MB941

**LGA775 Core™ 2 Duo
Intel® G41 Chipset
Micro-ATX Motherboard**

USER'S MANUAL

Version 1.0

Acknowledgments

Table of Contents

This page is intentionally left blank.

Introduction

Checklist

Your MB941 Core 2 Duo motherboard package should include the items listed below:

- The MB941 motherboard
- This User's manual
- 1x SATA cable (SATA-5)
- 1x COM port cable (PK1-56)
- 1x USB cable (USB2-9)
- 1x DVD containing the following:
    - Chipset Drivers
    - Flash Memory Utility

Remarks: PK1-20B dual-port COM cable is optional.

Product Description

The MB941 Micro-ATX motherboard is designed for either the Intel® Core™2 Duo or Core™2 Quad processors of up to 1333MHz FSB. It is based on the Intel G41 Express chipset and it comes with two dual-channel DDR3 memory slots and 8GB memory capacity for faster system responsiveness and support of 64-bit computing.

Dual independent display comes to life with the onboard Intel® G41 integrated graphics with CRT. LAN functionality is supported with two Gigabit Ethernet controllers.

MB941 is supports high-end features including high definition audio, eight fast USB ports, four SATAII, watchdog timer, digital I/O and four serial ports. Board dimensions are 244mm by 244mm.
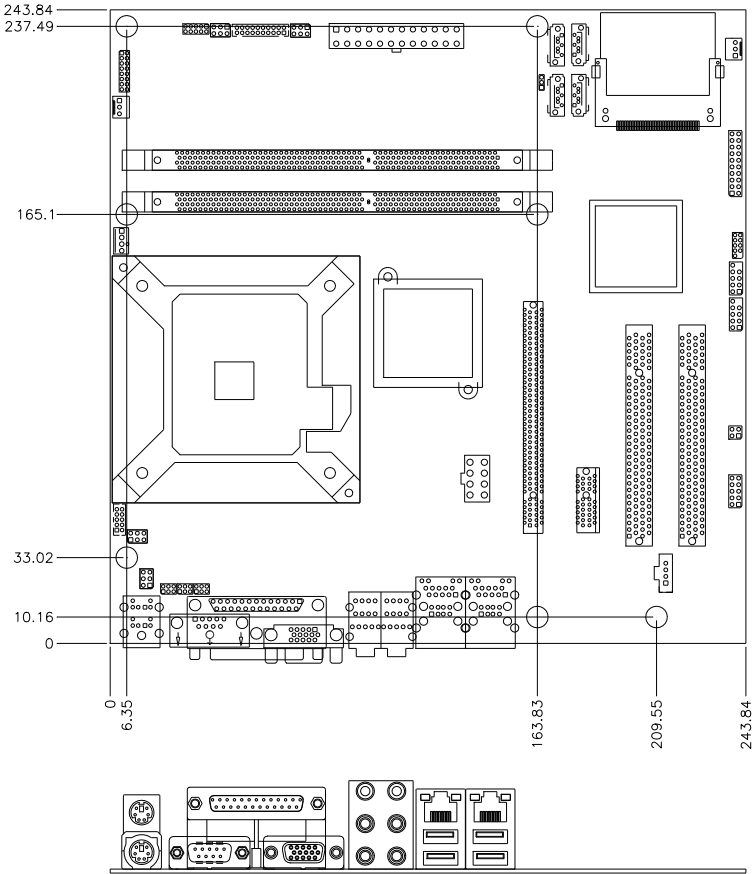
MB941 FEATURES

- Supports Intel® Core™2 Quad / Core™2 Duo / Pentium® Dual Core/ Celeron® processors
- 800MHz/1066MHz/1333MHz FSB
- DDR3 DIMM x2, Max. 8GB
- Single Gigabit LAN
- Integrated G41 VGA for CRT
- 8x USB 2.0, 4x SATA II, 4x COM
- HD audio, Watchdog timer, 2x PCI , 1x PCI-E x1, 1x PCI-E x16.

Specifications

Product	MB941
Form Factor	Micro-ATX
CPU Type	LGA775 socket for Intel® Core™2 Quad / Intel® Core™2 Duo / Celeron® processors 800/1066/1333MHz FSB (45nm/65nm Technology)
Green /APM	APM1.2
BIOS	**AMI** BIOS, support ACPI Function
Chipset	Intel® G41 (82G41) Express GMCH: 1254 FCBGA (25W) Intel® ICH7 (82801GB): 652 BGA (3.3W)
Memory Slot	240 DIMM x 2 (w/o ECC), for DDR3-1066 / DDR3-800 MHz Max. 4GB, dual channel
VGA	Intel® G41 Integrated Graphics Controller (GMA X4500) - Support DX10 and OpenGL2.1 - Dual independent display available through on-board VGA - and DVI-D (via IBP192 on PCIe x16 slot) ~ optional
LAN	Realtek RTL8111E PCI-E Gigabit LAN Controller x 2
USB	ICH7 built-in USB 2.0 host controller, support 8 ports - 4 ports on edge connectors - 4 ports from pin headers
Serial ATA Ports	ICH7 built-in SATA controller, supports 4 ports - SATA II (3.0 Gb/s)
Compact Flash Slot	ICH7 built-in one channel UDMA 33/66/100, for CF Type II (on component side)
Audio	Intel ICH7 built-in 7.1CH HD audio controller w/ Realtek ALC888 Codec - Edge connector (6 audio jacks)
LPC I/O	Winbond W83627UHG : - COM1 (RS232/422/485), COM2 (RS232) - COM3 (RS232), COM4 (RS232) [with pin-9 with power for 4 ports (500 mA for each port)] - CPU fan connector supports 4-wire fan with PWM control - System fan connector x1 DC fan control - System fan connector x1 w/o control - Supports two thermal sensor (CPU + on-board) - Voltage monitoring for VCC (processor), 3.3V, 5V, and 12V - Hardware monitor (2 thermal inputs, 4 voltage monitor inputs, VID0-4 & 1 x fan header)
Digital IO	4 in & 4 out
Expansion Slots	PCI-express (16x) slot x 1 PCI-Express(1x) slot x1 PCI slot x 2
IBASE Proprietary LPC Port	2 x 8 pins header x 1 for optional features: - ID395-R for TPM 1.2 ~ Optional (Custom BIOS needed to support this function) - ID394-R for 4 additional serial Ports ~ Optional (Custom BIOS needed to support this function)
Edge Connector	PS/2 KB/MS, COM1, CRT, LPT, HD audio RJ45 + dual USB stack connector x 2 for LAN1~2 & USB1~4
On Board Header/ Connector	Standard SATA (7-pin shrouded vertical) connector x4 2x5-pin header x2 for USB 5~8, 2x5-pin DF11 x1 for COM2 (RS232) 2x10-pin DF11 x1 for COM3~4 (RS232), 2x5-pin header x1 for Digital I/O 4-pins header x1 for CPU fan & 3-pim headers x2 for system fan 2x8-pin box header for power switch/power LED/HDD LED/reset switch
Watchdog Timer	Yes (256 segments, 0, 1, 2…255 sec/min)
Power Connector	ATX1 (24-pin connector for system) ATX_12 (8-pin connector for CPU operating voltage)
Board Size	244mm x 244mm

Board Dimensions

Installations

This section provides information on how to use the jumpers and connectors on the MB941 in order to set up a workable system. The topics covered are:

Installing the CPU

The MB941 motherboard supports an LGA 775 processor socket for Intel® Core 2 Duo processors.

The LGA 775 processor socket comes with a lever to secure the processor. Refer to the pictures below, from left to right, on how to place the processor into the CPU socket. *Please note that the cover of the LGA775 socket must always be installed during transport to avoid damage to the socket.*
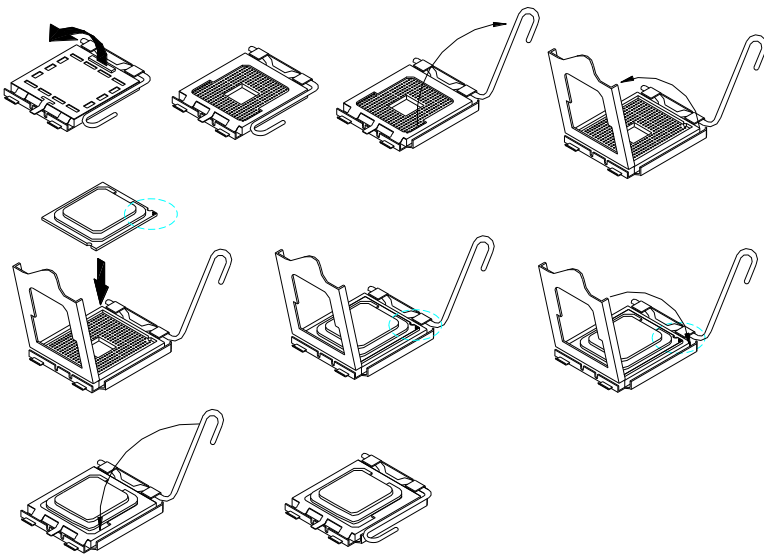
ATX Power Installation

The system power is provided to the motherboard with the ATX1 and ATX2 power connectors. ATX1 is a 24-pin power connector and ATX2 is a 8-pin 12V power connector.

The 24-pin power connector can to be connected to a standard 20-pin ATX power connector in a standard ATX power supply (Min. 400watt).

Note: The power supply 5VSB voltage must be at least 2A.

Installing the Memory

The MB941 motherboard supports two DDR3 memory sockets for a maximum total memory of 8GB. It supports DDR3 800/1066MHz.

Basically, the system memory interface has the following features:

Supports two 64-bit wide DDR data channels

Available bandwidth up to 6.4GB/s (DDR3 1066) for two-channel mode.

Supports 512Mb, 1Gb, 2Gb DDR3 technologies.

Supports only x8, x16, DDR3 devices with four banks

Supports only unbuffered DIMMs

Supports opportunistic refresh

Up to 32 simultaneously open pages (four per row, four rows maximum)

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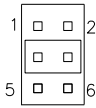
Jumpers are used on the motherboard are used to select various settings and features according to your needs and applications. Contact your supplier if you have doubts about the best configuration for your needs. The following lists the connectors and their respective functions.

Jumper Locations on MB941
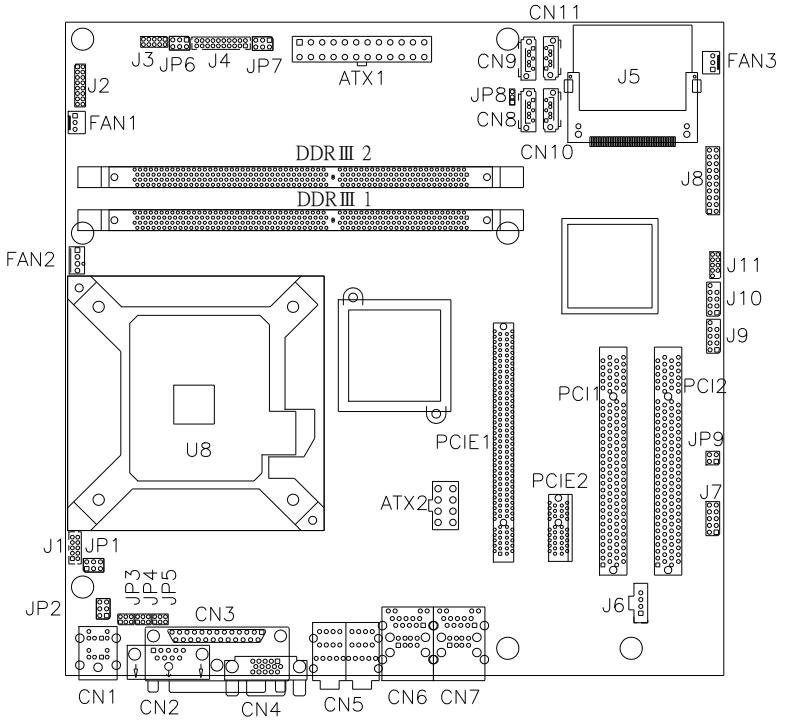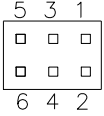
JP2, JP1, JP7, JP6: COM1/2/3/4 RS232 +5V/+12V Power Setting
JP3, JP4, JP5: RS232/422/485 (COM1) Selection
JP8: Clear CMOS Contents

JP2, JP1, JP7, JP6: COM1/2/3/4 RS232 +5V/+12V Power Setting

JP4/JP6	Setting	Function
	Pin 1-2 Short/Closed	+12V
	Pin 3-4 Short/Closed	Normal
	Pin 5-6 Short/Closed	+5V

JP3, JP4, JP5: RS232/422/485 (COM1) Selection
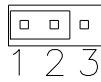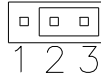
COM1 is selectable for RS232, RS-422 and RS-485.

COM1 Function	RS-232	RS-422	RS-485
Jumper Setting (pin closed)	JP3: 1-2	JP3: 3-4	JP3: 5-6
	JP4: 3-5 & 4-6	JP4: 1-3 & 2-4	JP4: 1-3 & 2-4
	JP5: 3-5 & 4-6	JP5: 1-3 & 2-4	JP5: 1-3 & 2-4

JP8: Clear CMOS Contents

Use JP8, a 3-pin header, to clear the CMOS contents.

Note that the ATX-power connector should be disconnected from the motherboard before clearing CMOS.

JP8	Setting	Function
	Pin 1-2 Short/Closed	Normal
	Pin 2-3 Short/Closed	Clear CMOS

Connectors on MB941

The connectors on MB941 allow you to connect external devices such as keyboard, floppy disk drives, hard disk drives, printers, etc. The following table lists the connectors on MB941 and their respective functions.

Connector Locations on MB941

ATX1: 24-pin ATX Power Connector
ATX2: ATX 12V Power Connector
DDRIII1: Channel A DDR3 Socket
DDRIII2: Channel B DDR3 Socket
FAN2: CPU Fan Power Connector
FAN1: System Fan1 Power Connector
FAN3: System Fan2 Power Connector
CN1: PS/2 Keyboard and PS/2 Mouse
Connectors
CN2: Serial Ports(COM1)
CN3: Parallel Port Connector
CN4: VGA CRT Connector
CN5: Audio Connector
CN6: GbE RJ-45 and USB0/1 Connector
CN7: GbE RJ-45 and USB2/3 Connector
CN11, CN9, CN10, CN8: SATA0/1/2/3
Connector

J1: COM2/RS232 Serial Port (DF11
Connector)
J2: For LPC I/F Adaptor Card
J3: Digital I/O Connector (4 in, 4 out)
J4: COM3, COM4 Serial Port (DF11
Connector)
J5: Compact Flash Connector
J6: Audio CD-IN Header
J7: Audio Front Header
J8: System Function Connector
J9, J10: USB 4/5/6/7 Connector
J11: SPI Debug Tools Port (Factory use only)
PCI1, PCI2: PCI Slot
PCIE1: x16 PCI Express Slot
PCIE2: x1 PCI Express Slot

ATX1: 24-pin ATX Power Connector
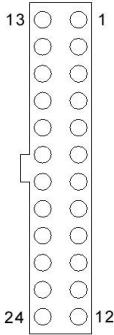
Signal Name	Pin #	Pin #	Signal Name
3.3V	13	1	3.3V
-12V	14	2	3.3V
Ground	15	3	Ground
PS-ON	16	4	+5V
Ground	17	5	Ground
Ground	18	6	+5V
Ground	19	7	Ground
-5V	20	8	Power good
+5V	21	9	5VSB
+5V	22	10	+12V
+5V	23	11	+12V
Ground	24	12	+3.3V

ATX2: ATX 12V Power Connector

This connector supplies the CPU operating voltage.
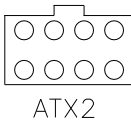
Signal Name	Pin #	Pin #	Signal Name
+12V	5	1	Ground
+12V	6	2	Ground
+12V	7	3	Ground
+12V	8	4	Ground

DDRIII1: Channel A DDR3 Socket

DDRIII2: Channel B DDR3 Socket

FAN2: CPU Fan Power Connector
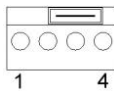
Pin #	Signal Name
1	Ground
2	+12V
3	Rotation detection
4	Control

FAN1: System Fan1 Power Connector
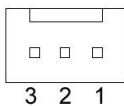
Pin #	Signal Name
1	Ground
2	+12V

3	Rotation detection

FAN3: System Fan2 Power Connector
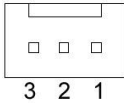
Pin #	Signal Name
1	Ground
2	+12V
3	NC

CN1: PS/2 Keyboard and PS/2 Mouse Connectors
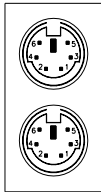
Mouse (top)
Keyboard (bottom)

Keyboard Signal	Pin #	Mouse Signal
Keyboard data	1	Mouse data
N.C.	2	N.C.
GND	3	GND
5V	4	5V
Keyboard clock	5	Mouse clock
N.C.	6	N.C.

CN2: COM1 RS232/RS422/RS485 Connector
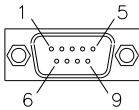
Pin #	Signal Name		
	RS-232	R2-422	RS-485
1	DCD	TX-	DATA-
2	RX	TX+	DATA+
3	TX	RX+	NC
4	DTR	RX-	NC
5	Ground	Ground	Ground
6	DSR	NC	NC
7	RTS	NC	NC
8	CTS	NC	NC
9	RI	NC	NC
10	NC	NC	NC

CN3: Parallel Port Connector

CN3 is a DB-25 external connector on top of the VGA and serial ports.
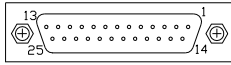
 CN3 Parallel Port

Signal Name	Pin #	Pin #	Signal Name
Line printer strobe	1	14	AutoFeed
PD0, parallel data 0	2	15	Error
PD1, parallel data 1	3	16	Initialize
PD2, parallel data 2	4	17	Slin
PD3, parallel data 3	5	18	Ground
PD4, parallel data 4	6	19	Ground
PD5, parallel data 5	7	20	Ground
PD6, parallel data 6	8	21	Ground
PD7, parallel data 7	9	22	Ground
ACK, acknowledge	10	23	Ground
Busy	11	24	Ground
Paper empty	12	25	Ground
Select	13	N/A	N/A

CN4: VGA CRT Connector

CN4 is a DB-15 VGA connector. The following table shows the pin-out assignments of this connector.
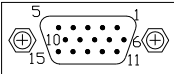
Signal Name	Pin #	Pin #	Signal Name
Red	1	2	Green
Blue	3	4	N.C.
GND	5	6	GND
GND	7	8	GND
VCC	9	10	GND
N.C.	11	12	DDCDATA
HSYNC	13	14	VSYNC
DDCCLK	15		

CN5: Audio Connector

Orange/Surr Rear	⊙ ⊙	Blue/Line In
Black/Surround	⊙ ⊙	Lime/Line out
Gray/CEN/LFE	⊙ ⊙	Pink/Microphone
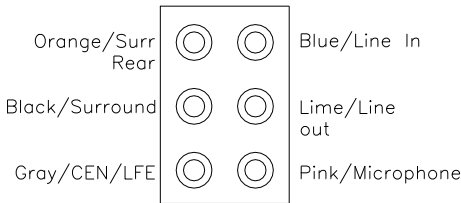
CN6: Gigabit RJ-45 &USB 0/1 Connector

CN7: Gigabit RJ-45 &USB 2/3 Connector

CN8, CN9, CN10, CN11: SATA HDD Connectors
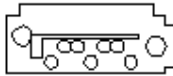
Pin #	Signal Name
1	Ground
2	TX+
3	TX-
4	Ground
5	RX-
6	RX+
7	Ground

J1: COM2/RS232 Serial Port
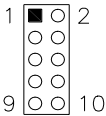
Signal Name	Pin #	Pin #	Signal Name
DCD, Data carrier detect	1	2	RXD, Receive data
TXD, Transmit data	3	4	DTR, Data terminal ready
GND, ground	5	6	DSR, Data set ready
RTS, Request to send	7	8	CTS, Clear to send
RI, Ring indicator	9	10	Not Used
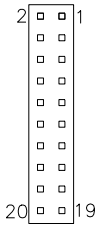
J2: For LPC I/F Adaptor Card

J3: Digital I/O Connector (4 in, 4 out)

J3 supports TTL levels and is used to control external devices requiring ON/OFF circuitry.

Signal Name	Pin #	Pin #	Signal Name
Ground	1	2	+5V
Out3	3	4	Out1
Out2	5	6	Out0
IN3	7	8	IN1
IN2	9	10	IN0
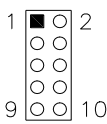
J4: COM3, COM4 Serial Port (DF11 Connector)

Signal Name	Pin #	Pin #	Signal Name
DSR3	2	1	DCD3
RTS3	4	3	RXD3
CTS3	6	5	TXD3
RI3	8	7	DTR3
NC	10	9	Ground
DSR4	12	11	DCD4
RTS4	14	13	RXD4
CTS4	16	15	TXD4
RI4	18	17	DTR4
NC	20	19	Ground

J5: Compact Flash Socket

J6: Audio CD-In Connector

Pin #	Signal Name
1	CD Audio L
2	Ground
3	Ground
4	CD Audio R

J7: Audio Front Header

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

J8: System Function Connector

J8 provides connectors for system indicators that provide light indication of the computer activities and switches to change the computer status. J8 is a 20-pin header that provides interfaces for the following functions.

Hard Disk Drive LED

Not Defined

Not Defined

Speaker

Reset Switch

ATX Power On Switch
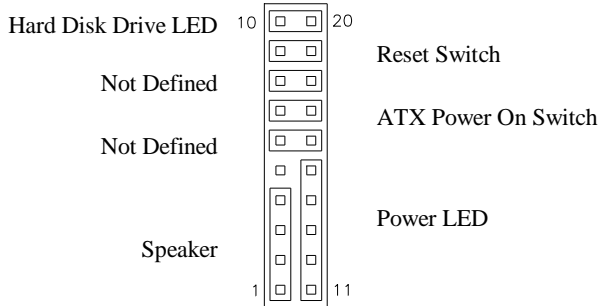
Power LED

Speaker: Pins 1 - 4

This connector provides an interface to a speaker for audio tone generation. An 8-ohm speaker is recommended.
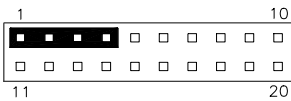
Pin #	Signal Name
1	Speaker out
2	No connect
3	Ground
4	+5V

Power LED: Pins 11 - 15

The power LED indicates the status of the main power switch.
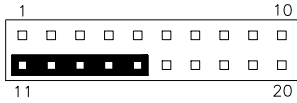
Pin #	Signal Name
11	Power LED
12	No connect
13	Ground
14	No connect
15	Ground

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.
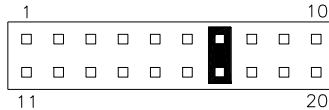
Reset Switch: Pins 9 and 19

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.
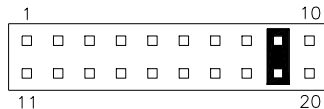
Hard Disk Drive LED Connector: Pins 10 and 20

This connector connects to the hard drive activity LED on control panel. This LED will flash when the HDD is being accessed.
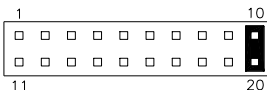
Pin #	Signal Name
10	HDD Active
20	5V

J9: USB4/USB5 Connector
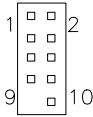
Signal Name	Pin	Pin	Signal Name
Vcc	1	2	Vcc
D-	3	4	D-
D+	5	6	D+
Ground	7	8	Ground
Protect Pin	9	10	NC

J10: USB6/USB7 Connector
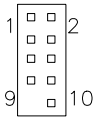
Signal Name	Pin	Pin	Signal Name
Vcc	1	2	Vcc
D-	3	4	D-
D+	5	6	D+
Ground	7	8	Ground
Protect Pin	9	10	NC

J11: SPI Debug Tools Port (Factory use only)

PCI1, PCI2: PCI Slot

PCIE1: x16 PCI Express Slot

PCIE2: x1 PCI Express Slot

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 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 view processor configuration used in your computer system and set the system time and date.

BIOS SETUP UTILITY

Main	Advanced	PCIPnP	Boot	Security	Chipset	Exit

System Overview

Processor
Intel(R) Core(TM) 2 cpu   6400 @ 2.13GHz
Speed   : 2133MHz
Count   : 1

System Memory
Size   : 2014MB

System Time   [02:29:50]
System Date   [Fri 01/02/2009]

Use[ENTER], [TAB]
or [SHIFT-TAB] to
select a field.

Use [+] or [-] to
configure system Time.

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

Note: *If the system cannot boot after making and saving system changes with Setup, the AMI 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 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.*

Advanced Settings

Main	**Advanced**	PCIPnP	Boot	Security	Chipset	Exit

Advanced Settings

Configure CPU.

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

► **CPU Configurations**
► IDE Configuration
► SuperIO Configuration
► Hardware Health Confguration
► ACPI Configuration
► APM Configuration
► EUP Power Saving Controller
► MPS Configuration
► Smbios Configuration
► USB Configuration

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

The Advanced BIOS Settings contains the following sections:
► CPU Configurations
► IDE Configuration
► SuperIO Configuration
► Hardware Health Confguration
► ACPI Configuration
► APM Configuration
► EUP Power Saving Controller
► MPS Configuration
► Smbios 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.

BIOS SETUP UTILITY

Main	Advanced	PCIPnP	Boot	Security	Chipset	Exit

Configure advanced CPU settings

Manufacturer: Intel

Intel® Core(TM)2 Duo CPU   6400 @ 2.13GHz

Frequency   : 2.13GHz

FSB Speed   : 1068MHz

Cache L1 : 64KB

Cache L2 : 2048KB

Ratio Actual Value: 8

Max CPU ID Value Limit	[Disabled]
Intel(R) Virtualization Tech	[Enabled]
Execute-Disabled Bit Capability	[Enabled]
Core Multi-Processing	[Enabled]
PECI	[Enabled]

Configure CPU.

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

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

		Options:
ATA/IDE Configuration	[Compatible]	**Disabled**
Legacy IDE Channels	[SATA Pri, PATA Sec]	**Compatible**
		Enhanced
► Primary IDE Master	: [Not Detected]	
► Primary Slave Master	: [Not Detected]	
► Secondary IDE Master	: [Hitachi HDS72]	<- Select Screen
► Secondary IDE Slave	: [Not Detected]	↑ ↓ Select Item
		+- Change Field
IDE Detect Time Out (Sec)	[35]	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.

BIOS SETUP UTILITY

Main	**Advanced**	PCIPnP	Boot	Security	Chipset	Exit

Configure Win627UHG Super IO Chipset

Serial Port1 Address	**[3F8/IRQ4]**	
Serial Port2 Address	[2F8/IRQ3]	
Serial Port2 Mode	[Normal]	
Serial Port3 Address	[3E8]	
Serial Port3 IRQ Select	[IRQ11]	<- Select Screen
Serial Port4 Address	[2E8]	↑ ↓ Select Item
Serial Port4 IRQ Select	[IRQ10]	+- Change Field
Parallel Port Address	[378]	Tab Select Field
Parallel Port Mode	[Normal]	F1 General Help
Parallel Port IRQ	[IRQ7]	F10 Save and Exit
Restore on AC Power Loss	[Power Off]	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	3E0/IRQ10
Serial Port 4	Disabled
Parallel Port	378/IRQ7

Parallel Port Mode

This field allows you to determine parallel port mode function.

SPP	Standard Printer Port
EPP	Enhanced Parallel Port
ECP	Extended Capabilities Port
ECP+EPP	Combination of ECP and EPP capabilities
Normal	Normal function

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.

BIOS SETUP UTILITY

Main	Advanced	PCIPnP	Boot	Security	Chipset	Exit

Hardware Health Configuration		Configure CPU.
System Temperature	:35°C/95°F	
CPU Temperature	:34°C/93°F	
VTIN Temperature	:37°C/98°F	
	:	
CPUFAN0 Speed	:2280 RPM	
Vcore(V)	:1.296 V	<- Select Screen
+5VS	: 5.304 V	↑ ↓ Select Item
+3SV	:3.178 V	+- Change Field
12V	:11.904 V	Tab Select Field
3.3V	: 3.360V	F1 General Help
1.5V	: 1.552V	F10 Save and Exit
VBAT (V)	:3.21 V	ESC Exit
SYSFAN Mode Setting	[Manual Mode]	
SYSTEM PWM Control	[256]	
CPUFAN0 Mode Setting	[Manual Mode]	
CPUFAN0 PWM Control	[256]	
ACPI Shutdown Temperature	**[Disabled]**	

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

ACPI Shutdown Temperature
The system will shut down automatically under OS with ACPI mode, when the CPU temperature reaches the configured temperature.

BIOS SETUP UTILITY

Main	Advanced	PCIPnP	Boot	Security	Chipset	Exit

ACPI Settings		General ACPI Configuration settings
►General ACPI Configuration		
		<- 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

General ACPI Configuration	**General ACPI Configuration settings**
Suspend mode [S3 (STR)]	
Repost Video on S3 Resume [No]	
	`<-  Select Screen`
	`↑↓ Select Item`
	`+-  Change Field`
	`Tab Select Field`
	`F1  General Help`
	`F10 Save and Exit`
	`ESC Exit`

Suspend Mode

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

Repost Video on S3 Resume

The default setting is *No*.

BIOS SETUP UTILITY

Main	Advanced	PCIPnP	Boot	Security	Chipset	Exit

```
 APM Configuration                                Disable/Enable
                                                  RI to generate
 Power Management/APM           [Disabled]        a wake event.
 Power Button Mode             [On/Off]

 Resume On Ring                 Disabled
 Resume On PME#                 Disabled          <-  Select Screen
 Resume On RTC Alarm            Disabled          ↑↓  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

BIOS SETUP UTILITY

Main	Advanced	PCIPnP	Boot	Security	Chipset	Exit

MPS Configuration

MPS Revision VT-d   [1.1]

Select MPS Revision

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

MPS Version Control for OS

This option is specifies the MPS (Multiprocessor Specification) version for your operating system.

BIOS SETUP UTILITY

Main	Advanced	PCIPnP	Boot	Security	Chipset	Exit

SMBIOS Configuration

Smbios SMI Support   [Enabled]

SMBIOS SMI Wrapper support for PnP Func 50h-54h

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

SMBIOS SMI Support

Use the SMBIOS SMI Support to enable the system to support the SMBIOS SMI wrapper for the PnP function 50h – 54h.

BIOS SETUP UTILITY

Main	Advanced	PCIPnP	Boot	Security	Chipset	Exit

USB Configuration

USB Devices Enabled:
  None

Legacy USB Support   [Enabled]
USB 2.0 Controller Mode   [HiSpeed]
BIOS EHCI Hand-Off   [Enabled]

Enables support for
legacy USB. AUTO
option disables
legacy support if
no USB devices are
connected.

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

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
This option is used to control USB device in HiSpeed (480Mbps) or FullSpeed (12Mbps).This option is enabled by HiSpeed.

BIOS EHCI Hand-Off
This feature allows you to enable support for operating system without EHCI hand-off feature.

PCIPnP Settings

This option configures the PCI/PnP settings.

BIOS SETUP UTILITY

Main	Advanced	**PCIPnP**	Boot	Security	Chipset	Exit

Advanced PCI/PnP Settings		Size of memory block to reserve for legacy ISA devices.
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]	
Off Board PCI?ISA IDE Card	[Auto]	
IRQ3	[Available]	
IRQ4	[Available]	
IRQ5	[Available]	
IRQ7	[Available]	
IRQ9	[Available]	
IRQ10	[Available]	
IRQ11	[Available]	
IRQ14	[Available]	<- Select Screen
IRQ15	[Available]	↑↓ Select Item
		+- Change Field
DMA Channel 0	[Available]	Tab Select Field
DMA Channel 1	[Available]	F1 General Help
DMA Channel 3	[Available]	F10 Save and Exit
DMA Channel 5	[Available]	ESC Exit
DMA Channel 6	[Available]	
DMA Channel 7	[Available]	
Reserved Memory Size	[Disabled]	

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.

OffBoard PCI/ISA IDE Card

Some PCI IDE cards may require this to be set to the PCI slot number that is holding the card.

IRQ#

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

Boot Settings

Main	Advanced	PCIPnP	**Boot**	Security	Chipset	Exit

Boot Settings Configuration

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]

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

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

Quick Boot
This allows BIOS to skp 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 ROM 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.

BIOS SETUP UTILITY

Main	Advanced	PCIPnP	Boot	**Security**	Chipset	Exit

Security Settings

Supervisor Password   : Not Installed
User Password   : Not Installed

Change Supervisor Password
Change User Password

Boot Sector Virus Protection   [Disabled]

Install or Change the Password.

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

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. -

Main	Advanced	PCIPnP	Boot	Security	**Chipset**	Exit

Advanced Chipset Settings

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

► North Bridge Configuration
► South Bridge Configuration

Configure North Bridge features.

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

Main	Advanced	PCIPnP	Boot	Security	**Chipset**	Exit

North Bridge Chipset Configuration

Memory Remap Feature	[Enabled]
PCI MMIO Allocation: 4GB To 3072MB	
DRAM Frequency	[Auto]
Configure DRAM Timing by SPD	[Enabled]
Memory Hole	[Disabled]
Initiate Graphics Adapter	[PEG/PCI]
IGD Graphics Mode Select	[Enabled, 32MB]
IGD GTT Graphics memory size	[No VT mode, 2MB]
PAVP Mode	[Lite]
PEG Port Configuration	
PEG Port	[Auto]
► Video Function Configuration	

ENABLE: Allow Remapping of Overlapped PCI memory above the total physical memory.

DISABLE: Do not allow remapping of memory

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

Memory Remap Feature
This feature allows remapping of overlapped PCI memory above the total physical memory.

DRAM Frequency
This option is, by default, set to Auto.

Configure DRAM Timing by SPD

When this item is enabled, the DRAM timing parameters are set according to the DRAM SPD (Serial Presence Detect). When disabled, you can manually set the DRAM timing parameters through the DRAM sub-items.

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.

Initiate Graphic Adapter

This option, by default, is set to PEG/PCI.

IGD Graphics Mode Select

Use the feature to set the amount of system memory to be used by the Internal. Graphics Devices. expansion cards that require a specified area of memory to work properly.

IGD GTT Graphics memory size

This feature allows the user to select the IGD GTT Graphics Size. The Default setting is No **VT** Mode, 2 MB

PAVP Mode

Use the feature to select the Protect Audio Video Path Mode.

Video Function Configuration

The configuration allows setting to DVMT/FIXED memory.

```
Main      Advanced      PCIPnP      Boot      Security      Chipset      Exit

  Video Function Configuration                 DVMT Mode

  DVMT Mode Select         [DVMT Mode]
   DVMT/FIXED Memory        [256MB]

                                              <-  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

South Bridge Chipset Configuration

USB Function	[6 USB Ports]	Disabled
USB 2.0 Controller	[Disabled]	2 USB Ports
Audio Controller	[Auto]	4 USB Ports
SMBUS Controller	[Enabled]	6 USB Ports

SLP_S4# Min. Assertion Width [4 to 5 seconds]

Enabled Onboard PCI option ROM [Disabled]

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

USB Function
This option enables the number of USB ports desired or disables the USB function.

USB 2.0 Controller
This option is disabled by default.

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

Exit Setup

The exit setup has the following settings which are:

BIOS SETUP UTILITY

Main	Advanced	PCIPnP	Boot	Security	Chipset	**Exit**

Exit Options

Save Changes and Exit
Discard Changes and Exit
Discard Changes

Load Optimal Defaults
Load Failsafe Defaults

Exit system setup after saving the changes.

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

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 the Windows XP and Windows Vista. 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/ Vista), 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 under Windows XP/Vista.

1. Insert the CD that comes with the board. Click *Intel* at the left side, then *Intel(R) G41 Chipset Drivers*.

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

3. When the welcome screen to the Intel(R) Chipset Software Installation
Utility appears, click **Next** to continue.

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

5. On Readme 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.

Intel G41 Chipset Family Graphics Driver Installation

To install the VGA drivers, follow the steps below to proceed with the installation.

1. Insert the CD that comes with the board. Click *Intel* at the left side, then *Intel(R) G41 Chipset Drivers*.

2. Click *Intel G41 Chipset Family Graphics Driver.*
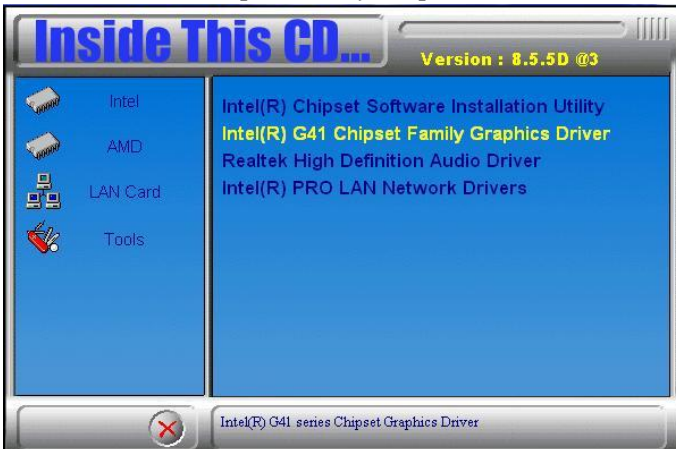
3. When the welcome screen of the Intel(R) Graphics Media Accelerator Driver appears, click *Next* to continue.

4. Click *Yes* to to agree with the license agreement and continue the installation.
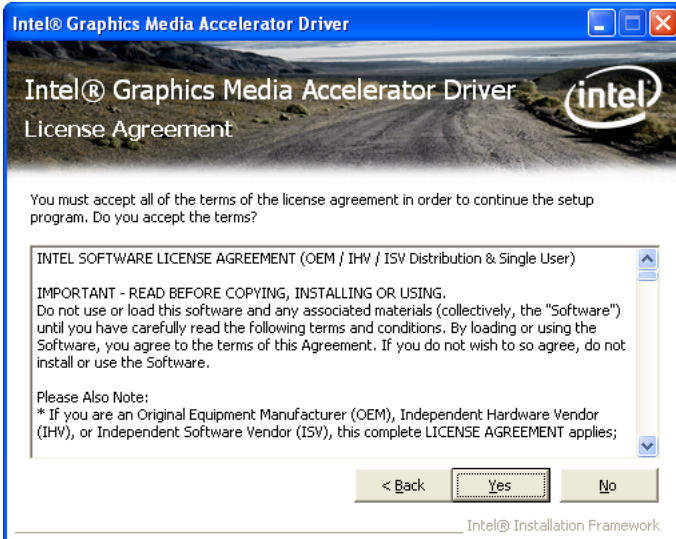
5. Click *Next* in the Readme File Information window.
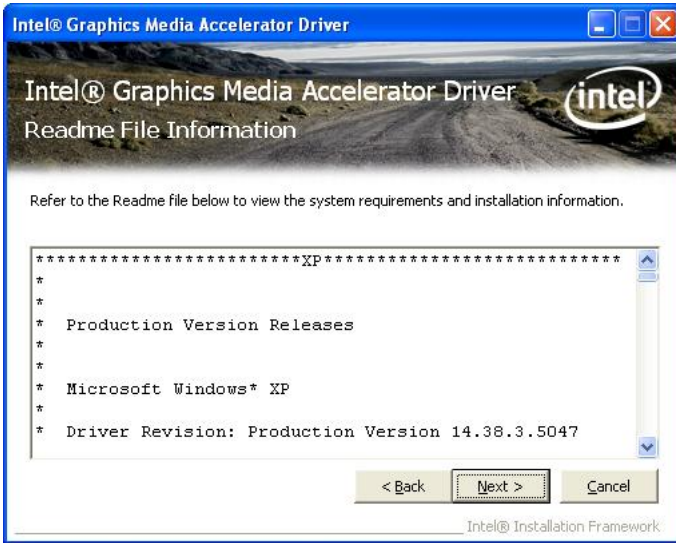
6. Click *Next* in the Setup Progress window.
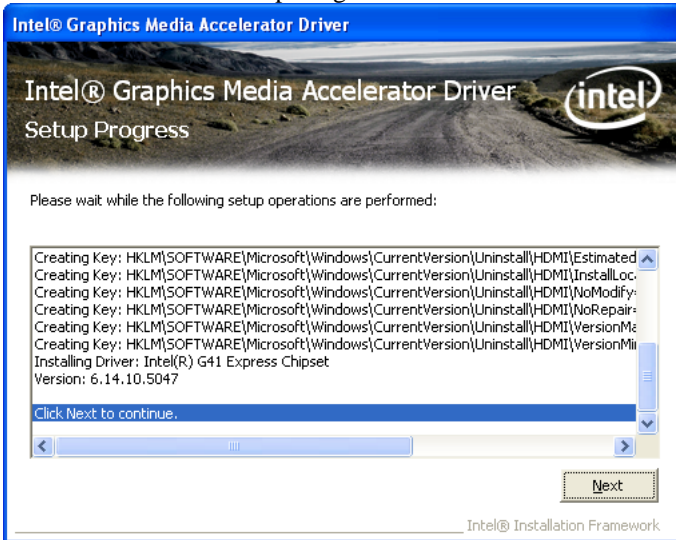
7. Setup is now complete. Click *Finish* to restart the computer and for changes to take effect.

Realtek High Definition Audio Driver Installation

Follow the steps below to install the Realtek HD Codec Audio Drivers.

1. Insert the CD that comes with the board. Click *Intel* and then *Intel(R) G41 Chipset Drivers*.

2. Click *Realtek High Definition Audio Driver.*
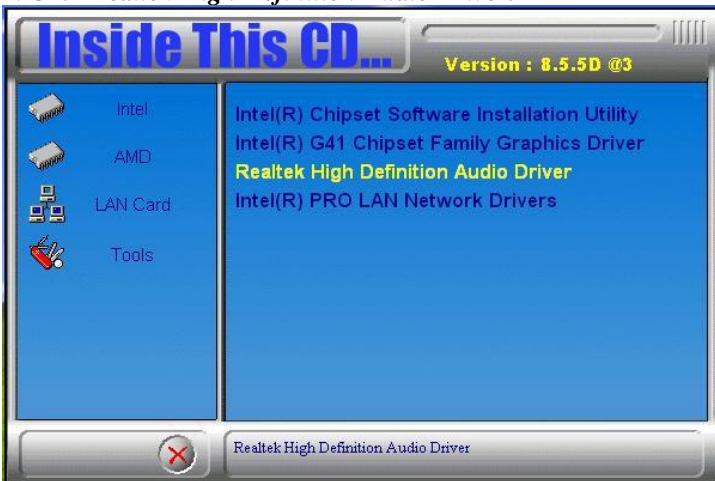
3.When the welcome screen to InstallShield Wizard for **Realtek High Definition Audio Driver** appears, click **Next** to start the installation.
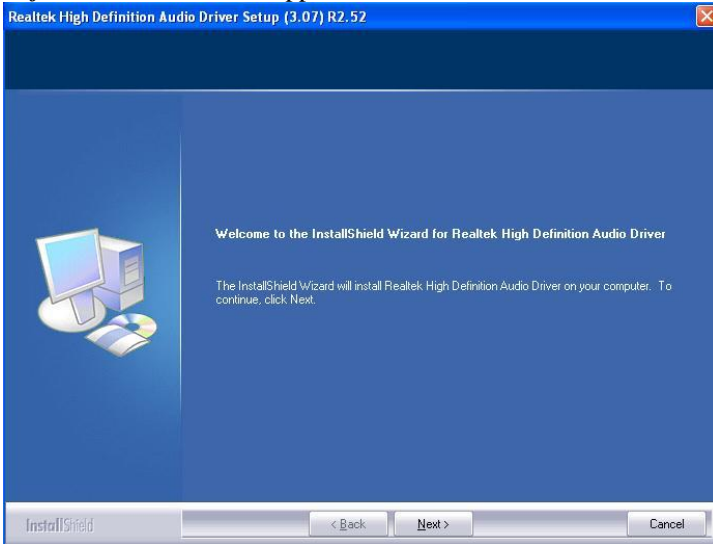
4.When the InstallShieldWizard has finished performing maintenance operations on Realtek High Definition Codec Audio Audio Driver, click **Finish** to restart the computer.
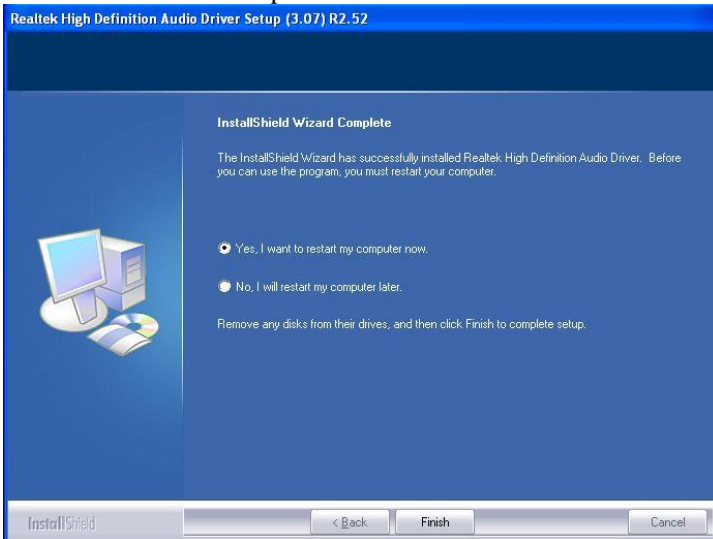
Realtek LAN Controller Drivers Installation

Follow the steps below to install the Realtek LAN Drivers.

1. Insert the CD that comes with the board. Click *Intel*, then *LAN Card,* and then *Realtek Lan Controller Drivers*.

2. Click *Realtek RTL8111E LAN Drivers.*

3.When the welcome screen to InstallShield Wizard appears, click *Next* to start the installation.
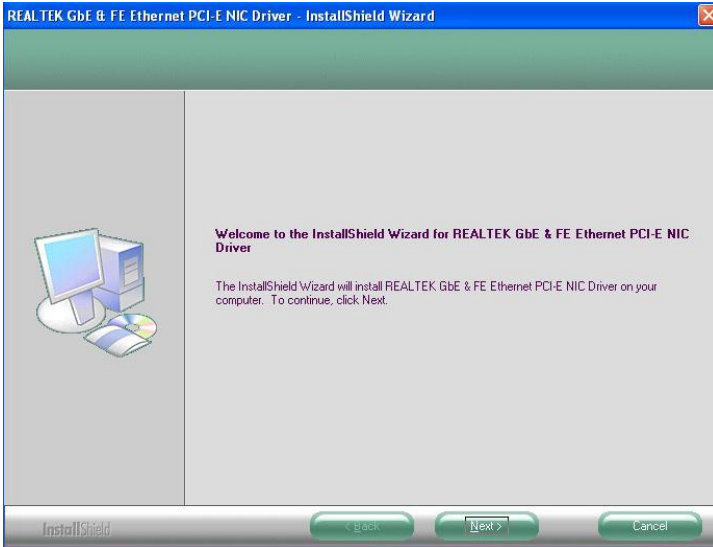
4.When the InstallShieldWizard has finished installing the Realtek LAN drivers, click *Finish*.
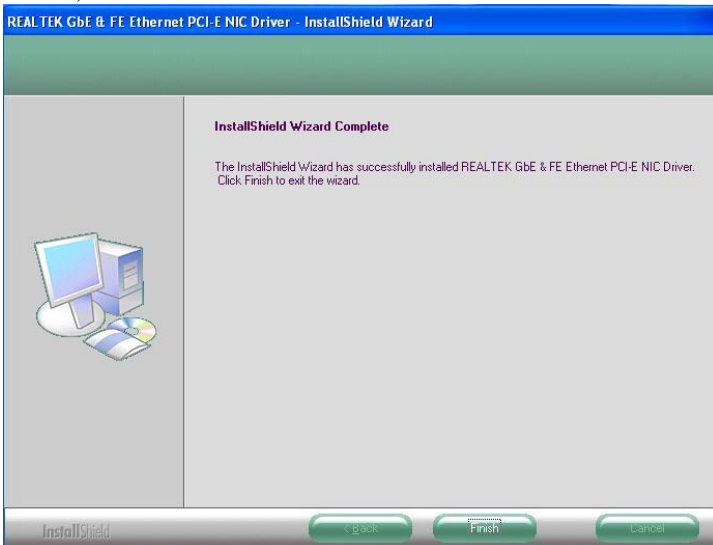
Appendix

A. I/O Port Address Map

Each peripheral device in the system is assigned a set of I/O port addresses that 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)
2F8h - 2FFh	Serial Port #2(COM2)
2B0h - 2DFh	Graphics adapter Controller
378h - 3FFh	Parallel Port #1(LPT1)
360h - 36Fh	Network Ports
3B0h - 3BFh	Monochrome & Printer adapter
3C0h - 3CFh	EGA adapter
3D0h - 3DFh	CGA adapter
3F0h - 3F7h	Floppy Disk Controller
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	Parallel Port #1
IRQ8	Real Time Clock
IRQ9	Reserved
IRQ10	Serial Port #4
IRQ11	Serial Port #3
IRQ12	PS/2 Mouse
IRQ13	80287
IRQ14	Primary 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);
}
//------------------------------------------------------------------------
```