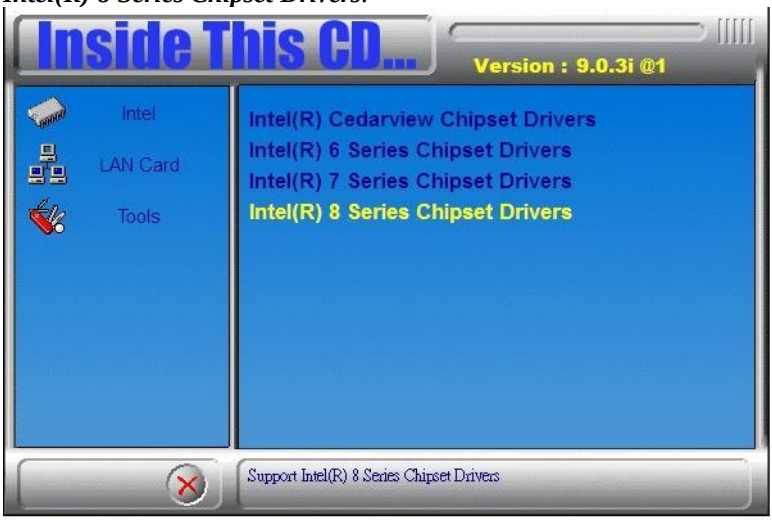
3. Click ***Yes*** to to agree with the license agreement.



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

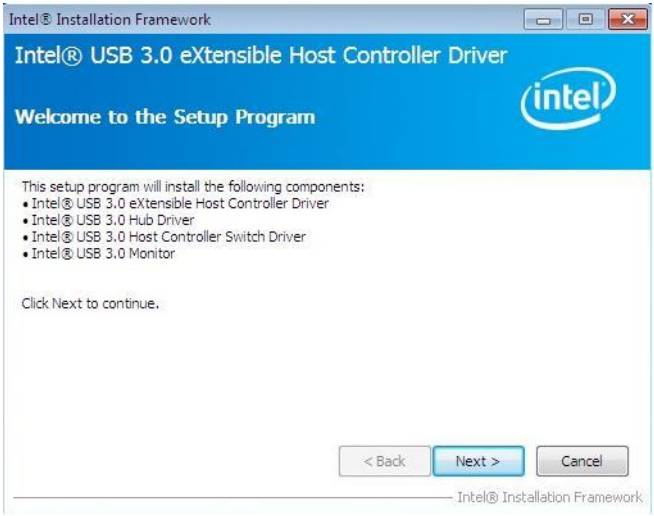
Intel® USB 3.0 Drivers

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



2. Click ***Intel(R) USB 3.0 Drivers.***

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



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

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

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

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
000h - 01Fh	DMA Controller #1
020h - 03Fh	Interrupt Controller #1
040h - 05Fh	Timer
060h - 06Fh	Keyboard Controller
070h - 07Fh	Real Time Clock, NMI
080h - 09Fh	DMA Page Register
0A0h - 0BFh	Interrupt Controller #2
0C0h - 0DFh	DMA Controller #2
0F0h	Clear Math Coprocessor Busy Signal
0F1h	Reset Math Coprocessor
1F0h - 1F7h	IDE Interface
278h - 27Fh	Parallel Port #2(LPT2)
2E8h – 2EFh	Serial Port #4(COM4)
2F8h - 2FFh	Serial Port #2(COM2)
2B0h- 2DFh	Graphics adapter Controller
360h - 36Fh	Network Ports
3B0h - 3BFh	Monochrome & Printer adapter
3C0h - 3CFh	EGA adapter
3D0h - 3DFh	CGA adapter
3E8h – 3EFh	Serial Port #3(COM3)
3F8h - 3FFh	Serial Port #1(COM1)

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 Output
IRQ1	Keyboard
IRQ2	Interrupt Cascade
IRQ3	Serial Port #2
IRQ4	Serial Port #1
IRQ5	Reserved
IRQ6	Reserved
IRQ7	Reserved
IRQ8	Real Time Clock
IRQ9	Reserved
IRQ10	Serial Port #3
IRQ11	Serial Port #4
IRQ12	PS/2 Mouse
IRQ13	80287
IRQ14	Primary IDE
IRQ15	Secondary IDE

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;
}
```