MI946F

Intel ® Core™ 2 Duo/GM45
Mini-ITX Motherboard

USER'S MANUAL

Version 1.0

Acknowledgments

AMI is a registered trademark of AMI Software International, Inc.
PS/2 is a trademark of International Business Machines Corporation.
Intel and Intel® Core 2 Duo and Intel® Celeron processors are registered trademarks of Intel Corporation.
Microsoft Windows is a registered trademark of Microsoft Corporation.
Winbond is a registered trademark of Winbond Electronics Corporation.
All other product names or trademarks are properties of their respective owners.

Table of Contents

IMPORTANT NOTE: When the system boots without the CRT being connected, there will be no image on screen when you insert the CRT/VGA cable. To show the image on screen, the hotkey must be pressed (CTRL-ALT-F1).

Introduction

Product Description

The MI946F Mini ITX board incorporates the Mobile Intel® GM45 Express Chipset for Embedded Computing, consisting of the Intel® GM45 Graphic Memory Controller Hub (GMCH) and Intel® I/O Controller Hub (ICH9-M), an optimized integrated graphics solution with a 1066MHz and 800MHz front-side bus. Dimensions of the board are 170mm x 170mm.

The integrated powerful 3D graphics engine, based on Intel® Graphics Media Accelerator X3500 (Intel® GMA4500MHD) architecture, operates at core speeds of up to 533 MHz. It features a low-power design, is validated with the Intel® Core 2 Duo processors on 45nm process. With two SoDIMM sockets supporting dual channel DDR3 1066MHz on board, the board supports up to 4GB of DDR2 system memory.

Intel® Graphics supports a unique intelligent memory management scheme called Dynamic Video Memory Technology (DVMT). DVMT handles diverse applications by providing the maximum (384MB) availability of system memory for general computer usage, while supplying additional graphics memory when a 3D-intensive application requests it. The Intel GMA4500MHD graphics architecture also takes advantage of the high-performance Intel processor. Intel GMA4500MHD graphics supports Dual Independent Display technology.

The main features of the board are:

- Supports Intel® Core™ 2 Duo (Penryn 1066MHz)
- Supports up to 2.53GHz, 1066MHz/800MHz FSB
- Two DDR3 SoDIMM, Max. 4GB memory
- Onboard Gigabit PHY and Intel PCI-Express Gigabit LAN
- Intel® GM45 Express VGA for CRT / LVDS
- 4x SATA, 8x USB 2.0, 4x COM, Watchdog timer
- 1x Mini PCI-E (Mini Card), 1x PCI, 1xPCI-E(x1) slots

Checklist

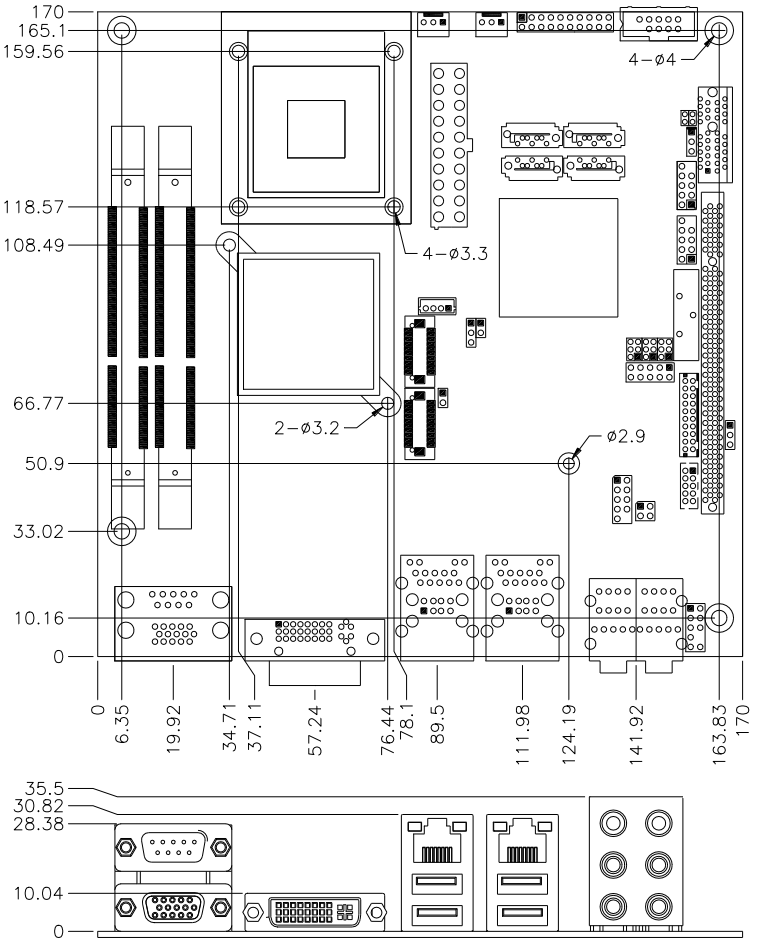
Your MI946F package should include the items listed below.

- The MI946F Mini-ITX motherboard
- This User's Manual
- 1 CD containing chipset drivers and flash memory utility
- Cable kit (Serial port, Serial ATA)

MI946F Specifications

CPU Supported	Intel® Core™ 2 Duo (Penryn), mobile processors
CPU Voltage	0.700V ~ 1.5V (IMVP-6)
System Speed	Up to 2.53GHz or above
CPU FSB	667MHz/800MHz/1066MHz FSB
Cache	1MB/2MB/4MB
Green /APM	APM1.2
CPU Socket	mPGA Socket 478
Chipset	Intel GM45 Chipset GMCH: GM45 1329-pin Micro-FCBGA ICH9M: 82801IBM 678-pin mBGA
BIOS	AMI BIOS, supports ACPI function
Memory	DDR3 800/1066 SoDIMM x2 (w/o ECC function), Max. 4GB
VGA	GM45 built-in, supports CRT
SDVO (Dual CH)	Chrontel 7307C, DVI Chrontel 7022A, CRT
LVDS LCD Panel	GM45 built-in, supports 24-bit, single or dual channel LVDS
LAN	1. ICH9M 10/100/gigabit MAC + PHY • Intel 82567L 10/100/1000 2. Intell 82583V PCI-e Gigabit LAN controller x1
USB	ICH9M built-in USB 2.0 host controller, support 8 ports
Serial ATA Ports	ICH9M built-in SATA controller, supports 4 ports
TPM1.2	ICH9M built-in iTPM version1.2 controller by firmware implement
Parallel IDE	JMicron JM368 (PCI-e to PATA) x1 for 1 PATA channel for CF
Audio	ICH9M built-in audio controller + AC97 Codec ALC892 w/ 7.1 channels
LPC I/O	Nuvoton NCT6627UD: COM1, COM2 (RS232/RS422/RS485), COM3 and COM4
Digital IO	4 in & 4 out
Keyboard/Mouse	Supports PS/2 keyboard/mouse onboard header type
Expansion Slots	PCI slot x1, PIC-E (x1) slot x1 and Mini PCIE socket x1
AMT	Intel ® Advanced Management Technology 4.0
Edge Connector	DVI-I connector x 1 for DVI/CRT Gigabit LAN RJ-45 + dual USB stack connector Gigabit LAN RJ-45 + dual USB stack connector DB9 x1 for COM 1; DB15 x1 for VGA RCA Jack 3x2 for Audio (Front-Out, Line-In, Mic, Center/LFE, Surround & Surround Back)
Onboard Header/ Connector	CF connector x1 @ solder side 10-pin headerx1 for Digital I/O; 10-pin header x1 for COM2 10-pin header x 2 for USB 5/6/7/8 DF13 connector x2 for LVDS; 10-pin header x1 for audio Line-Out & Mic 4-pin header x1 for CD in, SPDIF-out connector x1 SATA connector x4 for SATA ports
Watchdog Timer	Yes (256 segments, 0, 1, 2…255 sec/min)
System Voltage	+5V, +3.3V, +12V, -12V, 5VSB (2A)
Others	Modem Wakeup, LAN Wakeup
Board Size	170mm x 170mm (Mini ITX)

Board Dimensions

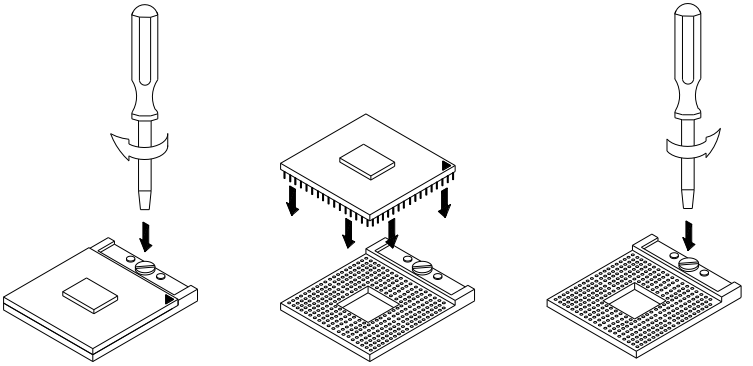


Installations

This section provides information on how to use the jumpers and connectors on the MI946F in order to set up a workable system. The topics covered are:

Installing the CPU

The MI946F board supports a Socket 478MN (Merom) processor socket for Intel Core™ 2 Duo (Penryn), mobile processors.
The processor socket comes with a screw to secure the processor. As shown in the left picture below, loosen the screw first before inserting the processor. Place the processor into the socket by making sure the notch on the corner of the CPU corresponds with the notch on the inside of the socket. Once the processor has slide into the socket, fasten the screw. Refer to the figures below.



***NOTE**: Ensure that the CPU heat sink and the CPU top surface are in total contact to avoid CPU overheating problem that would cause your system to hang or be unstable.*

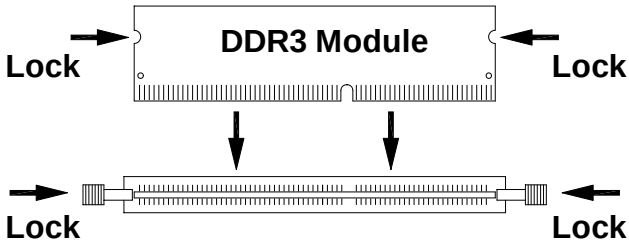
Installing the Memory

The MI946F board supports two DDR3 memory socket for a maximum total memory of 4GB.

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.

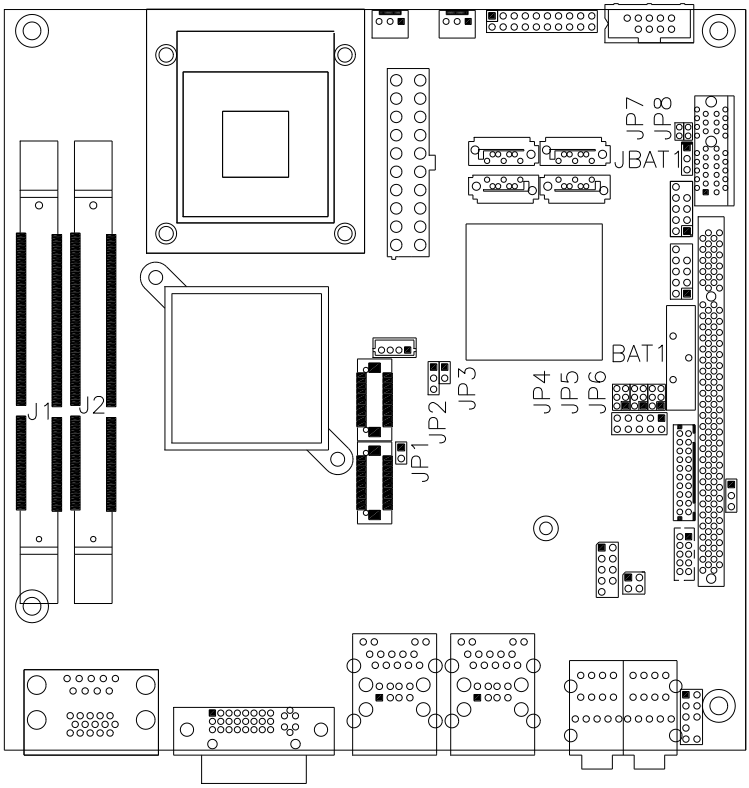


Setting the Jumpers

Jumpers are used on MI946F 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 MI946F and their respective functions.

IMPORTANT NOTE: When the system boots without the CRT being connected, there will be no image on screen when you insert the CRT/VGA cable. To show the image on screen, the hotkey must be pressed.

Jumper Locations on MI946F

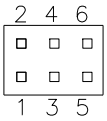


JP2: LCD Panel Power Selection

JP2	LCD Panel Power
	3.3V
	5V

JP4, JP5, JP6: RS232/422/485 (COM2) Selection

COM1 is fixed for RS-232 use only.
COM2 is selectable for RS232, RS-422 and RS-485.
The following table describes the jumper settings for COM2 selection.



COM2 Function	RS-232	RS-422	RS-485
Jumper Setting (pin closed)	JP4: 1-2 JP5: 3-5 & 4-6 JP6: 3-5 & 4-6	JP4: 3-4 JP5: 1-3 & 2-4 JP6: 1-3 & 2-4	JP4: 5-6 JP5: 1-3 & 2-4 JP6: 1-3 & 2-4

COM2 is jumper selectable for RS-232, RS-422 and RS-485.

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	RTS-	NC
7	RTS	RTS+	NC
8	CTS	CTS+	NC
9	RI	CTS-	NC
10	NC	NC	NC

JP8: CompactFlash Slave/Master Selection

JP8	CF Setting
	Master
	Slave

JP9: PCI/PCIE Riser Card Selection

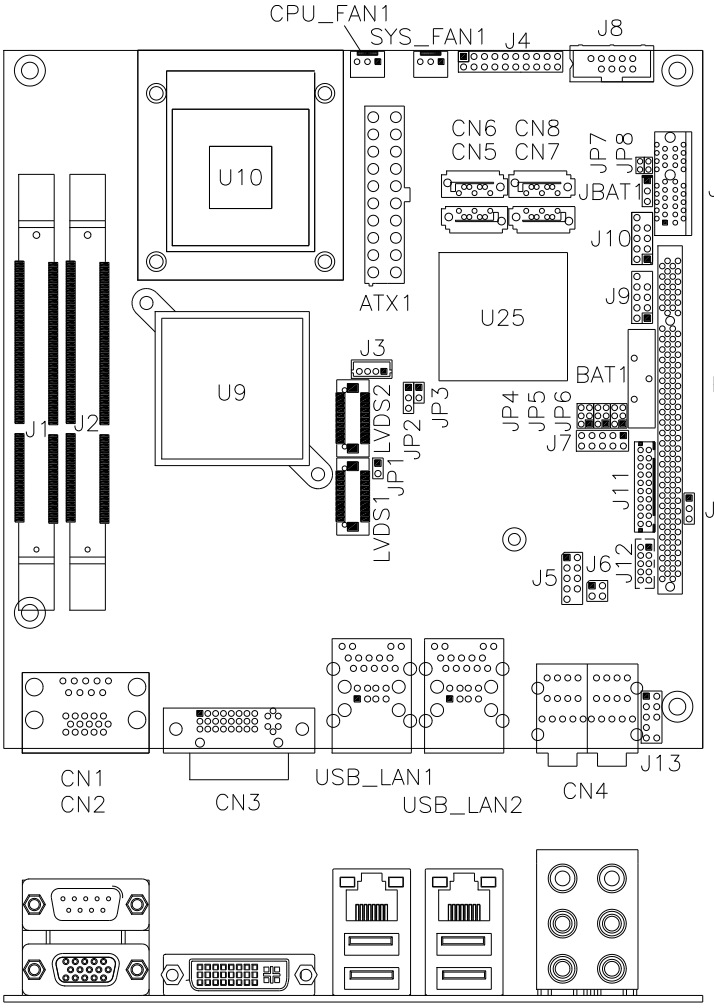
JP9	Riser Card
	IP390 Riser Card Install
	IP151, IP240 Riser Card Install

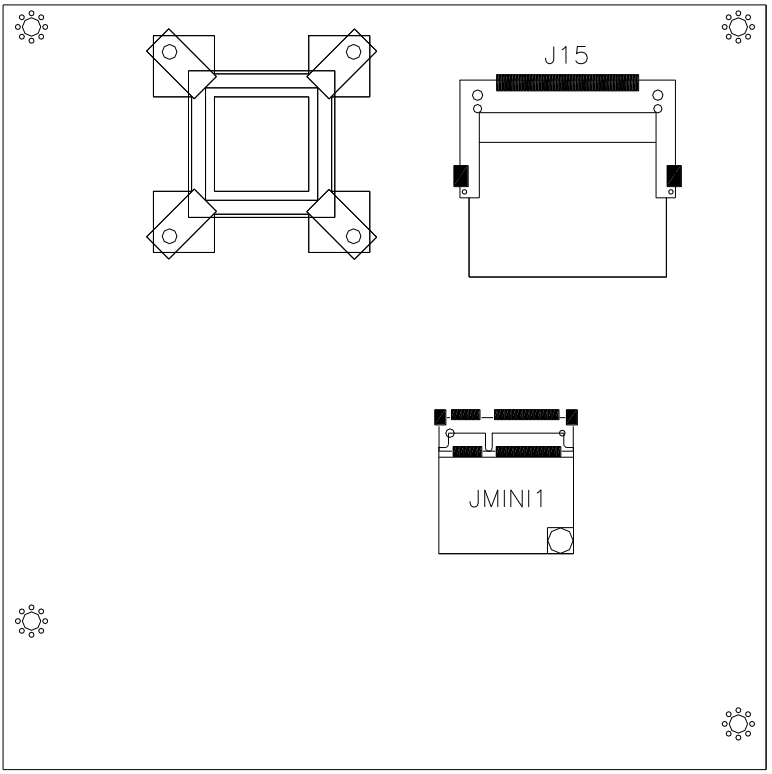
JBAT1: Clear CMOS Setting

JBAT1	Setting
	Normal
	Clear CMOS

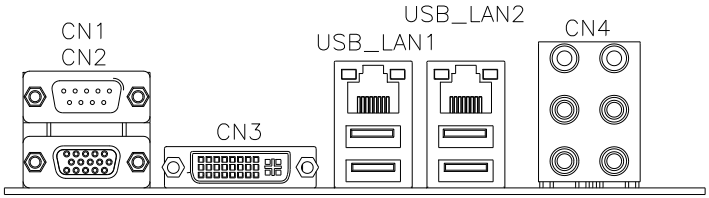
Connectors on MI946F

Connector Locations on MI946F

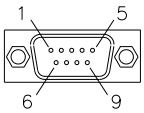




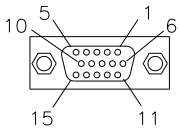
Solder Side of MI946F



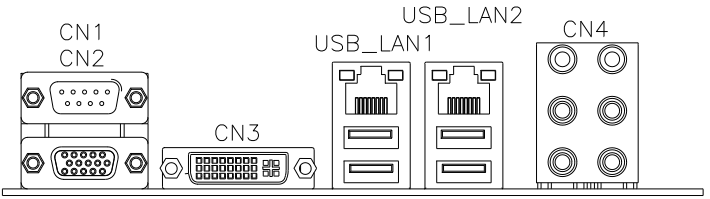
CN1, CN2: COM1 and VGA Connector



Signal Name	Pin #	Pin #	Signal Name
DCD	1	6	DSR
RXD	2	7	RTS
TXD	3	8	CTS
DTR	4	9	RI
GND	5	10	Not Used

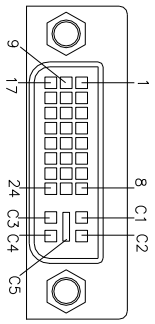


Signal Name	Pin #	Pin #	Signal Name
Red	1	2	Green
Blue	3	4	N.C.
GND	5	6	GND
GND	7	8	GND
N.C.	9	10	GND
N.C.	11	12	N.C.
HSYNC	13	14	VSYNC
NC	15		



CN3: DVI-I Connector

CRT interface is supported through the use of the DVI-I connector.

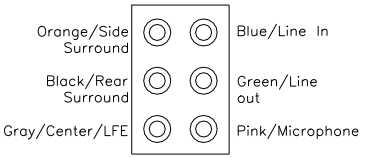


Signal Name	Pin #	Pin #	Signal Name
DATA 2-	1	16	HOT POWER
DATA 2+	2	17	DATA 0-
Shield 2/4	3	18	DATA 0+
DATA 4-	4	19	SHIELD 0/5
DATA 4+	5	20	DATA 5-
DDC CLOCK	6	21	DATA 5+
DDC DATA	7	22	SHIELD CLK
VSYNC	8	23	CLOCK -
DATA 1-	9	24	CLOCK +
DATA 1+	10	C1	Red.
SHIELD 1/3	11	C2	Green
DATA 3-	12	C3	Blue
DATA 3+	13	C4	HSYNC
DDC POWER	14	C5	Ground
A GROUND 1	15	C6	Ground

USB_LAN1: 10/100/1000 RJ-45 and USB1/2 Ports

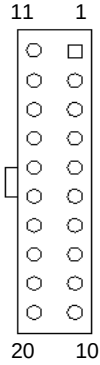
USB_LAN2: 10/100/1000 RJ-45 and USB3/4 Ports

CN4: Audio Connector



CN5, CN6, CN7, CN8: SATA Connectors

ATX1: ATX Power Supply Connector



Signal Name	Pin #	Pin #	Signal Name
3.3V	11	1	3.3V
-12V	12	2	3.3V
Ground	13	3	Ground
PS-ON	14	4	+5V
Ground	15	5	Ground
Ground	16	6	+5V
Ground	17	7	Ground
-5V	18	8	Power good
+5V	19	9	5VSB
+5V	20	10	+12V

SYS_FAN1: System Fan Power Connector

This is a 3-pin header for system fans. The fan must be a 12V (500mA).

Pin #	Signal Name
1	Ground
2	+12V
3	Rotation detection

CPU_FAN1: CPU Fan Power Connector

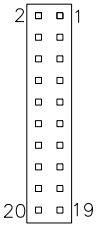
This is a 3-pin header for the CPU fan. The fan must be a 12V fan.

Pin #	Signal Name
1	Ground
2	+12V
3	Rotation detection

JMINI: Mini PCIE Connector

LVDS1, LVDS2: LVDS Connectors (1st channel, 2nd channel)

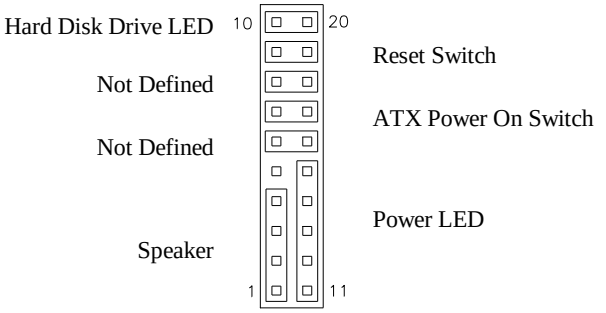
The LVDS connectors on board consist of the first channel (LVDS1) and second channel (LVDS2).



Signal Name	Pin #	Pin #	Signal Name
TX0-	2	1	TX0+
Ground	4	3	Ground
TX1-	6	5	TX1+
5V/3.3V	8	7	Ground
TX3-	10	9	TX3+
TX2-	12	11	TX2+
Ground	14	13	Ground
TXC-	16	15	TXC+
5V/3.3V	18	17	ENABKL
+12V	20	19	+12V

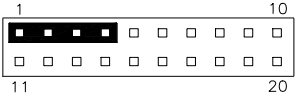
J4 (F_PANEL): System Function Connector

J4 provides connectors for system indicators that provide light indication of the computer activities and switches to change the computer status. J4 is a 20-pin header that provides interfaces for the following functions.



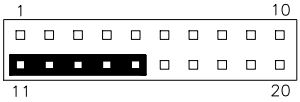
Speaker: Pins 1 - 4

This connector provides an interface to a speaker for audio tone generation. An 8-ohm speaker is recommended.



Pin #	Signal Name
1	Speaker out
2	No connect
3	Ground
4	+5V

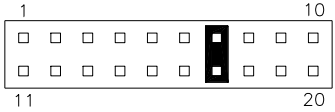
Power LED: Pins 11 - 15



Pin #	Signal Name
11	Power LED
12	No connect
13	Ground
14	No connect
15	Ground

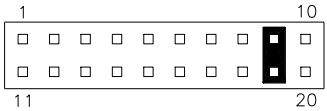
ATX Power ON Switch: Pins 7 and 17

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.

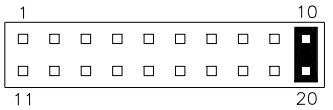


Reset Switch: Pins 9 and 19

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.

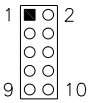


Hard Disk Drive LED Connector: Pins 10 and 20
This connector connects to the hard drive activity LED on control panel. This LED will flash when the HDD is being accessed.



Pin #	Signal Name
10	HDD Active
20	5V

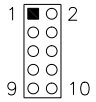
J7: Digital I/O



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

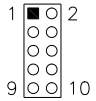
J8: SPI Flash Connector (factory use only)

J9: USB5/USB6 Connector



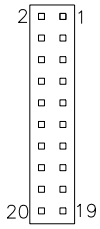
Signal Name	Pin	Pin	Signal Name
Vcc	1	2	Vcc
D0-	3	4	D1-
D0+	5	6	D1+
Ground	7	8	Ground
NC	9	10	Ground

J10: USB7/USB8 Connector



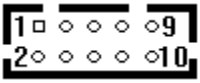
Signal Name	Pin	Pin	Signal Name
Vcc	1	2	Vcc
D0-	3	4	D1-
D0+	5	6	D1+
Ground	7	8	Ground
NC	9	10	Ground

J11: COM3, COM4 Serial Port



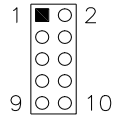
Signal Name	Pin #	Pin #	Signal Name
DSR	2	1	DCD
RTS	4	3	RXD
CTS	6	5	TXD
RI	8	7	DTR
NA	10	9	Ground
DSR	12	11	DCD
RTS	14	13	RXD
CTS	16	15	TXD
RI	18	17	DTR
NA	20	19	Ground

J12: COM2 Serial Port



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

J13: Front Audio Connector



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

J14: PCI-E(x1) Slot

PCI1: PCI Slot (supports 2 Master)

This page is intentionally left blank.

BIOS Setup

This chapter describes the different settings available in the Award 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 adds virus and 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:

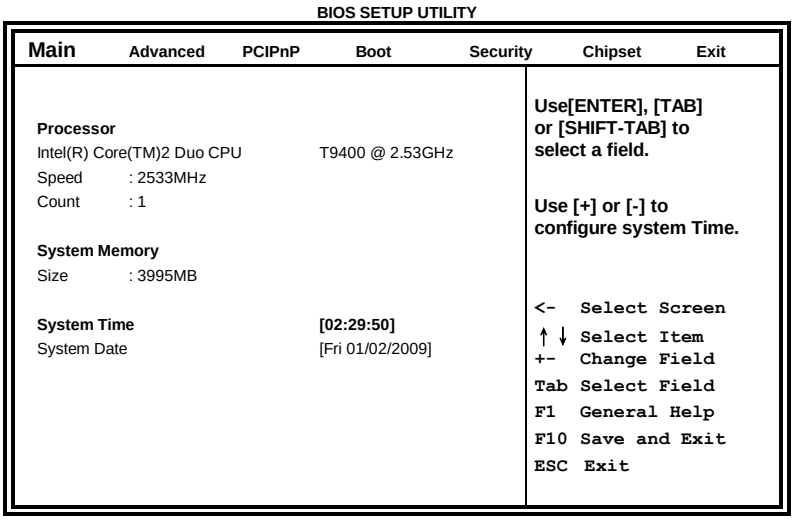
```
Press  <DEL>  to  Enter  Setup
```

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.

Main BIOS Setup

This setup allows you to record some basic hardware configurations in your computer system and set the system clock.

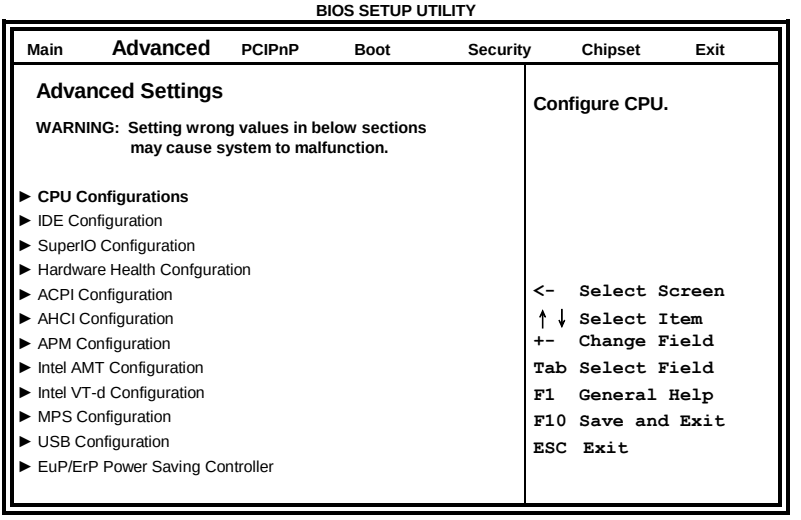


***Note**: If the system cannot boot after making and saving system changes with Setup, the Award BIOS supports an override to the CMOS settings that resets your system to its default.*

***Warning:** It is strongly recommended that you avoid making any changes to the chipset defaults. These defaults have been carefully chosen by both Award 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.*

Advanced Settings

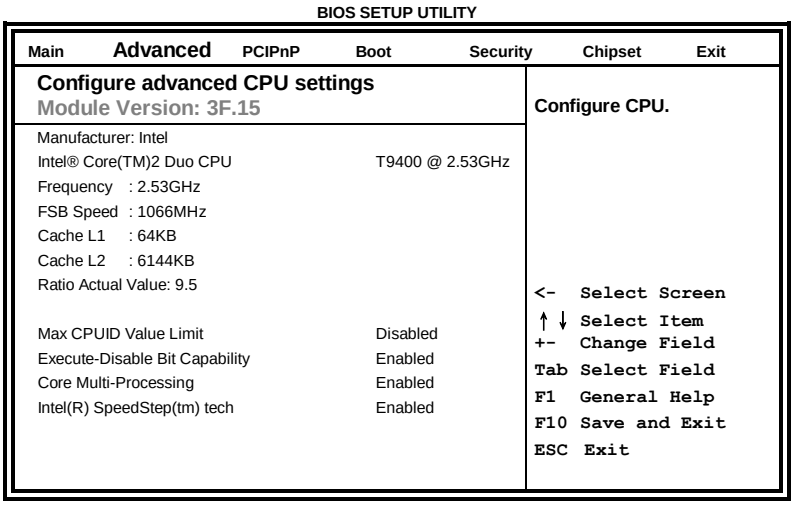
This section allows you to configure and improve your system and allows you to set up some system features according to your preference.



The Advanced BIOS Settings contains the following sections:

► CPU Configurations
► IDE Configuration
► SuperIO Configuration
► Hardware Health Confguration
► ACPI Configuration
► AHCI Configuration
► APM Configuration
► Intel AMT Configuration
► Intel VT-d Configuration
► MPS Configuration
► USB Configuration

The fields in each section are shown in the following pages, as seen in the computer screen. Please note that setting the wrong values may cause the system to malfunction. If unsure, please contact technical support of your supplier.



The CPU Configuration menu shows the following CPU details:
Manufacturer: the name of the CPU manufacturer
Brand String: the brand name of the CPU being used
Frequency: the CPU processing speed
FSB Speed: the FSB speed
Cache L1: the CPU L1 cache size
Cache L2: the CPU L2 cache

BIOS SETUP UTILITY

Main	Advanced	PCIPnP	Boot	Security	Chipset	Exit

IDE Configuration

Mirrored IDER Configuration	[Enabled]
SATA#1 Configuration	[Compatible]
Configure SATA#1 as	[IDE]
SATA#2 Configuration	[Enhanced]
► Primary IDE Master	: [Not Detected]
► Primary Slave Master	: [Not Detected]
► Secondary IDE Master	: [Not Detected]
► Secondary IDE Slave	: [Not Detected]
► Third IDE Master	: [Not Detected]
► Fourth IDE Master	: [Not Detected]
► Primary IDE Master	: [Not Detected]
► Fifth IDE Master	: [Not Detected]
► Fifth IDE Slave	: [Not Detected]
► Sixth IDE Master	: [Not Detected]
► Sixth IDE Slave	: [Not Detected]
► 7th IDE Master	: [Not Detected]
► 7th IDE Slave	: [Not Detected]
IDE Detect Time Out (Sec)	[35]
ATA(PI) 80Pin Cable Detection	[Host & Device]

Options:
Disabled
Compatible
Enhanced

<- Select Screen
↑ ↓ Select Item
+- Change Field
Tab Select Field
F1 General Help
F10 Save and Exit
ESC Exit

The IDE Configuration menu is used to change and/or set the configuration of the IDE devices installed in the system.
SATA#1 can be configured as IDE, Raid or AHCI Only With MI946RF.

BIOS SETUP UTILITY

Main	Advanced	PCIPnP	Boot	Security	Chipset	Exit

Configure Win627DHG Super IO Chipset

Serial Port1 Address	[3F8/IRQ4]
Serial Port2 Address	[2F8/IRQ3]
Serial Port3 Address	[Disabled]
Serial Port4 Address	[Disabled]
Restore on AC Power Loss	[Power Off]
Power On Function	[None]

<- Select Screen
↑ ↓ Select Item
+- Change Field
Tab Select Field
F1 General Help
F10 Save and Exit
ESC Exit

Onboard Serial Port/Parallel Port

These fields allow you to select the onboard serial ports and their addresses. The default values for these ports are:

Serial Port 1	3F8/IRQ4
Serial Port 2	2F8/IRQ3
Serial Port 3	Disabled
Serial Port 4	Disabled

Restore on AC Power Loss

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

Power On Function

This field is related to how the system is powered on . The options are *None, Mouse Left, Mouse Right,* and *Any Key.*

BIOS SETUP UTILITY

Main	**Advanced**	PCIPnP	Boot	Security	Chipset	Exit

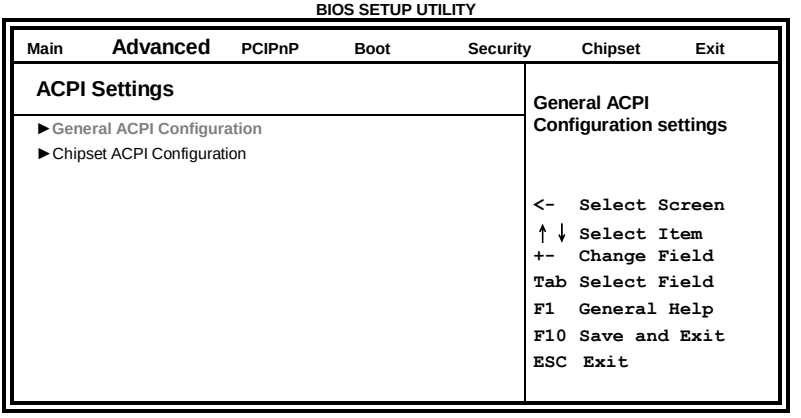
Hardware Health Configuration

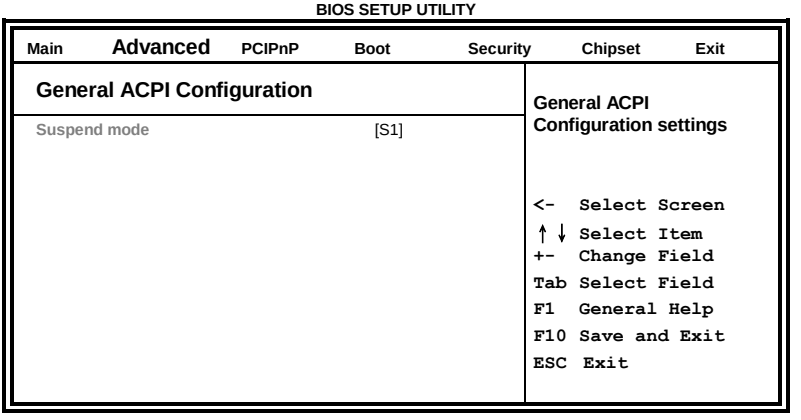
System Temperature	: 45°C/113°F
CPU Temperature	: 45°C/113°F
System FAN Speed	: 0 RPM
CPU FAN Speed	: 5400 RPM
Vcore	:1.160 V
+5VS	: 5.092 V
+3VS	: 3.392 V
12V	: 12.196 V
3.3V	: 3.392V
1.5V	: 1.521V
CPU Smart Fan	Disabled
ACPI Shut down Temperature	Disabled

Configure CPU.

<- Select Screen
↑ ↓ Select Item
+- Change Field
Tab Select Field
F1 General Help
F10 Save and Exit
ESC Exit

The Hardware Health Configuration menu is used to show the operating temperature, fan speeds and system voltages.





Suspend Mode

The options of this field are *S1*, *S3* and *Auto*.

BIOS SETUP UTILITY

Main **Advanced** PCIPnP Boot Security Chipset Exit

South Bridge ACPI Configuration

High Performance Event Timer [Disabled]

General ACPI Configuration settings

<- Select Screen
↑ ↓ Select Item
+- Change Field
Tab Select Field
F1 General Help
F10 Save and Exit
ESC Exit

BIOS SETUP UTILITY

Main **Advanced** PCIPnP Boot Security Chipset Exit

AHCI Settings

AHCI BIOS Support [Enabled]

► AHCI Port0 [Not Detected]
► AHCI Port1 [Not Detected]
► AHCI Port2 [Not Detected]
► AHCI Port3 [Not Detected]
► AHCI Port4 [Not Detected]
► AHCI Port5 [Not Detected]

General ACPI Configuration settings

<- Select Screen
↑ ↓ Select Item
+- Change Field
Tab Select Field
F1 General Help
F10 Save and Exit
ESC Exit

BIOS SETUP UTILITY

Main **Advanced** PCIPnP Boot Security Chipset Exit

APM Configuration

Power Management/APM Enabled

Resume On Ring Disabled
Resume On PME# Disabled
Resume On RTC Alarm Disabled

Disable/Enable RI to generate a wake event.

<- Select Screen
↑ ↓ Select Item
+- Change Field
Tab Select Field
F1 General Help
F10 Save and Exit
ESC Exit

Resume on Ring

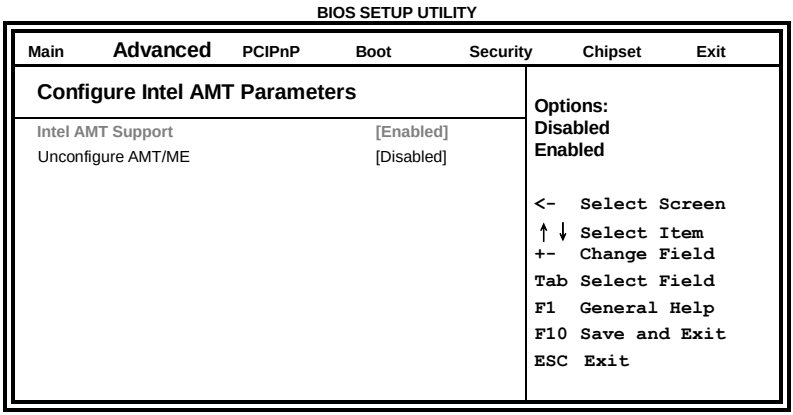
This option is used to enable activity on the RI (ring in) modem line to wake up the system from a suspend or standby state. That is, the system will be awakened by an incoming call on a modem.

Resume on PME#

This option is used enable activity on the PCI PME (power managementevent) controller to wake up the system from a suspend or standby state

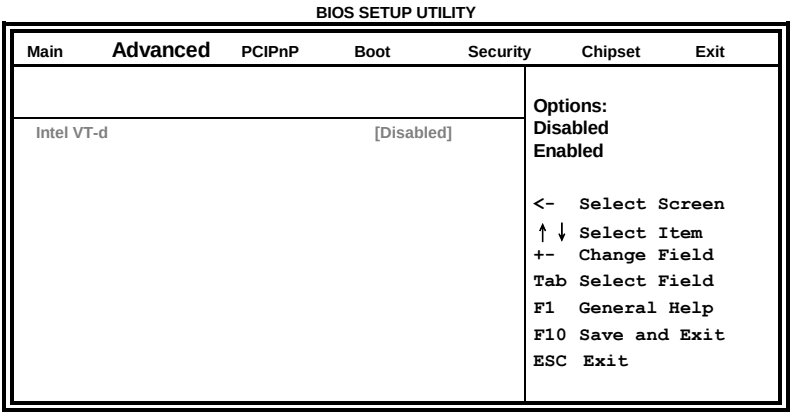
Resume On RTC Alarm

This option is used to specify the time the system should be awakened from a suspended state



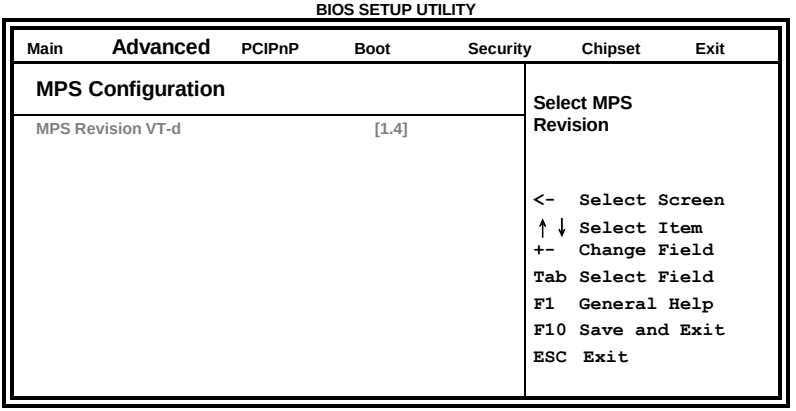
This configuration is supported only with MI946AF (with iAMT function).

The Intel AMT Configuration configures the Intel Active Management Technology (AMT) options.



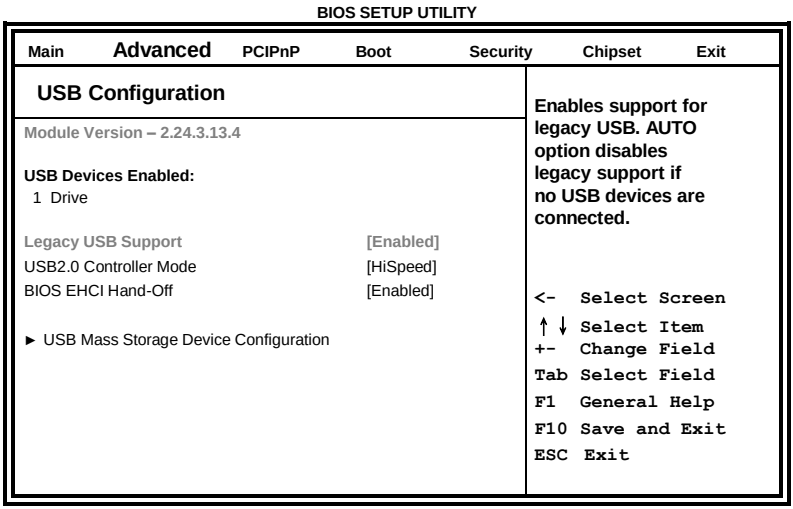
VT-d

This configuration is supported only with MI946AF.
Virtualization solutions allow multiple operating systems and applications to run in independent partitions all on a single computer. Using virtualization capabilities, one physical computer system can function as multiple "virtual" systems.



MPS Version Control for OS

This option is specifies the MPS (Multiprocessor Specification) version for your operating system. MPS version 1.4 added extended configuration tables to improve support for multiple PCI bus configurations and improve future expandability. The default setting is ***1.4***.



The USB Configuration menu is used to read USB configuration information and configure the USB settings.

Legacy USB Support

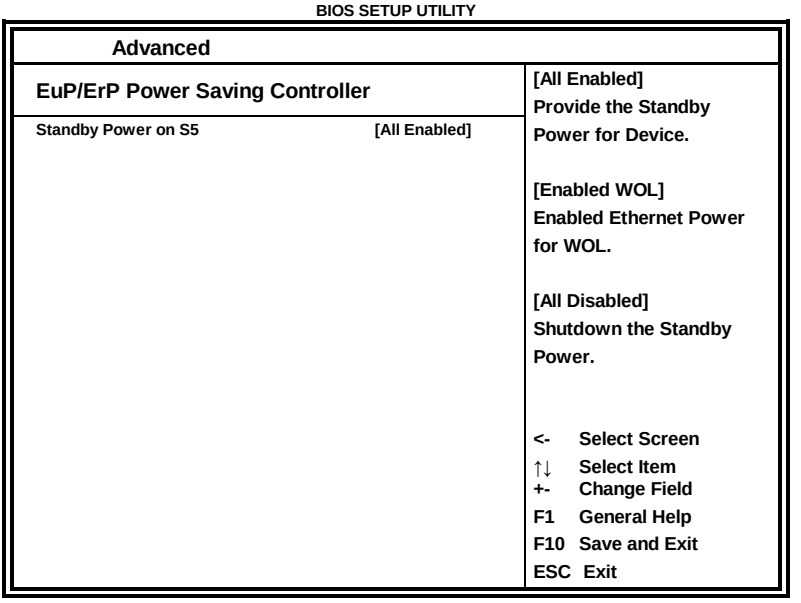
This option is used to enable the USB mouse and USB keyboard support. This option is enabled by default.

USB 2.0 Controller Mode

Configures the USB 2.0 controller in HiSpeed (480Mbps) or FullSpeed (12Mbps).This option is enabled by HiSpeed.

USB Mass Storage Device Configuration

This option is used to configure USB mass storage class devices.



Standby Power on S5

This configuration is supported only with MI946F/MI946RF.

PCIPnP Settings

This option configures the PCI/PnP settings.

BIOS SETUP UTILITY

Main	Advanced	PCIPnP	Boot	Security	Chipset	Exit

Advanced PCI/PnP Settings

WARNING: Setting wrong values in below sections may cause system to malfunction.

Clear NVRAM	[No]
Plug & Play O/S	[No]
PCI Latency Timer	[64]
Allocate IRQ to PCI VGA	[Yes]
Palette Snooping	[Disabled]
PCI IDE BusMaster	[Enabled]
IRQ3	[Available]
IRQ4	[Available]
IRQ5	[Available]
IRQ7	[Available]
IRQ9	[Available]
IRQ10	[Available]
IRQ11	[Available]
IRQ14	[Available]
IRQ15	[Available]
DMA Channel 0	[Available]
DMA Channel 1	[Available]
DMA Channel 3	[Available]
DMA Channel 5	[Available]
DMA Channel 6	[Available]
DMA Channel 7	[Available]
Reserved Memory Size	[Disabled]

Size of memory block to reserve for legacy ISA devices.

<- Select Screen
↑↓ Select Item
+- Change Field
Tab Select Field
F1 General Help
F10 Save and Exit
ESC Exit

Clear NVRAM

This item is used for clearing NVRAM during system boot.

Plug & Play O/S

This lets BIOS configure all devices in the system or lets the OS configure PnP devices not required for boot if your system has a Plug and Play OS.

PCI Latency Timer

This item sets value in units of PCI clocks for PCI device latency timer register. Options are: 32, 64, 96, 128, 160, 192, 224, 248.

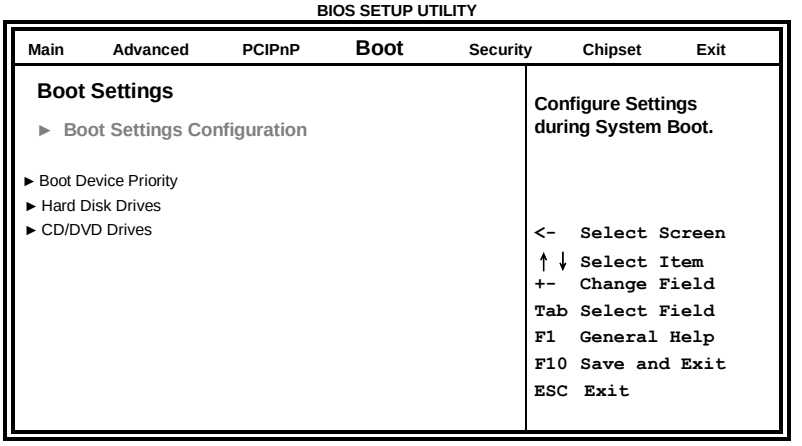
Allocate IRQ to PCI VGA
This assigns IRQ to PCI VGA card if card requests IRQ or doesn't assign IRQ to PCI VGA card even if card requests an IRQ.

Palette Snooping
This informs the PCI devices that an ISA graphics device is installed in the system so the card will function correctly.

PCI IDE BusMaster
This uses PCI busmastering for BIOS reading / writing to IDE devices.

IRQ#
Use the IRQ# address to specify what IRQs can be assigned to a particular peripheral device.

Boot Settings
This option configures the settings during system boot including boot device priority and HDD/CD/DVD drives.



BIOS SETUP UTILITY

Boot		
Boot Settings Configuration		**Allows BIOS to skip certain tests while booting. This will decrease the time needed to boot the system.**
Quick Boot	**[Enabled]**	
Quiet Boot	[Disabled]	
AddOn ROM Display Mode	[Force BIOS]	
Bootup Num-Lock	[On]	
PS/2 Mouse Support	[Auto]	
Wait for 'F1' If Error	[Enabled]	
Hit 'DEL' Message Display	[Enabled]	
Interrupt 19 Capture	[Disabled]	
		<- Select Screen
		↑↓ Select Item
		+- Change Field
		F1 General Help
		F10 Save and Exit
		ESC Exit

Quick Boot
This allows BIOS to skip certain tests while booting. This will decrease the time needed to boot the system.

Quite Boot
When disabled, this displays normal POST messages. When enabled, this displays OEM Logo instead of POST messages.

AddOn ROM Display Mode
This allows user to force BIOS/Option ROM of add-on cards to be displayed during quiet boot.

Bootup Num-Lock
This select the power-on state for numlock.

PS/2 Mouse Support
This select support for PS/w mouse.

Wait for 'F1' If Error
When set to Enabled, the system waits for the F1 key to be pressed when error occurs. This allows option ROM to trap interrupt 19.

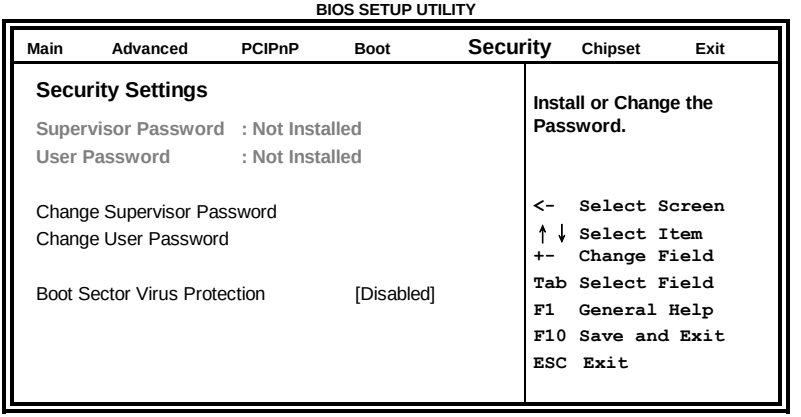
Hit <DEL> Message Display
This displays "Press <DEL> to run Setup" in POST.

Interrupt 19 Capture
This allows option ROMs to trap interrupt 19.

Security Settings

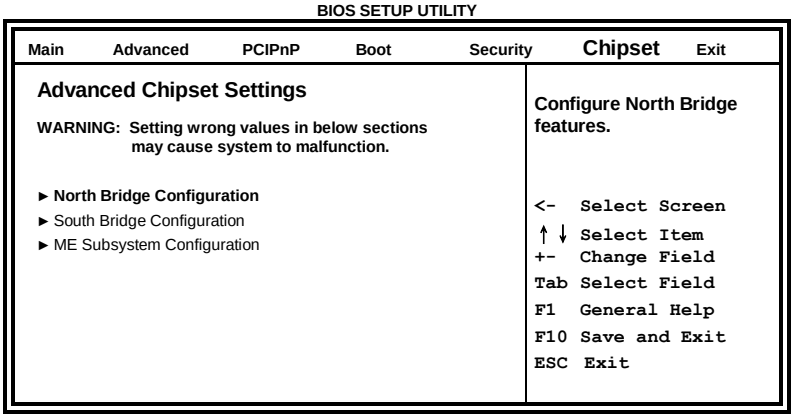
This setting comes with two options set the system password. Supervisor Password sets a password that will be used to protect the system and Setup utility. User Password sets a password that will be used exclusively on the system. To specify a password, highlight the type you want and press <Enter>. The Enter Password: message prompts on the screen. Type the password and press <Enter>. The system confirms your password by asking you to type it again. After setting a password, the screen automatically returns to the main screen.

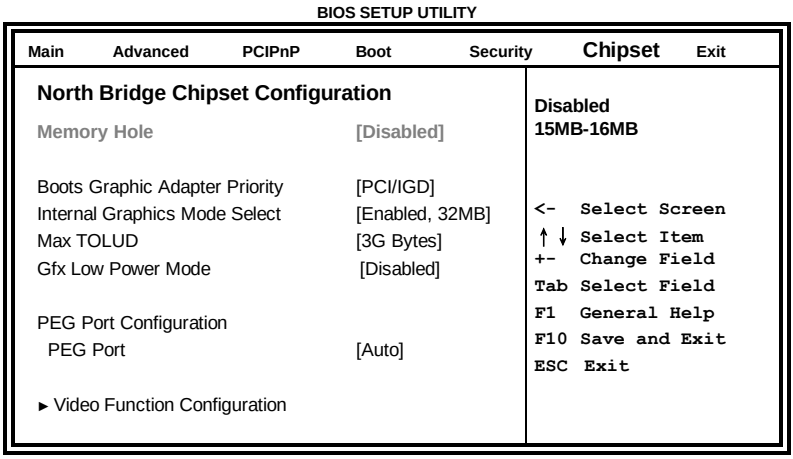
To disable a password, just press the <Enter> key when you are prompted to enter the password. A message will confirm the password to be disabled. Once the password is disabled, the system will boot and you can enter Setup freely.



Advanced Chipset Settings

This setting configures the north bridge, south bridge and the ME subsystem. WARNING! Setting the wrong values may cause the system to malfunction. -





Memory Hole

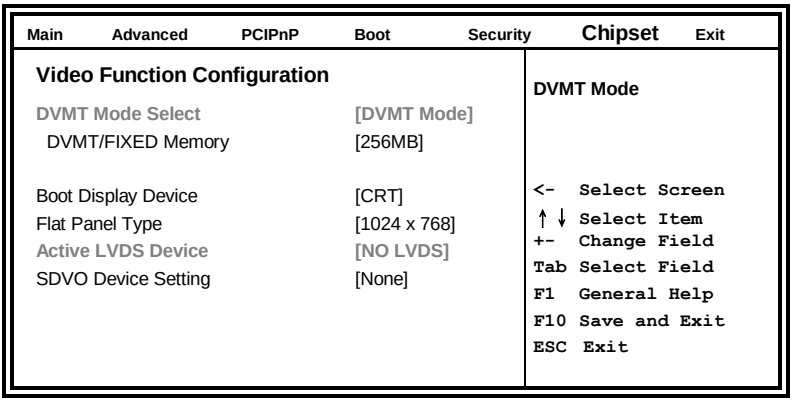
This option is used to reserve memory space between 15MB and 16MB for ISA expansion cards that require a specified area of memory to work properly.

Boots Graphics Adapter Priority

This option is used to select the graphics controller used as the primary boot device. Select either an integrated graphics controller (IGD) or a combination of PCI graphics controller, a PCI express (PEG) controller or an IGD.

Internal Graphics Mode Select

This option is used to specify the amount of system memory that can be used by the Internal graphics device.



Video Function Configuration

This option is used to configure the video device connected to the system.

Boot Display Device

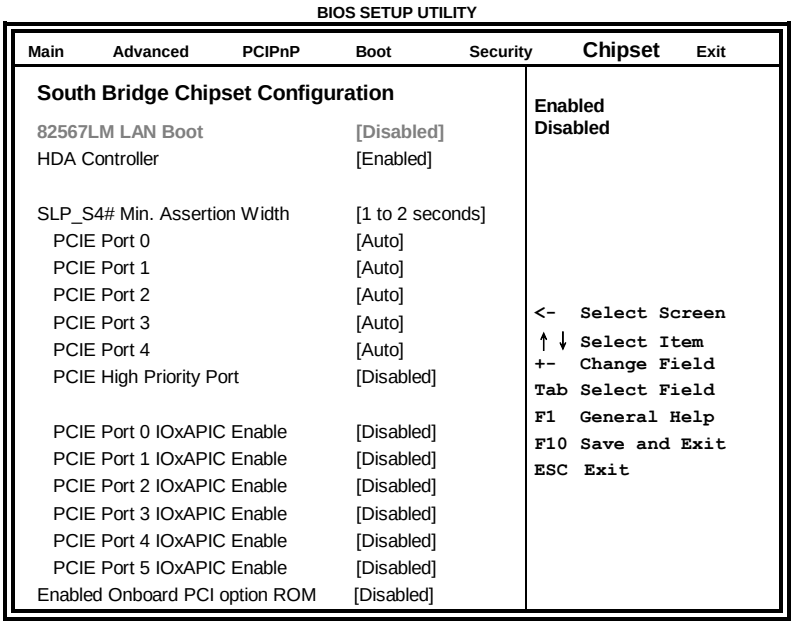
This option is used to select the display device used by the system when it boots.

Flat Panel Type

This option is used to select the type of flat panel connected to the system. Options include: 640x480 18b / 800x600 18b / 1024x768 18b / 1024x768 24b / 1280x1024 24b / 1600x1200 24b

Active LVDS Device

This option is set to Integrated LVDS, by default.

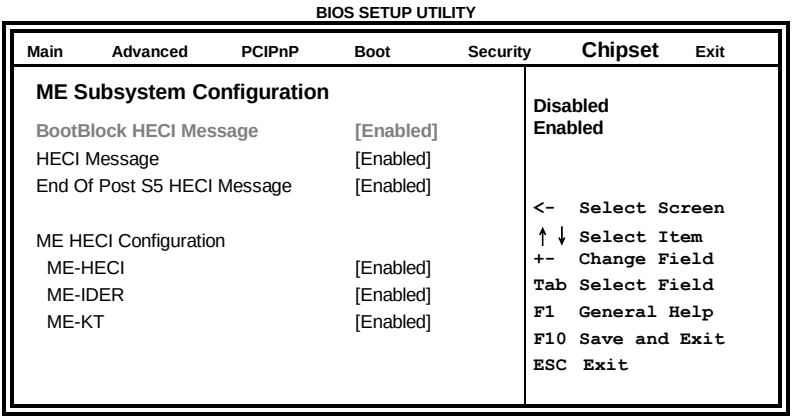


82567LM LAN Boot

This option is disabled by default. The system, then, will not boot using the Gigabit LAN interface.

HDA Controller

This option is used to enable the Southbridge high definition audio controller.



BootBlock HECI Message

This configuration is supported only with MI946AF (with iAMT function). This option allows you enable or disable HECI message when booting up the system..

HECI Message

This configuration is supported only with MI946AF (with iAMT function). This option allows you to enable or disable the HECI message.

End Of Post S5 HECI Message

This configuration is supported only with MI946AF (with iAMT function). This option allows you to enable or disable HECI message when the system is in the off (S5) state.

ME-HECI

This configuration is supported only with MI946AF (with iAMT function). This option is enabled by default and cannot be changed.

ME-IDER

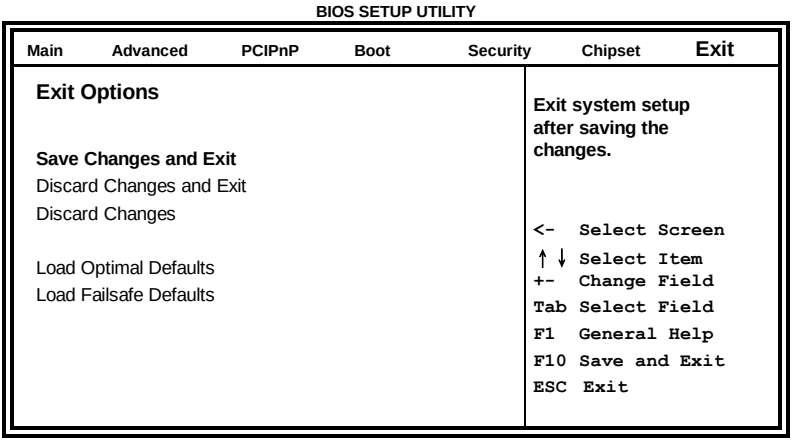
This configuration is supported only with MI946AF (with iAMT function). This option is used to enable or disable the IDE-Redirection (IDE-R) function on an AMT-capable system.

ME-KT

This configuration is supported only with MI946AF (with iAMT function). This option is used to enable or disable the Keyboard and Text redirection (KT) function on an AMT-capable system. KT is also known as Serial-Over-Lan (SOL). When enabled, the KT function allows a management system to control an IntelR AMT clientsystem remotely. The keyboard interface of a managed client system, such as BIOS menu, is displayed through the management system.

Exit Setup

The exit setup has the following settings which are:



Save Changes and Exit

This option allows you to determine whether or not to accept the modifications and save all changes into the CMOS memory before exit.

Discard Changes and Exit

This option allows you to exit the Setup utility without saving the changes you have made in this session.

Discard Changes

This option allows you to discard all the changes that you have made in this session.

Load Optimal Defaults

This option allows you to load the default values to your system configuration. These default settings are optimal and enable all high performance features.

Load Failsafe Defaults

This option allows you to load the troubleshooting default values permanently stored in the BIOS ROM. These default settings are non-optimal and disable all high-performance features.

Drivers Installation

This section describes the installation procedures for software and drivers under Windows XP. 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 (Windows XP), 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 CD that comes with the board. Click ***Intel*** and then ***Intel(R) GM45 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 CD that comes with the board. Click ***Intel*** and then ***Intel(R) GM45 Chipset Drivers***.

2. Click ***Intel(R) GM45 Chipset Family Graphics Driver.***

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

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® Graphics Media Accelerator Driver.

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

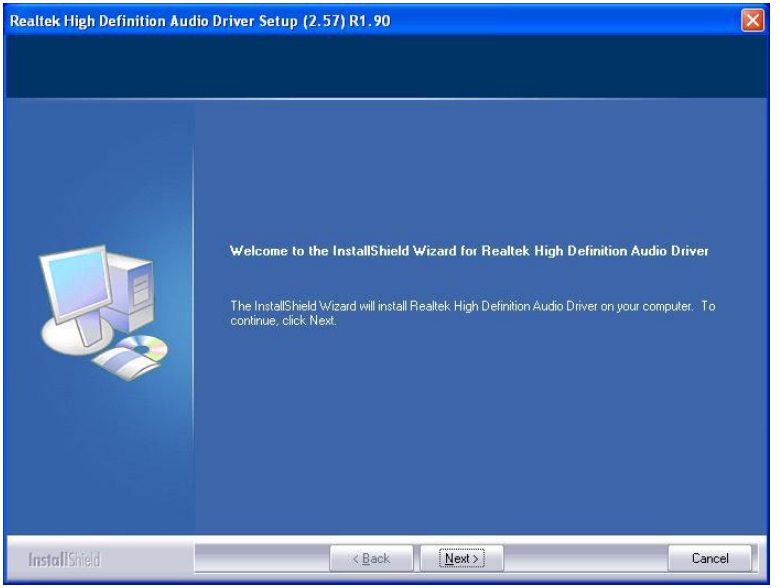
7. Setup complete. Click ***Finish*** to restart the computer and for changes to take effect.

Realtek Audio Driver Installation

1. Insert the CD that comes with the board. Click ***Intel*** and then ***Intel(R) GM45 Chipset Drivers***.

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

3. On the Welcome to the InstallShield Wizard screen, click ***Next***.



3. InstallShield Wizard is complete. Click ***Finish*** to restart the computer.

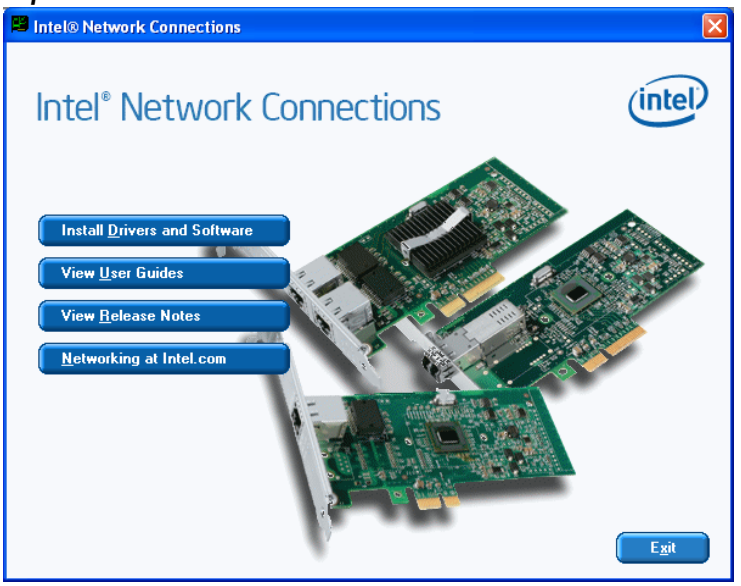
LAN Drivers Installation

Follow the steps below to install the Intel LAN drivers. ***This one installation will cover both 82574L and 82567LM LAN controllers.***

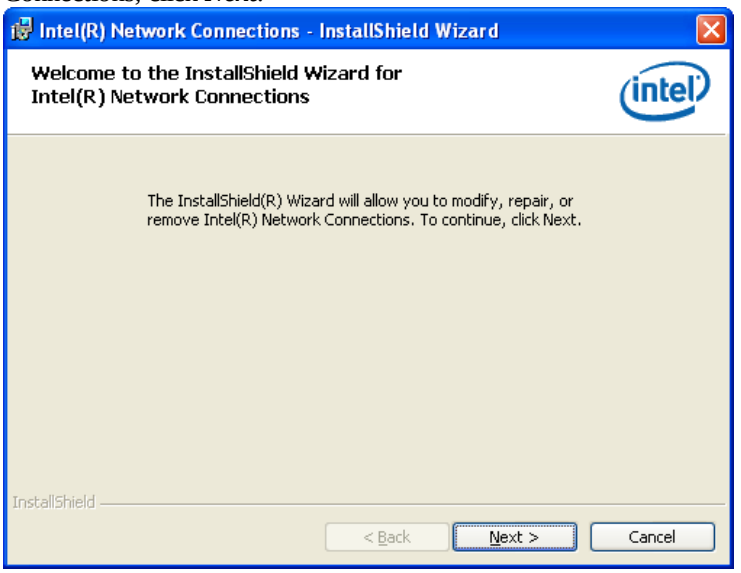
1. Insert the CD that comes with the board. Click ***Intel*** and then ***Intel(R) GM45 Chipset Drivers***.

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

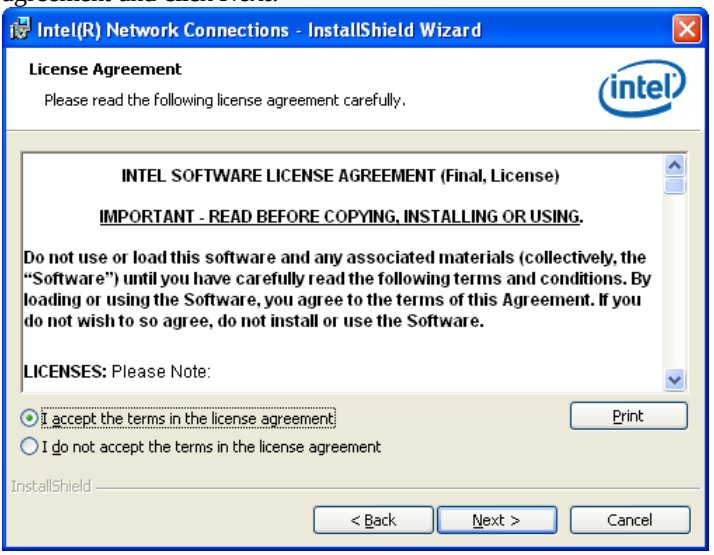
3. In the Intel® Network Connections screen*, click ***Install Drivers and Software***.



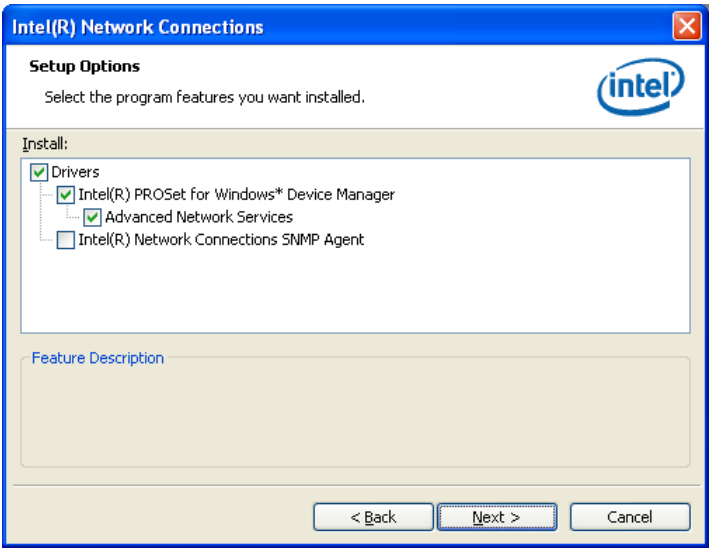
4. In the welcome screen of the InstallShield Wizard for Intel(R) Network Connections, click ***Next***.



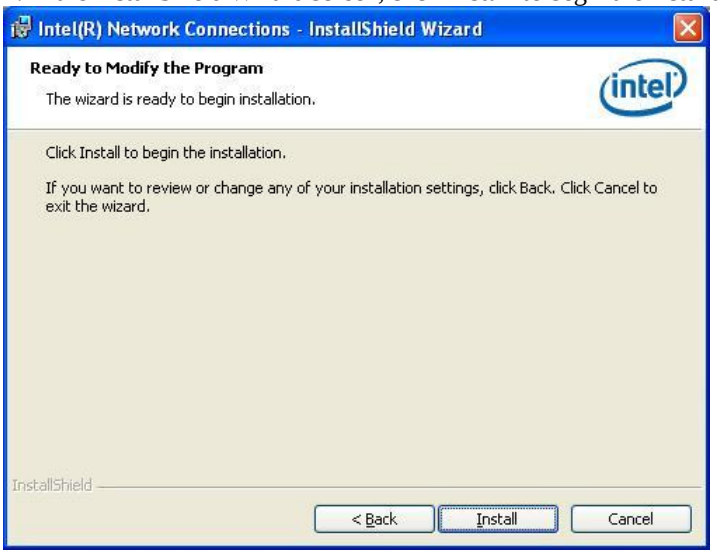
5. In the License Agreement, Select I accept the terms in the license agreement and click ***Next***.



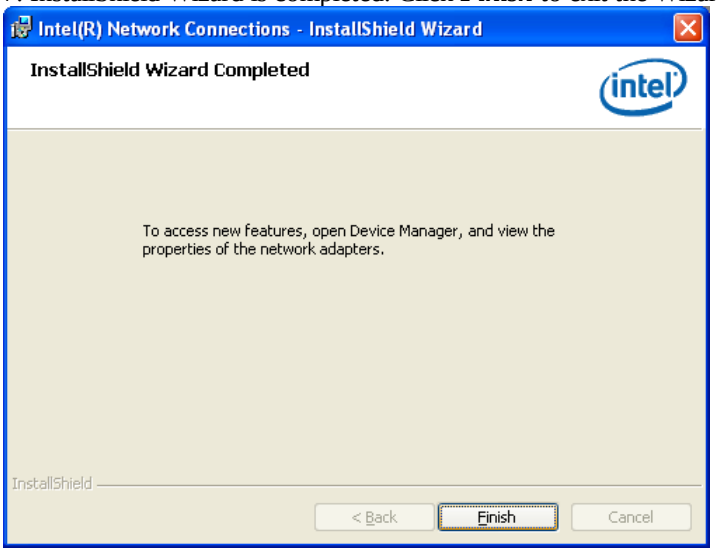
6. In the Setup Options, click the checkbox as shown below and click ***Next***.



7. In the InstallShield Wizard screen, click ***Install*** to begin the installation.



7. InstallShield Wizard is completed. Click ***Finish*** to exit the Wizard.



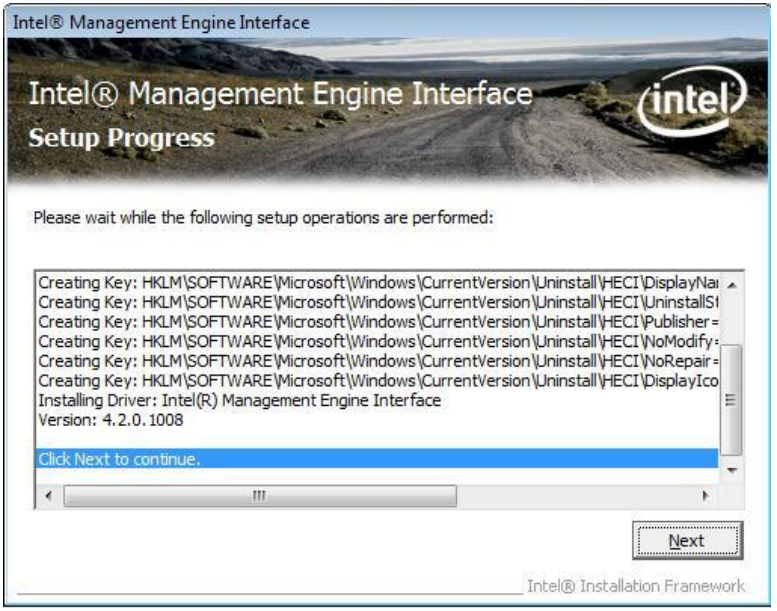
Intel® Management Engine Interface

REMARKS: The Intel iAMT 4.0 Drivers can be installed on MI946AF, not MI946F.

Follow the steps below to install the Intel Management Engine.

1. Insert the drivers disc that comes with the motherboard. Click ***Intel*** and then ***Intel(R) AMT 4.0 Drivers***, then ***Intel(R) MEI Driver***.

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



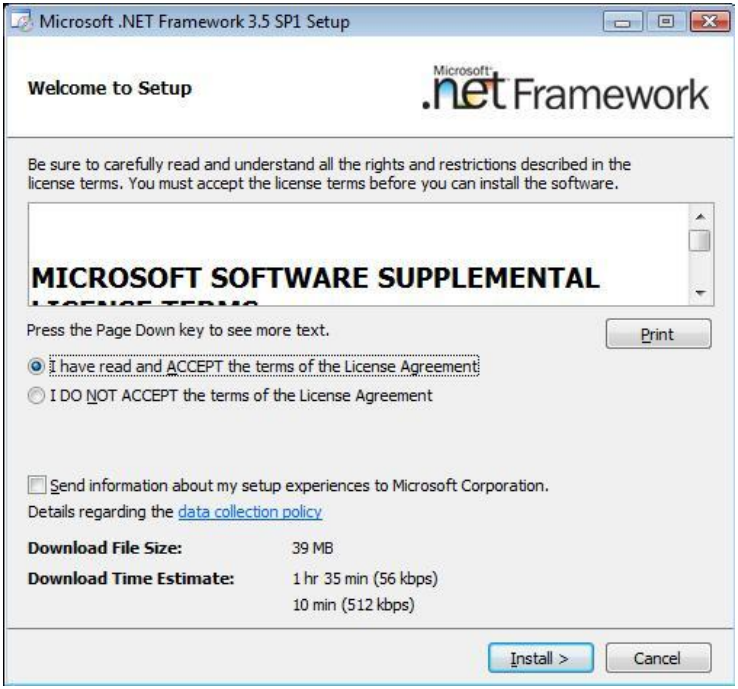
Intel® AMT SOL Driver Installation

REMARKS: The Intel iAMT 4.0 Drivers can be installed on MI946AF, not MI946F.

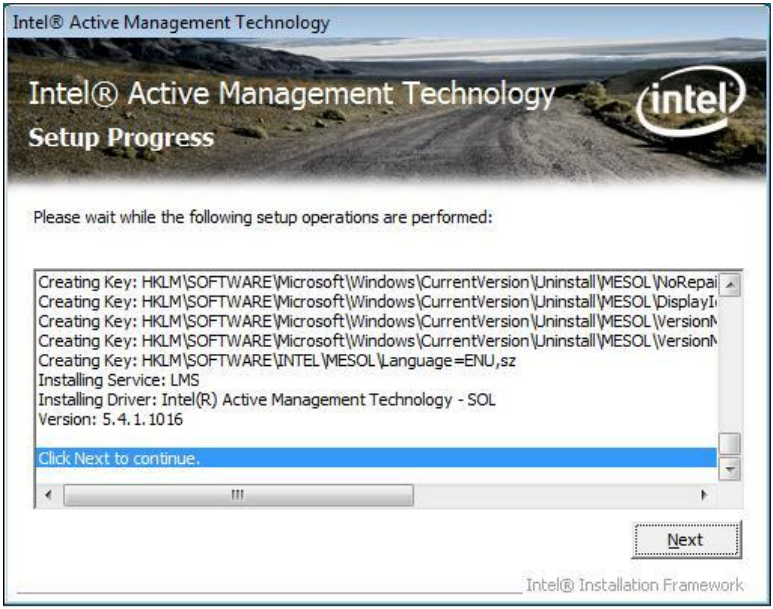
Follow the steps below to install the Intel Management Engine.

1. Insert the drivers disc that comes with the motherboard. Click ***Intel*** and then ***Intel(R) AMT 4.0 Drivers***, then ***Intel(R) LMS/SOL Driver***.

2. On the Setup screen for Microsoft .NET Framework 3.5 SPI, click ***Install***. When Setup is complete, click ***Exit***.



3. The next screen shows the Intel® Active Management Technology setup progress where the Intel AMT SOL driver is being installed. Click ***Next***. Click ***Finish*** when setup process is complete.



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
2F8h - 2FFh	Serial Port #2(COM2)
2B0 - 2DF	Graphics adapter Controller
360 - 36F	Network Ports
3B0 - 3BF	Monochrome & Printer adapter
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	Floppy Disk Controller
IRQ8	Real Time Clock
IRQ9	Reserved
IRQ10	Reserved
IRQ11	Reserved
IRQ12	PS/2 Mouse
IRQ13	80287
IRQ14	Primary IDE
IRQ15	Secondary IDE

C. 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 W627UHG.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 "W627UHG.H"
#include <dos.h>
//-----------------------------------------------------------------------------
unsigned int W627UHG_BASE;
void Unlock_W627UHG (void);
void Lock_W627UHG (void);
//-----------------------------------------------------------------------------
unsigned int Init_W627UHG(void)
{
        unsigned int result;
        unsigned char ucDid;

        W627UHG_BASE = 0x4E;
        result = W627UHG_BASE;

        ucDid = Get_W627UHG_Reg(0x20);
        if (ucDid == 0xA2)
                //W83627UHG??
        {       goto Init_Finish;               }

        W627UHG_BASE = 0x2E;
        result = W627UHG_BASE;

        ucDid = Get_W627UHG_Reg(0x20);
        if (ucDid == 0xA2)
                //W83627UHG??
        {       goto Init_Finish;               }

        W627UHG_BASE = 0x00;
        result = W627UHG_BASE;

Init_Finish:
        return (result);
}
//-----------------------------------------------------------------------------
void Unlock_W627UHG (void)
```

```
{
		outportb(W627UHG_INDEX_PORT, W627UHG_UNLOCK);
		outportb(W627UHG_INDEX_PORT, W627UHG_UNLOCK);
}
//-----------------------------------------------------------------------------
void Lock_W627UHG (void)
{
		outportb(W627UHG_INDEX_PORT, W627UHG_LOCK);
}
//-----------------------------------------------------------------------------
void Set_W627UHG_LD( unsigned char LD)
{
		Unlock_W627UHG();
		outportb(W627UHG_INDEX_PORT, W627UHG_REG_LD);
		outportb(W627UHG_DATA_PORT, LD);
		Lock_W627UHG();
}
//-----------------------------------------------------------------------------
void Set_W627UHG_Reg( unsigned char REG, unsigned char DATA)
{
		Unlock_W627UHG();
		outportb(W627UHG_INDEX_PORT, REG);
		outportb(W627UHG_DATA_PORT, DATA);
		Lock_W627UHG();
}
//-----------------------------------------------------------------------------
unsigned char Get_W627UHG_Reg(unsigned char REG)
{
		unsigned char Result;
		Unlock_W627UHG();
		outportb(W627UHG_INDEX_PORT, REG);
		Result = inportb(W627UHG_DATA_PORT);
		Lock_W627UHG();
		return Result;
}
//-----------------------------------------------------------------------------
```

File of the W627UHG.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 __W627UHG_H
#define __W627UHG_H                                    1
//---------------------------------------------------------------------------
#define         W627UHG_INDEX_PORT                     (W627UHG_BASE)
#define         W627UHG_DATA_PORT                      (W627UHG_BASE+1)
//---------------------------------------------------------------------------
#define         W627UHG_REG_LD                                 0x07
//---------------------------------------------------------------------------
#define W627UHG_UNLOCK                                         0x87
#define         W627UHG_LOCK                                   0xAA
//---------------------------------------------------------------------------
unsigned int Init_W627UHG(void);
void Set_W627UHG_LD( unsigned char);
void Set_W627UHG_Reg( unsigned char, unsigned char);
unsigned char Get_W627UHG_Reg( unsigned char);
//---------------------------------------------------------------------------
#endif          //__W627UHG_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 "W627UHG.H"
//---------------------------------------------------------------------------
int main (void);

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

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

        SIO = Init_W627UHG();
        if (SIO == 0)
        {
        .....printf("Can not detect Winbond 83627UHG, program abort.\n");
        ..... return(1);
        }

        WDTInitial();

        WDTEnable(10);

        WDTDisable();

        return 0;
}
//---------------------------------------------------------------------------
void WDTInitial(void)
{
        unsigned char bBuf;
        Set_W627UHG_LD(0x08);..... //switch to logic device 8
        bBuf = Get_W627UHG_Reg(0x30);
        bBuf &= (~0x01);
        Set_W627UHG_Reg(0x30, bBuf);..... //Enable WDTO
}
//---------------------------------------------------------------------------
void WDTEnable(unsigned char NewInterval)
{
        unsigned char bBuf;

        Set_W627UHG_LD(0x08);.....
        Set_W627UHG_Reg(0x30, 0x01);..... //enable timer
```

```
        bBuf = Get_W627UHG_Reg(0xF5);
        bBuf &= (~0x08);
        Set_W627UHG_Reg(0xF5, bBuf);.....//count mode is second

        Set_W627UHG_Reg(0xF6, NewInterval); .....//set timer
}
//-------------------------------------------------------------------------
void WDTDisable(void)
{
        Set_W627UHG_LD(0x08);.....
        Set_W627UHG_Reg(0xF6, 0x00);.....//clear watchdog timer
        Set_W627UHG_Reg(0x30, 0x00);.....
}
//-------------------------------------------------------------------------
```

D. Digital I/O Sample Code

File of the W627UHG.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 __W627UHG_H
#define __W627UHG_H                                    1
//---------------------------------------------------------------------------
#define         W627UHG_INDEX_PORT              (W627UHG_BASE)
#define         W627UHG_DATA_PORT               (W627UHG_BASE+1)
//---------------------------------------------------------------------------
#define         W627UHG_REG_LD                          0x07
//---------------------------------------------------------------------------
#define W627UHG_UNLOCK                                  0x87
#define         W627UHG_LOCK                            0xAA
//---------------------------------------------------------------------------
unsigned int Init_W627UHG(void);
void Set_W627UHG_LD( unsigned char);
void Set_W627UHG_Reg( unsigned char, unsigned char);
unsigned char Get_W627UHG_Reg( unsigned char);
//---------------------------------------------------------------------------
#endif          //__W627UHG_H
```

File of the W627UHG.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 "W627UHG.H"
#include <dos.h>
//---------------------------------------------------------------------------
unsigned int W627UHG_BASE;
void Unlock_W627UHG (void);
void Lock_W627UHG (void);
//---------------------------------------------------------------------------
unsigned int Init_W627UHG(void)
{
        unsigned int result;
        unsigned char ucDid;

        W627UHG_BASE = 0x4E;
        result = W627UHG_BASE;

        ucDid = Get_W627UHG_Reg(0x20);
        if (ucDid == 0xA2)
                //W83627UHG??
        {       goto Init_Finish;       }

        W627UHG_BASE = 0x2E;
        result = W627UHG_BASE;

        ucDid = Get_W627UHG_Reg(0x20);
        if (ucDid == 0xA2)
                //W83627UHG??
        {       goto Init_Finish;       }

        W627UHG_BASE = 0x00;
        result = W627UHG_BASE;

Init_Finish:
        return (result);
}
//---------------------------------------------------------------------------
void Unlock_W627UHG (void)
{
        outportb(W627UHG_INDEX_PORT, W627UHG_UNLOCK);
        outportb(W627UHG_INDEX_PORT, W627UHG_UNLOCK);
}
//---------------------------------------------------------------------------
void Lock_W627UHG (void)
{
        outportb(W627UHG_INDEX_PORT, W627UHG_LOCK);
}
//---------------------------------------------------------------------------
void Set_W627UHG_LD( unsigned char LD)
```

```
{
		Unlock_W627UHG();
		outportb(W627UHG_INDEX_PORT, W627UHG_REG_LD);
		outportb(W627UHG_DATA_PORT, LD);
		Lock_W627UHG();
}
//-----------------------------------------------------------------------
void Set_W627UHG_Reg( unsigned char REG, unsigned char DATA)
{
		Unlock_W627UHG();
		outportb(W627UHG_INDEX_PORT, REG);
		outportb(W627UHG_DATA_PORT, DATA);
		Lock_W627UHG();
}
//-----------------------------------------------------------------------
unsigned char Get_W627UHG_Reg(unsigned char REG)
{
		unsigned char Result;
		Unlock_W627UHG();
		outportb(W627UHG_INDEX_PORT, REG);
		Result = inportb(W627UHG_DATA_PORT);
		Lock_W627UHG();
		return Result;
}
//-----------------------------------------------------------------------
```

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 "W627UHG.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_W627UHG();
        if (SIO == 0)
        {
                printf("Can not detect Winbond 83627UHG, program abort.\n");
                return(1);
        }

        Dio5Initial();

        //for GPIO50..57
        Dio5SetDirection(0x0F);         //GP50..53 = input, GP54..57=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;

        Set_W627UHG_LD(0x08);
                                        //switch to logic device 8
        //enable the GP5 group
        ucBuf = Get_W627UHG_Reg(0x30);
        ucBuf |= 0x02;
        Set_W627UHG_Reg(0x30, ucBuf);
}
//---------------------------------------------------------------------------
void Dio5SetOutput(unsigned char NewData)
{
        Set_W627UHG_LD(0x08);
                //switch to logic device 8
        Set_W627UHG_Reg(0xE1, NewData);
}
//---------------------------------------------------------------------------
unsigned char Dio5GetInput(void)
{
        unsigned char result;

        Set_W627UHG_LD(0x08);
                //switch to logic device 8
        result = Get_W627UHG_Reg(0xE1);
        return (result);
}
//---------------------------------------------------------------------------
void Dio5SetDirection(unsigned char NewData)
{
        //NewData : 1 for input, 0 for output
        Set_W627UHG_LD(0x08);
                //switch to logic device 8
        Set_W627UHG_Reg(0xE0, NewData);
}
//---------------------------------------------------------------------------
unsigned char Dio5GetDirection(void)
{
        unsigned char result;

        Set_W627UHG_LD(0x08);
                //switch to logic device 8
        result = Get_W627UHG_Reg(0xE0);
        return (result);
}
//---------------------------------------------------------------------------
```

This page is intentionally left blank.