IB945

LGA775 Core™ 2 Duo
Intel® Q45 Chipset
Full Size CPU Card

USER'S MANUAL

Version 1.0A

Acknowledgments

Award is a registered trademark of Award Software International, Inc.

PS/2 is a trademark of International Business Machines Corporation.

Intel and Core 2 Duo are registered trademarks of Intel Corporation.

Microsoft Windows is a registered trademark of Microsoft Corporation.

Winbond is a registered trademark of Winbond Electronics Corporation.
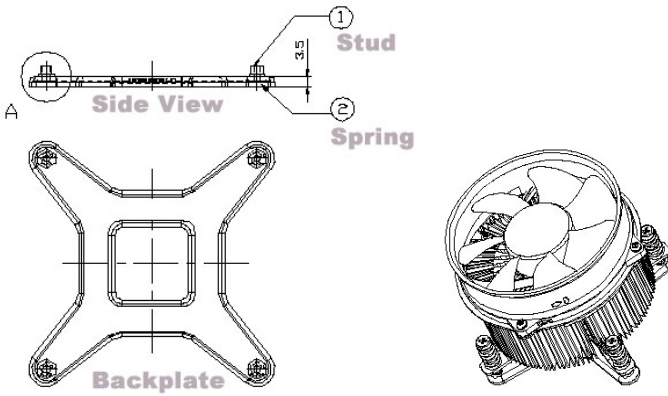
All other product names or trademarks are properties of their respective owners.

Table of Contents

ATTENTION:

It is strongly recommended that only heatsink that have corresponding (metal) backplate be used on the CPU card. This is to avoid the CPU card being bent/distorted, causing the CPU card to become damaged. A reference picture of a backplate and heatsink that has backplate are shown below.

Introduction

Specifications

Model Name	IB945 (Full Size CPU Card)
CPU Type Support	Intel® Core™2 Extreme processor QX9000 series Intel® Core™2 Quad processor Q9000 series, Intel® Core™2 Duo processor E8000 and E7000 series Intel® Pentium® Dual Core, Intel® Celeron® Dual Core
CPU Speed	Up to 3.0 GHz
CPU FSB	800/1066/1333MHz FSB
Green /APM	APM1.2
BIOS	Award BIOS, support ACPI Function
CPU Socket	LGA775
Chipset	Intel Eaglelake:Q45 + ICH10DO Chipset Q45 GMCH: AC82Q45, 34mm x 34mm, 1254-pin FC-BGA ICH10DO: AF82801JDO, 31mm x 31mm, 676-pin mBGA
Memory	DDRIII 800/1066MHz DIMM x 4 (w/o ECC), Max. 8GB
VGA	Intel Q45 GMCH integrated graphics device (GMA4500) Supports Direct X10 (Vista) & OpenGL 2.1
DVI	Chrontel CH7307C x1 for DVI
LVDS	N/A
LAN	1. ICH10DO Gigabit MAC + Intel Gigabit PHY: Intel 82567 x1 2. Intel 82574L PCI-e Gigabit LAN controller x1
USB	ICH10DO built-in USB 2.0 host controller, support 12 ports
Serial ATA Ports	ICH10DO built-in SATA controller, supports 6 ports
Parallel IDE	JMicron JM368 (PCI-e to PATA) x1 for 1 PATA channel
PCI-to-ISA bridge	ITE IT8888G x1 for high drive ISA bus
Audio	Intel ICH10DO built-in high definition audio w/ Realtek ALC888 Codec supports 5.1 CH audio (line-out, line-in & mic)
LPC I/O	Winbond 83627DHG COM1(RS232),COM2(RS232/422/485), IrDAx1, Parallel x 1, Floppy x 1; Hardware monitor (3 thermal inputs, 4 voltage monitor inputs, VID0-4 & 2 Fan Headers)
Digital IO	4 in & 4 out
Keyboard/Mouse	Supports PS/2 Keyboard/Mouse Connector
Expansion Slots	Mini PCI-express socket x1
Edge Connector	PS/2 Connector x1 for keyboard/mouse DB15 x1 for VGA, RJ45 x2 for LAN 1, 2
On Board Header/Connector	Standard SATA (7-pin shrouded vertical) connector x4 40-pin, 0.1" pitch, standard box-header x1 for IDE DF13-20 header x2 for LVDS, DF11-20 header x1 for DVI 34-pin box-header x1 for FDD, DF11 box-header x1 for printer DF11 10x2 box-header x2 for COM1-2 4-pin header x1 for CPU fan (PWM smart fan) 3-pin header x1 for system fan 4x2 header x3 for USB1-6, IrDA SATA connector x 6 ports, RAID function (0,1,10,5)
Watchdog Timer	Yes (256 segments, 0, 1, 2…255 sec/min)
System Voltage	+5V, +3.3V, +12V, -12V & 5VSB
Board Size	338mm x 122mm

Checklist

Your IB945 Core 2 Duo CPU Card package should include the items listed below:

- The IB945 Card
- This User's manual
- 1 x IDE cable
- 1 x Floppy cable
- 1 x SATA cable
- 1 x TMDS cable
- 2 Serial Port Ribbon Cable and 1 Parallel Port Attached to a Mounting Bracket
- 1 Y-Cable supporting a PS/2 Keyboard and a PS/2 Mouse
- Audio cable with bracket (Audio-18K)
- USB cable with bracket (USB2K-4)
- 1 CD containing the following:
  - Chipset Drivers
  - Flash Memory Utility
- Optional SATA Power Cord (PW34)

Board Dimensions
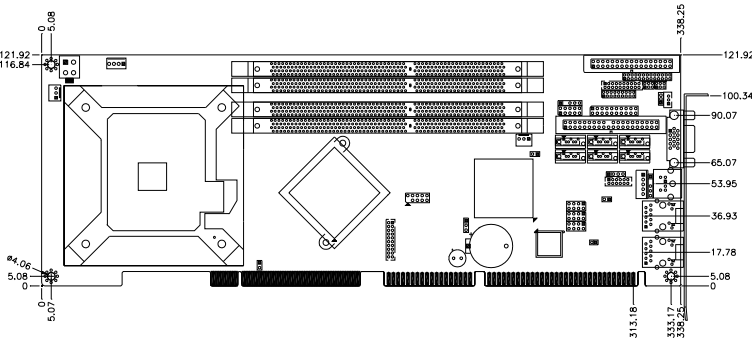
Installations

This section provides information on how to use the jumpers and connectors on the IB945 in order to set up a workable system. The topics covered are:

Installing the CPU

The IB945 CPU Card 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.*
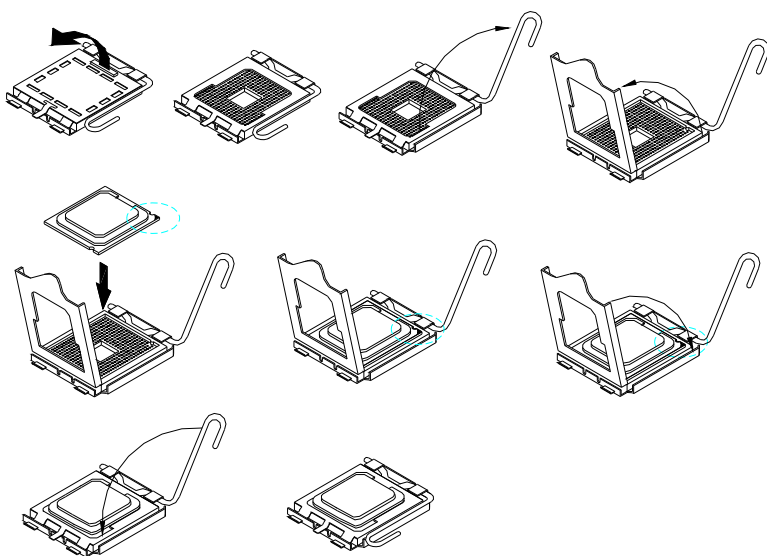
IMPORTANT: Ensure that the power supply is OFF during CPU and memory module installation.

Installing the Memory

The IB945 CPU Card supports four DDR3 memory sockets for a maximum total memory of 8GB. It supports DDRIII 800/1066.

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 align 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 DDR module touches the bottom of the slot.

3. To remove the DDR3 module, press the clips with both hands.
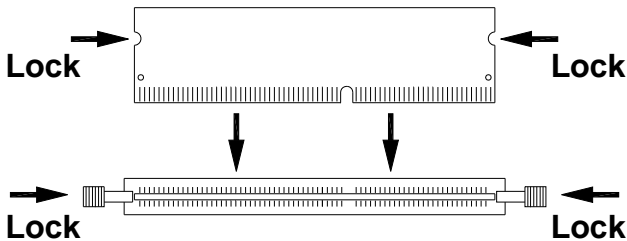
IMPORTANT: Ensure that the power supply is OFF during CPU and memory module installation.

Setting the Jumpers

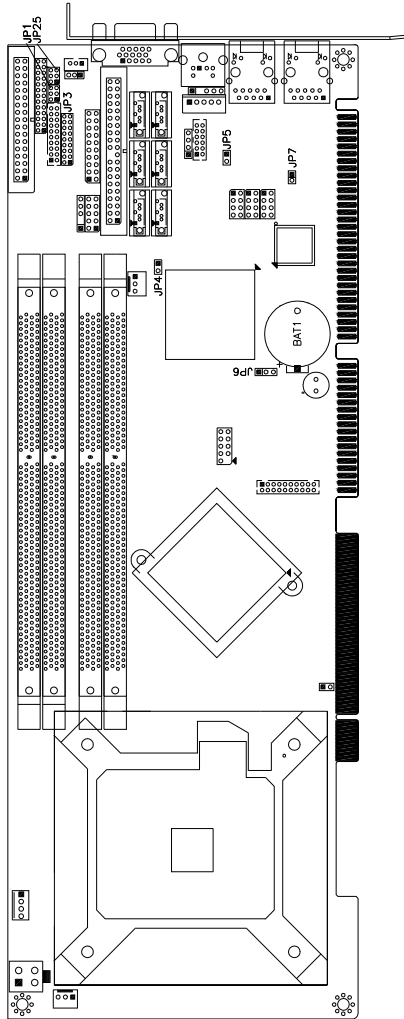Jumpers are used on IB945 to select various settings and features according to your needs and applications. Contact your supplier if you have doubts about the best configuration for your needs. The following lists the connectors on IB945 and their respective functions.
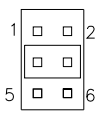
Jumper Locations on IB945

JP1/JP2 COM2/COM1 RS232 +5V/+12V Power Setting

JP1/JP2	Setting	Function
1 □ □ 2 □ □ 5 □ □ 6	Pin 1-2 Short/Closed	+12V
	Pin 3-4 Short/Closed	Normal
	Pin 5-6 Short/Closed	+5V

JP3: 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.
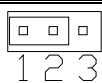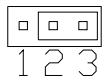
COM2 Function	RS-232	RS-422	RS-485
Jumper Setting (pin closed)	Short: 1-2 9-11 10-12 15-17 16-18	Short: 3-4 7-9 8-10 13-15 14-16	Short: 5-6 7-9 8-10 13-15 14-16

Note: Factory is default RS-232

J4: Power Supply Type Select

Use J4, a 3-pin header, to select between AT and ATX power supply.

J4	Setting	Power Supply Type
□ □ □ 1 2 3	Pin 1-2 Short/Closed	ATX
□ □ □ 1 2 3	Pin 2-3 Short/Closed	AT

JP6: Clear CMOS Contents

Use JP6, a 3-pin header, to clear the CMOS contents. *Note that the ATX-power connector should be disconnected from the motherboard before clearing CMOS.*

JP6	Setting	Function
□ □ □ 1 2 3	Pin 1-2 Short/Closed	Normal
□ □ □ 1 2 3	Pin 2-3 Short/Closed	Clear CMOS

Connectors on IB945

The connectors on IB945 allows you to connect external devices such as keyboard, floppy disk drives, hard disk drives, printers, etc. The following table lists the connectors on IB945 and their respective functions.

Connector Locations on IB945

ATX1: ATX 12V/+12V Power Connector
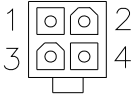
Pin #	Signal Name
1	Ground
2	Ground
3	+12V
4	+12V

IDE1: Primary IDE Connectors
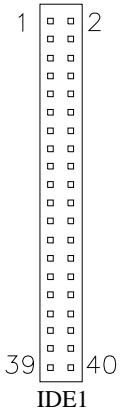
Signal Name	Pin #	Pin #	Signal Name
Reset IDE	1	2	Ground
Host data 7	3	4	Host data 8
Host data 6	5	6	Host data 9
Host data 5	7	8	Host data 10
Host data 4	9	10	Host data 11
Host data 3	11	12	Host data 12
Host data 2	13	14	Host data 13
Host data 1	15	16	Host data 14
Host data 0	17	18	Host data 15
Ground	19	20	Protect pin
DRQ0	21	22	Ground
Host IOW	23	24	Ground
Host IOR	25	26	Ground
IOCHRDY	27	28	Host ALE
DACK0	29	30	Ground
IRQ14	31	32	No connect
Address 1	33	34	No connect
Address 0	35	36	Address 2
Chip select 0	37	38	Chip select 1
Activity	39	40	Ground

Note: The CompactFlash interface cannot be used simultaneously with the IDE interface.

SATA1~SATA6: SATAII Connectors

Pin #	Signal Name
1	Ground
2	TX+
3	TX-
4	Ground
5	RX-
6	RX+
7	Ground

CPU_FAN1: CPU Fan Power Connector
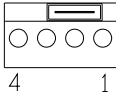
Pin #	Signal Name
1	Ground
2	+12V
3	Rotation detection
4	Control

FAN1: SYSTEM Fan Power Connector
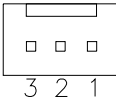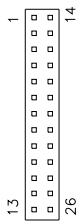
Pin #	Signal Name
1	Ground
2	+12V
3	Rotation detection

FDD1: Floppy Drive Connector
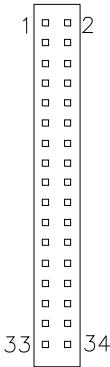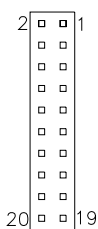
Signal Name	Pin #	Pin #	Signal Name
Ground	1	2	RM/LC
Ground	3	4	No connect
Ground	5	6	No connect
Ground	7	8	Index
Ground	9	10	Motor enable 0
Ground	11	12	Drive select 1
Ground	13	14	Drive select 0
Ground	15	16	Motor enable 1
Ground	17	18	Direction
Ground	19	20	Step
Ground	21	22	Write data
Ground	23	24	Write gate
Ground	25	26	Track 00
Ground	27	28	Write protect
Ground	29	30	Read data
Ground	31	32	Side 1 select
Ground	33	34	Diskette change

J1: Parallel Port Connector
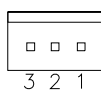
Signal Name	Pin #	Pin #	Signal Name
Line printer strobe	1	14	AutoFeed
PD0, parallel data 0	2	15	Error
PD1, parallel data 1	3	16	Initialize
PD2, parallel data 2	4	17	Select
PD3, parallel data 3	5	18	Ground
PD4, parallel data 4	6	19	Ground
PD5, parallel data 5	7	20	Ground
PD6, parallel data 6	8	21	Ground
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

J2: COM1, COM2 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

J3: External ATX Power Connector

Pin #	Signal Name
1	Ground
2	PS-ON (soft on/off)
3	5VSB (Standby +5V)

J7: IrDA Connector
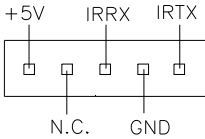
Pin #	Signal Name
1	+5V
2	No connect
3	Ir RX
4	Ground
5	Ir TX

J8: Digital I/O Connector (4 in, 4 out)

This 10-pin digital I/O connector 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

J9: System Function Connector

J9 provides connectors for system indicators that provide light indication of the computer activities and switches to change the computer status.
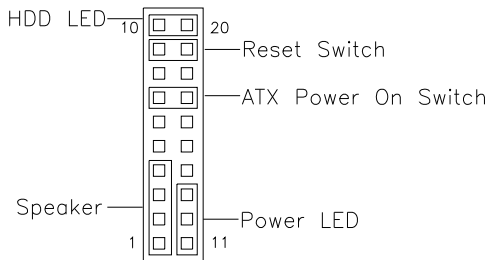
Speaker: Pins 1 - 4

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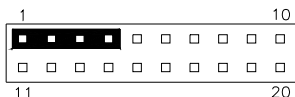
Pin #	Signal Name
1	Speaker out
2	No connect
3	Ground
4	+5V

Power LED: Pins 11 - 13

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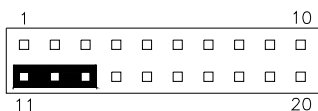
Pin #	Signal Name
11	Power LED
12	No connect
13	Ground

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.
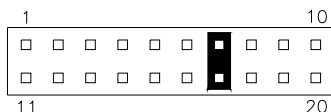
Reset Switch: Pins 9 and 19

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.
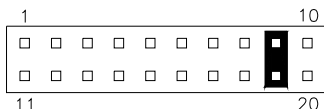
Hard Disk Drive LED Connector: Pins 10 and 20

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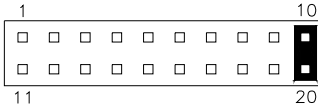
Pin #	Signal Name
10	HDD Active
20	5V

J11: Audio Connector for 5.1 channel
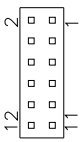
Signal Name	Pin #	Pin #	Signal Name
LINEOUT_L	1	2	LINEOUT_R
LINEOUT Detect	3	4	Ground
LINEIN L	5	6	LINEIN R
LINEIN Detect	7	8	Ground
MIC1	9	10	VREFOUT
MIC1 Detect	11	12	Ground

J13, J14: External PS/2 Keyboard and Mouse Connector
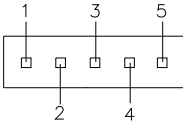
Pin #	J13	J14
1	KB clock	Mouse data
2	KB data	N.C.
3	N.C.	Ground
4	Ground	Vcc
5	Vcc	Mouse clock

J16, J17, J18: USB0/USB1/USB2 Connector

Signal Name	Pin	Pin	Signal Name
Vcc	1	2	Ground
USB0-	3	4	USB1+
USB0+	5	6	USB1-
Ground	7	8	Vcc

VGA1: VGA CRT Connector

VGA1 is a DB-15 VGA connector located beside the COM1 port. The following table shows the pin-out assignments of this connector.
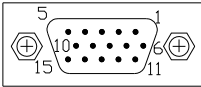
Signal Name	Pin #	Pin #	Signal Name
Red	1	2	Green
Blue	3	4	N.C.
GND	5	6	GND
GND	7	8	GND
VCC	9	10	GND
N.C.	11	12	DDCDATA
HSYNC	13	14	VSYNC
DDCCLK	15		

J12: PS/2 Keyboard and Mouse Connector

J12 uses a Y-cable with dual D-connectors for a PS/2 keyboard and a PS/2 mouse.
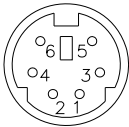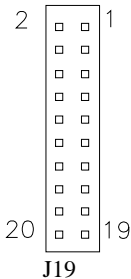
Pin #	Signal Name
1	Keyboard data
2	Mouse data
3	Ground
4	Vcc
5	Keyboard Clock
6	Mouse Clock

CN1: Intel PCI Express GbE RJ45 Connector Intel 82567LM
CN2: Intel PCI Express GbE RJ45 Connector Intel 82574L

J19: TMDS Panel Connector

TMDS stands Transition Minimized Differential Signaling.

Signal Name	Pin #	Pin #	Signal Name
TXIN	2	1	TX1P
GND	4	3	GND
TXCN	6	5	TXCP
+5V	8	7	GND
NC	10	9	HTPG
TX2N	12	11	TX2P
GND	14	13	GND
TX0N	16	15	TX0P
NC	18	17	NC
DDCCLK	20	19	DDCDATA

J19

J15: SPI Debug Tools Port (Factory use only)

U31: x1 Mini PCI Express Slot

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 Award 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 Award 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 Award 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.

Phoenix - AwardBIOS CMOS Setup Utility

Standard CMOS Features	Frequency/Voltage Control
Advanced BIOS Features	Load Fail-Safe Defaults
Advanced Chipset Features	Load Optimized Defaults
Integrated Peripherals	Set Supervisor Password
Power Management Setup	Set User Password
PnP/PCI Configurations	Save & Exit Setup
PC Health Status	Exit Without Saving
ESC : Quit	↑ ↓ → ← : Select Item
F10 : Save & Exit Setup	
Time, Date, Hard Disk Type…	

The section below the setup items of the Main Menu displays the control keys for this menu. At the bottom of the Main Menu just below the control keys section, there is another section, which displays information on the currently highlighted item in the list.

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.*

Standard CMOS Features

"Standard CMOS Features" choice allows you to record some basic hardware configurations in your computer system and set the system clock and error handling. If the motherboard is already installed in a working system, you will not need to select this option. You will need to run the Standard CMOS option, however, if you change your system hardware configurations, the onboard battery fails, or the configuration stored in the CMOS memory was lost or damaged.

Phoenix - AwardBIOS CMOS Setup Utility
Standard CMOS Features

Date (mm:dd:yy)	Tue, Apr 21, 2009	Item Help
Time (hh:mm:ss)	16 : 11 : 00	Menu Level >
IDE Channel 0 Master	None	Change the day, month,
IDE Channel 0 Slave	None	Year and century
IDE Channel 1 Master	None	
IDE Channel 1 Slave	None	
IDE Channel 2 Master	None	
IDE Channel 3 Master	None	
Drive A	1.44M, 3.5 in.	
Drive B	None	
Video	EGA/VGA	
Halt On	All, but keyboard	
Base Memory	640K	
Extended Memory	2086912K	
Total Memory	2087936K	

At the bottom of the menu are the control keys for use on this menu. If you need any help in each item field, you can press the <F1> key. It will display the relevant information to help you. The memory display at the lower right-hand side of the menu is read-only. It will adjust automatically according to the memory changed. The following describes each item of this menu.

Date
The date format is:

Day :	**Sun to Sat**
Month :	**1 to 12**
Date :	**1 to 31**
Year :	**1999 to 2099**

To set the date, highlight the "Date" field and use the PageUp/ PageDown or +/- keys to set the current time.

Time

The time format is: **Hour   : 00 to 23**
                                    Minute : 00 to 59
                                    Second : 00 to 59

To set the time, highlight the "Time" field and use the <PgUp>/ <PgDn> or +/- keys to set the current time.

IDE Channel Master/Slave

IB945F with ICH10DO supports 6 Serial ATA connectors

The onboard Serial ATA connectors provide Primary and Secondary channels for connecting up to four Serial ATA hard disks . Each channel can support up to two hard disks; the first is the "Master" and the second is the "Slave".

Press <Enter> to configure the hard disk. The selections include Auto, Manual, and None. Select 'Manual' to define the drive information manually. You will be asked to enter the following items.

Capacity :	Capacity/size of the hard disk drive
Cylinder :	Number of cylinders
Head :	Number of read/write heads
Precomp :	Write precompensation
Landing Zone :	Landing zone
Sector :	Number of sectors

The Access Mode selections are as follows:
CHS   (HD < 528MB)
LBA   (HD > 528MB and supports Logical Block Addressing)
Large   (for MS-DOS only)
Auto

Video

This field selects the type of video display card installed in your system.
You can choose the following video display cards:

EGA/VGA	For EGA, VGA, SEGA, SVGA or PGA monitor adapters. (default)
CGA 40	Power up in 40 column mode.
CGA 80	Power up in 80 column mode.
MONO	For Hercules or MDA adapters.

Halt On

This field determines whether or not the system will halt if an error is
detected during power up.

No errors	The system boot will not be halted for any error that may be detected.
All errors	Whenever the BIOS detects a non-fatal error, the system will stop and you will be prompted.
All, But Keyboard	The system boot will not be halted for a keyboard error; it will stop for all other errors
All, But Diskette	The system boot will not be halted for a disk error; it will stop for all other errors.
All, But Disk/Key	The system boot will not be halted for a keyboard or disk error; it will stop for all others.

Advanced BIOS Features

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

CPU Feature	Press Enter	ITEM HELP
Hard Disk Boot Priority	Press Enter	
Virus Warning	Disabled	Menu Level >
Quick Power On Self Test	Enabled	
First Boot Device	Floppy	
Second Boot Device	Hard Disk	
Third Boot Device	CDROM	
Boot Other Device	Enabled	
Swap Floppy Device	Disabled	
Boot Floppy Seek	Disabled	
Boot Up NumLock Status	On	
Gate A20 Option	Fast	
Typematic Rate Setting	Disabled	
Typematic Rate (Chars/Sec)	6	
Typematic Delay (Msec)	250	
Security Option	Setup	
APIC Mode	Enabled	
MPS Version Control for OS	1.4	
OS Select For DRAM>64MB	Non-OS2	
Report No FDD For WIN 95	No	
Small Logo (EPA) Show	Disabled	

CPU Feature
Press Enter to configure the settings relevant to CPU Feature.

Hard Disk Boot Priority
With the field, there is the option to choose, aside from the hard disks connected, "Bootable add-in Cards" which refers to other external devices.

Virus Warning
If this option is enabled, an alarm message will be displayed when trying to write on the boot sector or on the partition table on the disk, which is typical of the virus.

Quick Power On Self Test
When enabled, this field speeds up the Power On Self Test (POST) after the system is turned on. If it is set to *Enabled*, BIOS will skip some items.

First/Second/Third Boot Device

These fields determine the drive that the system searches first for an operating system. The options available include *Floppy*, *LS120*, *Hard Disk*, *CDROM, ZIP100, USB-FDD, USB-ZIP, USB-CDROM, Legacy LAN* and *Disabled*.

Boot Other Device

These fields allow the system to search for an OS from other devices other than the ones selected in the First/Second/Third Boot Device.

Swap Floppy Drive

This item allows you to determine whether or not to enable Swap Floppy Drive. When enabled, the BIOS swaps floppy drive assignments so that Drive A becomes Drive B, and Drive B becomes Drive A. By default, this field is set to ***Disabled.***

Boot Up Floppy Seek

This feature controls whether the BIOS checks for a floppy drive while booting up. If it cannot detect one (either due to improper configuration or its absence), it will flash an error message.

Boot Up NumLock Status

This allows you to activate the NumLock function after you power up the system.

Gate A20 Option

This field allows you to select how Gate A20 is worked. Gate A20 is a device used to address memory above 1 MB.

Typematic Rate Setting

When disabled, continually holding down a key on your keyboard will generate only one instance. When enabled, you can set the two typematic controls listed next. By default, this field is set to ***Disabled.***

Typematic Rate (Chars/Sec)

When the typematic rate is enabled, the system registers repeated keystrokes speeds. Settings are from 6 to 30 characters per second.

Typematic Delay (Msec)

When the typematic rate is enabled, this item allows you to set the time interval for displaying the first and second characters. By default, this item is set to ***250msec.***

Security Option
This field allows you to limit access to the System and Setup. The default value is *Setup*. When you select *System,* the system prompts for the User Password every time you boot up. When you select *Setup,* the system always boots up and prompts for the Supervisor Password only when the Setup utility is called up.

APIC Mode
APIC stands for Advanced Programmable Interrupt Controller. The default setting is *Enabled*.

MPS Version Control for OS
This option is specifies the MPS (Multiprocessor Specification) version for the OS. 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*.

OS Select for DRAM > 64MB
This option allows the system to access greater than 64MB of DRAM memory when used with OS/2 that depends on certain BIOS calls to access memory. The default setting is *Non-OS/2*.

Report No FDD For WIN 95
If you are using Windows 95/98 without a floppy disk drive, select Enabled to release IRQ6. This is required to pass Windows 95/98's SCT test. You should also disable the Onboard FDC Controller in the Integrated Peripherals screen when there's no floppy drive in the system. If you set this feature to Disabled, the BIOS will not report the missing floppy drive to Win95/98.

Small Logo (EPA) Show
The EPA logo appears at the right side of the monitor screen when the system is boot up. The default setting is *Disabled*.

Advanced Chipset Features

This Setup menu controls the configuration of the chipset.

Phoenix - AwardBIOS CMOS Setup Utility
Advanced Chipset Features

System BIOS Cacheable	Enabled	ITEM HELP
Memory Hole at 15M-16M	Disabled	Menu Level >
PCI Express Root Port Func	Press Enter	
Disable MCHBAR MMIO	Enabled	
VT-d	Disabled	
** VGA Setting **		
PEG/On Chip VGA Control	Auto	
On-Chip Frame Buffer Size	32MB	
DVMT Mode	Enabled	
Total GFX Memory	256MB	
PAVP Mode	Lite	
SDVO Device Setting	None	
Boot Display	CRT	

System BIOS Cacheable

The setting of *Enabled* allows caching of the system BIOS ROM at F000h-FFFFFh, resulting in better system performance. However, if any program writes to this memory area, a system error may result.

Memory Hole At 15M-16M

In order to improve performance, certain space in memory can be reserved for ISA cards. This memory must be mapped into the memory space below 16 MB. The choices are *Enabled* and *Disabled*.

PCI Express Root Port Func

Press Enter to configure this field.

Disable MCHBAR MMIO

By default, this feature is Enabled.

VT-d

By default, this feature is Disabled.

VGA Setting

The fields under the On-Chip VGA Setting and their default settings are:
PEG/On Chip VGA Control: Auto
On-Chip Frame Buffer Size: 32MB
DVMT Mode: Enabled
Total GFX memory: 256MB
PAVP Mode: Lite
SDVO Device Setting: None
Boot Display: CRT

Integrated Peripherals

This section sets configurations for your hard disk and other integrated peripherals. The first screen shows three main items for user to select. Once an item selected, a submenu appears. Details follow.

Phoenix - AwardBIOS CMOS Setup Utility
Integrated Peripherals

		ITEM HELP
OnChip IDE Device	Press Enter	
Onboard Device	Press Enter	Menu Level >
SuperIO Device	Press Enter	
USB Device Setting	Press Enter	

Phoenix - AwardBIOS CMOS Setup Utility
OnChip IDE Device

		ITEM HELP
IDE HDD Block Mode	Enabled	
IDE DMA transfer access	Enabled	
IDE Primary Master PIO	Auto	Menu Level >
IDE Primary Slave PIO	Auto	
IDE Primary Master UDMA	Auto	
IDE Primary Slave UDMA	Auto	
On-Chip Secondary PCI IDE	Enabled	
IDE Secondary Master PIO	Auto	
IDE Secondary Slave PIO	Auto	
IDE Secondary Master UDMA	Auto	
IDE Secondary Slave UDMA	Auto	
SATA Mode	IDE	
LEGACY Mode Support	Enabled	

Phoenix - AwardBIOS CMOS Setup Utility
Onboard Device

		ITEM HELP
High Definition Audio	Enabled	
Onboard LAN Boot ROM	Disabled	

Phoenix - AwardBIOS CMOS Setup Utility
SuperIO Device

		ITEM HELP
Onboard FDC Controller	Enabled	
Onboard Serial Port 1	3F8/IRQ4	Menu Level >
Onboard Serial Port 2	2F8/IRQ3	
UART Mode Select	Normal	
RxD , TxD Active	Hi, Lo	
IR Transmission Delay	Enabled	
UR2 Duplex Mode	Half	
Use IR Pins	IR-Rx2Tx2	
Onboard Parallel Port	378/IRQ7	
Parallel Port Mode	SPP	
EPP Mode Select	EPP1.7	
ECP Mode Use DMA	3	
PWRON After PWR-Fail	Off	

Phoenix - AwardBIOS CMOS Setup Utility
USB Device Setting

USB 1.0 Controller	Enabled	ITEM HELP
USB 2.0 Controller	Enabled	Menu Level >
USB Operation Mode	High Speed	
USB Keyboard Function	Enabled	
USB Mouse Function	Enabled	
USB Storage Function	Enabled	
*** USB Mass Storage Device Boot Setting ***		

IDE HDD Block Mode

This field allows your hard disk controller to use the fast block mode to transfer data to and from your hard disk drive.

IDE DMA Transfer Access

This field, by default, is Enabled

OnChip Secondary PCI IDE

This field, by default, is Enabled

IDE Primary/Secondary Master/Slave PIO

These fields allow your system hard disk controller to work faster. Rather than have the BIOS issue a series of commands that transfer to or from the disk drive, PIO (Programmed Input/Output) allows the BIOS to communicate with the controller and CPU directly.

The system supports five modes, numbered from 0 (default) to 4, which primarily differ in timing. When Auto is selected, the BIOS will select the best available mode.

IDE Primary/Secondary Master/Slave UDMA

These fields allow your system to improve disk I/O throughput to 33Mb/sec with the Ultra DMA/33 feature. The options are *Auto* and *Disabled*.

SATA Mode (The item is for IB945F only)

The setting choices for the SATA Mode are IDE, RAID and AHCI Mode. Select [IDE] if you want to have SATA function as IDE.
Select [AHCI] for Advanced Host Controller Interface (AHCI) feature, with improved SATA performance with native command queuing & native hot plug. Select [RAID] to use SATA as RAID function. RAID function is supported on the board if it uses ICH10DO. *(IB945F supports 6 x SATA with RAID.)*

LEGACY Mode Support

When the Serial ATA (SATA) is set with the legacy mode enabled, then the SATA is set to the conventional IDE mode. Legacy mode is otherwise known as compatible mode.

High Definition Audio

This field, by default, is Enabled

Onboard Lan Boot ROM

This field, by default, is Disabled

Onboard FDC Controller

Select *Enabled* if your system has a floppy disk controller (FDC) installed on the motherboard and you wish to use it. If you install an add-in FDC or the system has no floppy drive, select Disabled in this field. This option allows you to select the onboard FDD port.

Onboard Serial/Parallel Port

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

Serial Port 1	3F8/IRQ4
Serial Port 2	2F8/IRQ3
Parallel Port	378H/IRQ7

UART Mode Select

This field determines the UART 2 mode in your computer. The default value is *Normal*. Other options include *IrDA* and *ASKIR*.

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

PWRON After PWR-Fail

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

USB 1.0 Controller

The options for this field are *Enabled* and *Disabled*. By default, this field is set to *Enabled*.

USB 2.0 Controller
The options for this field are *Enabled* and *Disabled*. By default, this field is set to ***Enabled***. In order to use USB 2.0, necessary OS drivers must be installed first. ***Please update your system to Windows 2000 SP4 or Windows XP SP2.***

USB Keyboard/Mouse/Storage Function
The options for this field are *Enabled* and *Disabled*. By default, this field is set to ***Enabled***.

Power Management Setup

		ITEM HELP
PCI Express PM Function	Press Enter	
ACPI Function	Enabled	Menu Level >
ACPI Suspend	S1(POS)	
Run VGABIOS if S3 Resume	Auto	
Power Management	User Define	
Video Off Method	V/H SYNC+Blank	
Video Off In Suspend	Yes	
Suspend Type	Stop Grant	
Modem Use IRQ	3	
Suspend Mode	Disabled	
HDD Power Down	Disabled	
Soft-Off by PWR-BTTN	Instant-Off	
CPU THRM-Throttling	50.0%	
Wake-Up by PCI Card	Disabled	
Power On by Ring	Disabled	
Resume by Alarm	Disabled	
Date (of Month) Alarm	0	
Time (hh:mm:ss) Alarm	0 : 0 : 0	
** Reload Global Timer Events **		
Primary IDE 0	Enabled	
Primary IDE 1	Enabled	
Secondary IDE 0	Enabled	
Secondary IDE 1	Enabled	
FDD, COM, LPT Port	Enabled	
PCI PIRQ[A-D] #	Enabled	
HPET Support	Enabled	
HPET Mode	32-bit mode	

ACPI Function

Enable this function to support ACPI (Advance Configuration and Power Interface).

ACPI Suspend

The default setting of the ACPI Suspend mode is *S1(POS)*.

RUN VGABIOS if S3 Resume

The default setting of this field is *Auto*.

Power Management

This field allows you to select the type of power saving management modes. There are four selections for Power Management.

Min. Power Saving	Minimum power management
Max. Power Saving	Maximum power management.
User Define	Each of the ranges is from 1 min. to 1hr. Except for HDD Power Down which ranges from 1 min. to 15 min.

Video Off Method

This field defines the Video Off features. There are three options.

V/H SYNC + Blank	Blank the screen and turn off vertical and horizontal scanning.
DPMS	Default setting, allows BIOS to control the video display.
Blank Screen	Writes blanks to the video buffer.

Video Off In Suspend

When enabled, the video is off in suspend mode.

Suspend Type

The default setting for the Suspend Type field is *Stop Grant*.

Modem Use IRQ

This field sets the IRQ used by the Modem. By default, the setting is *3*.

Suspend Mode

When enabled, and after the set time of system inactivity, all devices except the CPU will be shut off.

HDD Power Down

When enabled, and after the set time of system inactivity, the hard disk drive will be powered down while all other devices remain active.

Soft-Off by PWRBTN

This field defines the power-off mode when using an ATX power supply. The *Instant Off* mode allows powering off immediately upon pressing the power button. In the *Delay 4 Sec* mode, the system powers off when the power button is pressed for more than four seconds or enters the suspend mode when pressed for less than 4 seconds.

CPU THRM-Throttling

When the system enters Doze mode, the CPU clock runs only part of the time. You may select the percent of time that the clock runs.

Wake up by PCI Card

By default, this field is *Disabled.*

Power On by Ring

This field enables or disables the power on of the system through the modem connected to the serial port or LAN.

Resume by Alarm

This field enables or disables the resumption of the system operation. When enabled, the user is allowed to set the *Date* and *Time*.

Reload Global Timer Events

The HDD, FDD, COM, LPT Ports, and PCI PIRQ are I/O events that can prevent the system from entering a power saving mode or can awaken the system from such a mode. When an I/O device wants to gain the attention of the operating system, it signals this by causing an IRQ to occur. When the operating system is ready to respond to the request, it interrupts itself and performs the service.

HPET Support

HPET stands for High Precision Event Timer. HPET can produce periodic interrupts at a much higher resolution than the RTC and is used to synchronize multimedia streams, providing smooth playback and reducing the need to use other timestamp calculations such as an x86 CPU's RDTSC instruction. HPET is not supported in Windows XP, Windows Server 2003, or earlier Windows versions. HPET is supported under Linux and Windows Vista. By default, this field is enabled*.*

HPET Mode

By default, this field is set to *32-bit mode*.

PNP/PCI Configurations

This option configures the PCI bus system. All PCI bus systems on the system use INT#, thus all installed PCI cards must be set to this value.

Phoenix - AwardBIOS CMOS Setup Utility
PnP/PCI Configurations

		ITEM HELP
PNP OS Install	No	
Init Display First	PCI Slot	Menu Level
Reset Configuration Data	Disabled	
		Select Yes if you are
Resources Controlled By	Auto (ESCD)	using a Plug and Play
IRQ Resources	Press Enter	capable operating
DMA Resources	Press Enter	system Select No if
		you need the BIOS to
PCI/VGA Palette Snoop	Disabled	configure non-boot
		devices
PCI Express relative items		
Maximum Payload Size	128	

PNP OS Install
Enable the PNP OS Install option if it is supported by the operating system installed. The default value is *No.*

Init Display First
The default setting is ***PCI Card***.

Reset Configuration Data
The default value is ***Disabled.***

Resources Controlled by
This PnP BIOS can configure all of the boot and compatible devices with the use of a PnP operating system such as Windows 95.

PCI/VGA Palette Snoop
Some non-standard VGA display cards may not show colors properly. This field allows you to set whether or not MPEG ISA/VESA VGA cards can work with PCI/VGA. When this field is enabled, a PCI/VGA can work with an MPEG ISA/VESA VGA card. When this field is disabled, a PCI/VGA cannot work with an MPEG ISA/VESA card.

Maximum Payload Size
The default setting of the PCI Express Maximum Payload Size is 128.

PC Health Status

Phoenix - AwardBIOS CMOS Setup Utility
PC Health Status

Shutdown Temperature	Disabled	ITEM HELP
CPU Warning Temperature	Disabled	
Current System Temp	32°C/89°F	Menu Level >
Current CPU Temp	39°C/102°F	
System Fan Speed	0 RPM	
CPU Fan Speed	4440 RPM	
Vcore	1.29 V	
+12 V	12.19 V	
+1.5V	1.56 V	
+ 5 V	5.14 V	
3.3 V	3.32 V	
VBAT (V)	3.21 V	
3.3VSB(V)	3.31 V	

CPU Warning Temperature

This field allows the user to set the temperature so that when the temperature is reached, the system sounds a warning. This function can help prevent damage to the system that is caused by overheating.

Shutdown Temperature

This field allows the user to set the temperature by which the system automatically shuts down once the threshold temperature is reached. This function can help prevent damage to the system that is caused by overheating.

Temperatures/Fan Speeds/Voltages

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

Frequency/Voltage Control

Phoenix - AwardBIOS CMOS Setup Utility
Frequency/Voltage Control

Auto Detect PCI Clk	Disabled	ITEM HELP
Spread Spectrum	Disabled	Menu Level >

Auto Detect PCI Clk
This field enables or disables the auto detection of the PCI clock.

Spread Spectrum
This field sets the value of the spread spectrum. The default setting is *Disabled*. This field is for CE testing use only.

Load Fail-Safe 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.

Load Optimized 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.

Set Supervisor/User Password

These 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, up to eight characters in length, 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.

Save & Exit Setup

This option allows you to determine whether or not to accept the modifications. If you type "Y", you will quit the setup utility and save all changes into the CMOS memory. If you type "N", you will return to Setup utility.

Exit Without Saving

Select this option to exit the Setup utility without saving the changes you have made in this session. Typing "Y" will quit the Setup utility without saving the modifications. Typing "N" will return you to Setup utility.

Drivers Installation

This section describes the installation procedures for software and drivers under the Windows 2000, Windows XP and Windows Vista. The software and drivers are included with the board. 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 2000/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 2000/XP/Vista. (Before installing this utility, please update your system to Windows 2000 SP4 or Windows XP SP2)

1. Insert the drivers DVD into the DVD drive. Click *Intel* and then *Intel(R) Q45 Chipset Drivers*. Click *Intel(R) Chipset Software Installation Utility.*

2. When the Welcome screen appears, click *Next* to continue. Click *Yes* to accept the software license agreement and proceed with the installation process.

3. On the Readme Information screen, click *Next* to continue. When the Setup Progress screen appears, click *Next* to continue.

4. Setup process is now complete. Click *Finish* to restart the computer.

Intel Graphics Driver Installation

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

1. Insert the drivers DVD into the DVD drive. Click *Intel* and then *Intel(R) Q45 Chipset Drivers*. Click *Intel(R) Q45 Chipset Family Graphics Driver.*

2. When the Welcome screen appears, click *Next* to continue. Click *Yes* to accept the software license agreement and proceed with the installation process.

3. On the Readme Information screen, click *Next* to continue.

4. When the Setup Progress screen appears, click *Next* to continue.

5. Setup is complete. Click *Finish* to restart the computer.
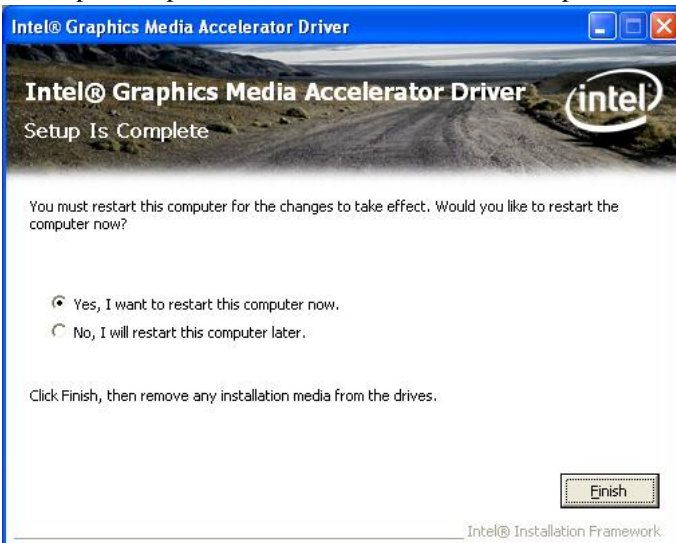
Realtek HD Code Audio Driver Installation

Follow the steps below to install the Realtek High Definition Codec Audio Driver.

1. Insert the drivers DVD into the DVD drive. Click *Intel* and then *Intel(R) Q45 Chipset Drivers*. Click *Realtek High Definition Codec Audio Driver.*

2. When the Welcome screen appears, click *Next* to continue.
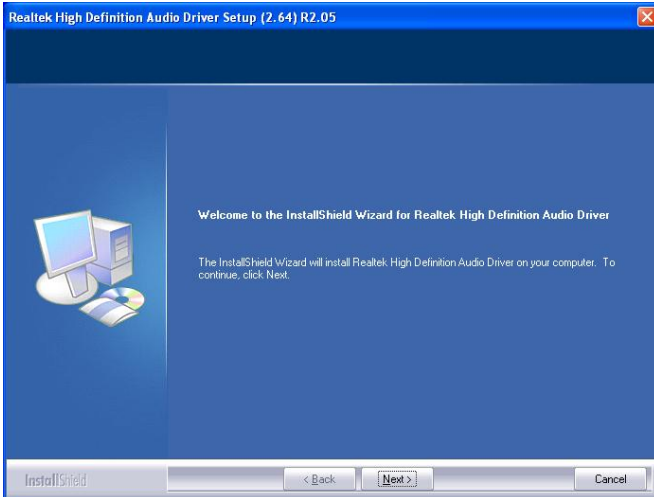
3. InstallShield Wizard is complete. Click *Finish* to restart the computer.

LAN Drivers Installation

Follow the steps below to start installing the Intel PRO LAN drivers.

1. Insert the drivers DVD into the DVD drive. Click *Intel* and then *Intel(R) Q45 Chipset Drivers*. Click *Intel(R) PRO LAN Network Drivers.*
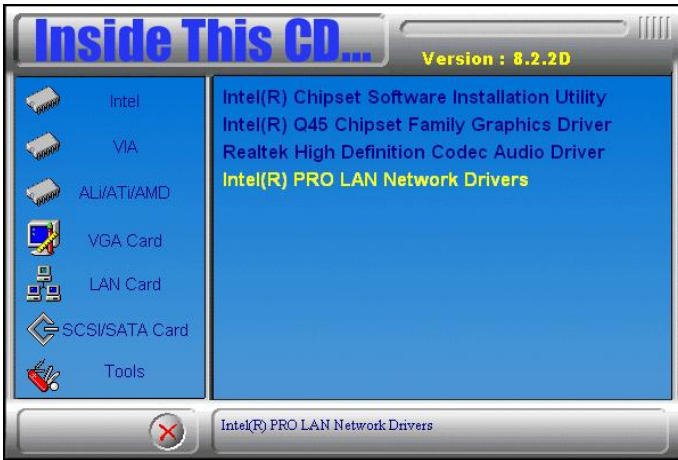
2. When the Welcome screen appears, click *Next* to continue. Click *Yes* to accept the software license agreement and proceed with the process.
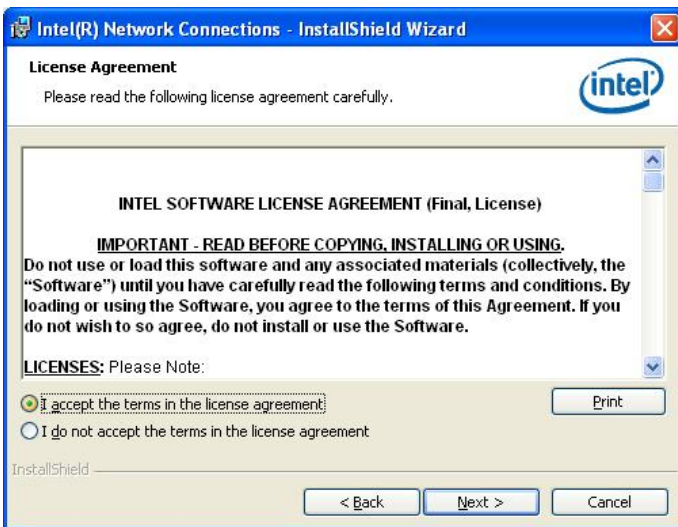
3. On the Setup Options screen, the checkbox for Drivers should be checked. Now, click *Next* to continue.
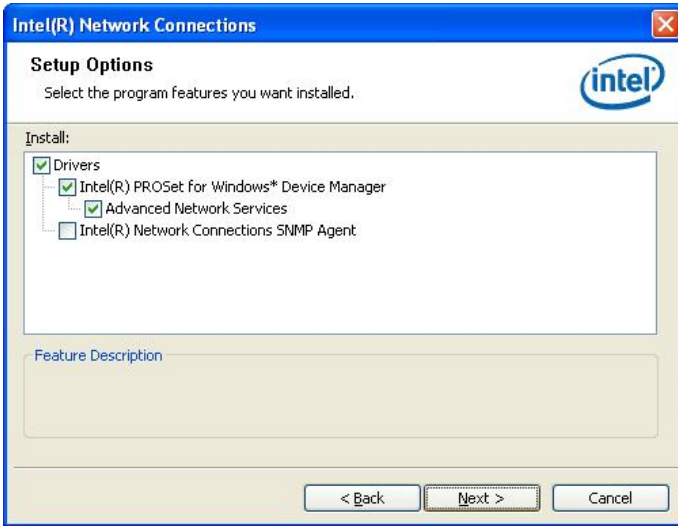
4. The wizard is ready to begin installation Now, click *Install* to continue.

5. InstallShield wizard is complete. 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	Floppy Disk Controller
IRQ7	Parallel Port #1
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:

```
//-------------------------------------------------------------------------
//
// 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 "W627DHG.H"
//-------------------------------------------------------------------------
int main (void);

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

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

    SIO = Init_W627DHG();
    if (SIO == 0)
    {
       printf("Can not detect Winbond 83627DHG, program abort.\n");
       return(1);
    }

    WDTInitial();

    WDTEnable(10);

    WDTDisable();

    return 0;
}
//-------------------------------------------------------------------------
void WDTInitial(void)
```

```
{
    unsigned char bBuf;

    bBuf = Get_W627DHG_Reg(0x2D);
    bBuf &= (~0x01);
    Set_W627DHG_Reg(0x2D, bBuf);                    //Enable WDTO
}
//---------------------------------------------------------------------------
void WDTEnable(unsigned char NewInterval)
{
    unsigned char bBuf;

    Set_W627DHG_LD(0x08);                           //switch to logic device 8
    Set_W627DHG_Reg(0x30, 0x01);                    //enable timer
    bBuf = Get_W627DHG_Reg(0xF5);
    bBuf &= (~0x08);
    Set_W627DHG_Reg(0xF5, bBuf);                    //count mode is second

    Set_W627DHG_Reg(0xF6, NewInterval);             //set timer
}
//---------------------------------------------------------------------------
void WDTDisable(void)
{
    Set_W627DHG_LD(0x08);                           //switch to logic device 8
    Set_W627DHG_Reg(0xF6, 0x00);                    //clear watchdog timer
    Set_W627DHG_Reg(0x30, 0x00);                    //watchdog disabled
}
//---------------------------------------------------------------------------
```

```
//------------------------------------------------------------------------
//
// 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 "W627DHG.H"
#include <dos.h>
//------------------------------------------------------------------------
unsigned int W627DHG_BASE;
void Unlock_W627DHG (void);
void Lock_W627DHG (void);
//------------------------------------------------------------------------
unsigned int Init_W627DHG(void)
{
     unsigned int result;
     unsigned char ucDid;

     W627DHG_BASE = 0x4E;
     result = W627DHG_BASE;

     ucDid = Get_W627DHG_Reg(0x20);
     if (ucDid == 0xA0)                          //W83627DHG??
     {  goto Init_Finish; }
     else if (ucDid == 0xB0)                     //W83627DHG-P??
     {  goto Init_Finish; }

     W627DHG_BASE = 0x2E;
     result = W627DHG_BASE;

     ucDid = Get_W627DHG_Reg(0x20);
     if (ucDid == 0xA0)                          //W83627DHG??
     {  goto Init_Finish; }
     else if (ucDid == 0xB0)                     //W83627DHG-P??
     {  goto Init_Finish; }

     W627DHG_BASE = 0x00;
     result = W627DHG_BASE;
Init_Finish:
     return (result);
}
//------------------------------------------------------------------------
void Unlock_W627DHG (void)
{
     outportb(W627DHG_INDEX_PORT, W627DHG_UNLOCK);
     outportb(W627DHG_INDEX_PORT, W627DHG_UNLOCK);
}
//------------------------------------------------------------------------
void Lock_W627DHG (void)
{
     outportb(W627DHG_INDEX_PORT, W627DHG_LOCK);
}
//------------------------------------------------------------------------
void Set_W627DHG_LD( unsigned char LD)
```

```
{
     Unlock_W627DHG();
     outportb(W627DHG_INDEX_PORT, W627DHG_REG_LD);
     outportb(W627DHG_DATA_PORT, LD);
     Lock_W627DHG();
}
//-------------------------------------------------------------------------
void Set_W627DHG_Reg( unsigned char REG, unsigned char DATA)
{
     Unlock_W627DHG();
     outportb(W627DHG_INDEX_PORT, REG);
     outportb(W627DHG_DATA_PORT, DATA);
     Lock_W627DHG();
}
//-------------------------------------------------------------------------
unsigned char Get_W627DHG_Reg(unsigned char REG)
{
     unsigned char Result;
     Unlock_W627DHG();
     outportb(W627DHG_INDEX_PORT, REG);
     Result = inportb(W627DHG_DATA_PORT);
     Lock_W627DHG();
     return Result;
}
//-------------------------------------------------------------------------




//-------------------------------------------------------------------------
//
// 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 __W627DHG_H
#define __W627DHG_H              1
//-------------------------------------------------------------------------
#defineW627DHG_INDEX_PORT         (W627DHG_BASE)
#defineW627DHG_DATA_PORT          (W627DHG_BASE+1)
//-------------------------------------------------------------------------
#defineW627DHG_REG_LD             0x07
//-------------------------------------------------------------------------
#define W627DHG_UNLOCK            0x87
#defineW627DHG_LOCK              0xAA
//-------------------------------------------------------------------------
unsigned int Init_W627DHG(void);
void Set_W627DHG_LD( unsigned char);
void Set_W627DHG_Reg( unsigned char, unsigned char);
unsigned char Get_W627DHG_Reg( unsigned char);
//-------------------------------------------------------------------------
#endif        //__W627DHG_H
```

D. Digital I/O Sample Code

```
//-------------------------------------------------------------------------
//
// 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 "W627DHG.H"
//-------------------------------------------------------------------------
int main (void);

void Dio3Initial(void);
void Dio3SetOutput(unsigned char);
unsigned char Dio3GetInput(void);
void Dio3SetDirection(unsigned char);
unsigned char Dio3GetDirection(void);
//-------------------------------------------------------------------------
int main (void)
{
     char SIO;

     SIO = Init_W627DHG();
     if (SIO == 0)
     {
        printf("Can not detect Winbond 83627DHG, program abort.\n");
        return(1);
     }

     Dio3Initial();

     //for GPIO30..37
     Dio3SetDirection(0x0F);  //GP30..33 = input, GP34..37=output
     printf("Current DIO direction = 0x%X\n", Dio3GetDirection());

     printf("Current DIO status = 0x%X\n", Dio3GetInput());

     printf("Set DIO output to high\n");
     Dio3SetOutput(0x0F);

     printf("Set DIO output to low\n");
     Dio3SetOutput(0x00);

     return 0;
}
//-------------------------------------------------------------------------
void Dio3Initial(void)
{
     unsigned char ucBuf;

     //switch GPIO multi-function pin
     ucBuf = Get_W627DHG_Reg(0x2C);
```

```
    ucBuf &= 0x1F;
    Set_W627DHG_Reg(0x2C, ucBuf);                            //clear

    Set_W627DHG_LD(0x09);                                    //switch to logic device 9

    //enable the GP3 group
    ucBuf = Get_W627DHG_Reg(0x30);
    ucBuf |= 0x02;
    Set_W627DHG_Reg(0x30, ucBuf);

    //input detect type
    Set_W627DHG_Reg(0xFE, 0xFF);
}
//---------------------------------------------------------------------------
void Dio3SetOutput(unsigned char NewData)
{
    Set_W627DHG_LD(0x09);                                    //switch to logic device 9
    Set_W627DHG_Reg(0xF1, NewData);
}
//---------------------------------------------------------------------------
unsigned char Dio3GetInput(void)
{
    unsigned char result;

    Set_W627DHG_LD(0x09);                                    //switch to logic device 9
    result = Get_W627DHG_Reg(0xF1);
    return (result);
}
//---------------------------------------------------------------------------
void Dio3SetDirection(unsigned char NewData)
{
    //NewData : 1 for input, 0 for output
    Set_W627DHG_LD(0x09);                                    //switch to logic device 9
    Set_W627DHG_Reg(0xF0, NewData);
}
//---------------------------------------------------------------------------
unsigned char Dio3GetDirection(void)
{
    unsigned char result;

    Set_W627DHG_LD(0x09);                                    //switch to logic device 9
    result = Get_W627DHG_Reg(0xF0);
    return (result);
}
//---------------------------------------------------------------------------
```

```
//------------------------------------------------------------------------
//
// 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 "W627DHG.H"
#include <dos.h>
//------------------------------------------------------------------------
unsigned int W627DHG_BASE;
void Unlock_W627DHG (void);
void Lock_W627DHG (void);
//------------------------------------------------------------------------
unsigned int Init_W627DHG(void)
{
      unsigned int result;
      unsigned char ucDid;

      W627DHG_BASE = 0x4E;
      result = W627DHG_BASE;

      ucDid = Get_W627DHG_Reg(0x20);
      if (ucDid == 0xA0)                        //W83627DHG??
      {  goto Init_Finish; }
      else if (ucDid == 0xB0)                   //W83627DHG-P??
      {  goto Init_Finish; }

      W627DHG_BASE = 0x2E;
      result = W627DHG_BASE;

      ucDid = Get_W627DHG_Reg(0x20);
      if (ucDid == 0xA0)                        //W83627DHG??
      {  goto Init_Finish; }
      else if (ucDid == 0xB0)                   //W83627DHG-P??
      {  goto Init_Finish; }

      W627DHG_BASE = 0x00;
      result = W627DHG_BASE;

Init_Finish:
      return (result);
}
//------------------------------------------------------------------------
void Unlock_W627DHG (void)
{
      outportb(W627DHG_INDEX_PORT, W627DHG_UNLOCK);
      outportb(W627DHG_INDEX_PORT, W627DHG_UNLOCK);
}
//------------------------------------------------------------------------
void Lock_W627DHG (void)
{
      outportb(W627DHG_INDEX_PORT, W627DHG_LOCK);
}
//------------------------------------------------------------------------
void Set_W627DHG_LD( unsigned char LD)
{
      Unlock_W627DHG();
      outportb(W627DHG_INDEX_PORT, W627DHG_REG_LD);
```

```
     outportb(W627DHG_DATA_PORT, LD);
     Lock_W627DHG();
}
//-------------------------------------------------------------------------
void Set_W627DHG_Reg( unsigned char REG, unsigned char DATA)
{
     Unlock_W627DHG();
     outportb(W627DHG_INDEX_PORT, REG);
     outportb(W627DHG_DATA_PORT, DATA);
     Lock_W627DHG();
}
//-------------------------------------------------------------------------
unsigned char Get_W627DHG_Reg(unsigned char REG)
{
     unsigned char Result;
     Unlock_W627DHG();
     outportb(W627DHG_INDEX_PORT, REG);
     Result = inportb(W627DHG_DATA_PORT);
     Lock_W627DHG();
     return Result;
}
//-------------------------------------------------------------------------



//-------------------------------------------------------------------------
//
// 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 __W627DHG_H
#define __W627DHG_H                 1
//-------------------------------------------------------------------------
#defineW627DHG_INDEX_PORT          (W627DHG_BASE)
#defineW627DHG_DATA_PORT           (W627DHG_BASE+1)
//-------------------------------------------------------------------------
#defineW627DHG_REG_LD              0x07
//-------------------------------------------------------------------------
#define W627DHG_UNLOCK             0x87
#defineW627DHG_LOCK                0xAA
//-------------------------------------------------------------------------
unsigned int Init_W627DHG(void);
void Set_W627DHG_LD( unsigned char);
void Set_W627DHG_Reg( unsigned char, unsigned char);
unsigned char Get_W627DHG_Reg( unsigned char);
//-------------------------------------------------------------------------
#endif //__W627DHG_H
```